Author(s): Averbuch, Amir; Neittaanmäki, Pekka; Zheludev, Valery; Salhov, Moshe; Hauser, Jonathan

Title: Image inpainting using directional wavelet packets originating from polynomial splines

Year: 2021

Version: Accepted version (Final draft)

Please cite the original version:

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, Article 116334. https://doi.org/10.1016/j.image.2021.116334

Image inpainting using directional wavelet packets originating from polynomial splines

Amir Averbuch[1] Pekka Neittaanmäki[2] Valery Zheludev[1] Moshe Salhov[1] Jonathan Hauser[3]
[1]School of Computer Science, [3]School of Electrical Engineering
Tel Aviv University, Israel
[2]Faculty of Mathematical Information Technology
University of Jyväskylä, Finland

Abstract

The paper presents a new algorithm for the image inpainting problem. The algorithm uses a recently designed versatile library of quasi-analytic complex-valued wavelet packets (qWPs) which originate from polynomial splines of arbitrary orders. Tensor products of 1D qWPs provide a diversity of 2D qWPs oriented in multiple directions. For example, a set of the fourth-level qWPs comprises 62 different directions. The properties of these qWPs such as refined frequency resolution, directionality of waveforms with unlimited number of orientations, (anti-)symmetry of waveforms and windowed oscillating structure of waveforms with a variety of frequencies, make them efficient in image processing applications, in particular, in dealing with the inpainting problem addressed in the paper. The obtained results for this problem are quite competitive with the best state-of-the-art algorithms. The inpainting is implemented by an iterative scheme, which expands the *Split Bregman Iteration* (SBI) procedure by supplying it with an adaptive variable soft thresholding based on the *Bivariate Shrinkage* algorithm. In the inpainting experiments, performance comparison between the qWP-based methods and the state-of-the-art algorithms is presented.

1 Introduction

Multimedia images as well as biomedical, seismic and hyper-spectral images, to name a few, comprise smooth regions, oriented edges in various directions and texture that can have an oscillating structure. One of the main goals of image processing is to recover the image from a degraded input data. The degradation can result, for example, from missing pixels, noise, blurring and from different combinations of these factors. Another goal is the extraction of a limited number of characteristic features from images for pattern recognition and machine learning tasks. Achievement of the above goals relies on the fact that practically all processed images have a sparse representation in a proper transform domain. A sparse representation of an image means that the image can be approximated by a linear combination of a relatively small number of 2D "basic" elements from some dictionary, while retaining the above mentioned components of the image. The dictionary of such elements should comprise waveforms that

1. Are oriented in multiple directions (for edges capturing);
2. Have an oscillating structure with multiple frequencies (for retaining texture patterns);

3. Have vanishing moments, at least locally (for sparse representation of smooth regions);
4. Good localization in the spatial domain;
5. Produce a refined frequency separation.
6. Fast implementation of the corresponding transforms is important.

In recent years, a number of elements' dictionaries, which meet some of the above requirements, were described such as pseudo-polar processing [1, 2], contourlets [11], curvelets [8, 7] and shearlets [20, 12]. These dictionaries were used in various image processing applications. For example, one successful application of shearlet transforms to the image processing is the methodology based on the *Digital Affine Shear transforms* (DAS-1) [25]. However, while successfully capturing edges in images, those dictionaries did not demonstrate a satisfactory texture restoration due to the lack of oscillatory waveforms in the dictionaries' libraries.

Another approach to design directional dictionaries consists of the tensor multiplication of complex wavelets ([19, 22]), wavelet frames and wavelet packets (WPs) ([17, 6, 15, 14, 16], to name a few). The tight tensor-product complex wavelet frames (TP_$\mathbb{C}$TF$_n$) with different number of directions, are designed in [15, 16, 14] (here n refers to the number of filters in the underlying 1D complex tight framelet filter bank). Some of them, in particular TP_$\mathbb{C}$TF$_6$ and its modifications like cptTP_$\mathbb{C}$TF$_6$ with compactly supported framelets and TP_$\mathbb{C}$TF$_6^{\downarrow}$ with the reduced redundancy, 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 possess some oscillatory properties.

Some of the disadvantages in the above 2D TP_$\mathbb{C}$TF$_6$, cptTP_$\mathbb{C}$TF$_6$ and TP_$\mathbb{C}$TF$_6^{\downarrow}$ frames such as, for example, limited and fixed number of directions (14 directions at each decomposition level) are overcome in [9] using the algorithm *Digital Affine Shear Filter Transform with 2-Layer Structure (DAS-2)*. It is done by the incorporation of the two-layer structure, which is inherent in the TP_$\mathbb{C}$TF$_6$ frames, into directional filter banks introduced in [25]. This incorporation improves the performance of DAS-2 compared to TP_$\mathbb{C}$TF$_6$ on texture-rich images such as "Barbara", which is not the case on smoother images like "Lena".

Recently, we designed a family of dictionaries that maximally meet the requirements 1–6 [1] and utilize them in image processing applications. Building blocks for such a design are orthonormal WPs originated from discretized polynomial splines of multiple orders (see Chapter 4 in [4]). The designed dictionaries consist of 2D quasi-analytic directional wavelet packets (qWPs). The design scheme is briefly outlined in Section 2.

The qWPs are applied to image inpainting to restore images from corrupted data where many pixels are missing and additive noise is present. This paper introduces two qWP-based iterative methods: Method1 (**M1**) and Method2 (**M2**) to deal with the inpainting problem. The performances of **M1** and **M2** are compared with the performance of **Algorithm I** introduced in [23] (see Section 3) that uses the SET-4[2] filter banks. The computational scheme in **M1** is close to the scheme in **Algorithm I**. The difference is that the qWP transforms are used instead of the SET-4 filter banks. **M2** couples **M1** with the *split Bregman iteration* (SBI) scheme, which uses the so-called *å-based* approach ([13, 18]). Practically, in all the conducted experiments, **M1** and especially **M2** outperform the DET-4 algorithms in the Structural Similarity Index (SSIM) sense.

[1] The design of the dictionaries and implementation of the corresponding transforms are described in detail in [3].

[2] SET-4 is the set of filter banks DAS-2, DAS-1, TP-$\mathbb{C}$TF$_6$ and TP-$\mathbb{C}$TF$_6^{\downarrow}$.

$N=2^j$, $\delta[n]$– N-periodic Kronecker delta	$\omega \overset{\mathrm{def}}{=} e^{2\pi\, i/N}$	$\cdot^*$ – complex conjugate
$\Pi[N]$ – space of N-periodic signals	**M1** – Method1	**M2** – Method2
$\Pi[N,N]$– space of 2D N-periodic arrays	2D – two-dimensional	WP – wavelet packet
DFT – Discrete Fourier transform	FFT – Fast Fourier transform	WPT – WP transform
cWP – complimentary WP	qWP – quasi-analytic WP	HT – Hilbert transform
$H(\mathbf{x})$ – HT of $\mathbf{x} \in \Pi[N]$	DTS – discrete-time spline	DTSWP – WP from DTS
BSA – Bivariate Shrinkage algorithm	p-filter – periodic filter	SBI – split Bregman iteration
SSIM – Structural Similarity Index	SET-4 – filter banks: DAS-2,	DAS-1, TP-$\mathbb{C}$TF$_6$, TP-$\mathbb{C}$TF$_6^{\downarrow}$
PSNR – Peak Signal-to-Noise ratio	in decibels (dB)	$10\log_{10}\left(\frac{N\,255^2}{\sum_{k=1}^{N}(x_k-\tilde{x}_k)^2}\right)\ dB.$

Table 1.1: Notations and abbreviations

$u^p[n]$ and $v^p[n]$ – DFTs of sampled B-spline –Eq. (2.1)	sequences $\alpha[n]$, $\beta[n]$ – Eq.(2.2)
$\psi^p_{[m],l}$ – DTSWP of order p No. l from level m – Eqs.(2.2), (2.6))	$\tilde{\mathbf{H}} = \mathbf{H} = \{\mathbf{h}_0, \mathbf{h}_1\}$ – p-filter banks
$\mathbf{H}_{[m]} = \left\{\mathbf{h}_{[m],0}, \mathbf{h}_{[m],1}\right\}$ –m-level p-filter bank , Eq. (2.5)	$\psi^p_{[m],j,l}[k,n] \overset{\mathrm{def}}{=} \psi^p_{[m],j}[k]\, \psi^p_{[m],l}[n]$– 2D DTSWPs

Table 2.1: Notations for Section 2.1

In many experiments, especially on texture rich images, **M2** produces higher PSNR values than all the SET-4 algorithms.

The paper is organized as follows. In order for the paper to be self-contained, Section 2 briefly outlines the design and implementation of the qWP transforms in one and two dimensions. Section 3 describes the qWP-based **M1** and **M2** for image inpainting. Section 4 presents the experimental results from the restoration of images degraded by missing up to 80% of its pixels while additive noise is present. The performances of the qWP-based methods are compared with the performances of the algorithms that use the SET-4 filter banks. Section 5 discusses the results.

Notations and abbreviations are given in Table 1.1

2 Quasi-analytic WPs

In this section, we briefly outline the design and properties of quasi-analytic WPs (qWPs). For details see [3]. Notations for this section are listed in Table 2.1.

2.1 Orthonormal WPs originated from polynomial splines

In this section, we list the main properties of periodic discrete-time wavelet packets originated from polynomial splines (DTSWPs) and the corresponding transforms. For details see Chapter 4 in [4].

