



An hybrid denoising algorithm based on directional wavelet packets

Amir Averbuch¹ · Pekka Neittaanmäki² · Valery Zheludev¹ · Moshe Salhov¹ · Jonathan Hauser³

Received: 12 January 2022 / Revised: 10 May 2022 / Accepted: 22 May 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

The paper presents an image denoising algorithm by combining a method that is based on directional quasi-analytic wavelet packets (qWPs) with the popular BM3D algorithm. The qWP-based denoising algorithm (qWPdn) consists of decomposition of the degraded image, application of adaptive localized soft thresholding to the transform coefficients using the *Bivariate Shrinkage* methodology, and restoration of the image from the thresholded coefficients from several decomposition levels. The combined method consists of several iterations of qWPdn and BM3D algorithms, where at each iteration the output from one algorithm updates the input to the other. The proposed methodology couples the qWPdn capabilities to capture edges and fine texture patterns even in the severely corrupted images with utilizing the sparsity in real images and self-similarity of patches in the image that is inherent in the BM3D. Multiple experiments, which compared the proposed methodology performance with the performance of six state-of-the-art denoising algorithms, confirmed that the combined algorithm was quite competitive.

Keywords Denoising · Directional wavelet packet · BM3D · Hybrid

✉ Amir Averbuch
amir1@tauex.tau.ac.il

Pekka Neittaanmäki
pekka.neittaanmaki@jyu.fi

Valery Zheludev
zhel@tauex.tau.ac.il

Moshe Salhov
moshebar-s@013.net

Jonathan Hauser
thehauser@gmail.com

¹ School of Computer Science, Tel Aviv University, Ramat Aviv, Tel Aviv, Israel

² Faculty of Mathematical Information Technology, University of Jyväskylä, Agora, Jyväskylä, Finland

³ School of Electrical Engineering, Tel Aviv University, Ramat Aviv, Tel Aviv, Israel

1 Introduction

High quality denoising is one of the main challenges in image processing. It tries to achieve suppression of noise while capturing and preserving edges and fine structures in the image. A huge number of publications related to a variety of denoising methods (see, for example the reviews Goyal et al., 2020; Ilesanmi & Ilesanmi, 2021; Tian et al., 2020) exist.

Currently, two main groups of image denoising methods exist:

1. “Classical” schemes, which operate on single images;
2. Methodologies based on Deep Learning.

In turn, most up to date “classical” schemes, where the proposed algorithm belongs to, are based on one of two approaches.

Utilization of non-local self-similarity (NLSM) in images: Starting from the introduction of the Non-local mean (NLM) filter in Buades et al. (2005), which is based on the similarity between pixels in different parts of the image, the exploitation of various forms of the non-local self-similarity in images has resulted in a remarkable progress in image denoising. It is reflected in multiple publications (Dabov et al., 2007; 2009; Dong et al., 2013; Gu et al., 2014, Hou & Shen 2018; Liu & Osher, 2019, Zhou et al., 2021, to name a very few). The non-local self-similarity is explored in some denoising schemes based on Deep Learning (Chen & Pock, 2017; Zhang et al., 2017, for example).

A kind of benchmark in image denoising remains the BM3D algorithm (Dabov et al., 2007), which was presented as far as 2007. The algorithm exploits the self-similarity of patches and sparsity of the image in a transform domain. It collects similar patches in the image into a 3D array, which is subjected to a decorrelating 3D transform followed by either hard thresholding or Wiener filtering. After the inverse transforms, the processed patches are returned to their original locations with corresponding weights. This method is highly efficient in restoration of moderately noised images. However, the BM3D tends to over-smooth and smear the image fine structure and edges when noise is strong. Also, the BM3D is not successful when the image contains many edges oriented in multiple directions.

Some improvement of the original BM3D algorithm was achieved by using shape adaptive neighborhoods and the inclusion of the PCA into the 3D transform (BM3D-SAPCA algorithm, Dabov et al. 2009). Even better results compared to BM3D and BM3D-SAPCA are demonstrated by the so-called WNNM method (Gu et al., 2014), which is based on the assumption that, by stacking the nonlocal similar patch vectors into a matrix, this matrix should be a low rank matrix and, as such, must have sparse singular values. The low rank matrix approximation in Gu et al. (2014) is achieved by an adaptive weighted thresholding SVD values of such matrices. Many denoising algorithms presented in recent years, which are based on the NLSM concept, report results close to the results produced by BM3D-SAPCA and WNNM. At the same time, they share, to some extent, the shortcomings of the BM3D algorithm, especially blurring the fine structure of images restored from the severely degraded inputs.

Transform domain filtering using directional filters: A way to capture lines, edges and texture pattern while restoring degraded images is to use directional filters and, respectively, dictionaries of waveforms oriented in multiple directions. A number of dictionaries are reported in the literature and applied to image processing. We mention contourlets (Do & Vetterli, 2008), curvelets (Candés & Donoho, 2004; Candés et al., 2006), pseudo-polar Fourier transforms (Averbuch et al., 2008a, b) and related to them shearlets (Kutyniok & Labate, 2012; Lim et al., 2012). However, while these transforms successfully capture edges in images,

these dictionaries did not demonstrate a satisfactory texture restoration due to the shortage of oscillating waveforms in the dictionaries.

A number of publications (Bayram & Selesnick, 2008; Han & Zhao, 2014; Han et al., 2019, 2016; Jalobeanu et al., 2000; Ji et al., 2017, 2018), to name a few, present directional dictionaries by the tensor multiplication of complex wavelets (Kingsbury, 1999; Selesnick et al., 2005), wavelet frames and wavelet packets (WPs). The tight tensor-product complex wavelet frames (TP_CTF_n)¹ with different numbers of directions, are designed in Han and Zhao (2014), Han et al. (2016, 2019) and some of them, in particular cptTP_CTF_6 , TP_CTF_6 and $\text{TP_CTF}_6^\downarrow$, demonstrate impressive performance for image denoising and inpainting. The waveforms in these frames are oriented in 14 directions and, due to the 2-layer structure of their spectra, they possess certain, although limited, oscillatory properties.

In Che and Zhuang (2018) [algorithm *Digital Affine Shear Filter Transform with 2-Layer Structure (DAS-2)*] the two-layer structure, which is inherent in the TP_CTF_6 frames, is incorporated into shearlet-based directional filter banks introduced in Zhuang (2016). This improves the performance of DAS-2 in comparison to TP_CTF_6 on texture-rich images such as “Barbara”, which is not the case for smoother images like “Lena”.

Recently, we designed a family of complex WPs (Averbuch et al., 2020, brief outline of the design is in Averbuch et al., 2021), which are referred to as quasi-analytic WPs (qWPs). As a base for the design, the family of WPs originated from polynomial splines of different orders, which are described in Averbuch et al. (2019) (Chapter 4), is used. The two-dimensional (2D) qWPs are derived by a standard tensor products of 1D qWPs. The real parts of the 2D qWPs possess a combination of properties valuable for image processing: They are oriented in multiple directions (see Table 1). The waveforms are close to directional cosines with multiple frequencies modulated by localized low-frequency 2D signals. Their DFT spectra form a refined split of the frequency domain. Both one- and two-dimensional qWP transforms are implemented in a very fast ways by using the Fast Fourier transform (FFT). The directional qWPs are successfully applied to image inpainting (Averbuch et al., 2021).

Due to the above properties, a qWP-based denoising algorithm (qWPdn), which utilizes a version of the Bivariate Shrinkage algorithm (BSA Şendur and Selesnick 2002b,a), proved to be efficient for image denoising. Experiments with the qWPdn demonstrate its ability to restore edges and texture details even from severely degraded images. In most experiments, the qWPdn to be described in Sect. 3.1 provides better resolution of edges and fine structures compared to the cptTP_CTF_6 , DAS-2 and NLSM-based algorithms, which is reflected in getting higher SSIM (Wang et al., 2004; Wang & Bovik, 2009) values. On the other hand, the NLSM-based algorithms proved to be superior in the noise suppression, especially in smooth regions of images, thus producing the highest PSNR values in almost all the experiments. However, some over-smoothing effect on the edges and fine texture persisted under the BM3D, BM3D-SAPCA, NCSR (Dong et al., 2013) and WNNM algorithms. Especially, this is the case for severely degraded images.

Therefore, we propose to combine the qWPdn and BM3D algorithms in order to retain strong features of both algorithms and to get rid of their drawbacks. The combined qWPdn–BM3D algorithms presented in the paper consists of the iterated execution of the qWPdn and BM3D algorithms in a way that at each iteration, the output from one algorithm updates the input to the other. Typically, 2–3 (rarely more than 4) iterations are needed to get an excellent result. In multiple experiments, part of which is reported in Sect. 3.3, the qWPdn–BM3D

¹ The index n refers to the number of filters in the underlying one-dimensional complex tight framelet filter bank.

algorithms performance is compared with the performance of the NLSM-based algorithms BM3D, BM3D-SAPCA, NCSR and WNNM algorithms and the algorithms cptTP-CTF₆ and DAS-2 using directional waveforms. The hybrid qWPdn-BM3D algorithms demonstrated noise suppression efficiency that is quite competitive with BM3D. Thus, it produces PSNR values higher than or very close to the values BM3D produces. On the other hand, its performance related to the edge and fine structures resolution is much better than the performance of the NLSM-based algorithms, thus, producing significantly higher SSIM values. In almost all the experiments the performance of the cptTP-CTF₆ and DAS-2 algorithms were inferior to that of the qWPdn-BM3D algorithms.

Contribution of the paper:

- Development of fast image denoising algorithm qWPdn based on recently designed directional quasi-analytic wavelet packets.
- Design of the fast iterative hybrid qWPdn-BM3D algorithms, which are highly efficient in capturing edges and fine structures even in severely degraded images.
- Experimental comparison of the hybrid qWPdn-BM3D algorithms performance with the performance of multiple state-of-the-art algorithms, which demonstrated a decisive advantage of the hybrid algorithms in a SSIM sense and visual perception.

The paper is organized as follows: Sect. 2 briefly outlines properties of the qWP transforms in one and two dimensions. Section 3.1 describes the qWPdn algorithm. Section 3.2 presents the combined qWPdn-BM3D algorithms. In the multiple experiments in Sect. 3.3, the performance of these algorithms is compared with the performance of the cptTP-CTF₆, DAS-2, BM3D, BM3D-SAPCA, NCSR and WNNM algorithms. Section 3.4 briefly discusses the relation of the proposed methodology to the recently published Deep Learning denoising methods. Section 4 provides an overview of the achieved results. The Matlab code which counts the directions number of 2D qWPs for different decomposition levels is placed in “Appendix”.

Notation and abbreviations: $N = 2^j$, $\omega \stackrel{\text{def}}{=} e^{2\pi i/N}$ and $\Pi[N]$ is a space of real-valued N -periodic signals. $\Pi[N, N]$ is the space of two-dimensional N -periodic arrays in both vertical and horizontal directions.

The abbreviations WP, dWP and qWP mean wavelet packet, orthonormal spline-based wavelet packet $\psi_{[m],l}^p$ and quasi-analytic wavelet packets $\Psi_{\pm[m],l}^p$, respectively, in a 1D case, and wavelet packet $\psi_{[m],j,l}^p$ and quasi-analytic wavelet packets $\Psi_{\pm[m],l,j}^p$, respectively, in a 2D case.

qWPdn designates the qWP-based image denoising algorithm. qWPdn-BM3D means an hybrid image denoising algorithm combining the qWPdn with the BM3D.

PSNR means Peak Signal-to-Noise ratio in decibels (dB). SSIM means Structural Similarity Index (Wang et al., 2004). BSA stands for Bivariate Shrinkage algorithm (Sendur & Selesnick, 2002a, b).

BM3D stands for *Block-matching and 3D filtering* (Dabov et al., 2007), SAPCA means Shape-Adaptive Principal Component Analysis (Dabov et al., 2009), NCSR means Non-locally Centralized Sparse Representation (Dong et al., 2013), WNNM means Weighted Nuclear Norm Minimization (Gu et al., 2014), cptTP-CTF stands for *Compactly Supported Tensor Product Complex Tight Framelets with Directionality* (Zhuang & Han, 2019) and DAS-2 stands for *Digital Affine Shear Filter Transform with 2-Layer Structure* (Che & Zhuang, 2018).

2 Preliminaries: quasi-analytic directional wavelet packets

Recently we designed a family of wavelet packets (qWPs), which possess a collection of properties indispensable for image processing. A brief outline of the qWPs design and the implementation of corresponding transforms is provided in the paper Averbuch et al. (2021), which describes successful application of qWPs to image inpainting. A detailed description of the design and implementation is given in Averbuch et al. (2020). In this section we list properties of qWPs and present some illustrations.

2.1 Properties of qWPs

One-dimensional qWPs The qWPs are derived from the periodic WPs originating from orthonormal discretized polynomial splines of different orders (dWPs), which are described in Chapter 4 in Averbuch et al. (2019) (a brief outline is given in Averbuch et al. 2020). The dWPs are denoted by $\psi_{[m],l}^p$, where p means the generating spline's order, m is the decomposition level and $l = 0, \dots, 2^m - 1$, is the index of an m -level wavelet packets. The 2^m -sample shifts $\left\{ \psi_{[m],l}^p(\cdot - 2^m k) \right\}$, $l = 0, \dots, 2^m - 1$, $k = 0, \dots, N/2^m - 1$, of the m -level dWPs form an orthonormal basis of the space $\Pi[N]$ of N -periodic discrete-time signals. Surely, other orthonormal bases using the wavelet packets $\psi_{[m],l}^p$ are possible, for example, wavelet and Best bases (Coifman & Wickerhauser, 1992).

The waveforms $\psi_{[m],l}^p[k]$ are symmetric, well localized in the spatial domain and have oscillatory structure, their DFT spectra form a refined split of the frequency domain. The shapes of magnitude spectra tend to rectangular as the spline's order p grows. A common way to extend 1D WP transforms to multiple dimensions is by the tensor-product extension. The 2D dWPs from the level m are: $\psi_{[m],j,l}^p[k, n] \stackrel{\text{def}}{=} \psi_{[m],j}^p[k] \psi_{[m],l}^p[n]$. Their 2^m -sample shifts along vertical and horizontal directions form orthonormal bases of the space $\Pi[N, N]$ of 2D signals N -periodic in both directions. The drawback is the lack of directionality. The directionality can be achieved by switching to complex wavelet packets.

For this, we start with application of the Hilbert transform (HT) to the dWPs $\psi_{[m],l}^p$, thus getting the signals $\tau_{[m],l}^p = H(\psi_{[m],l}^p)$, $m = 1, \dots, M$, $l = 0, \dots, 2^m - 1$. A slight correction of those signals' DFT:

$$\hat{\phi}_{[m],l}^p[n] \stackrel{\text{def}}{=} \hat{\psi}_{[m],l}^p[n] + \hat{\psi}_{[m],l}^p[N/2] + \hat{\tau}_{[m],l}^p[n]. \quad (2.1)$$

provides us with a set of signals from the space $\Pi[N]$, whose properties are similar to the properties of the dWPs $\psi_{[m],l}^p$. In particular, their shifts form orthonormal bases in $\Pi[N]$ and the magnitude spectra of $\hat{\phi}_{[m],l}^p$ coincide with the magnitude spectra of the dWPs $\psi_{[m],l}^p$. However, unlike the symmetric dWPs $\psi_{[m],l}^p$, the signals $\hat{\phi}_{[m],l}^p$ are antisymmetric for all l except for $l_0 = 0$ and $l_m = 2^m - 1$. We refer to the signals $\hat{\phi}_{[m],l}^p$ as the complementary orthonormal WPs (cWPs).

The sets of complex-valued WPs, which we refer to as the quasi-analytic wavelet packets (qWP), are defined as $\Psi_{\pm[m],l}^p \stackrel{\text{def}}{=} \psi_{[m],l}^p \pm i \hat{\phi}_{[m],l}^p$, $m = 1, \dots, M$, $l = 0, \dots, 2^m - 1$, where $\hat{\phi}_{[m],l}^p$ are the cWPs defined by Eq. (2.1). The qWPs $\Psi_{\pm[m],l}^p$ differ from the analytic WPs by adding two values $\pm i \hat{\psi}_{[m],l}^p[0]$ and $\pm i \hat{\psi}_{[m],l}^p[N/2]$ into their DFT spectra, respectively. The DFT spectra of the qWPs $\Psi_{+[-m],l}^p$ are located within positive half-band of the frequency domain and vice versa for the qWPs $\Psi_{-[-m],l}^p$.

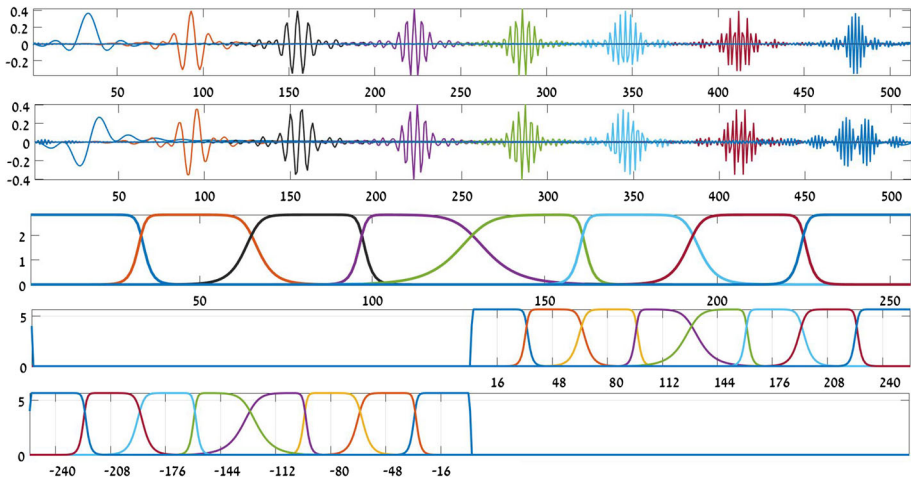


Fig. 1 Top to bottom: signals $\psi_{[3],l}^9$; signals $\phi_{[3],l}^9$, $l = 0, \dots, 7$; their magnitude DFT spectrum (right half-band); magnitude DFT spectra of complex qWPs $\Psi_{+[3],l}^9$; same for $\Psi_{-[3],l}^9$, $l = 0, \dots, 7$

Figure 1 displays the signals $\psi_{[3],l}^9$ and $\phi_{[3],l}^9$, $l = 0, \dots, 7$ derived from ninth-order splines, from the third decomposition level and their magnitude spectra (right half-band), that coincide with each other. Addition of $\hat{\psi}_{[3],l}^9[0]$ and $\hat{\psi}_{[3],l}^9[N/2]$ to the spectra of $\phi_{[3],l}^9$, $l = 0, 7$, results in an antisymmetry distortion. These WPs provide a collection of diverse symmetric and antisymmetric well localized waveforms, which range from smooth wavelets for $l = 0, 1$ to fast oscillating transients for $l = 5, 6, 7$. Thus, this collection is well suited to catching smooth as well as oscillating local patterns in signals.

In the 2D case, these valuable properties of the spline-based wavelet packets are completed by the directionality of the tensor-product waveforms.

Two-dimensional qWPs Similarly to the 2D dWPs $\psi_{[m],j,l}^p[k, n]$, the 2D cWPs are defined as the tensor products of 1D WPs such that $\phi_{[m],j,l}^p[k, n] \stackrel{\text{def}}{=} \phi_{[m],j}^p[k] \phi_{[m],l}^p[n]$. The 2^m -sample shifts of the WPs $\{\phi_{[m],j,l}^p\}$, $j, l = 0, \dots, 2^m - 1$, in both directions form an orthonormal basis for the space $\Pi[N, N]$ of arrays that are N -periodic in both directions.

2D qWPs and their spectra The 2D dWPs $\{\psi_{[m],j,l}^p\}$ as well as the cWPs $\{\phi_{[m],j,l}^p\}$ lack the directionality property which is needed in many applications that process 2D data. However, real-valued 2D wavelet packets oriented in multiple directions can be derived from tensor products of complex quasi-analytic qWPs $\Psi_{\pm[m],\rho}^p$. The complex 2D qWPs are defined as follows:

$$\begin{aligned} \Psi_{++[m],j,l}^p[k, n] &\stackrel{\text{def}}{=} \Psi_{+[m],j}^p[k] \Psi_{+[m],l}^p[n], \\ \Psi_{+-[m],j,l}^p[k, n] &\stackrel{\text{def}}{=} \Psi_{+[m],j}^p[k] \Psi_{-[m],l}^p[n], \end{aligned}$$

where $m = 1, \dots, M$, $j, l = 0, \dots, 2^m - 1$, and $k, n = 0, \dots, N - 1$. The real parts of these 2D qWPs are

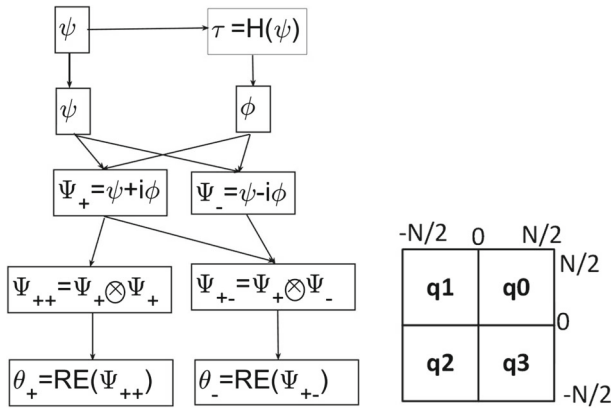


Fig. 2 Diagram of the qWP design (left) and quadrants of frequency domain (right)

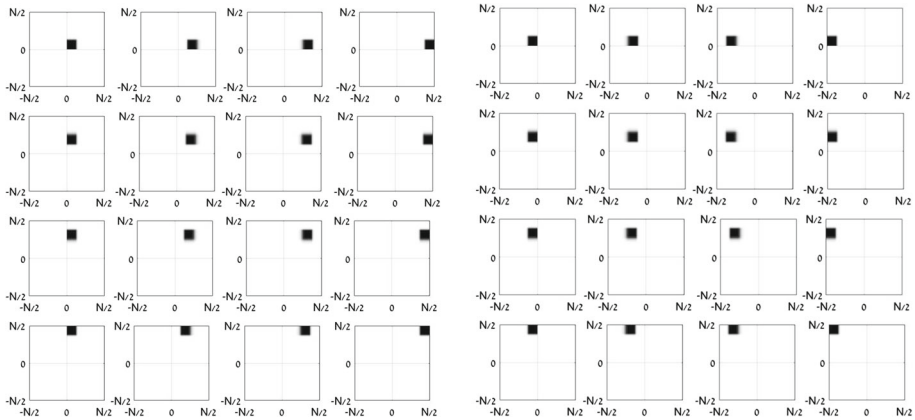


Fig. 3 Magnitude spectra of 2D qWPs $\Psi_{++[2],j,l}^9$ (left) and $\Psi_{+-[2],j,l}^9$ (right) from the second decomposition level

$$\theta_{+[m],j,l}^p[k,n] \stackrel{\text{def}}{=} \Re(\Psi_{+[m],j,l}^p[k,n]) = \psi_{[m],j,l}^p[k,n] - \phi_{[m],j,l}^p[k,n],$$

$$\theta_{-[m],j,l}^p[k,n] \stackrel{\text{def}}{=} \Re(\Psi_{-[m],j,l}^p[k,n]) = \psi_{[m],j,l}^p[k,n] + \phi_{[m],j,l}^p[k,n],$$

The diagram in Fig. 2 illustrates the design of qWPs. The DFT spectra of the 2D qWPs $\Psi_{++[m],j,l}^p$, $j, l = 0, \dots, 2^m - 1$, are tensor products of the one-sided spectra of the qWPs $\hat{\Psi}_{++[m],j,l}^p[p,q] = \hat{\Psi}_{+[m],j}^p[p] \hat{\Psi}_{+[m],l}^p[q]$ and, as such, they fill the quadrant q_0 of the frequency domain, while the spectra of $\Psi_{+-[m],j,l}^p$, $j, l = 0, \dots, 2^m - 1$, fill the quadrant q_1 (see Fig. 2). Figure 3 displays magnitude spectra of the ninth-order 2D qWPs $\Psi_{++[2],j,l}^9$ and $\Psi_{+-[2],j,l}^9$ from the second decomposition level. Figure 3 shows that the DFT spectra of the qWPs $\Psi_{\pm[m],j,l}^p$ effectively occupy relatively small squares in the frequency domain. For deeper decomposition levels, sizes of the corresponding squares decrease as geometric progression. Such configuration of the spectra leads to the directionality of the real-valued 2D WPs $\theta_{\pm[m],j,l}^p$. The directionality of the WPs $\theta_{\pm[m],j,l}^p$ is discussed in Averbuch et al. (2020). It is established that if the spectrum of a WP $\Psi_{\pm[m],j,l}^p$ occupies a square whose

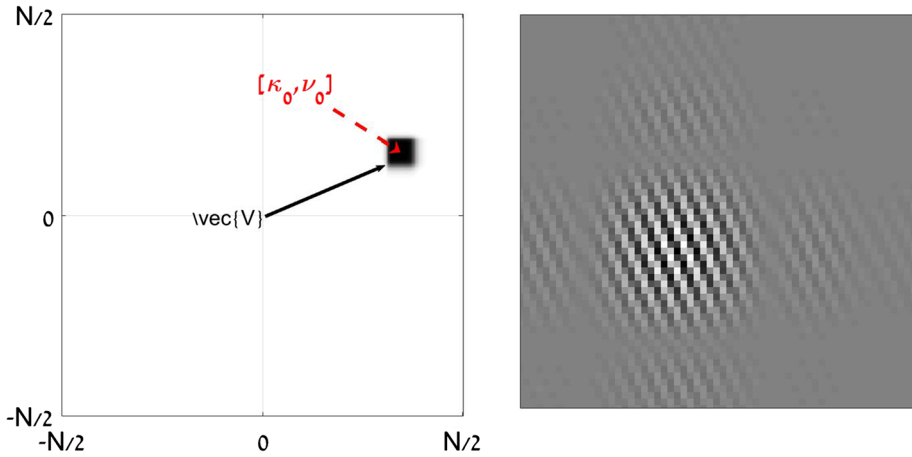


Fig. 4 Left: magnitude spectra of 2D qWP $\Psi_{++[3],2,5}^p[k, n]$. Right WP $\theta_{++[3],2,5}^p = \Re(\Psi_{++[3],2,5}^p)$

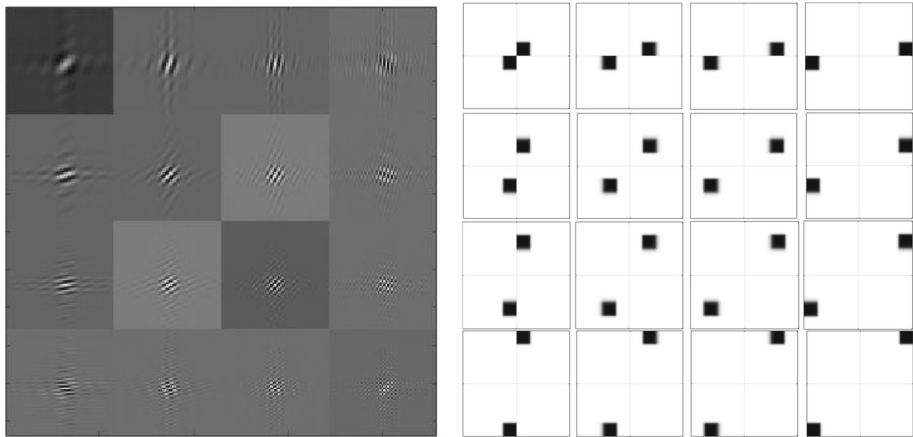


Fig. 5 WPs $\theta_{+[2],j,l}^9$ from the second decomposition level and their magnitude spectra

center lies in the point $[\kappa_0, \nu_0]$, then the respective real-valued WP $\theta_{\pm[m],j,l}^p$ is represented by $\theta_{\pm[m],j,l}^p[k, n] \approx \cos \frac{2\pi(\kappa_0 k + \nu_0 n)}{N} \underline{\theta}[k, n]$, where $\underline{\theta}[k, n]$ is a spatially localized low-frequency waveform which does not have a directionality. But the 2D signal $\cos \frac{2\pi(\kappa_0 k + \nu_0 n)}{N}$ is oscillating in the direction $\mathbf{D}$, which is orthogonal to the vector $\vec{V} = \kappa_0 \vec{i} + \nu_0 \vec{j}$. Therefore, WP $\theta_{\pm[m],j,l}^p$ can be regarded as the directional cosine modulated by the localized low-frequency signal $\underline{\theta}$. The cosine frequencies in the vertical and horizontal directions are determined by the indices j and l , respectively, of the WP $\theta_{\pm[m],j,l}^p$. The bigger is the index, the higher is frequency in the respective direction. The situation is illustrated in Fig. 4. The imaginary parts of the qWPs $\Psi_{\pm[m],j,l}^p$ have a similar structure.

Figure 5 displays WPs $\theta_{+[2],j,l}^9$, $j, l = 0, 1, 2, 3$, from the second decomposition level and their magnitude spectra.

Figure 6 displays WPs $\theta_{-[2],j,l}^9$, $j, l = 0, \dots, 7$, from the second decomposition level and their magnitude spectra.