2.1.1 One-dimentional DTSWPs

The centered N-periodic polynomial B-spline $B^p(t)$ of order p is an N-periodization of the function $b^p(t) = \frac{1}{(p-1)!}\sum_{k=0}^{p}(-1)^k\binom{p}{k}\left(t+\frac{p}{2}-k\right)_+^{p-1}$ where $x_+ \overset{\mathrm{def}}{=} \max\{x,0\}$. The B-spline $B^p(t)$ is supported on the interval $(-p/2, p/2)$ up to periodization. It is strictly positive inside this interval and symmetric about zero, where it has its single maximum and has $p-2$ continuous derivatives. The functions $S^p(t) \overset{\mathrm{def}}{=} \sum_{l=0}^{N-1} q[l]\, B^p[t-l]$ are referred to as the order-p periodic splines. The following two sequences will be used later:

$$u^p[n] \stackrel{\mathrm{def}}{=} \sum_{k=-N/2}^{N/2-1} \omega^{-kn}\, b^p\left(k\right), \quad v^p[n] \stackrel{\mathrm{def}}{=} \omega^{-n/2} \sum_{k=-N/2}^{N/2-1} \omega^{-kn}\, b^p\left(k+\frac{1}{2}\right). \tag{2.1}$$

The two-times dilation of the B-spline $b^p(t)$ is denoted by $b^p_d(t) = b^p(t/2)/2$.

Definition 2.1 *The span-two discrete-time B-spline* $\mathbf{b}^p_{[1]}$ *of order* p *is defined as an* N*-periodization of the sampled B-spline* $b^p_d(t)$*. It is an* N*-periodic signal from* $\Pi[N]$*:*

$$b^p_{[1]}[k] \stackrel{\mathrm{def}}{=} b^p_d(k),\ k=-N/2,...,N/2-1 (\mathrm{mod}\, N), \quad \hat{b}^p_{[1]}[n] = \frac{u^p[2n]+v^p[2n]}{2}.$$

Linear combinations $s^p_{[1]}[k] = \sum_{l=0}^{N/2-1} q[l]\, b^p_{[1]}[k-2l]$ of two-sample shifts of the B-splines are referred to as periodic discrete-time splines (DTSs) of span 2. Their DFTs are $\hat{s}^p_{[1]}[n] = \hat{q}[n]_1\, \hat{b}^p_{[1]}[n]$. The $N/2$-dimensional space of the DTSs is denoted by ${}^p\mathcal{S}^0_{[1]} \subset \Pi[N]$ and its orthogonal complement in the signal space $\Pi[N]$ is denoted by ${}^p\mathcal{S}^1_{[1]}$.

Define the DTS $\psi^p_{[1],0}$ and the signal $\psi^p_{[1],1}$ by their DFTs:

$$\hat{\psi}^p_{[1],0}[n] = \beta[n] \stackrel{\mathrm{def}}{=} 2\frac{u^p[2n]+v^p[2n]}{\sqrt{u^p[2n]^2+v^p[2n]^2}}, \quad \hat{\psi}^p_{[1],1}[n] = \alpha[n] \stackrel{\mathrm{def}}{=} 2\omega^n\,\frac{u^p[2n]-v^p[2n]}{\sqrt{u^p[2n]^2+v^p[2n]^2}}. \tag{2.2}$$

Two-sample shifts of the discrete-time-spline $\psi^p_{[1],0}[k]$ and of the signal $\psi^p_{[1],1}[k]$ form orthonormal bases of the subspaces ${}^p\mathcal{S}^0_{[1]}$ and ${}^p\mathcal{S}^1_{[1]}$, respectively.

The signals $\psi^p_{[1],0}[k]$ and $\psi^p_{[1],1}[k]$ are real-valued and symmetric about $k_0 = 0$ and $k_1 = -1$, respectively. They are referred to as the discrete-time-spline wavelet packets (DTSWPs) of order p from the first decomposition level

The one-level DTSWP transform of a signal $\mathbf{x} \in \Pi[N]$ and its inverse are implemented by the application of the analysis $\tilde{\mathbf{H}} = \left\{\tilde{\mathbf{h}}_0, \tilde{\mathbf{h}}_1\right\}$ and synthesis $\mathbf{H} = \{\mathbf{h}_0, \mathbf{h}_1\}$ p-filter banks to the signal $\mathbf{x}$ and the transform coefficients, respectively. The synthesis p-filters are $\mathbf{h}_s = \tilde{\mathbf{h}}_s,\ s=0,1$. Their impulse responses and frequency responses are

$$h_s[k] = \psi^p_{[1],s}[k], \qquad \hat{h}_0[n] = \beta[n], \quad \hat{h}_1[n] = \alpha[n]. \tag{2.3}$$

The transforms can be represented in a matrix form:

$$\left(\begin{array}{c} \hat{y}^0_{[1]}[n] \\ \hat{y}^1_{[1]}[n] \end{array}\right) = \frac{1}{2}\tilde{\mathbf{M}}[-n]\cdot\left(\begin{array}{c} \hat{x}[n] \\ \hat{x}[\vec{n}] \end{array}\right), \quad \left(\begin{array}{c} \hat{x}[n] \\ \hat{x}[\vec{n}] \end{array}\right) = \mathbf{M}[n]\cdot\left(\begin{array}{c} \hat{y}^0_{[1]}[n] \\ \hat{y}^1_{[1]}[n] \end{array}\right),$$

where $\vec{n} = n + N/2$ and $\tilde{\mathbf{M}}[n]$ and $\mathbf{M}[n]$ are the modulation matrices of the analysis and synthesis p-filter banks:

$$\mathbf{M}[n] = \sqrt{2}\left(\begin{array}{cc} \beta[n] & \alpha[n] \\ \beta\left[n+\frac{N}{2}\right] & \alpha\left[n+\frac{N}{2}\right] \end{array}\right) = \tilde{\mathbf{M}}[n]^T. \tag{2.4}$$

The sequences $\beta[n]$ and $\alpha[n]$ are defined in Eq. (2.2).

Extension of transforms to deeper decomposition levels. The p-filter bank is defined as $\mathbf{H}_{[m]} = \left\{\mathbf{h}_{[m],0}, \mathbf{h}_{[m],1}\right\}$, where the p-filters $\mathbf{h}_{[m],s},\ s = 0,1,\ m = 1,...,M$, are defined by their frequency responses:

$$\hat{h}_{[m],0}[n] = \hat{h}_0[2^m n] = \beta[2^m n], \quad \hat{h}_{[m],1}[n] = \hat{h}_1[2^m n] = \alpha[2^m n]. \tag{2.5}$$

The WP transforms to deeper decomposition levels are implemented iteratively, while the transform coefficients $\left\{\mathbf{y}^r_{[m+1]}\right\}$ are derived by filtering the coefficients $\left\{\mathbf{y}^l_{[m]}\right\}$ with the time-reversed p-filters $\mathbf{h}_{[m],s}$, where $l = 0,...,2^m-1,\ s = 0,1$ and $r = \begin{cases} 2l+s, & \text{if } l \text{ is even;} \\ 2l+(1-s), & \text{if } l \text{ is odd.} \end{cases}$

The transform coefficients are $y^l_{[m]}[lk] = \left\langle \mathbf{x}, \psi^p_{[m],l}[\cdot, -2^m k]\right\rangle$, where the signals $\psi^p_{[m],l}$ are normalized, orthogonal to each other in the space $\Pi[N]$, and their $2^m k-$sample shifts are mutually orthogonal. They are referred to as the level-m DTSWPs of order p. The set $\left\{\psi^p_{[m],l}[\cdot, -2^m k]\right\}$, $l = 0,...,2^m-1,\ k = 0,...N/2^m-1$, constitutes an orthonormal basis of the space $\Pi[N]$ and generates its split into 2^m orthogonal subspaces. The next-level WPs $\psi^p_{[m+1],r}$ are derived by filtering the wavelet packets $\psi^p_{[m],l}$ with the p-filters $\mathbf{h}^s_{[m]}$ such that

$$\psi^p_{[m+1],r}[n] = \sum_{k=0}^{N/2^m-1} h_{[m],s}[k]\, \psi^p_{[m],l}[n-2^m k]. \tag{2.6}$$

The DTSWPs originated from ninth-order splines and their spectra are displayed in Fig. 2.2. The transforms are executed in the spectral domain with the Fast Fourier transform (FFT) by using the modulation matrices $\tilde{\mathbf{M}}[2^m\, n]$ and $\mathbf{M}[2^m\, n]$ of the p-filter bank $\mathbf{H}_{[m]}$.

2.1.2 Two-dimentional DTSWPs

A standard way to extend one-dimensional (1D) WPTs to multiple dimensions is by tensor-product extension. The 2D one-level WPT of a signal $\mathbf{x} = \{x[k,n]\},\ k,n = 0,...,N-1$, which belongs to $\Pi[N,N]$, consists of the application of 1D WPT to columns of $\mathbf{x}$, which is followed by the application of the transform to rows of the coefficients array. As a result of the 2D WPT of signals from $\Pi[N,N]$, the space split into four mutually orthogonal subspaces $\Pi[N,N] = \bigoplus_{j,l=0}^{1}\, {}^p\mathcal{S}^{j,l}_{[1]}$.