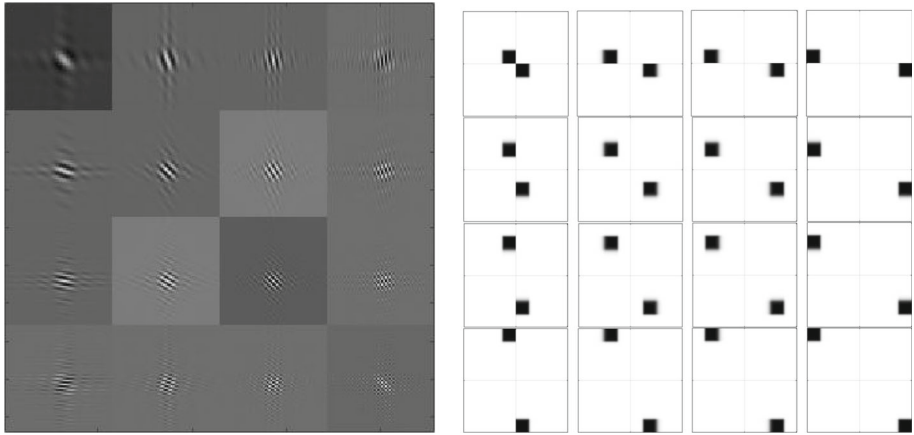


Fig. 6 WPs $\theta_{-[2],j,l}^9$ from the second decomposition level and their magnitude spectra

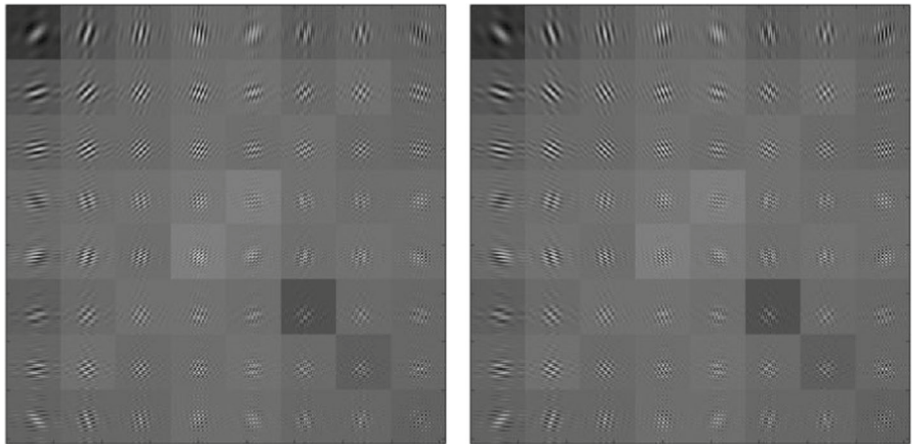


Fig. 7 WPs $\theta_{+[3],j,l}^9$ (left) and $\theta_{-[3],j,l}^9$ (right) from the third decomposition level

Figure 7 displays WPs $\theta_{+[3],j,l}^9$ and $\theta_{-[3],j,l}^9$, $j, l = 0, 1, 2, 3$, from the third decomposition level.

Remark 2.1 Note that all the WPs $\theta_{+[m],j,l}^p$ whose spectra are located along a vector $\vec{V}$ have approximately the same orientation. It is seen in Figs. 6, 5 and 7. Consequently, the number of different orientations of the m -th level WPs is less than the number of WPs, which is $2 \cdot 4^m$. For example, all the “diagonal” qWPs $\{\theta_{\pm[m],j,j}^p\}$, $j = 0, \dots, 2^m - 1$, are oscillating with different frequencies in the directions of either 135° (for θ_+) or 45° (for θ_-). These orientation numbers are given in Table 1. The Matlab code for the calculation of these numbers is contained in “Appendix”.

Table 1 Numbers of different orientations of qWPs

$\{\theta_{\pm[m],j,l}^p\}$, $j, l = 0, \dots, 2^m - 1$, for different decomposition levels

Level m	1	2	3	4	5	6	...
# of directions	6	22	86	318	1290	5030	...

2.2 Outline of the implementation scheme for 2D qWP transforms

The spectra of 2D qWPs $\{\Psi_{++[m],j,l}^p\}$, $j, l = 0, \dots, 2^m - 1$ fill the quadrant $\mathbf{q}_0$ of the frequency domain (see Fig. 2), while the spectra of 2D qWPs $\{\Psi_{+-[m],j,l}^p\}$ fill the quadrant $\mathbf{q}_1$. Consequently, the spectra of the real-valued 2D WPs $\{\theta_{+[m],j,l}^p\}$, $j, l = 0, \dots, 2^m - 1$, and $\{\theta_{-[m],j,l}^p\}$ fill the pairs of quadrant $\mathbf{q}_+ = \mathbf{q}_0 \cup \mathbf{q}_2$ and $\mathbf{q}_- = \mathbf{q}_1 \cup \mathbf{q}_3$, respectively.

By this reason, none linear combination of the WPs $\{\theta_{+[m],j,l}^p\}$ and their shifts can serve as a basis in the signal space $\Pi[N, N]$. The same is true for WPs $\{\theta_{-[m],j,l}^p\}$. However, combinations of the WPs $\{\theta_{\pm[m],j,l}^p\}$ provide frames of the space $\Pi[N, N]$.

The transforms are implemented in the frequency domain using modulation matrices of the filter banks, which are built from the corresponding wavelet packets. It is important to mention that the structure of the filter banks $\mathbf{Q}_+$ and $\mathbf{Q}_-$ for the first decomposition level is different for the transforms with the “positive” $\Psi_{+[m],l}^p$ and “negative” $\Psi_{-[m],l}^p$ qWPs, respectively. However, the transforms from the first to the second and further decomposition levels are executed using the same filter bank $\mathbf{H}_m$ for the “positive” and “negative” qWPs. This fact makes it possible a parallel implementation of the transforms.

The one-level 2D qWP transforms of a signal $\mathbf{X} = \{X[k, n]\} \in \Pi[N, N]$ are implemented by a tensor-product scheme. To be specific, for the transform with $\Psi_{++[1]}^p$, the 1D transform of columns from the signal $\mathbf{X}$ is executed using the filter bank $\mathbf{Q}_+$, which is followed by the 1D transform of rows of the produced coefficient arrays using the same filter bank $\mathbf{Q}_+$. These operation results in the transform coefficient array $\mathbf{Z}_{+[1]} = \bigcup_{j,l=0}^1 \mathbf{Z}_{+[1]}^{j,l}$ comprising of four blocks.

The transform with $\Psi_{+-[1]}^p$ is implemented by the subsequent application of the filter banks $\mathbf{Q}_+$ and $\mathbf{Q}_-$ to columns from the signal $\mathbf{X}$ and rows of the produced coefficient arrays, respectively. This results in the coefficient array $\mathbf{Z}_{-[1]} = \bigcup_{j,l=0}^1 \mathbf{Z}_{-[1]}^{j,l}$.

The further transforms starting from the arrays $\mathbf{Z}_{+[1]}$ and $\mathbf{Z}_{-[1]}$ produce two sets of the coefficients $\{\mathbf{Z}_{+[m]} = \bigcup_{j,l=0}^{2^m-1} \mathbf{Z}_{+[m]}^{j,l}\}$ and $\{\mathbf{Z}_{-[m]} = \bigcup_{j,l=0}^{2^m-1} \mathbf{Z}_{-[m]}^{j,l}\}$, $m = 2, \dots, M$. The transforms are implemented by application of the same filter banks $\mathbf{H}_m$, $m = 2, \dots, M$ to rows and columns of the “positive” and “negative” coefficient arrays. The coefficients from a level m comprise of 4^m “positive” blocks of coefficients $\{\mathbf{Z}_{+[m]}^{j,l}\}$, $j, l = 0, \dots, 2^{m-1}$, and the same number of “negative” blocks $\{\mathbf{Z}_{-[m]}^{j,l}\}$.

Coefficients from a block are inner products of the signal $\mathbf{X} = \{X[k, n]\} \in \Pi[N, N]$ with the shifts of the corresponding wavelet packet:

$$Z_{\pm[m]}^{j,l}[k, n] = \sum_{\lambda, \mu=0}^{N-1} X[\lambda, \mu] \Psi_{\pm[m],j,l}^p[\lambda - 2^m k, \mu - 2^m n],$$

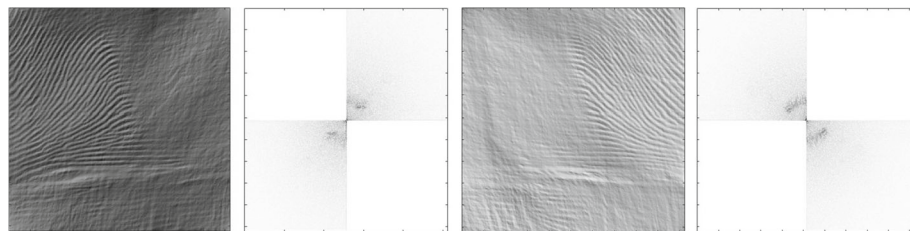


Fig. 8 Left to right: 1. Image $\Re(\mathbf{X}_+)$. 2. Its magnitude DFT spectrum. 3. Image $\Re(\mathbf{X}_-)$. 4. Its magnitude DFT spectrum

$$\begin{aligned} Y_{\pm[m]}^{j,l}[k, n] &\stackrel{\text{def}}{=} \Re(Z_{\pm[m]}^{j,l}[k, n]) \\ &= \sum_{\lambda, \mu=0}^{N-1} X[\lambda, \mu] \theta_{\pm[m], j, l}^p[\lambda - 2^m k, \mu - 2^m n]. \end{aligned} \quad (2.2)$$

The inverse transforms are implemented accordingly. Prior to the reconstruction, some structures, possibly different, are defined in the sets $\{\mathbf{Z}_{+[m]}^{j,l}\}$ and $\{\mathbf{Z}_{-[m]}^{j,l}\}$, $m = 1, \dots, M$, (for example, 2D wavelet, Best Basis or single-level structures) and some manipulations on the coefficients, (for example, thresholding, shrinkage, l_1 minimization) are executed. The reconstruction produces two complex arrays $\mathbf{X}_+$ and $\mathbf{X}_-$. The signal $\mathbf{X}$ is restored by $\tilde{\mathbf{X}} = \Re(\mathbf{X}_+ + \mathbf{X}_-)/8$.

Figure 8 illustrates the image “Fingerprint” restoration by the 2D signals $\Re(\mathbf{X}_{\pm})$. The signal $\Re(\mathbf{X}_-)$ captures oscillations oriented to *north-east*, while $\Re(\mathbf{X}_+)$ captures oscillations oriented to *north-west*. The signal $\tilde{\mathbf{X}} = \Re(\mathbf{X}_+ + \mathbf{X}_-)/8$ perfectly restores the image achieving PSNR = 312.3538 dB.

3 Image denoising

In this section, we describe application of the directional qWP transforms presented in Sect. 2 to the restoration of an image $\mathbf{X}$ from the data $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}$, where $\mathbf{E}$ is the Gaussian zero-mean noise whose STD = σ .

3.1 Denoising scheme for 2D qWPs

The degraded image $\tilde{\mathbf{X}}$ is decomposed into two sets $\{\tilde{\mathbf{Z}}_{+[m]}^{j,l}\}$ and $\{\tilde{\mathbf{Z}}_{-[m]}^{j,l}\}$, $m = 1, \dots, M$, $j, l = 0, 2^m - 1$, of the qWP transform coefficients, then a version of the Bivariate Shrinkage algorithm (BSA) (Sendur & Selesnick, 2002a, b) is implemented and the image $\tilde{\mathbf{X}} \approx \mathbf{X}$ is restored from the shrunken coefficients. The restoration is executed separately from the sets of coefficients belonging to several decomposition levels and the results are averaged with some weights.

3.1.1 Image restoration from a single-level transform coefficients

Consider the estimation of an image $\mathbf{X} \in \Pi[N, N]$ from the *fourth*-level transform coefficients of the degraded array $\tilde{\mathbf{X}}$ of size $N \times N$.

The denoising algorithm, which we refer to as qWpDn, is implemented by the following steps:

1. In order to eliminate boundary effects, the degraded image $\check{\mathbf{X}}$ is symmetrically extended to image $\check{\mathbf{X}}_T$ of size $N_1 \times N_1$, where $N_1 = N + 2T$. Typically, either $T = N/4$ or $T = N/8$.
2. The Bivariate Shrinkage (BSA) utilizes the interscale dependency of the transform coefficients. Therefore, the direct 2D transforms of the image $\check{\mathbf{X}}_T$ with the complex qWPs Ψ_{++}^p and Ψ_{+-}^p are executed down to the *fifth* decomposition level. As a result, two sets $\check{\mathbf{Z}}_{+[m]}^{j,l} = \{\check{\mathbf{Z}}_{+[m]}^{j,l}[k, n]\}$ and $\check{\mathbf{Z}}_{-[m]}^{j,l} = \{\check{\mathbf{Z}}_{-[m]}^{j,l}[k, n]\}$, $m = 1, \dots, 5$, $j, l = 0, \dots, 2^m - 1$, $k, n = 0, \dots, N_1/2^m - 1$, of the qWP transform coefficients are produced.
3. The noise variance is estimated by $\tilde{\sigma}_e^2 = \frac{\text{median}(\text{abs}(\check{\mathbf{Z}}_{+[1]}^{1,l}[k, n]))}{0.6745}$.
4. Let $\check{c}_4[k, n] \stackrel{\text{def}}{=} \check{\mathbf{Z}}_{+[4]}^{j,l}[k, n]$ denote a coefficient from a block $\check{\mathbf{Z}}_{+[4]}^{j,l}$ at the fourth decomposition level. The following operations are applied to the coefficient $\check{c}_4[k, n]$:

- (a) The averaged variance $\bar{\sigma}_c[k, n]^2 = \frac{1}{W_4^2} \sum_{\kappa, v=-W_4/2}^{W_4/2-1} \check{c}_4[k + \kappa, n + v]^2$ is calculated.

The integer W_4 determines the neighborhood of $\check{c}_4[k, n]$ size.