The 2D DTSWPs are $\psi^p_{[1],j,l}[k,n] \stackrel{\text{def}}{=} \psi^p_{[1],j}[k]\, \psi^p_{[1],l}[n],\quad j,l = 0,1$. They are normalized and orthogonal to each other in the space $\Pi[N,N]$. It means that

$$\sum_{k,n=0}^{N-1} \psi^p_{[1],j_1,l_1}[k,n]\, \psi^p_{[1],j_2,l_2}[k,n] = \delta[j_1-j_2]\, \delta[l_1-l_2].$$

Their two-sample shifts in both directions are mutually orthogonal. The subspace ${}^p\mathcal{S}^{j,l}_{[1]}$ is a linear hull of two-sample shifts of the 2D wavelet packets $\left\{\psi^p_{[1],j,l}[\cdot-2k, \cdot-2n]\right\}$, $k,n, = 0,...,N/2-1$, that form an orthonormal basis of ${}^p\mathcal{S}^{j,l}_{[1]}$.

By the application of the above transforms iteratively to blocks of the transform coefficients down to m-th level, we get that the space $\Pi[N,N]$ is decomposed into 4^m mutually orthogonal subspaces $\Pi[N,N] = \bigoplus_{j,l=0}^{2^m-1}\, {}^p\mathcal{S}^{j,l}_{[m]}$.

$\tau^p_{[m],l} = H(\psi^p_{[m],l})$ – HT of DTSWP $\psi^p_{[m],l}$	$\psi^p_{\pm[m],l} = \psi^p_{[m],l} \pm i\tau^p_{[m],l}$ – analytic DTSWPs
$\hat{\phi}^p_{[m],l}$ – complimentary WP (cWP) –Eq. (2.7)	$\Psi^p_{\pm[m],l}$ – complex quasi-analytic WPs (qWPs) –Eq. (2.8)
$\mathbf{Z}_{[m]}$ – m-level transform coefficients with the qWPs $\left\{\Psi_{\pm[m]}\right\}$, Eq. (2.11)	$\mathbf{H}$, $\mathbf{F}$, $\mathbf{Q}_\pm$ – p-filter banks for qWP transforms, Eq. (2.12)
$\Psi^p_{++[m],j,l}[k,n]$ and $\Psi^p_{+-[m],j,l}[k,n]$ – 2D complex qWPs, Eq. (2.9)	$\theta^p_{+[m],j,l}[k,n]$ and $\theta^p_{-[m],j,l}[k,n]$ – directional qWPs, Eq. (2.10)

Table 2.2: Notations for Section 2.2

The 2D tensor-product wavelet packets $\psi^p_{[m],j,l} \stackrel{\text{def}}{=} \psi^p_{[m],j}[k]\,\psi^p_{[m],l}[n]$, which provide orthonormal bases to the respective subspaces ${}^p\mathcal{S}^{j,l}_{[m]}$, are well localized in the spatial domain, their 2D DFT spectra produce a refined tiling of the frequency domain of signals from $\Pi[N,N]$.[3] The drawback for image processing is that the WPs are oriented in ether horizontal or vertical directions or are not oriented at all.

2.2 Quasi-analytic and complementary WPs

In this section, we define the so-called quasi-analytic WPs related to the DTSWPs discussed in Section 2.1. In addition, we introduce an orthonormal set of waveforms which are complementary to the DTSWPs. Notations for this section are listed in Table 2.2.

Briefly, the design scheme consists of the following steps: 1. Application of the Hilbert transform (HT) to the set $\{\psi\}$ of orthonormal WPs producing the set $\{\tau = H(\psi)\}$. 2. A slight correction of the set $\{\tau\}$ provides an orthonormal set $\{\phi\}$ of the so-called complimentary WPs (cWPs), which are anti-symmetric and whose magnitude spectra coincides with the magnitude spectra of the respective WPs from the set $\{\psi\}$. 3. Two sets of complex quasi-analytic WPs (qWPs) $\left\{\Psi_+ \stackrel{\text{def}}{=} \psi + i\,\phi\right\}$ and $\left\{\Psi_- \stackrel{\text{def}}{=} \psi - i\,\phi\right\}$, whose spectra are localized in the positive and negative half-bands of the frequency domain, respectively, are defined. 4. Two sets of 2D complex qWPs produced by the tensor products of the qWPs $\{\Psi_\pm\}$ as: $\left\{\Psi_{++} \stackrel{\text{def}}{=} \Psi_+ \bigotimes \Psi_+\right\}$ and $\left\{\Psi_{+-} \stackrel{\text{def}}{=} \Psi_+ \bigotimes \Psi_-\right\}$ are defined. 5. The dictionaries we use are the real parts of these qWPs: $\left\{\theta_\pm \stackrel{\text{def}}{=} \mathfrak{Re}(\Psi_{+\pm})\right\}$.

The diagram in Fig. 2.1 illustrates the design of qWPs.

2.2.1 Analytic periodic signals

A signal $\mathbf{x} \in \Pi[N]$ is represented by its inverse DFT which can be written as follows:

$$x[k] \quad = \quad \frac{\hat{x}[0] + (-1)^k \hat{x}[N/2]}{N} + \frac{2}{N}\sum_{n=1}^{N/2-1} \frac{\hat{x}[n]\,\omega^{kn} + (\hat{x}[n]\,\omega^{kn})^*}{2}.$$

Define the real-valued signal $\mathbf{t} \in \Pi[N]$ and two complex-valued signals $\mathbf{x}_+$ and $\mathbf{x}_-$ such that

$$\begin{aligned}
t[k] &\stackrel{\text{def}}{=} \tfrac{2}{N}\textstyle\sum_{n=1}^{N/2-1} \tfrac{\hat{x}[n]\,\omega^{kn} - \hat{x}[n]^*\,\omega^{-kn}}{2i},\\
x_\pm[k] &\stackrel{\text{def}}{=} x[k] \pm i\,t[k] = \tfrac{\hat{x}[0]+(-1)^k\hat{x}[N/2]}{N}\\
&+ \tfrac{2}{N}\textstyle\sum_{n=1}^{N/2-1} \begin{cases} \hat{x}[n]\,\omega^{kn}, & \text{for } \bar{x}_+;\\ \hat{x}[-n]\,\omega^{-kn} = \hat{x}[N-n]\,\omega^{-k(N-n)}, & \text{for } \bar{x}_-. \end{cases}
\end{aligned}$$

[3] It is true especially for WPs derived from higher-order DTSs.

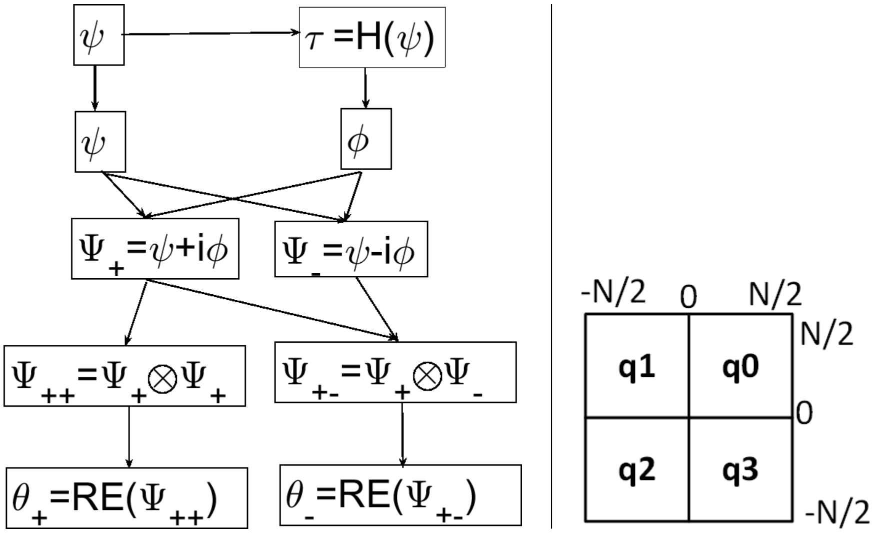

Figure 2.1: Diagram of the qWP design (left) and quadrants of frequency domain (right)

The spectrum of $\mathbf{x}_+$ comprises only non-negative frequencies and vice versa for $\mathbf{x}_-$[4]. We have $\mathbf{x} = \mathfrak{Re}(\mathbf{x}_\pm)$ and $\mathfrak{Im}(\mathbf{x}\pm) = \pm\mathbf{t}$. The signals $\mathbf{x}_\pm$ are referred to as the periodic discrete-time analytic signals.

The signal $\mathbf{t}$ is a discrete periodic version of the Hilbert transform (HT) of a discrete-time periodic signal $\mathbf{x}$, that is $\mathbf{t} = H(\mathbf{x})$ (see [21], for example). Note that the DFT spectrum $\hat{t}[n]$ lacks the values $\hat{t}[0]$ and $\hat{t}[N/2]$.

2.2.2 Analytic and quasi-analytic WPs

The analytic WPs and their DFT spectra are derived from DTSWPs $\left\{\psi^p_{[m],l}\right\}$, $m = 1, ..., M,\ l = 0, ..., 2^m - 1$, in line with the scheme in Section 2.2.1. Recall that for all $l \neq 0$, the DFT $\hat{\psi}^p_{[m],l}[0] = 0$ and for all $l \neq 2^m - 1$, the DFT $\hat{\psi}^p_{[m],l}[N/2] = 0$.