- (b) The marginal variance for $\check{c}_4[k, n]$ is estimated by $\tilde{\sigma}[k, n]^2 = (\bar{\sigma}_c[k, n]^2 - \tilde{\sigma}_e^2)_+^{.2}$.
- (c) In order to estimate the clean transform coefficients from the fourth decomposition level, it is needed to utilize coefficients from the fifth level. The size of the coefficient block $\check{\mathbf{Z}}_{+[4]}^{j,l}$ is $N_1/16 \times N_1/16$. The coefficients from that block are related to the qWP $\Psi_{++[4],j,l}^p$, whose spectrum occupies, approximately, the square $\mathbf{S}_{+[4]}^{j,l}$ of size $N_1/32 \times N_1/32$ within the quadrant $\mathbf{q}_0$ (see Fig. 2). The spectrum's location determines the directionality of the waveform $\Psi_{++[4],j,l}^p$. On the other hand, four coefficient blocks $\{\check{\mathbf{Z}}_{+[5]}^{2j+\iota, 2l+\lambda}\}$, $\iota, \lambda = 0, 1$, of size $N_1/32 \times N_1/32$ are derived by filtering the block $\check{\mathbf{Z}}_{+[4]}^{j,l}$ coefficients followed by downsampling. The coefficients from those blocks are related to the qWPs $\Psi_{++[5],2j+\iota, 2l+\lambda}^p$, whose spectra occupy, approximately, the squares $\mathbf{S}_{+[5]}^{2j+\iota, 2l+\lambda}$ of size $N_1/64 \times N_1/64$, which fill the square $\mathbf{S}_{+[4]}^{j,l}$. Therefore, the orientations of the waveforms $\Psi_{++[5],2j+\iota, 2l+\lambda}^p$ are close to the orientation of $\Psi_{++[4],j,l}^p$. Keeping this in mind, we form the joint fifth-level array $\mathbf{c}_5^{j,l}$ of size $N_1/16 \times N_1/16$ by interleaving the coefficients from the arrays $\{\check{\mathbf{Z}}_{+[5]}^{2j+\iota, 2l+\lambda}\}$. To be specific, the joint array $\mathbf{c}_5^{j,l}$ consists of the quadruples:

$$\mathbf{c}_5^{j,l} = \left\{ \left[\begin{array}{cc} \check{\mathbf{Z}}_{+[5]}^{2j, 2l}[\kappa, v] & \check{\mathbf{Z}}_{+[5]}^{2j, 2l+1}[\kappa, v] \\ \check{\mathbf{Z}}_{+[5]}^{2j+1, 2l}[\kappa, v] & \check{\mathbf{Z}}_{+[5]}^{2j+1, 2l+1}[\kappa, v] \end{array} \right] \right\}, \quad \kappa, v = 0, \dots, N_1/32 - 1.$$

- (d) Let $\check{c}_5[k, n]$ denote a coefficient from the joint array $\mathbf{c}_5^{j,l}$. Then, the transform coefficient $\check{\mathbf{Z}}_{+[4]}^{j,l}[k, n]$ from the fourth decomposition level is estimated by the bivariate shrinkage of the coefficients $\check{\mathbf{Z}}_{+[4]}^{j,l}[k, n]$:

$$\check{\mathbf{Z}}_{+[4]}^{j,l}[k, n] \approx \check{\mathbf{Z}}_{+[4]}^{j,l}[k, n] = \frac{\left(\sqrt{\check{c}_4[k, n]^2 + \check{c}_5[k, n]^2} - \frac{\sqrt{3} \tilde{\sigma}_e^2}{\tilde{\sigma}[k, n]} \right)_+}{\sqrt{\check{c}_4[k, n]^2 + \check{c}_5[k, n]^2}} \check{c}_4[k, n].$$

² $s_+ \stackrel{\text{def}}{=} \max\{s, 0\}$.

5. As a result of the above operations, the fourth-level coefficient array $\tilde{\mathbf{Z}}_{+[4]} = \left\{ \tilde{\mathbf{Z}}_{+[4],j,l} \right\}$, $j, l = 0, \dots, 15$, is estimated, where $\tilde{\mathbf{Z}}_{+[4],j,l} = \left\{ \tilde{Z}_{+[4]}^{j,l}[k, n] \right\}$, $k, n = 0, \dots, N_1/16 - 1$.
6. The inverse qWP transform is applied to the coefficient array $\tilde{\mathbf{Z}}_{+[4]}$ and the result shrinks to the original image size $N \times N$. Thus, the sub-image $\tilde{\mathbf{X}}_+^4$ is obtained.
7. The same operations are applied to $\tilde{\mathbf{Z}}_{-[m]}^{j,l}$, $m = 4, 5$, thus resulting in the sub-image $\tilde{\mathbf{X}}_-^4$.
8. The clean image is estimated by $\mathbf{X} \approx \tilde{\mathbf{X}}^4 = \frac{\Re(\tilde{\mathbf{X}}_+^4 + \tilde{\mathbf{X}}_-^4)}{8}$.

3.1.2 Image restoration from several decomposition levels

More stable estimation of the image $\mathbf{X}$ is derived by the weighted average of several single-level estimations $\left\{ \tilde{\mathbf{X}}^m \right\}$. In most cases, the estimations from the second, third and fourth levels are combined, so that $m = 2, 3, 4$.

The approximated image $\tilde{\mathbf{X}}^3$ is derived from the third-level coefficients $\tilde{\mathbf{Z}}_{\pm[3]}^{j,l}$. The fourth-level coefficients that are needed for the Bivariate Shrinkage of the coefficients $\tilde{\mathbf{Z}}_{\pm[3]}^{j,l}$ are taken from the “cleaned” arrays $\tilde{\mathbf{Z}}_{\pm[4],j,l}$ rather than from the “raw” ones $\tilde{\mathbf{Z}}_{\pm[4],j,l}$.

Similarly, the image $\tilde{\mathbf{X}}^2$, is derived from the coefficient arrays $\tilde{\mathbf{Z}}_{\pm[2]}^{j,l}$ and $\tilde{\mathbf{Z}}_{\pm[3]}^{j,l}$. The final operation is the weighted averaging such as

$$\tilde{\mathbf{X}} = \frac{\alpha_2 \tilde{\mathbf{X}}^2 + \alpha_3 \tilde{\mathbf{X}}^3 + \alpha_4 \tilde{\mathbf{X}}^4}{\alpha_2 + \alpha_3 + \alpha_4}. \quad (3.1)$$

Remark 3.1 Matlab implementation of all the operations needed to transform the degraded array $\tilde{\mathbf{X}}$ of size 512×512 into the estimation $\tilde{\mathbf{X}}$ given by Eq. (3.1) takes 1 s. Note that the noise STD is not a part of the input. It is evaluated as indicated in Item 3.

Remark 3.2 In some cases, restoration from third, fourth and fifth levels is preferable. Then, the degraded array $\tilde{\mathbf{X}}_T$ is decomposed down to the sixth level.

Remark 3.3 The algorithm comprises a number of free parameters which enable a flexible adaptation to the processed object. These parameters are the order “p” of the generating spline, integers W_4 , W_3 and W_2 , which determine the sizes of neighborhoods for the averaged variances calculation, and the weights α_2 , α_3 and α_4 .

Remark 3.4 Fragments of the Matlab functions `denoising_dwt.m` and `bishrink.m` from the websites http://eeweb.poly.edu/iselesni/WaveletSoftware/denoising_dwt.html and <http://eeweb.poly.edu/iselesni/WaveletSoftware/denoise2.html>, respectively, were used as patterns while compiling our own denoising software.

3.2 qWPdn–BM3D: hybrid algorithm

Equation (2.2) imply that the coefficients of the qWP transforms reflect the correlation of the image under processing with the collection of waveforms, which are well localized in the spatial domain and are oscillating in multiple directions with multiple frequencies. By this reason, these transforms are well suited for capturing edges and texture patterns oriented in different directions. Experiments with the qWPdn image denoising demonstrate its ability to restore edges and texture details even from severely degraded images. In most conducted

experiments, the qWPdn provides better resolution of edges and fine structures compared to the most state-of-the-art algorithms including the BM3D algorithm, which is reflected in higher SSIM values. On the other hand, the BM3D algorithm proved to be superior in noise suppression especially in smooth regions in images, thus producing high PSNR values in almost all the experiments. However, some over-smoothing effect on the edges and fine texture persisted with the application of the BM3D algorithm. Especially, this was the case for severely degraded images.

In this section, we propose to combine qWPdn with BM3D algorithms to benefit from the strong features of both algorithms.

Denote by $\mathbf{W}$ and $\mathbf{B}$ the operators of application of the qWPdn, which is described in Sect. 3, and BM3D denoising algorithms, respectively, to a degraded array $\mathbf{A}$: $\mathbf{W}\mathbf{A} = \mathbf{D}_W$ and $\mathbf{B}\mathbf{A} = \mathbf{D}_B$.

Assume that we have an array $\check{\mathbf{X}}^0 = \mathbf{X} + \mathbf{E}$, which represents an image $\mathbf{X}$ degraded by additive Gaussian noise $\mathbf{E}$ whose STD is σ . The denoising processing is implemented along the following boosting scheme.

First step Apply the operators $\mathbf{W}$ and $\mathbf{B}$ to the input array $\check{\mathbf{X}}^0$: $\mathbf{Y}_W^1 = \mathbf{W}\check{\mathbf{X}}^0$ and $\mathbf{Y}_B^1 = \mathbf{B}\check{\mathbf{X}}^0$.
Iterations $i = 1, \dots, I - 1$

1. Form new input arrays $\check{\mathbf{X}}_W^i = \frac{\check{\mathbf{X}}^0 + \mathbf{Y}_W^i}{2}$, $\check{\mathbf{X}}_B^i = \frac{\check{\mathbf{X}}^0 + \mathbf{Y}_B^i}{2}$.
2. Apply the operators $\mathbf{W}$ and $\mathbf{B}$ to the new input arrays: $\mathbf{Y}_W^{i+1} = \mathbf{W}\check{\mathbf{X}}_B^i$, $\mathbf{Y}_B^{i+1} = \mathbf{B}\check{\mathbf{X}}_W^i$.

Estimations of the clean image Three estimations are used:

1. The updated BM3D estimation $\tilde{\mathbf{X}}_{uB} \stackrel{\text{def}}{=} \mathbf{Y}_B^I$ (**upBM3D**).
2. The updated qWPdn estimation $\tilde{\mathbf{X}}_{uW} \stackrel{\text{def}}{=} \mathbf{Y}_W^I$ (**upqWP**).
3. The hybrid estimation $\tilde{\mathbf{X}}_H \stackrel{\text{def}}{=} (\mathbf{Y}_B^I + \mathbf{Y}_W^I)/2$ (**hybrid**).

3.3 Experimental results

In this section, we compare the performance of our denoising schemes designated as **upBM3D**, **upqWP** and **hybrid** on the restoration of degraded images with the performances of the state-of-the-art algorithms such as BM3D (Dabov et al., 2007), BM3D-SAPCA (Dabov et al., 2009), WNNM (Gu et al., 2014), NCSR (Dong et al., 2013), cptTP-CTF₆ (Zhuang & Han, 2019) and DAS-2 (Che & Zhuang, 2018). To produce results for the comparison, we used the software available at the websites http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_software (BM3D and BM3D-SAPCA), <http://staffweb1.cityu.edu.hk/xzhuang7/softs/index.html#bdTPCTF> (cptTP-CTF₆ and DAS-2), https://github.com/csjunxu/WNNM_CVPR2014 (WNNM), and <https://www4.comp.polyu.edu.hk/~cslzhang/NCSR.htm> (NCSR).

The restored images were evaluated by the visual perception, by Peak Signal-to-Noise ratio (PSNR) [see Eq. (3.2)]³ and by the Structural Similarity Index (SSIM) (Wang et al. 2004, `ssim.m` Matlab 2020b function). SSIM measures the structural similarity of small moving windows in two images. It varies from 1 for fully identical windows to -1 for completely dissimilar ones. The global index is the average of local indices. Currently, SSIM

³
$$\text{PSNR}(\mathbf{x}, \tilde{\mathbf{x}}) \stackrel{\text{def}}{=} 10 \log_{10} \left(\frac{K 255^2}{\sum_{k=1}^K (x_k - \tilde{x}_k)^2} \right) \text{ dB.} \quad (3.2)$$



Fig. 9 Clean images: “Lena”, “Boat”, “Hill”, “Barbara”, “Mandrill”, “Bridge”, “Man”, “Fabric”, “Fingerprint” and “Seismic”

is regarded as more informative characteristics of the image quality compared to PSNR and Mean Square Error (MSE) (see discussion in Wang and Bovik 2009).

For the experiments, we used a standard set of benchmark images: “Lena”, “Boat”, “Hill”, “Barbara”, “Mandrill”, “Bridge”, “Man”, “Fabric” and “Fingerprint”. One image that represents a stacked seismic section is designated as “Seismic”. The “clean” images are displayed in Fig. 9.

The images were corrupted by Gaussian zero-mean noise whose STD was $\sigma = 5, 10, 25, 40, 50, 80$ and 100 dB. Then, the BM3D, BM3D-SAPCA, NCSR, WNNM, upBM3D, upqWP, hybrid, cptTP-CT F_6 and DAS-2 denoising algorithms were applied to restore the images. In most experiments the algorithm **upBM3D** performed better than **upqWP**. However, this was not the case in experiments with the “Seismic” image. Therefore, in the “Seismic” block in Table 3 and pictures in Fig. 16 we provide results from experiments with the **upqWP** rather than with **upBM3D** algorithm.

Table 2 summarizes experimental results from the restoration of the “Barbara”, “Boat”, “Fingerprint”, “Lena” and “Mandrill” images corrupted by additive Gaussian noise. PSNR and SSIM values for each experiment are given.

Table 3 summarizes experimental results on the restoration of the “Hill”, “Seismic”, “Fabric”, “Bridge” and “Man” images corrupted by additive Gaussian noise. The PSNR and SSIM values for each experiment are given.

It is seen from Tables 2 and 3 that, while our methods **upBM3D** and **hybrid** produce PSNR values close to those produced by other methods, in the overwhelming majority of cases the SSIM values produced by **upBM3D** and **hybrid** significantly exceed the SSIM from all other methods. Especially it is true for the restoration of images in presence of a strong noise. This observation reflects the fact that directional qWPs have exceptional capabilities to capture fine structures even in severely degraded images.