Denote by $\tau^p_{[m],l} \stackrel{\text{def}}{=} H(\psi^p_{[m],l})$ the HT of the wavelet packet $\psi^p_{[m],l}$. Then, the analytic WPs are $\psi^p_{\pm[m],l} = \psi^p_{[m],l} \pm i\tau^p_{[m],l}$.

Complementary orthonormal WPs The values $\hat{\tau}^p_{[m],l}[0]$ and $\hat{\tau}^p_{[m],l}[N/2]$ are missing in the DFT spectra of the HT signals $\tau^p_{[m],l}$. We upgrade the set $\left\{\tau^p_{[m],l}\right\}$, $l = 0, ...2^m - 1$, in the following way. The set $\left\{\phi^p_{[m],l}\right\}$, $m = 1, ..., M,\ l = 0, ..., 2^m - 1$, of signals from the space $\Pi[N]$ is defined via their DFTs:

$$\hat{\phi}^p_{[m],l}[n] = \hat{\tau}^p_{[m],l}[n] + \hat{\psi}^p_{[m],l}[0] + \hat{\psi}^p_{[m],l}[N/2]. \tag{2.7}$$

For all $l \neq 0, 2^m - 1$, the signals $\phi^p_{[m],l}$ coincide with $\tau^p_{[m],l} = H(\psi^p_{[m],l})$.

Proposition 2.2

- *The magnitude spectra* $\left|\hat{\phi}^p_{[m],l}[n]\right|$ *of cWPs coincide with* $\left|\hat{\psi}^p_{[m],l}[n]\right|$.

[4] Recall that $\hat{x}[N/2] = \hat{x}[-N/2]$.

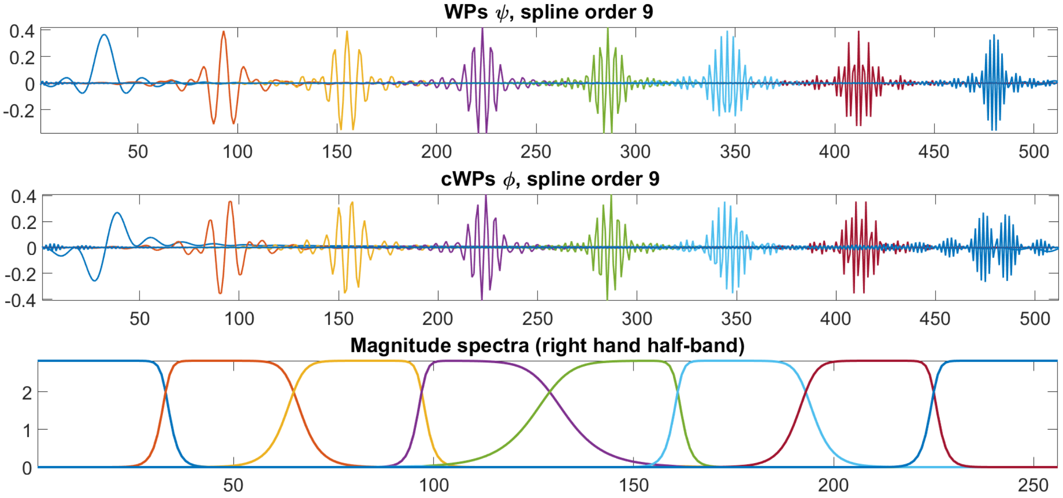

Figure 2.2: Top: signals $\psi^9_{[3],l}$, $l = 0, ..., 7$. Center: signals $\phi^9_{[3],l}$, $l = 0, ..., 7$. Bottom: their magnitude DFT spectra, respectively

- *For any $m = 1, ..., M$, and $l = 1, ..., 2^m - 2$, the signals $\phi^p_{[m],l}$ are antisymmetric oscillating waveforms. For $l = 0,\ 2^m - 1$, the shapes of the signals are near antisymmetric.*
- *The orthonormality properties that are similar to the properties of WPs $\psi^p_{[m],l}$ hold for the signals $\phi^p_{[m],l}$ such that* $\left\langle \phi^p_{[m],l}[\cdot - k\,2^m], \phi^p_{[m],\lambda}[\cdot - n\,2^m] \right\rangle = \delta[\lambda, l]\,\delta[k, n], \quad k, n = 0, ..., N/2^m - 1.$

Figure 2.2 displays the signals $\psi^9_{[3],l}$ and $\phi^9_{[3],l}$, $l = 0, ..., 7$, from the third decomposition level and their magnitude spectra. Addition of $\hat{\psi}^p_{[3],l}[0]$ and $\hat{\psi}^p_{[3],l}[N/2]$ to the spectra of $\phi^9_{[3],l}$, $l = 0, 7$ results in an antisymmetry distortion.

We call the signals $\left\{\phi^p_{[m],l}\right\}$, $m = 1, ..., M$, $l = 0, ..., 2^m - 1$, the *complementary wavelet packets* (cWPs). Similarly to the WPs $\left\{\psi^p_{[m],l}\right\}$, differenent combinations of the cWPs can provide differenent orthonormal bases for the space $\Pi[N]$.

Quasi-analytic WPs: The sets of complex-valued WPs, which we refer to as the quasi-analytic wavelet packets (qWP), are defined by

$$\Psi^p_{\pm[m],l} = \psi^p_{[m],l} \pm i\phi^p_{[m],l}, \quad m = 1, ..., M,\ l = 0, ..., 2^m - 1, \tag{2.8}$$

where $\phi^p_{[m],l}$ are the cWPs from Eq. (2.7). qWPs $\Psi^p_{\pm[m],l}$ differ from the analytic WPs $\psi^p_{\pm[m],l}$ by the addition of the two values $\pm i\,\hat{\psi}^p_{[m],l}[0]$ and $\pm i\,\hat{\psi}^p_{[m],l}[N/2]$ into their DFT spectra, respectively. For a given decomposition level m, these values are zero for all l except for $l_0 = 0$ and $l_m = 2^m - 1$. It means that for all l except for l_0 and l_m, the qWPs $\Psi^p_{\pm[m],l}$ are analytic. The DFTs of qWPs are

$$\hat{\Psi}^p_{+[m],l}[n] = \begin{cases} (1+i)\hat{\psi}^p_{[m],l}[n], & \text{if } n = 0, N/2; \\ 2\hat{\psi}^p_{[m],l}[n], & \text{if } 0 < n < N/2; \\ 0 & \text{if } N/2 < n < N, \end{cases} \qquad \hat{\Psi}^p_{-[m],l}[n] = \begin{cases} (1-i)\hat{\psi}^p_{[m],l}[n], & \text{if } n = 0, N/2; \\ 0 & \text{if } 0 < n < N/2; \\ 2\hat{\psi}^p_{[m],l}[n], & \text{if } N/2 < n < N. \end{cases}$$

2.2.3 Design of 1D cWPs and qWPs

The DFTs of the first-level DTSWPs are $\hat{\psi}^p_{[1],0}[n] = \beta[n], \quad \hat{\psi}^p_{[1],1}[n] = \omega^n \, \beta[n + N/2] = \alpha[n]$, where the sequences $\beta[n]$ and $\alpha[n]$ are defined in Eq. (2.2). The DFTs of the first-level cWPs are

$$\hat{\phi}^p_{[1],0}[n] = \begin{cases} -i\,\beta[n], & \text{if } 0 < n < N/2; \\ i\,\beta[n], & \text{if } N/2 < n < N; \\ \sqrt{2}, & \text{if } n = 0; \\ 0, & \text{if } n = N/2, \end{cases} \qquad \hat{\phi}^p_{[1],1}[n] = \begin{cases} -i\,\alpha[n], & \text{if } 0 < n < N/2; \\ i\,\alpha[n], & \text{if } N/2 < n < N; \\ 0, & \text{if } n = 0; \\ -\sqrt{2}, & \text{if } n = N/2. \end{cases}$$

The cWPs and qWPs from the second and further decomposition levels are derived iteratively using the same p-filter banks as the DTFWPs.

Proposition 2.3 ([3]) *Assume that for a DTSWP $\psi^p_{[m+1],r}$ the relation in Eq. (2.6) holds. Then, for the cWP $\phi^p_{[m+1],r}$ and qWP $\Psi^p_{\pm[m+1],r}$ we have*

$$\begin{aligned} \phi^p_{[m+1],r}[k] &= \sum_{j=0}^{N/2^m-1} h_{[m],s}[j] \, \phi^p_{[m],l}[k - 2^m j] \Longleftrightarrow \hat{\phi}^p_{[m+1],r}[n] = \hat{h}_s[2^m n]_m \, \hat{\phi}^p_{[m],l}[n], \\ \Psi^p_{\pm[m+1],r}[k] &= \sum_{j=0}^{N/2^m-1} h_{[m],s}[j] \, \Psi^p_{\pm[m],l}[k - 2^m j] \Longleftrightarrow \hat{\Psi}^p_{\pm[m+1],r}[n] = \hat{h}_s[2^m n]_m \, \hat{\Psi}^p_{[m],l}[n], \\ \hat{h}^0_{[1]}[\nu] &= \hat{\psi}^p_{[1],0}[\nu] = \beta[\nu], \quad \hat{h}^1_{[1]}[\nu] = \hat{\psi}^p_{[1],1}[\nu] = \alpha[\nu], \quad r = \begin{cases} 2l + s, & \textit{if } l \textit{ is even;} \\ 2l + (1 - s), & \textit{if } l \textit{ is odd.} \end{cases} \end{aligned}$$