Diagrams in Fig. 10 illustrate the results reported in Tables 2 and 3. Top frame in Fig. 10 shows PSNR values for restoration of images degraded by Gaussian noise with STD $\sigma = 5, 10, 25, 40, 50, 80$ and 100 dB, by all eight methods. The PSNR values are averaged over ten images participated in the experiments (see Fig. 9). Bottom frame in Fig. 10 does the same for the SSIM values. We can observe that the averaged PSNR values for our methods **upBM3D** and **hybrid** practically coincide with each other and are very close to the values produced by BM3D and NCSR. The averaged PSNR values for the methods BM3D-SAPCA and WNNM exceed the values produced by all other methods. They practically coincide with each other

Table 2 PSNR/SSIM values for restoration of “Barbara”, “Boat”, “Fingerprint”, “Lena” and “Mandrill” images

σ	5	10	25	40	50	80	100
<i>Barbara</i>							
Noised	34.16/0.77	28.14/0.6	20.18/0.35	16.09/0.23	14.16/0.18	10.07/0.09	8.14/0.06
WNNM	38.79 /0.8416	35.5 /0.775	31.25 /0.68	28.76 /0.5985	27.8 /0.5614	25.48 / 0.4474	24.36 /0.3884
SAPCA	38.37/0.8445	35.1/0.7756	31/0.6797	28.68/0.6086	27.5/0.5605	24.3/0.402	23.09/0.339
NCSR	38.37/0.8302	354/0.7674	30.62/0.6579	28.19/0.5687	27/0.5176	24.37/0.3902	23.21/0.3296
cpTP-CTF ₆	37.71/0.8382	34.04/0.763	29.24/0.6358	26.75/0.5375	25.61/0.4833	23.44/0.3633	22.55/0.3092
DAS-2	37.7/0.8438	33.98/0.764	29.4/0.6422	27.09/0.5595	26.01/0.5121	23.72/0.3987	22.63/0.3398
BM3D	38.3/0.8387	34.97/0.7738	30.71/0.6707	27.98/0.5788	27.23/0.5389	24.79/0.4192	23.63/0.3608
upBM3D	38.36/ 0.8562	34.85/0.7846	30.62/ 0.6807	28.41/0.6075	27.29/ 0.5623	24.78/0.4326	23.98/ 0.3952
Hybrid	38.44/0.8543	34.99/ 0.7851	30.64/0.679	28.33/0.6031	27.25/0.5593	24.88/0.4461	23.9/0.3936
<i>Boat</i>							
Noised	34.16/0.79	28.14/0.58	20.18/0.29	16.09/0.18	14.16/0.13	10.07/0.07	8.14/0.05
WNNM	37.37/0.8172	34.08/0.6835	30.04 /0.5254	27.96 /0.4303	26.98 /0.3807	25.02 /0.2811	24.1 /0.2349
SAPCA	37.51 /0.8288	34.1 /0.6925	30.03/0.5282	27.92/0.4384	26.89/0.3931	24.68/0.2993	23.69/0.2609
NCSR	37.33/0.8101	33.92/0.6848	29.78/0.5053	27.64/0.4003	26.4/0.3551	24.61/0.257	23.69/0.217
cpTP-CTF ₆	36.86/0.8044	33.34/0.6713	29.12/0.504	27.02/0.4044	26.09/0.3567	24.27/0.2626	23.48/0.2232
DAS-2	36.85/0.8294	33.15/0.688	28.88/0.5182	26.78/0.4228	25.81/0.3763	23.84/0.2792	22.9/0.2361
BM3D	37.29/0.8065	33.91/0.6805	29.9/0.5296	27.74/0.4395	26.77/0.3899	24.87/0.2952	23.96/0.2533
upBM3D	37.23/ 0.8322	33.86/0.706	29.8/0.5505	27.79/0.4618	26.64/ 0.4223	24.61/0.3293	23.88/ 0.2862
Hybrid	37.13/0.8193	33.73/ 0.7105	29.56/ 0.5525	27.58/ 0.4633	26.48/0.4217	24.56/ 0.33	23.78/0.2862
<i>Fingerprint</i>							
Noised	34.16/0.97	28.14/0.91	20.18/0.67	16.09/0.48	14.16/0.38	10.07/0.21	8.14/0.15
WNNM	36.95 / 0.9864	32.81 / 0.9655	27.9 /0.8991	25.3 /0.8367	24.66 /0.8065	22.75 /0.7238	21.81 /0.672
SAPCA	36.66/0.9859	32.64/0.9648	27.8/0.8981	25.54/0.839	24.53/0.8057	22.3/0.7068	21.18/0.6449
NCSR	36.8/0.9860	32.69/0.9645	27.83/0.8968	25.52/0.8290	24.48/0.7894	22.37/0.6869	21.4/0.6295

Table 2 continued

σ	5	10	25	40	50	80	100
cpTP-CTF ₆	36/0.9845	32.23/0.9619	27.33/0.889	25.07/0.8219	24.04/0.7815	21.98/0.6746	21.03/0.6129
DAS-2	36.25/0.9843	32.03/0.9601	27.07/0.886	24.86/0.8246	23.88/0.7895	21.87/0.7013	20.96/0.652
BM3D	36.5/0.9854	32.45/0.9634	27.71/0.8955	25.3/0.8334	24.53/0.8019	22.2/0.7165	21.61/0.6643
upBM3D	36.73/0.986	32.5/0.9644	27.7/0.9008	25.43/0.8453	24.53/0.8155	22.5/0.7326	21.64/ 0.7019
Hybrid	36.5/0.9861	32.58/0.9649	27.76/ 0.9013	25.47/ 0.8474	24.5/ 0.8175	22.59/ 0.7341	21.58/0.7015
<i>Lena</i>							
Noised	34.16/0.65	28.14/0.43	20.18/0.2	16.09/0.12	14.16/0.09	10.07/0.04	8.14/0.03
WNNM	38.8/0.7131	36.06/0.6228	32.26 /0.5031	30.1/0.4247	29.24 /0.3935	27.25 /0.3206	26.2 /0.2779
SAPCA	38.85 /0.7174	36.07 /0.6233	32.23/0.5077	30.11 /0.4411	29.07/0.408	26.5/0.314	25.1/0.2771
NCSR	38.73/0.7123	35.85/0.6214	31.92/0.4841	29.91/0.4155	28.9/0.382	26.72/0.3025	25.71/0.2685
cpTP-CTF ₆	38/0.705	35.46/0.616	31.53/0.4992	29.41/0.4313	28.41/0.3965	26.36/0.3199	25.41/0.2833
DAS-2	38.18/0.7272	35.2/0.625	31.09/0.4920	28.9/0.4148	27.84/0.3757	25.53/0.29	24.4/0.2285
BM3D	38.72/0.7078	35.92/0.6233	32.07/0.5026	29.87/0.4265	29.04/0.3957	26.98/0.3214	25.95/0.2851
upBM3D	38.73/ 0.7405	35.7/ 0.6459	31.87/ 0.5185	29.91/0.4433	28.7/0.4096	26.77/0.3303	25.91/0.3005
Hybrid	38.75/0.7348	35.79/ 0.6459	31.87/0.5176	29.77/ 0.4473	28.8/ 0.4137	26.69/ 0.3384	25.82/ 0.306
<i>Mandrill</i>							
Noised	34.16/0.7	28.14/0.52	20.18/0.51	16.09/0.34	14.16/0.26	10.07/0.14	8.14/0.1
WNNM	35.19/0.9232	30.53/0.8197	25.56 /0.6235	23.1 /0.4827	22.64 /0.4145	21.1 /0.2656	20.43 /0.1911
SAPCA	35.2 /0.9281	30.59 /0.8278	25.54/0.6244	23.4/0.4843	22.54/0.413	20.92/0.2649	20.3/0.2048
NCSR	35.07/0.9129	30.38/0.7986	25.36/0.5908	23.18/0.4325	22.35/0.3669	20.82/0.2215	20.23/0.162
cpTP-CTF ₆	35.06/0.9252	30.32/0.8198	25.3/0.6082	23.16/0.4531	22.25/0.371	20.72/0.2148	20.2/0.1617
DAS-2	35.02/ 0.9301	30.24/0.8329	25.24/0.64	23.2/0.51	22.34/0.4423	20.8/0.2997	20.14/0.2376
BM3D	34.98/0.9209	30.34/0.8135	25.27/0.6095	23/0.4613	22.27/0.3813	20.9/0.2439	20.39/0.1956
upBM3D	34.4/0.9221	30.38/ 0.8379	25.38/ 0.6617	23/0.5415	22.35/0.4802	20.85/0.3205	20.38/0.2559
Hybrid	33.2/0.9062	30.04/0.8307	25.42/0.6607	23.25/ 0.5471	22.28/ 0.4861	20.56/ 0.3405	20.04/ 0.2789

Boldface highlights the best results

Table 3 PSNR/SSIM values for restoration of “Hill”, “Seismic”, “Fabric”, “Bridge” and “Man” images

σ	5	10	25	40	50	80	100
<i>Hill</i>							
Noised	34.16/0.8	28.14/0.59	20.18/0.27	16.09/0.15	14.16/0.11	10.07/0.05	8.14/0.03
WNNM	37.24/0.8399	33.78/0.7235	29.96 /0.5266	28.13 /0.4289	27.34 /0.3711	25.62 /0.278	24.76 /0.2317
SAPCA	37.3 /0.8457	33.84 /0.7316	29.95/0.5335	28.09/0.4281	27.2/0.3815	25.18/0.2867	24.29/0.2491
NCSR	37.17/0.8350	33.7/0.7246	29.72/0.5138	27.84/0.3936	26.99/0.3463	25.16/0.2480	24.35/0.2089
cptTP-CTF ₆	36.83/0.8388	33.17/0.7157	29.13/0.4992	27.37/0.3903	26.59/0.3425	25/0.2495	24.29/0.2108
DAS-2	36.68/0.8447	33.02/0.7292	29.15/0.5327	27.31/0.4245	26.47/0.3747	24.62/0.2733	23.7/0.2272
BM3D	37.14/0.8384	33.61/0.7193	29.85/0.5311	27.99/0.4305	27.19/0.3825	25.42/0.2878	24.59/0.2448
upBM3D	37.21/ 0.8526	33.68/0.7474	29.88/0.5625	28.02/0.4642	27.2/0.4161	25.34/0.3207	24.65/0.2761
Hybrid	37.04/0.8510	33.57/0.7498	29.72/ 0.5672	27.88/ 0.466	27.1/ 0.4181	25.38/ 0.3256	24.62/ 0.278
<i>Seismic</i>							
Noised	34.16/0.8	28.14/0.55	20.18/0.22	16.09/0.12	14.16/0.08	10.07/0.04	8.14/0.03
WNNM	39.22/0.9173	34.87/0.765	30.59/0.4473	28.87/0.303	28.28/0.2448	27.07 /0.1552	26.55 /0.1248
SAPCA	39.09/0.9167	35.04/0.7871	30.8/0.5016	29.04/0.3654	28.2/0.3124	26.34/0.23	25.4/0.1934
NCSR	38.97/0.9131	34.86/0.78	30.62/0.4643	29/0.2948	28.27/0.2374	26.81/0.139	26.25/0.1089
cptTP-CTF ₆	38.81/0.9107	34.83/0.7696	30.72/0.4616	29.1/0.3188	28.4/0.2618	26.91/0.1616	26.16/0.127
DAS-2	38.8/0.9141	34.83/0.795	30.67/0.5383	28.87/0.3938	28.05/0.3318	26.22/0.2222	25.29/0.1785
BM3D	39/0.9126	34.9/0.77	30.8/0.4984	29.08/0.3606	28.45/0.2853	26.97/0.19	26.31/0.1547
upqWP	39.54/0.931	35.39/0.83	30.8/ 0.5903	28.83/ 0.4548	27.99/ 0.3947	25.97/ 0.285	25.1/ 0.2382
Hybrid	39.41/0.9283	35.36/0.8228	30.88 /0.5836	28.99 /0.4513	28.19 /0.3916	26.29/0.2847	25.64/0.2381
<i>Fabric</i>							
Noised	34.16/0.8	28.14/0.55	20.18/0.22	16.09/0.12	14.16/0.08	10.07/0.04	8.14/0.03
WNNM	38.75 /0.8332	34.94/0.7333	30.76/0.549	28.83 /0.4478	28.02 /0.4107	26.24 /0.3369	25.31 /0.3002
SAPCA	38.72/ 0.8351	35/0.7432	30.8/0.5678	28.82/ 0.4726	27.89/0.3905	25.54/0.3275	24.51/0.2915
NCSR	38.58/0.8263	34.78/0.7298	30.53/0.5408	28.54/0.4384	27.69/0.3988	25.77/0.3185	24.8/0.2838

Table 3 continued

σ	5	10	25	40	50	80	100
cpTP-CTF ₆	37.99/0.81	34.07/0.6978	29.73/0.5127	27.83/0.4266	26.94/0.3905	25.12/0.3185	24.23/0.2856
DAS-2	37.74/0.8017	33.78/0.6985	29.48/0.5223	27.48/0.4283	26.52/0.3858	24.45/0.3017	23.42/0.2641
BM3D	38.47/0.8303	34.63/0.7262	30.57/0.5476	28.65/0.4548	27.85/0.4182	26/0.3402	25.06/0.3032
upqWP	38.47/0.8329	34.67/0.7363	30.52/0.5577	28.61/0.4674	27.75/0.426	25.78/0.3467	24.92/0.316
Hybrid	38.4/0.8275	34.6/0.7302	30.33/0.5549	28.4/0.4665	27.5/0.4249	25.58/0.3486	24.77/0.3065
<i>Bridge</i>							
Noised	34.16/0.7	28.14/0.52	20.18/0.51	16.09/0.34	14.16/0.26	10.07/0.14	8.14/0.1
WNNM	35.89 /0.9423	31.3/0.8627	26.4/0.6531	24.49/0.505	23.7/0.4385	22.24 /0.3085	21.59/0.2532
SAPCA	35.89 /0.9439	31.35 /0.8671	26.44 /0.6587	24.5 /0.5152	23.7/0.4465	22.17/0.3273	21.48/0.2783
NCSR	35.78/0.9375	31.2/0.8551	26.29/0.6389	24.31/0.4814	23.54/0.4169	22.02/0.2832	21.36/0.2344
cpTP-CTF ₆	35.56/0.9388	30.94/0.8575	26/0.6362	24.01/0.4803	23.22/0.4067	21.78/0.2744	21.17/0.2263
DAS-2	35.44/0.9373	30.76/0.8573	25.91/0.6609	24.03/0.5287	23.22/0.4625	21.66/0.3316	20.94/0.2768
BM3D	35.78/0.9415	31.17/0.8597	26.22/0.641	24.3/0.5007	23.58/0.4286	22.22/0.3168	21.61 /0.2716
upBM3D	35.73/0.9416	31.21/0.8689	26.27/0.6918	24.22/0.5712	23.72 /0.4963	22.12/0.3707	21.53/0.3186
Hybrid	35.42/0.9366	31.15/0.8656	26.24/0.6908	24.04/0.5669	23.47/0.5047	21.85/0.3831	21.2/0.3311
<i>Man</i>							
Noised	34.16/0.75	28.14/0.55	20.18/0.27	16.09/0.16	14.16/0.12	10.07/0.06	8.14/0.04
WNNM	38.05/0.8351	34.26/0.7281	29.83/0.5415	27.83/0.4322	26.98 /0.3831	25.22 /0.2878	24.39 /0.2419
SAPCA	38.09 /0.8387	34.28 /0.7325	29.84 /0.5488	27.86 /0.445	26.97/0.3997	24.92/0.3078	23.9/0.2677
NCSR	37.88/0.8216	34.08/0.7212	29.62/0.5289	27.58/0.4107	26.7/0.3632	24.89/0.2671	24.05/0.2276
cpTP-CTF ₆	37.44/0.8302	33.59/0.7235	29.14/0.5341	27.14/0.4241	26.25/0.3725	23.99/0.2713	23.73/0.229
DAS-2	37.1/0.7525	33.16/0.7184	28.81/0.5374	26.84/0.4326	25.93/0.3815	23.99/0.2783	23.06/0.2336
BM3D	37.85/0.8306	34.02/0.7225	29.65/0.5429	27.68/0.442	26.84/0.3919	25.09/0.3009	22.26/0.2598
upBM3D	35.74/0.8432	33.98/0.7422	29.67/0.5648	27.69/0.4637	26.62/0.4204	24.74/0.3244	24.16/0.2813
Hybrid	35.62/0.8416	33.89 /0.7424	29.54/0.5658	27.51/0.4652	26.46/0.4191	24.67/0.326	23.93/0.2835