2.2.4 Design of 2D directional WPs

The 2D wavelet packets, whicht are defined by the tensor products of 1D DTSWPs such that $\psi^p_{[m],j,l}[k,n] = \psi^p_{[m],j}[k] \, \psi^p_{[m],l}[n]$ and cWPs $\phi^p_{[m],j,l}[k,n] = \phi^p_{[m],j}[k] \, \phi^p_{[m],l}[n]$, possess many valuable properties but they lack the directionality, which is needed in many applications that process 2D data. However, real-valued 2D WPs oriented in multiple directions are derived from tensor products of complex qWPs $\Psi^p_{\pm[m],l}$. The complex 2D qWPs are defined as follows:

$$\Psi^p_{++[m],j,l}[k,n] \overset{\text{def}}{=} \Psi^p_{+[m],j}[k] \, \Psi^p_{+[m],l}[n], \quad \Psi^p_{+-[m],j,l}[k,n] \overset{\text{def}}{=} \Psi^p_{+[m],j}[k] \, \Psi^p_{-[m],l}[n], \tag{2.9}$$

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

$$\begin{aligned} \theta^p_{+[m],j,l}[k,n] &\overset{\text{def}}{=} \mathfrak{Re}(\Psi^p_{++[m],j,l}[k,n]) = \psi^p_{[m],j,l}[k,n] - \phi^p_{[m],j,l}[k,n], \\ \theta^p_{-[m],j,l}[k,n] &\overset{\text{def}}{=} \mathfrak{Re}(\Psi^p_{+-[m],j,l}[k,n]) = \psi^p_{[m],j,l}[k,n] + \phi^p_{[m],j,l}[k,n], \end{aligned} \tag{2.10}$$

The DFT spectra of the 2D qWPs $\Psi^p_{++[m],j,l}, \; j, l = 0, ..., 2^m - 1$, are the tensor products of the one-sided spectra of the qWPs: $\hat{\Psi}^p_{++[m],j,l}[\kappa, \nu] = \hat{\Psi}^p_{+[m],j}[\kappa] \, \hat{\Psi}^p_{+[m],l}[\nu]$ and, as such, they tile the quadrant $\mathbf{q}_0$ of the frequency domain, while the spectra of $\Psi^p_{+-[m],j,l}, \; j, l = 0, ..., 2^m - 1$, tile the quadrant $\mathbf{q}_1$ (see Fig. 2.1). Consequently, the spectra of the real-valued 2D qWPs $\theta^p_{+[m],j,l}$ and $\theta^p_{-[m],j,l}$ tile the pairs of quadrants $\mathbf{q}_+ \overset{\text{def}}{=} \mathbf{q}_0 \bigcup \mathbf{q}_2$ and $\mathbf{q}_- \overset{\text{def}}{=} \mathbf{q}_1 \bigcup \mathbf{q}_3$, respectively, by relatively

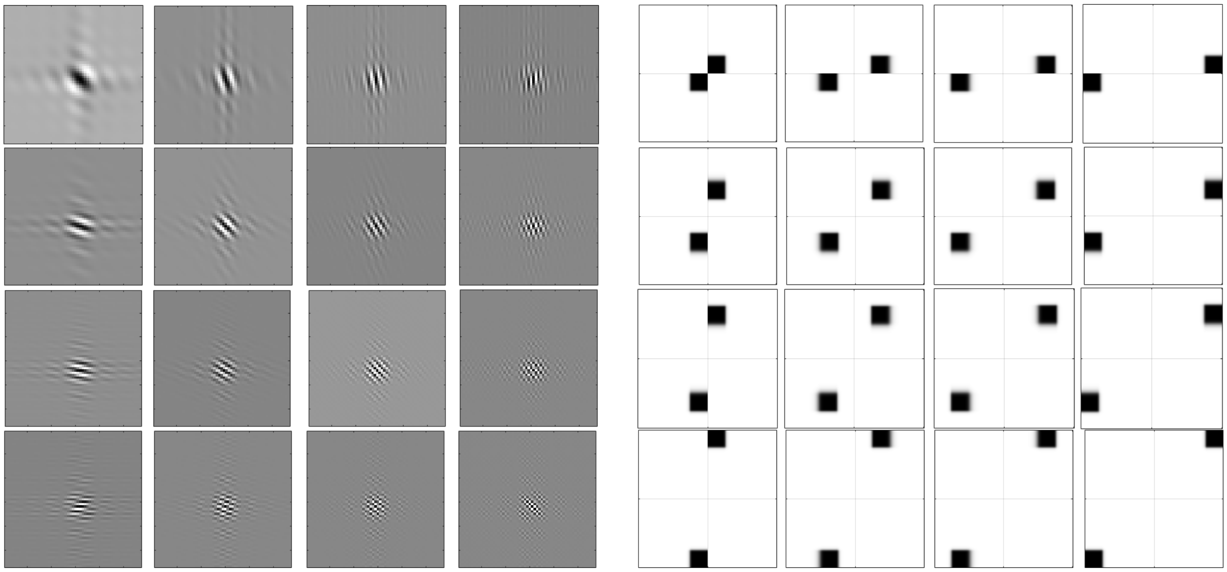

Figure 2.3: WPs $\theta^{9}_{+[2],j,l}$ from the second decomposition level and their magnitude spectra

small pairs of squares, which are symmetric about the origin. Such localizations and shapes of the spectra determine the directionality and shapes of the waveforms $\theta^{p}_{\pm[m],j,l}$. They are, approximately, windowed cosines with multiple frequencies, which are oriented, at the level m, in $2(2^{m+1}-1)$ directions ([3]). Figures 2.3 and 2.4 display real qWPs $\theta^{9}_{+[2],j,l}$ and $\theta^{9}_{-[2],j,l}$, $j,l=0,1,2,3$, respectively, from the second decomposition level and their magnitude spectra.

2.2.5 Implementation of qWP transforms

The qWP transforms are executed in the frequency domain using the FFT. Assume that the signal $\mathbf{x}$ to be transformed belongs to the space $\Pi[N]$ of $N-$periodic discrete-time signals.

The sets $\mathbf{Z}_{[m]}$ of the transform coefficients with the qWPs $\left\{\Psi_{\pm[m]}\right\}$ from the decomposition level m consist of 2^{m+1} blocks $\mathbf{Z}_{[m]}=\mathbf{Z}_{+[m]}\bigcup\mathbf{Z}_{-[m]}$, where $\mathbf{Z}_{\pm[m]}=\bigcup_{l=0}^{2^m-1}\mathbf{Z}_{\pm[m],l}$. The blocks $\mathbf{Z}_{\pm[m],l}$ are related to the qWPs $\left\{\Psi_{\pm[m],l}\right\}$, $l=0,...,2^m-1$, respectively. The qWP transform coefficients of a signal $\mathbf{x}=\{x[k]\}\in\Pi[N]$ are the inner products of $\mathbf{x}$ with the translations of the corresponding wavelet packets:

$$\begin{aligned} z_{\pm[m],l}[k] = \ & \left\langle\mathbf{x},\Psi_{\pm[m],l}[\cdot-2^m k]\right\rangle=\textstyle\sum_{j=0}^{N-1}\Psi^{*}_{\pm[m],l}[j-2^m k]\,x[j]=y_{[m],l}[k]\mp i\,c_{[m],l}[k],\\ y_{[m],l}[k] = \ & \left\langle\mathbf{x},\psi_{[m],l}[\cdot-2^m k]\right\rangle,\quad c_{[m],l}[k]=\left\langle\mathbf{x},\phi_{[m],l}[\cdot-2^m k]\right\rangle,\quad l=0,...,2^m-1.\end{aligned}\tag{2.11}$$

The transform s are implemented in a multiresolution mode by multirate filtering. It follows from Proposition 2.3 that the structure of the filter bank for the transform of a signal $\mathbf{x}$ to the first decomposition level differs from the structure of the filter banks for subsequent levels. Define the filter banks $\mathbf{H}=\{\mathbf{h}_s\},\mathbf{F}=\{\mathbf{f}_s\},\mathbf{Q}_{\pm}=\{\mathbf{q}_{\pm s}\},\ s=0,1$, by their impulse responses:

$$h_s[k]=\psi_{[1],s}[k],\quad f_s[k]=\phi_{[1],s}[k],\quad q_{\pm s}[k]=h_s[k]\pm if_s[k],\quad k=0,...,N-1,\ s=0,1.\tag{2.12}$$