Boldface highlights the best results

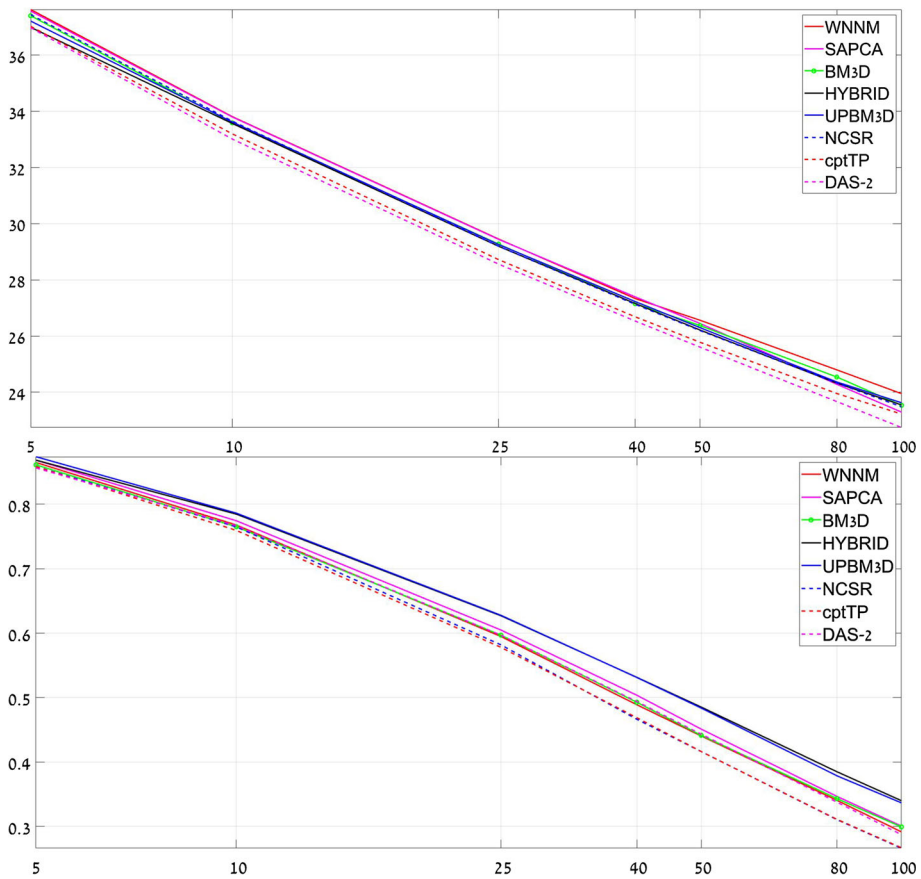


Fig. 10 Diagrams of PSNR (top frame) and SSIM (bottom frame) values averaged over ten images for restoration of the images by eight different methods. Labels on x-axes indicate $\text{STD} = \sigma$ values of the additive noise (logarithmic scale)

for $\sigma \leq 40$ dB, but for $\sigma > 40$ dB, WNNM produces significantly higher PSNR values. The values from the methods cptTP-CTF₆ and DAS-2 are inferior.

A different situation we see in the bottom frame, which displays averaged SSIM values. Again, the averaged values for our methods **upBM3D** and **hybrid** practically coincide with each other but they strongly override the values produced by all other methods. BM3D-SAPCA demonstrates some advantage over the rest of the methods. It is worth mentioning relatively good results by DAS-2, which are better than those by NCSR.

Figures 11, 12, 13, 14, 15 and 16 display results of restoration of several images corrupted by strong Gaussian noise. In all those cases the SSIM values from the **upBM3D** or/and **hybrid** algorithms significantly exceed values from all other algorithms. Respectively, the restoration of the images' fine structure by the **upBM3D** or/and **hybrid** algorithms is much better compared to the restoration by other algorithms.



Fig. 11 Restoration of the “Hill” image corrupted by Gaussian noise with STD $\sigma = 50$ dB. PSNR/SSIM for BM3D—27.19/0.3825; for WNNM—27.34/0.3711; for BM3D-SAPCA—27.2/0.3815; for **upBM3D**—27.2/0.4161; for **hybrid**—27.1/0.4181

Each of the mentioned figures comprises 12 frames, which are arranged in a 4×3 order: $\begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \\ f_{41} & f_{42} & f_{43} \end{pmatrix}$. Here frame f_{11} displays noised image; frame f_{21} —image restored by

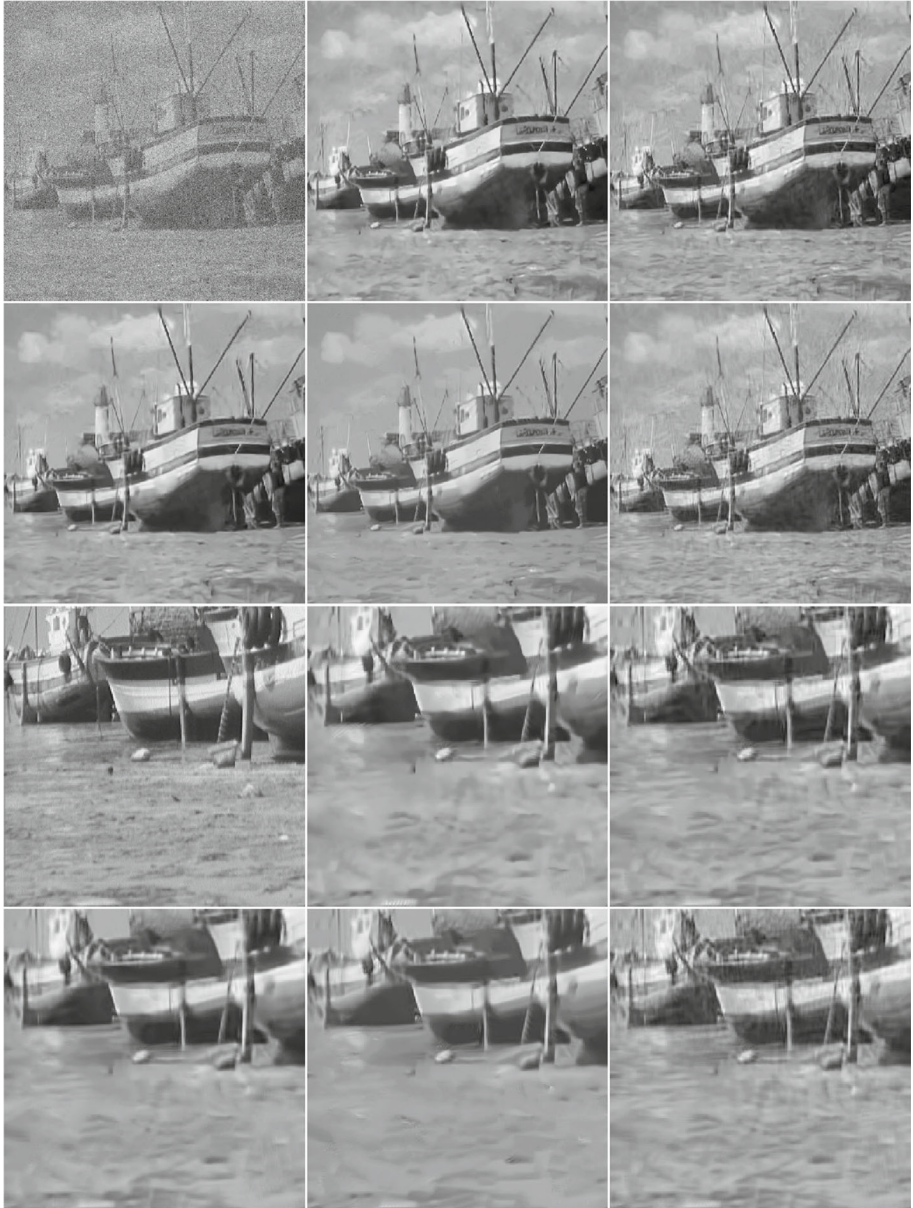


Fig. 12 Restoration of “Boat” image corrupted by Gaussian noise with STD $\sigma = 50$ dB. PSNR/SSIM for BM3D—26.77/0.3899; for WNNM—26.98/0.3807; for BM3D-SAPCA—26.89/0.3931; for **upBM3D**—26.66/0.4223; for **hybrid**—26.49/0.4217

BM3D; f_{12} —image restored by BM3D-SAPCA; f_{22} —image restored by WNNM; f_{13} —image restored by **upBM3D**⁴; f_{23} —image restored by **hybrid**. Frame f_{31} displays a fragment of the original image. The remaining frames $\{f_{32}, f_{33}, f_{41}, f_{42}, f_{43}\}$ display the fragments

⁴ For the “Seismic” image **upqWP** instead of **upBM3D**

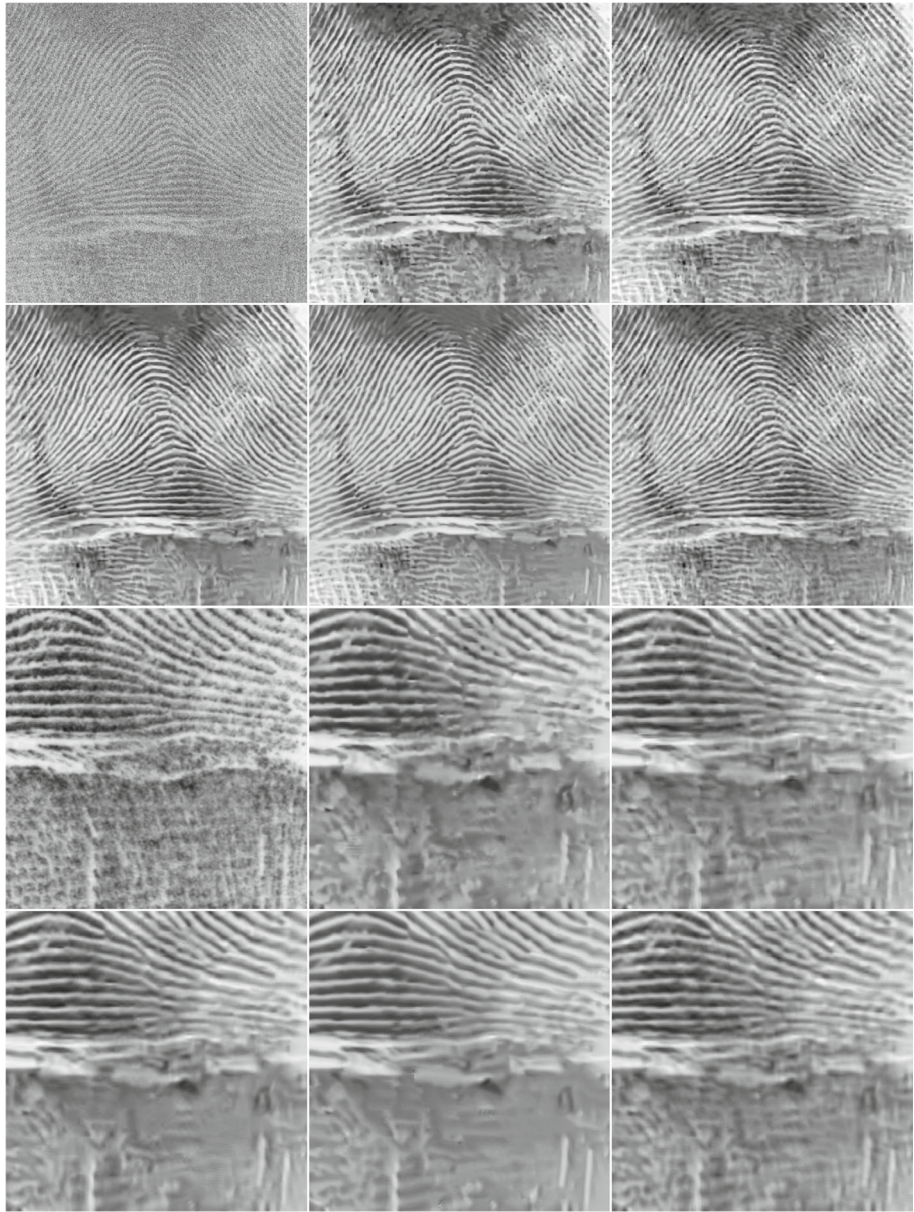


Fig. 13 Restoration of “Fingerprint” image corrupted by Gaussian noise with STD $\sigma = 80$ dB. PSNR/SSIM for BM3D—22.2/0.7165; for WNNM—22.75/0.7238; for BM3D-SAPCA—22.3/0.7068; for upBM3D—22.5/0.7326; for hybrid—22.59/0.7341

of the restored images shown in frames $\{f_{12}, f_{13}, f_{21}, f_{22}, f_{23}\}$, which are arranged in the same order.

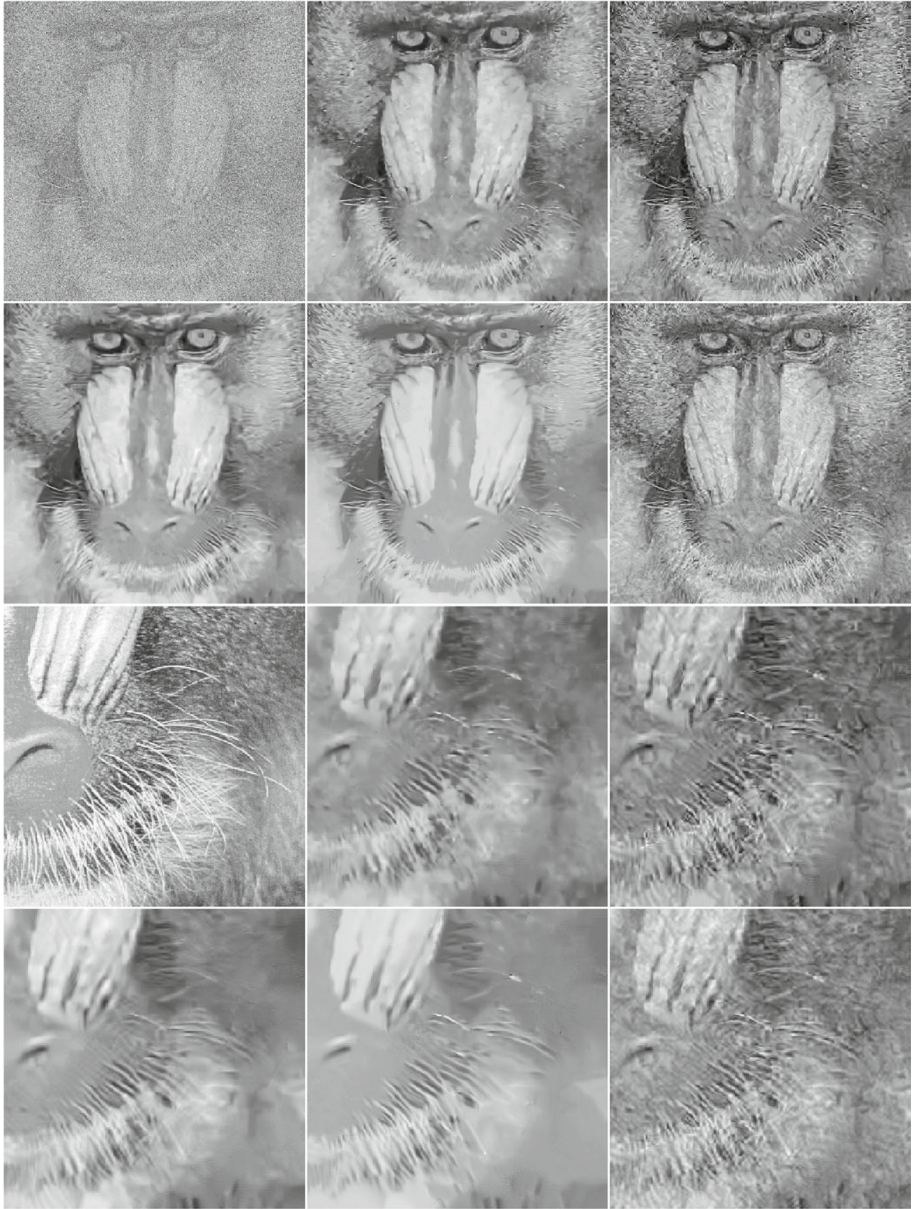


Fig. 14 Restoration of “Mandrill” image corrupted by Gaussian noise with STD $\sigma = 80$ dB. PSNR/SSIM for BM3D—20.9/0.2439; for WNNM—21.1/0.2656; for BM3D-SAPCA—20.92/0.2649; for **upBM3D**—20.85/0.3205; for **hybrid**—20.56/0.3402

3.4 Relation of the proposed algorithms to the deep learning methods

In recent years, the focus of the image denoising research shifted to the Deep Learning methods, which resulted in a huge amount of publications. We mention a few of them: Chen

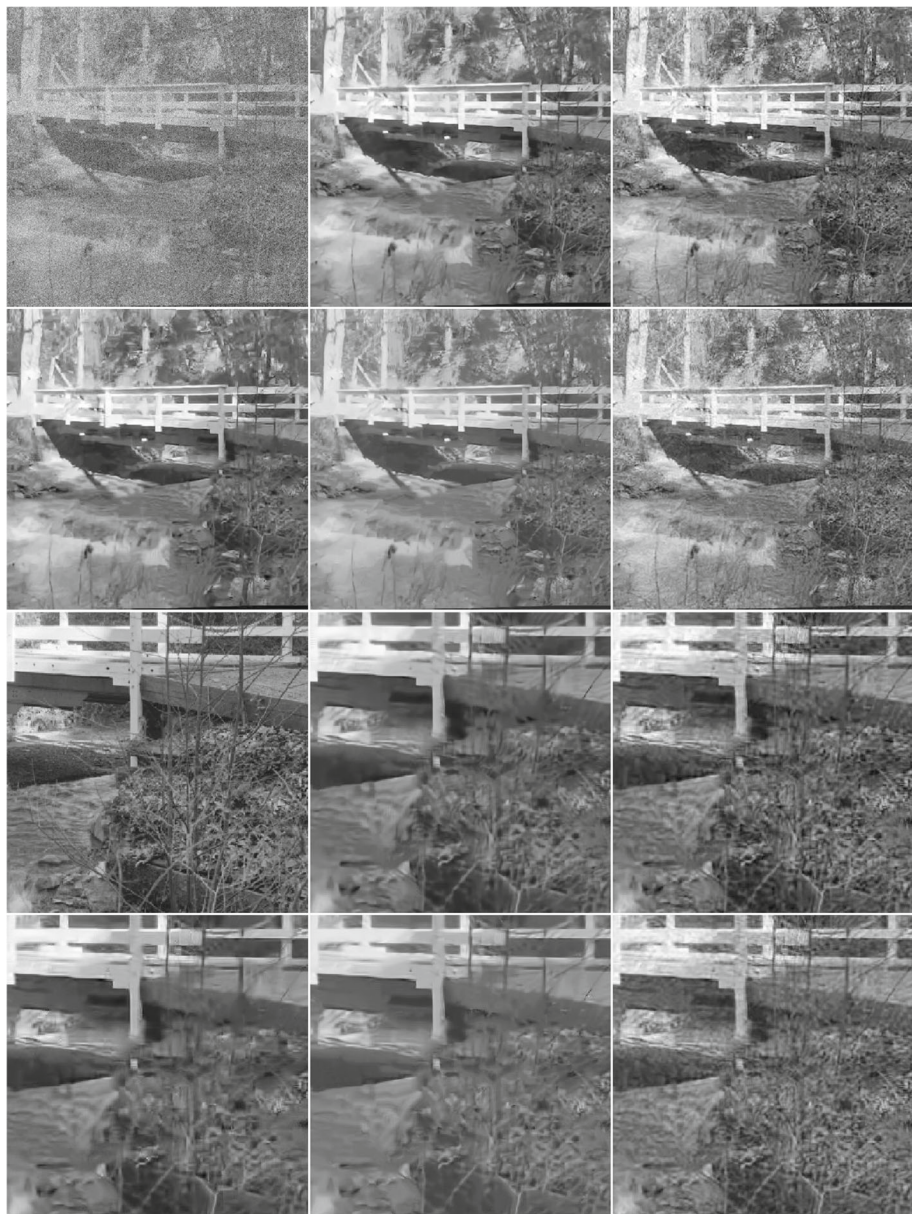


Fig. 15 Restoration of “Bridge” image corrupted by Gaussian noise with STD $\sigma = 40$ dB. PSNR/SSIM for BM3D—24.3/0.5007; for WNNM—24.49/0.505; for BM3D-SAPCA—24.5/0.5152; for **upBM3D**—24.22/0.5712; for **hybrid**—24.04/0.5669

and Pock (2017); Zhang et al. (2017); Cruz et al. (2018); Fang and Zeng (2020); Shi et al. (2019); Li et al. (2020); Quan et al. (2021); Binh et al. (2021). Much more references can be found in the reviews (Tian et al., 2020; Ilesanmi & Ilesanmi, 2021). One of advantages of the Deep Learning methods is that, once the Neural Net is trained (which can involve extended

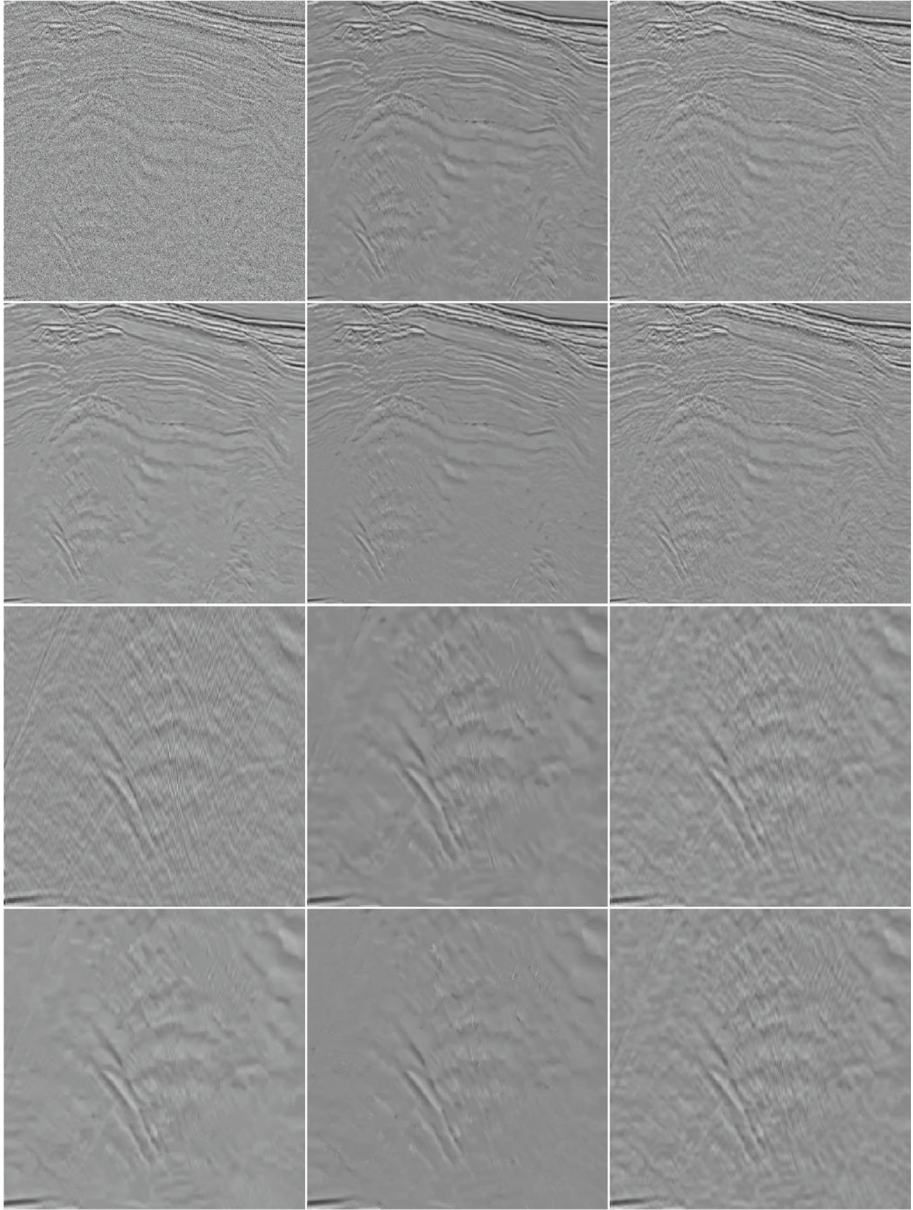


Fig. 16 Restoration of “Seismic” image corrupted by Gaussian noise with STD $\sigma = 25$ dB. PSNR/SSIM for BM3D—30.8/0.4984; for WNNM—30.59/0.4473; for BM3D-SAPCA—30.8/0.5016; for **upqWP**—30.8/0.5903; for **hybrid**—30.88/0.5836

datasets and take several days), its application to test images is very fast. Therefore, the experimental results in most related publications are presented via the PSNR and, sometimes, the SSIM values averaged over some test datasets such as, for example, Set12 introduced in Zhang et al. (2017).

Set12 partially overlap with the set of 10 (Set10) images that we used in our experiments. Namely the images “Barbara”, “Boat”, “Fingerprint”, “Hill”, “Lena” and “Man” participate in both sets. The structure of the remaining images “Seismic”, “Fabric”, “Mandrill” and “Bridge” from Set10 is more complicated compared to the images “Camera”, “Couple”, “House”, “Monarch”, “Pepper” and “Straw” from Set12. Therefore, the averaged results comparison from these two datasets is quite justified. For even better compatibility, we compare the gains of results from different methods over the corresponding results from BM3D: $P_{method} \stackrel{def}{=} \frac{PSNR_{method}}{PSNR_{BM3D}}$ and $S_{method} \stackrel{def}{=} \frac{SSIM_{method}}{SSIM_{BM3D}}$. Recall that for the calculation of SSIM we use the function `ssim` from Matlab 2020b, whereas in most publications SSIM is computed by some other schemes.

We compare results from the recent state-of-the-art algorithms Cola_Net (Mou et al., 2022), CDNet (Quan et al., 2021), FLCNN (Binh et al., 2021), DRCNN (Li et al., 2020), and the non-Deep Learning algorithm presented in Zhou et al. (2021), which we mark as SRENS, averaged over Set12 with the results from WNNM (Gu et al., 2014) and the proposed **hybrid** algorithm averaged over Set10.⁵ The PSNR and SSIM values for all methods except for WNNM and **hybrid** are taken from the corresponding publications. Table 4 shows the results of this comparison.

We can observe from the table that all participated up-to-date schemes, including the non-Deep Learning algorithm SRENS, demonstrate a moderate gain over BM3D in both PSNR and SSIM values averaged over Set12 (far from being a breakthrough). The values from WNNM averaged over Set10 are very close to those from BM3D averaged over the same set. The same can be said for the PSNR values from the **hybrid** algorithm. However, the SSIM values from the **hybrid** algorithm demonstrate a strong gain over BM3D, which is clearly seen in Fig. 10. Especially it is true for a strong Gaussian noise with $\sigma = 80, 100$ dB. This fact highlights the ability of the qWP-based algorithms to restore edges and fine structures even in severely damaged images.

Note that denoising results presented in the overwhelming majority of the Deep Learning publications deal with the noise level not exceeding 50 dB.

4 Discussion

We presented a denoising scheme that combines the qWPDn algorithm based on the directional quasi-analytic wavelet packets, which are designed in Averbuch et al. (2020), with the popular BM3D algorithm (Dabov et al., 2007) considered to be one of the best in the field. The qWPDn and BM3D methods complement each other. Therefore, the idea to combine these methods is natural. In the iterative hybrid scheme qWPDn–BM3D, which is proposed in Sect. 3.2, the output from one algorithm updates the input to the other. The scheme proved to be highly efficient. It is confirmed by a series of experiments on restoration of 10 multimedia images of various structure, which were degraded by Gaussian noise of various intensity. In the experiments, the performance of two combined qWPDn–BM3D algorithms was compared with the performance of six advanced denoising algorithms BM3D, BM3D-SAPCA (Dabov et al., 2009), WNNM (Gu et al., 2014), NCSR (Dong et al., 2013), cptTP-CTF₆ (Zhuang & Han, 2019) and DAS-2 (Che & Zhuang, 2018). In the overwhelming majority of the experiments reported in Sect. 3.3, the two combined algorithms produce PSNR value, which are very close to the values produced by BM3D but give in to the values from WNNM and BM3D-SAPCA. Their noise suppression efficiency is competitive with that of the BM3D.

⁵ Averaged results from **upBM3D** are almost identical to those from **hybrid** algorithm.