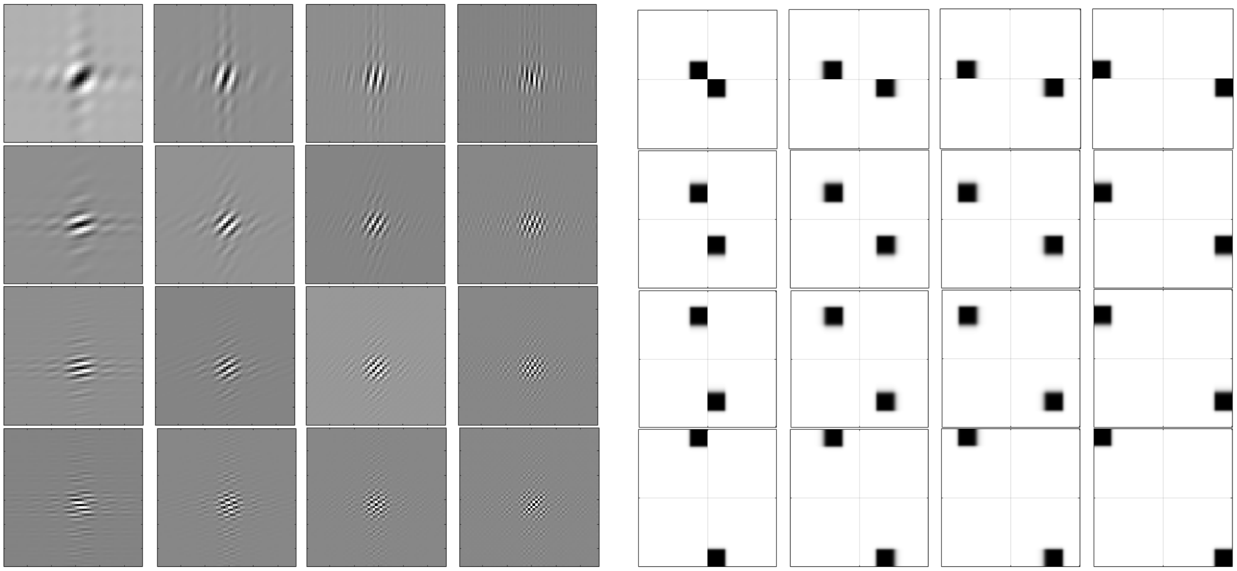

Figure 2.4: WPs $\theta^{9}_{-[2],j,l}$ from the second decomposition level and their magnitude spectra

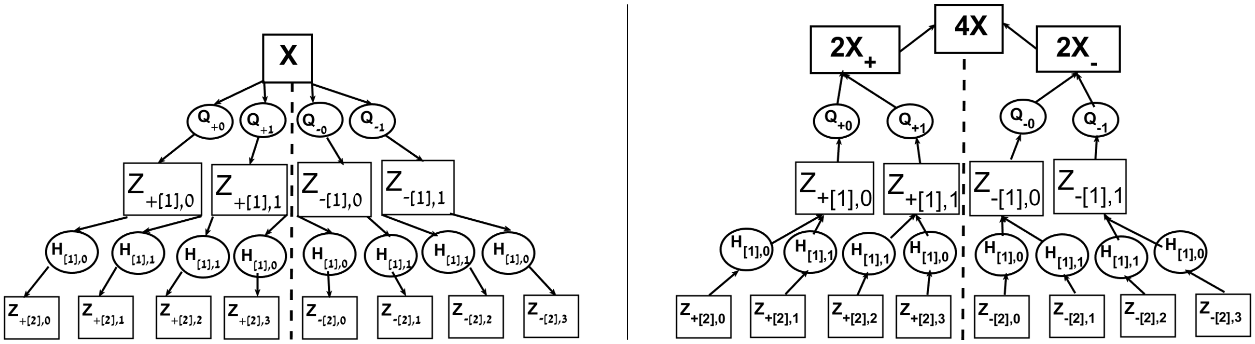

Figure 2.5: Diagrams of the two-level 1D qWP transforms. Left: direct ⌢Right: inverse ⌢

The four blocks $\left\{\mathbf{Z}_{+[1],0}, \mathbf{Z}_{+[1]1}, \mathbf{Z}_{-[1]0}, \mathbf{Z}_{-[1]1}\right\}$ of the first-level transform coefficients are derived by filtering the signal $\mathbf{x}$ by the time reversed filters $\mathbf{Q}^{*}_{\pm s} = \mathbf{Q}_{\mp s}$, which is followed by downsampling, such as $z_{\pm[1],l}[k] = \sum_{j=0}^{N-1} q_{\mp\lambda}[j-2k]\,x[j],\ l = 0, 1.$

The frequency responses of the filters $\mathbf{h}_s$ are $\hat{h}_s[n] = \sum_{k=0}^{N-1} e^{2\pi ikn/N} h_s[k] = \hat{\psi}_{[1],s}[n],\ n = 0, ..., N-1,\ s = 0, 1.$ The filters $\mathbf{h}_{[m],s},\ m = 1, ..., M,\ s = 0, 1,$ for the transform from the first to the subsequent decomposition levels are defined via their frequency responses: $\hat{h}_{[m],s}[n] = \hat{h}_s[2^m n],\ m = 1, ..., M,\ s = 0, 1.$ Thus, the qWP transforms from the first to the second decomposition level are

$$\begin{array}{ll} z_{\pm[2],0}[k] = \sum_{l=0}^{N/2-1} h_{[1],0}[l-4k]\, z_{\pm[1],0}[l], & z_{\pm[2],1}[k] = \sum_{l=0}^{N/2-1} h_{[1],1}[l-4k]\, z_{\pm[1],0}[l], \\ z_{\pm[2],2}[k] = \sum_{l=0}^{N/2-1} h_{[1],1}[l-4k]\, z_{\pm[1],1}[l], & z_{\pm[2],3}[k] = \sum_{l=0}^{N/2-1} h_{[1],0}[l-4k]\, z_{\pm[1],1}[l], \quad . \end{array}$$

The transforms to the subsequent decomposition levels are executed similarly using the filters $\mathbf{H}_{[m],s},\ m = 2, ..., M,\ s = 0, 1.$ The diagrams in Fig. 2.5 illustrate the qWP transform implementation.

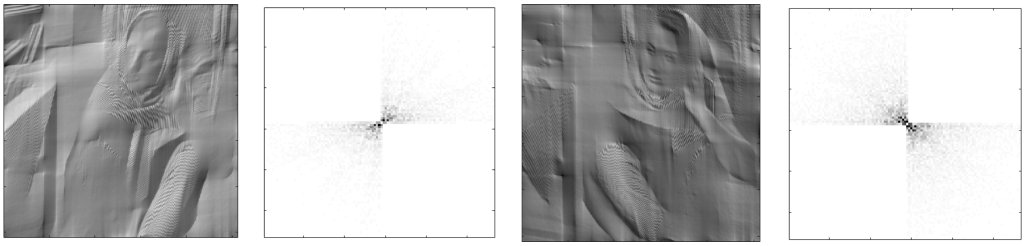

Figure 2.6: Left to right: 1.Image $\mathfrak{Re}(\mathbf{X}_+)$. 2.Its magnitude DFT spectrum. 3.Image $\mathfrak{Re}(\mathbf{X}_-)$. 4.Its magnitude DFT spectrum

The inverse transforms from the level-M coefficients $\mathbf{Z}_{\pm[M]}$ to the coefficients of the first decomposition level $\left\{\mathbf{Z}_{+[1],0}, \mathbf{Z}_{+[1]1}, \mathbf{Z}_{-[1]0}, \mathbf{Z}_{-[1]1}\right\}$ are executed in a standard way by upsampling and filtering the upsampled arrays by the filters $\mathbf{H}_{[m],s},\ m = 2, ..., M, l = 0, 1$. Then, arrays $\mathbf{Z}_{+[1],0}$ and $\mathbf{Z}_{+[1]1}$ are upsampled and filtered by the filters $\mathbf{Q}_{+0}$ and $\mathbf{Q}_{+1}$, respectively, thus producing a complex signal $2\mathbf{x}_+$. Similarly, the signal $2\mathbf{x}_-$ is derived from the coefficients $\mathbf{Z}_{-[1]l}$ using the filters $\mathbf{Q}_{-s},\ s = 0, 1$. The signal $\mathbf{x}$ is restored by $\mathbf{x} = \mathfrak{Re}(\mathbf{x}_+ + \mathbf{x}_-)/4$.

Practically, the direct and inverse transforms are implemented in the frequency domain using modulation matrices (see [3]).

The 2D transforms of an image $\mathbf{X}$ of size $N \times N$ are executed in a tensor product mode that means application of 1D transforms with the filters $\mathbf{Q}_{\pm s},\ s = 0, 1$, to rows of the image $\mathbf{X}$, which is followed by application of the transforms with the filters $\mathbf{Q}_{+s},\ s = 0, 1$, to the columns of the derived coefficient arrays. Thus the eight coefficient arrays of size $N/2 \times N/2$ are produced: $\left\{\mathbf{Z}_{+[1],\lambda,\mu},\ \mathbf{Z}_{-[1],\lambda,\mu},\ \lambda, \mu = 0, 1\right\}$. Then, the coefficient arrays are processed in the same way but using the filters $\mathbf{P}_{[m],\lambda},\ m = 2, ..., M, \lambda = 0, 1$ instead of $\mathbf{Q}_{\pm\lambda},\ \lambda = 0, 1$. The arrays related to the "positive" qWPs Ψ_{++} and arrays related to the "negative" qWPs Ψ_{+-} are processed separately forming a double-tree structure. At the second level we have 32 coefficient arrays $\left\{\mathbf{Z}_{+[2],\lambda,\mu},\ \mathbf{Z}_{-[2],\lambda,\mu},\ \lambda, \mu = 0, 1, 2, 3\right\}$ of size $N/4 \times N/4$ and so on.

The inverse transforms from the level-M coefficients $\left\{\mathbf{Z}_{+[M],\lambda,\mu},\ \mathbf{Z}_{-[M],\lambda,\mu},\ \lambda, \mu = 0, ..., 2^M - 1\right\}$ to the coefficients of the first decomposition level $\left\{\mathbf{Z}_{+[1],\lambda,\mu},\ \mathbf{Z}_{-[1],\lambda,\mu},\ \lambda, \mu = 0, 1\right\}$ are executed in a standard tensor product way using filters $\mathbf{H}_{[m],s},\ m = 2, ..., M, l = 0, 1$. Then, the complex sub-images $\mathbf{X}_+$ and $\mathbf{X}_-$ are derived from the coefficients $\mathbf{Z}_{+[1],\lambda,\mu}$ and $\mathbf{Z}_{-[1],\lambda,\mu},\ \lambda, \mu = 0, 1$, by the filters $\mathbf{Q}_{+s}$ and $\mathbf{Q}_{-s},\ s = 0, 1$, respectively. Then, the signal $\mathbf{X} \in \Pi[N, N]$ is restored by $\mathbf{X} = \mathfrak{Re}(\mathbf{X}_+ + \mathbf{X}_-)/8$.

Figure 2.6 illustrates the "Barbara" image restoration by the 2D signals $\mathfrak{Re}(\mathbf{X}_\pm)$. The signal $\mathfrak{Re}(\mathbf{X}_+)$ captures edges oriented to *north-east*, while $\mathfrak{Re}(\mathbf{X}_-)$ captures edges oriented to *north-west*. The signal $\tilde{\mathbf{X}} = \mathfrak{Re}(\mathbf{X}_+ + \mathbf{X}_-)/8$ perfectly restores the image achieving PSNR=313.8596 dB.

3 Image inpainting methodology

This section presents two qWP-based methods: Method1 (**M1**) and Method2 (**M2**) that are applied to image inpainting. Image inpainting means restoration of an image that was degraded by having many missing pixels and, possibly, was corrupted by Gaussian noise. Similarly to **Algorithm I** (**AI**) in [24, 23], both methods use decreasing thresholding values that are determined by the sequences $\Lambda_1[i]$ and $\Lambda_2[i]$ described in Eq. (3.4) and rely on the redundancy of the qWP transforms and

M1 – Method1, **M2** – Method2, **A1** – **Algorithm I**,	$\Lambda_1[i]$ and $\Lambda_2[i]$ – threshold sequences (Eq. (3.4))
$\mathbf{B}^{\lambda}_{W[m]}$ – Bivariate shrinkage operator (Eq. (3.2))	$\bar{\sigma}_{W[m]}[k,n]^2$ – averaged variance for $c_{[m]}[k,n]$ (Eq. (3.1))
$\tilde{\sigma}_{W[m]}[k,n]^2$ – marginal variance for $c_{[m]}[k,n]$	R_1 and R_2 –free parameters for thresholding
$\tilde{\mathbf{F}}_{[m]}$ – operator of transform $\tilde{\mathbf{F}}_{[m]}\,\mathbf{X} = \mathbf{Z}_{[m]} = \mathbf{Z}_{+[m]} \bigcup \mathbf{Z}_{-[m]}$	$\mathbf{F}_{[m]}$ – reconstruction operator from level m: $\mathbf{F}_{[m]}\,\mathbf{Z}_{[m]} = \mathbf{X}_{[m]}$
Θ – mask matrix, Θ_1 – extended mask	$\rho = \frac{\sum_{l,n=1}^{N_1}\Theta_1[l,n]}{N_1^2}$, tol_1 and tol_2 –tolerance parameters

Table 3.1: Notations for Section 3

on the interdependency between the transform coefficients in the horizontal (between neighboring coefficients) and vertical (parent-child) directions. This interdependency is utilized via the Bivariate Shrinkage algorithm (BSA) ([10]).

We refer to the set of the filter banks DAS-2, DAS-1, TP-$\mathbb{C}TF_6$ and TP-$\mathbb{C}TF_6^{\downarrow}$ as SET-4.

AI uses the TP-$\mathbb{C}$TF$_6$ filter bank that produces excellent image inpainting results that outperform (in a PSNR sense) most state-of-the-art methods (see [23] for methods comparison). In [9], **AI** is implemented using the SET-4 filter banks. On the cartoon-type images such as "Lena" and "Boat", TP-$\mathbb{C}$TF$_6$ achieves better performance than the rest of the members in SET-4. However, images with texture such as "Fingerprint" and, especially "Barbara", are inpainted better by the filter bank DAS-2 with increased directionality compared to TP-$\mathbb{C}$TF$_6$. Unlike [9, 24, 23], both **MI** and **M2** are using 2D qWP transforms. The transform coefficients from either two or three decomposition levels are utilized.

Bivariate shrinkage operator (BSA): Assume that $c_{[m]}[k,n]$ is a transform coefficient from a decomposition level m. The averaged variance for the coefficient $c_{[m]}[k,n]$ is determined by the equation

$$\bar{\sigma}_{W[m]}[k,n]^2 = \frac{1}{(2W_{[m]})^2} \sum_{\kappa,\nu=-W_{[m]}}^{W_{[m]}-1} c_{[m]}[k+\kappa, n+\nu]^2, \tag{3.1}$$

where $W_{[m]}$ is an integer.

The marginal variance for the coefficient $c_{[m]}[k,n]$ is estimated: $\tilde{\sigma}_{W[m]}[k,n]^2 = (\bar{\sigma}_{W[m]}[k,n]^2 - \lambda^2)_+$, where λ is a parameter taken from either the sequence Λ_1 or Λ_2 by the rule explained below. Then, the BSA operator $\mathbf{B}^{\lambda}_{W[m]}$ is defined by

$$\begin{aligned} \mathbf{B}^{\lambda}_{W[m]}\, c_{[m]}[k,n] &\overset{\text{def}}{=} \begin{cases} c_{[m]}[k,n] - \lambda_{W[m]}(c)\,\frac{c_{[m]}[k,n]}{|c_{[m]}[k,n]|}, & \text{if } |c_{[m]}[k,n]| > \lambda_W(c); \\ 0, & \text{otherwise,} \end{cases} \qquad (3.2) \\ \lambda_{W[m]}(c) &= \frac{\sqrt{3}\,\lambda^2}{\tilde{\sigma}_{W[m]}[k,n]\,\sqrt{1 + |c_{[m+1]}[k,n]/c_{[m]}[k,n]|^2}}, \end{aligned}$$

where $c_{[m+1]}[k,n]$ is the transform coefficient from the coarser decomposition level $m+1$ which is located at approximately the same spatial position as the coefficient $c_{[m]}[k,n]$. In our case, it is related to a waveform with approximately the same directionality as the waveform related to $c_{[m]}[k,n]$. If $\mathbf{C}_m = \left\{c_{[m]}[k,n]\right\}$ is the set of all transform coefficients from the decomposition level m, then $\mathbf{B}^{\lambda}_{W}\,\mathbf{C}_m$ is the application of the operator $\mathbf{B}^{\lambda}_{W}$ to each coefficient $c_{[m]}[k,n] \in \mathbf{C}_m$.

Thresholding parameters λ: Assume that the image to be restored comprises M pixels while K pixels are missing. Denote by $\rho = K/M$ the percentage of missing pixels. Assume that the noise STD=σ is known. Let $\lambda_{max} \overset{\text{def}}{=} 512$,

$$\lambda_{min} \overset{\text{def}}{=} \max\left\{1, \sigma\left(1 - \frac{\rho^2}{2}\right)\right\}, \quad \lambda_{mid} \overset{\text{def}}{=} \min\left\{\max\left\{2\,\lambda_{min} + 10\right\}, 20\right\}. \tag{3.3}$$

Let $r_1 \overset{\text{def}}{=} \lambda_{mid}/\lambda_{max} < 1$, $r_2 \overset{\text{def}}{=} \lambda_{min}/\lambda_{mid} < 1$. The sequences $\Lambda_1[i]$ and $\Lambda_2[i]$ are defined by

$$\Lambda_1[j] = \sqrt{2}\, r_1^{\frac{j-R_1}{R_1-1}}\, \lambda_{mid},\ j = 1, ..., R_1, \quad \Lambda_2[j] = \sqrt{2}\, r_2^{\frac{j-R_2}{R_2}}\, \lambda_{min},\ j = 1, ..., R_2 \tag{3.4}$$

where R_1 and R_2 are free parameters. In our experiments, we use $R_1 = 5$, $R_2 = 8$.

The parameters λ in Eq. (3.2) are taken from the sequences $\Lambda_1[j]$ and $\Lambda_2[j]$.

Preliminaries common to M1 and M2: The problem to be solved is to estimate an image $\mathbf{X}$ of size $N \times N$ from the degraded array $\check{\mathbf{X}} = \Theta \cdot (\mathbf{X} + \mathbf{n})$. Here Θ, which is referred to as the inpainting mask, is an $N \times N$ matrix consisting of ones and zeros and $\mathbf{n}$ is zero-mean Gaussian noise with STD=σ. We assume that the mask Θ and σ are known.