Table 4 P_{method}/S_{method} , where the PSNR and SSIM values are averaged over either Set12 or Set10. Noise STD = σ

σ	10	15	25	50	70	80	100
Cola_Net	–	1.028/1.016	1.031/1.025	1.04/1.046v	1.041/1.06	–	–
CDNet	–	1.019/1.016	1.016/1.01	1.025/1.074	1.027/1.057	1.027/1.065	–
FLCNN	–	1.022/1.02	1.017/1.011	1.028/1.039	–	–	–
DRCNN	–	1.016/1.036	1.02/1.033	1.02/1.039	–	–	–
SRENSS	1.044/1.011	–	–	1.02/1.028	1.011/1.028	–	1.006/1.0038
WNNM	1.007/1.004	–	1.006/0.997	1.007/0.998	–	1.01/0.992	1.018/0.974
hybrid	0.999/1.026	–	0.997/1.051	0.994/1.099	–	0.991/1.122	1.001/1.136

On the other hand, their results in the resolution of edges and fine structures are much better than the results from all other algorithms participating in the experiments. This is seen in the images presented in Sect. 3.3. Consequently, the SSIM values produced by the combined algorithms qWpDn–BM3D are significantly higher than the values produced by BM3D and exceed the values produced by all other participated algorithms. The combined algorithm runs fast. In most cases, it is sufficient to conduct two to three iterations (very rarely five to six iterations) in order to get an excellent result. For example, the Matlab implementation of three-iterations processing of a 512×512 image takes about 20 s, where 11 s are consumed by the BM3D (MEX-files) and 3 s are consumed by a non-compiled version of the qWP denoising algorithm. For comparison, processing of the same image by the WMMN algorithm takes 470 s and BM3D-SAPCA processing may take from 3 min to several hours.

The discussion in Sect. 3.4 shows that the qWpDn–BM3D algorithm can compete, in some aspects, with denoising methods based on Deep Learning. Especially it is true for the restoration of images contaminated by strong noise.

It is worth noting that our combined methods have some distant relation to the SOS boosting scheme presented in Romano and Elad (2015). The main distinction between the qWpDn–BM3D and the SOS boosting is that each of the qWpDn and BM3D algorithms is “boosted” by the output from the other algorithm. Such a scheme can be regarded as a *CrossBoosting*.

Summarizing, having such a versatile and flexible tool as the directional qWPs that proved to be highly efficient in image inpainting and denoising, we are in a position to address data processing problems such as deblurring, superresolution, segmentation and classification, target detection (here the directionality is of utmost importance). Another potential application to be addressed is the extraction of characteristic features in multidimensional data in the deep learning framework. The 3D directional wavelet packets, whose design is underway, will be beneficial for seismic and hyper-spectral processing.

Acknowledgements This research was partially supported by the Israel Science Foundation (ISF, 1556/17), Blavatnik Computer Science Research Fund Israel Ministry of Science and Technology 3-13601 and 3-14481.

Appendix: Matlab code for calculation of number of different orientations of real qWPs

```
function D=dir_count(m)
% m -- decomposition level
% D -- number of different orientations of real qWPs from level ``m"
%P(m)--prime numbers;P(3)=2;P(4)=3;P(5)=5;P(6)=7;P(7)=11;...
N=2^m;
M=N/2;
R=N-1;
for kk=1:(M-1)
    for nn=(kk+1):M
        for ll=2:M
            if nn*ll<N+1&&(rem(kk,2)~=0 || rem(nn,2)~=0)...
                &&(rem(kk,3)~=0 || rem(nn,3)~=0)...
                &&(rem(kk,5)~=0 || rem(nn,5)~=0) &&(rem(kk,7)~=0 || rem(nn,7)~=0)
                %&&(rem(kk,11)~=0 || rem(nn,11)~=0)...
                %&&(rem(kk,P(m))~=0 || rem(nn,P(m))~=0)
                    R=R+2;
            end
        end
    end
end
```

```

end
end
end
D=(N^2-R)*2;
end

```

References

- Averbuch, A., Neittaanmäki, P., & Zheludev, V. (2020). Directional wavelet packets originating from polynomial splines. [arXiv:2008.05364](#)
- Averbuch, A., Neittaanmäki, P., Zheludev, V., Salhov, M., & Hauser, J. (2021). Image inpainting using directional wavelet packets originating from polynomial splines. *Signal Processing: Image Communication*, 97. [arXiv:2001.04899](#)
- Averbuch, A., Coifman, R. R., Donoho, D. L., Israeli, M., & Shkolnisky, Y. (2008). A framework for discrete integral transformations I—The pseudopolar Fourier transform. *SIAM Journal on Scientific Computing*, 30(2), 764–784.
- Averbuch, A., Coifman, R. R., Donoho, D. L., Israeli, M., Shkolnisky, Y., & Sedelnikov, I. (2008). A framework for discrete integral transformations II—The 2d discrete Radon transform. *SIAM Journal on Scientific Computing*, 30(2), 785–803.
- Averbuch, A., Neittaanmäki, P., & Zheludev, V. (2019). *Splines and spline wavelet methods with application to signal and image processing, Volume III: Selected topics*. Springer.
- Bayram, I., & Selesnick, I. W. (2008). On the dual-tree complex wavelet packet and m-band transforms. *IEEE Transactions on Signal Processing*, 56, 2298–2310.
- Binh, P. H. T., Cruz, C., & Egiazarian, K. (2021). Flashlight CNN image denoising. In *2020 28th European signal processing conference (EUSIPCO)* (pp. 670–674).
- Buades, A., Coll, B., & Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2), 490–530.
- Candès, E., Demanet, L., Donoho, D., & Ying, L. X. (2006). Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5, 861–899.
- Candès, E., & Donoho, D. (2004). New tight frames of curvelets and optimal representations of objects with piecewise c^2 singularities. *Communications on Pure and Applied Mathematics*, 57, 219–266.
- Chen, Y., & Pock, T. (2017). Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1256–1272.
- Che, Z., & Zhuang, X. (2018). Digital affine shear filter banks with 2-layer structure and their applications in image processing. *IEEE Transactions on Image Processing*, 27(8), 3931–3941.
- Coifman, R. R., & Wickerhauser, V. M. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2), 713–718.
- Cruz, Cristóvão, Foi, Alessandro, Katkovnik, Vladimir, & Egiazarian, Karen. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, 25(8), 1216–1220.
- Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. O. (2009). BM3D image denoising with shape adaptive principal component analysis. In *Proceedings of the workshop on signal processing with adaptive sparse structured representations (SPARS'09)*.
- Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8), 2080–2095.
- Dong, W. S., Zhang, L., Shi, G. M., & Li, X. (2013). Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4), 1620–1630.
- Do, M. N., & Vetterli, M. (2008). Contourlets. In G. V. Welland (Ed.), *Beyond wavelets*. Academic Press.
- Fang, Y., & Zeng, T. (2020). Learning deep edge prior for image denoising. *Computer Vision and Image Understanding*, 200, 103044.
- Goyal, B., Dogra, A., Agrawal, S., Sohi, B. S., & Sharma, A. (2020). Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, 55, 220–244.
- Gu, S., Zhang, L., Zuo, W., & Feng, X. (2014). Weighted nuclear norm minimization with application to image denoising. In *2014 IEEE conference on computer vision and pattern recognition* (pp. 2862–2869).
- Han, B., Mo, Q., Zhao, Z., & Zhuang, X. (2019). Directional compactly supported tensor product complex tight framelets with applications to image denoising and inpainting. *SIAM Journal on Imaging Sciences*, 12(4), 1739–1771.
- Han, B., & Zhao, Z. (2014). Tensor product complex tight framelets with increasing directionality. *SIAM Journal on Imaging Sciences*, 7(2), 997–1034.

- Han, B., Zhao, Z., & Zhuang, X. (2016). Directional tensor product complex tight framelets with low redundancy. *Applied and Computational Harmonic Analysis*, 41(2), 603–637.
- Hou, Y., & Shen, D. (2018). Image denoising with morphology- and size-adaptive block-matching transform domain filtering. *EURASIP Journal on Image and Video Processing*, 59.
- Ilesanmi, A. E., & Ilesanmi, T. O. (2021). Methods for image denoising using convolutional neural network: A review. *Complex & Intelligent Systems*, 7, 2179–2198.
- Jalobeanu, A., Blanc-Féraud, L., & Zerubia, J. (2000). Satellite image deconvolution using complex wavelet packets. In *Proc. IEEE Int. Conf. Image Process. (ICIP)* (pp. 809–812).
- Ji, H., Shen, Z., & Zhao, Y. (2017). Directional frames for image recovery: Multi-scale discrete Gabor frames. *Journal of Fourier Analysis and Applications*, 23(4), 729–757.
- Ji, H., Shen, Z., & Zhao, Y. (2018). Digital Gabor filters with MRA structure. *SIAM Journal on Multiscale Modeling and Simulation*, 16(1), 452–476.
- Kingsbury, N. G. (1999). Image processing with complex wavelets. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 357(1760), 2543–2560.
- Kutyniok, G., & Labate, D. (2012). *Shearlets: Multiscale analysis for multivariate data*. Birkhäuser.
- Lim, W.-Q., Kutyniok, G., & Zhuang, X. (2012). Digital shearlet transforms. In *Shearlets: multiscale analysis for multivariate data* (pp. 239–282). Birkhäuser.
- Liu, J., & Osher, S. (2019). Block matching local SVD operator based sparsity and TV regularization for image denoising. *Journal of Scientific Computing*, 78, 607–624.
- Li, Xiaoxia, Xiao, Juan, Zhou, Yingyue, Ye, Yuanzheng, Lv, Nianzu, Wang, Xueyuan, et al. (2020). Detail retaining convolutional neural network for image denoising. *Journal of Visual Communication and Image Representation*, 71, 102774.
- Mou, Chong, Zhang, Jian, Fan, Xiaopeng, Liu, Hangfan, & Wang, Ronggang. (2022). Cola-net: Collaborative attention network for image restoration. *IEEE Transactions on Multimedia*, 24, 1366–1377.
- Quan, Yuhui, Chen, Yixin, Shao, Yizhen, Teng, Huan, Xu, Yong, & Ji, Hui. (2021). Image denoising using complex-valued deep CNN. *Pattern Recognition*, 111, 107639.
- Romano, Y., & Elad, M. (2015). Boosting of image denoising algorithms. *SIAM Journal on Imaging Sciences*, 8(2), 1187–1219.
- Selesnick, I. W., Baraniuk, R. G., & Kingsbury, N. G. (2005). The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6), 123–151.
- Şendur, L., & Selesnick, I. W. (2002a). Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Transactions on Signal Processing*, 50, 2744–2756.
- Şendur, L., & Selesnick, I. (2002b). Bivariate shrinkage with local variance estimation. *IEEE Signal Processing Letters*, 9(12), 438–441.
- Shi, Wuzhen, Jiang, Feng, Zhang, Shengping, Wang, Rui, Zhao, Debin, & Zhou, Huiyu. (2019). Hierarchical residual learning for image denoising. *Signal Processing: Image Communication*, 76, 243–251.
- Tian, C., Fei, L., Zheng, W., Zuo, W., Xu, Y., & Lin, C.-W. (2020). Deep learning on image denoising: An overview. *Neural Networks*, 131.
- Wang, Z., & Bovik, A. C. (2009). Mean squared error?: Love it or leave it? *IEEE Signal Processing Magazine*, 26, 98–117.
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.
- Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7), 3142–3155.
- Zhou, T., Li, C., Zeng, X., & Zhao, Y. (2021). Sparse representation with enhanced nonlocal self-similarity for image denoising. *Machine Vision and Applications*, 32(5), 1–11.
- Zhuang, X. (2016). Digital affine shear transforms: Fast realization and applications in image/video processing. *SIAM Journal on Imaging Sciences*, 9(3), 1437–1466.
- Zhuang, X., & Han, B. (2019). Compactly supported tensor product complex tight framelets with directionality. In *2019 International conference on sampling theory and applications (SampTA)*, Bordeaux, France.



Amir Averbuch is an emeritus professor in the school of Computer Science, Tel Aviv University <http://www.cs.tau.ac.il/~amir1/>. His research interests include applied and computational harmonic analysis, machine learning, big data analytics and processing, wavelets, signal/image processing and scientific computing. We have developed theories, algorithms, software to extract intelligence from diverse data. These have been achieved with 30 PhD and 125 M.Sc students, 19 post-docs and senior researchers. He got B.Sc and M.Sc from Hebrew University in Jerusalem and PhD from Columbia University, New York. Before joining Tel Aviv University, he was 10 years Research Staff Member at IBM T.J.Watson Research Center.



Pekka Neittaanmäki Born 14.4.1951 in Saarijärvi Finland. Studies in Mathematics and Computer Science 1970–74 at University Jyväskylä, PhD in Mathematics 1978 at University of Jyväskylä. Post doctoral studies at university of Bonn 1978–79 and at University of Jyväskylä 1979–82. Associate professor at Lappeenranta University 1982 and University of Jyväskylä 1986 and professor 1986–2019, Unesco chair 2019–23. Supervisor of 125 Phd theses. Research interests applied Mathematics and Computer Science. Author or co-author on more than 300 scientific papers and 20 monographs.



Valery Zheludev received his M.S. degree on Math. Physics from Leningrad University, USSR, Ph.D. degree on Math. Physics from Steklov Math. Inst. of Acad. Sci. USSR and the prize “For the best Ph.D. thesis” at Leningrad University. He received Dr. Sci. degree (habilitation) on Comput. Math. from Siberia Branch of Russian Acad. Sci. In 1975–95 Professor, St.-Petersburg Military Institute for Construction Engineering. From 1995 -present with School of Computer Sci., Tel Aviv University, Israel and part-time work with Jyväskylä Univ., Finland University <http://www.cs.tau.ac.il/~zhel/>. Cooperated with the companies Paradigm Geophysical Ltd, ELTA Systems Ltd. Tieto-Saab, ABB, Metso Paper. Fields of research: Wavelet Analysis, Approximation Theory, Signal and Image Processing, Geophysics, Pattern Recognition.



Moshe Salhov is a researcher in the field of geometric data analysis with more than 30 relevant publications in leading journals and conferences (full publication list can be found here: https://scholar.google.com/citations?hl=en&user=oxtRoo8AAAAJ&view_op=list_works&sortby=pubdate). Moshe holds a Ms.C in EE from the Technion Israel Institute of Technology (2006) and a Ph.D in CS from Tel-Aviv University (2016). Moshe has extensive academic and industrial experience in developing algorithms in various fields: Deep learning, Signal Processing, Imaging, Econometrics and abstract/unstructured data analysis.



Jonathan Hauser received the B.Sc., M.Sc. degrees in electrical engineering from the Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel, in 2011, 2016, and 2022, respectively, under the supervision of Prof. Menachem Nathan and the late Prof. Michael A. Golub. His research interests include optics, image processing, and computational imaging, with emphasis on computational snapshot spectral imaging.