Denote by $\tilde{\mathbf{F}}^{+}_{[m]}$ and $\tilde{\mathbf{F}}_{-[m]}$ the operators of the qWP transforms of an image $\mathbf{X}$ to the coefficients of decomposition level m such that $\tilde{\mathbf{F}}_{\pm[m]}\,\mathbf{X} = \mathbf{Z}_{\pm[m]}$. The transforms with the complex qWPs $\Psi^{p}_{++[m],j,l}$ and $\Psi^{p}_{+-[m],j,l}$, respectively. Denote by $\tilde{\mathbf{F}}_{[m]}$ the combined operator such that $\tilde{\mathbf{F}}_{[m]}\,\mathbf{X} = \mathbf{Z}_{[m]} = \mathbf{Z}_{+[m]} \bigcup \mathbf{Z}_{-[m]}$. Denote by $\mathbf{F}_{+[m]}$ and $\mathbf{F}_{-[m]}$ the operators of the reconstruction of images $\mathbf{X}_{\pm[m]}$ from the sets of the m-level transform coefficients $\mathbf{Z}+[m]$ and $\mathbf{Z}_{-[m]}$, respectively: $\mathbf{X}_{\pm[m]} = \mathbf{F}_{\pm[m]}\mathbf{Z}\pm[m]$. Denote by $\mathbf{F}_{[m]}$ the operator of the full reconstruction of the image from the coefficient array $\mathbf{Z}_{[m]} = \mathbf{Z}_{+[m]} \bigcup \mathbf{Z}_{-[m]}$. It means that $\mathbf{F}_{[m]}\,\mathbf{Z}_{[m]} = \mathbf{X}_{[m]} = \mathfrak{Re}(\mathbf{X}_{+[m]} + \mathbf{X}_{-[m]})/8$. The solution is obtained by a weighted average of reconstruction results from several levels such as $\tilde{\mathbf{X}} = \frac{\sum_m \beta_m \mathbf{X}_{[m]}}{\sum_m \beta_m}$, where either $m = 3, 4$ or $m = 2, 3, 4$.

Settings common to M1 and M2: The following free parameters' values are set in advance: 1. Order p of the DTS that generates qWPs (in most cases $p = 3$, 4 or 5). 2. Integers R_1 and R_2 (Eq. (3.4)) and tolerance parameters tol_1 and tol_2 (typically, $tol_1 = 0.05$ and $tol_2 = 0.01$). 3. Iterations limits L_1, L_2 and L_3; 4. Windows spans $\{W_m\}$ for the averaged variances' $\bar{\sigma}_{W[m]}[k, n]$ calculation (Eq. (3.1)) and balance weights $\{\beta_m\}$, where $m = (2,)3, 4$.

In order to eliminate boundary effects, the degraded image $\check{\mathbf{X}}$ of size $N \times N$ is symmetrically extended to the image $\mathbf{Y}$ of size $N_1 \times N_1$, where $N_1 = N + 2T$. Typically, $T = N/8$. Respectively, the mask Θ is extended to Θ_1.

3.1 Method1

Assume that $m = 3, 4$. Extension to $m = 2, 3, 4$ is straightforward. This method is a slight modification of **AI**. The solution scheme is based on the assumption that the original image $\mathbf{X}$ can be sparsely represented in the qWP transform domain.

AI (and **M1**) consist of iterative thresholding of the transform coefficients with decreasing localized thresholds. After setting a few parameters, the threshold levels are determined automatically by using BSA.

Initialization: Let $\mathbf{X}^0 = \mathbf{X}^1 = 0$, $k = \nu = 1$. Compute the parameter $\rho = \frac{\sum_{l,n=1}^{N_1} \Theta_1[l,n]}{N_1^2}$ and sequences Λ_1 and Λ_2 (Eqs. (3.3) and (3.4)). The thresholding parameter is $\lambda = \Lambda_1[1]$. Then, the iterations have the following steps:

1. $\mathbf{Y}^k = \Theta_1 \cdot Y + (\mathbf{X}^k - \Theta_1 \cdot \mathbf{X}^k)$;
2. Compute $\mathbf{Z}^k_{\pm[m]} = \tilde{\mathbf{F}}_{\pm[m]}\,\mathbf{Y}^k,\ m = 3, 4, 5$[5];
3. Select the thresholding parameter λ/(stop the iterations) by the **Select/stop rule** (below);
4. Update the arrays $\mathbf{Z}^k_{\pm[m]}$ by the application of BSA operators $\mathbf{B}^\lambda_{W[m]}$, $m = 3, 4$, to these arrays: $\mathbf{Z}^{k+1}_{\pm[m]} = \mathbf{B}^\lambda_{W[m]}\,\mathbf{Z}^k_{\pm[m]}, \quad m = 3, 4$;
5. Apply the inverse qWP transforms to the arrays $\mathbf{Z}^{k+1}_{[m]}$, where $m = 3, 4$,
$$\mathbf{X}^{k+1}_{[m]} = \mathbf{F}_{[m]}\,\mathbf{Z}^{k+1}_{[m]}, \quad \mathbf{X}^{k+1} \stackrel{\text{def}}{=} \frac{\beta_3 \mathbf{X}^k_{[3]} + \beta_4 \mathbf{X}^k_{[4]}}{\beta_3 + \beta_4};$$
6. $k = k + 1$. Go to 1.

Select/stop rule common to M1 and M2: The rule, to some extent, is similar to the rule in **AI** in [23]. Iterations limits are L_1, L_2 and L_3 (not present in **AI**). Initial thresholding parameter is $\lambda = \Lambda_1[1]$ and $\nu = 1$. Denote the number of iterations with a certain index ν by K_ν.

- The l_2 norm of the difference $\Delta^k = \left\|\mathbf{X}^k - \mathbf{X}^{k-1}\right\|_2$ is computed;
- If $\nu < R_1$ **and** $\left\{K_\nu > L_1 \text{ **or** } \Delta^k < tol_1\right\}$, then $\nu = \nu + 1$ and $\lambda = \Lambda_1[\nu]$;
- If $R_1 \le \nu < R_1 + R_2$ **and** $\left\{K_\nu > L_2 \text{ **or** } \Delta^k < tol_2\right\}$, then $\nu = \nu + 1$ and $\lambda = \Lambda_2[\nu - R_1]$;
- If $\nu = R_1 + R_2$ **and** $\left\{K_\nu > L_3 \text{ **or** } \Delta^k < tol_2\right\}$, then **STOP** iterations;
- The array $\mathbf{X}^k$ is shrunk to the original size $N \times N$ as $\mathbf{X}^k \succ \tilde{\mathbf{X}}$, and the array $\tilde{\mathbf{X}}$ is taken as a solution to the inpainting problem.

3.2 Method2

An image restoration scheme is developed in Chapter 18 in [5] by utilizing non-directional 2D wavelet frames designed in Chapter 17 in [5]. Images are restored in [5] by the application of the *split Bregman iteration* (SBI) scheme [13] that uses the so-called *å-based* approach (see for example [18]).

[5] The coefficients $\mathbf{Z}^k_{\pm[5]}$ from the fifth decomposition level are used in the BSA operators.

M2 to be described couples **M1** with the SBI scheme. In essence, it is the SBI algorithm that uses qWPs and is supplied with decreasing localized thresholds determined by BSA and the *Select/stop rule.* In addition to the settings that are common to **M1** and **M2**, we set the regularization parameter μ.

Initialization: Let $k = \nu = 1$, $\mathbf{X}^0 = 0$, $\mathbf{b}_{+[m]}^0 = \mathbf{b}_{-[m]}^0 = 0$, $\mathbf{b}_{[m]}^0 = \mathbf{b}_{+[m]}^0 \bigcup \mathbf{b}_{-[m]}^0$, $\mathbf{d}_{+[m]}^0 = \mathbf{d}_{-[m]}^0 = 0$, $\mathbf{d}_{[m]}^0 = \mathbf{d}_{+[m]}^0 \bigcup \mathbf{d}_{-[m]}^0$, $m = 3, 4, 5$[6]. Compute the parameter $\rho = \frac{\sum_{l,n=1}^{N_1} \Theta_1[l,n]}{N_1^2}$ and sequences Λ_1 and Λ_2 (Eqs. (3.3) and (3.4)). The thresholding parameter is $\lambda = \Lambda_1[1]$. Then, the iterations have the following steps:

1. Apply the inverse qWP transforms to the arrays $\mathbf{D}_{[m]}^{k-1} = \mathbf{d}_{[m]}^{k-1} - \mathbf{b}_{[m]}^{k-1}$, where $m = 3, 4, 5$:
$$\left\{\mathbf{x}_{[m]}^{k-1} \stackrel{\text{def}}{=} \mathbf{F}_{[m]}\,\mathbf{D}_{[m]}^{k-1},\right\}, \quad \mathbf{x}^{k-1} \stackrel{\text{def}}{=} \frac{\beta_3 \mathbf{x}_{[3]}^{k-1} + \beta_4 \mathbf{x}_{[4]}^{k-1}}{\beta_3 + \beta_4};$$

2. The next iteration $\mathbf{X}^k$ is derived from Eq. (3.5) that is solved by the *conjugate gradient* algorithm:
$$\mathbf{X}^k := \Theta_1 \cdot \mathbf{x} + \mu\,\mathbf{x} = \mathbf{Y} + \mu\,\mathbf{x}^{k-1}; \tag{3.5}$$

3. Select the thresholding parameter λ/(stop the iterations) by the **Select/stop rule** (above).

4. Apply the forward qWP transforms to the array $\mathbf{X}^k$: $\tilde{\mathbf{F}}_{[m]}^{\pm}\,\mathbf{X^k} = \mathbf{Z}_{\pm[m]}^k$, and denote $\tilde{\mathbf{Z}}_{\pm[m]}^k = \mathbf{Z}_{\pm[m]}^k + \mathbf{b}_{\pm[m]}^{k-1}$, where $m = 3, 4, 5$;

5. Update the arrays $\mathbf{d}_{\pm[m]}^{k-1}$ by application of the BSA operators $\mathbf{B}_{W[m]}^{\lambda}$, $m = 3, 4$, to the arrays $\tilde{\mathbf{Z}}_{\pm[m]}^k$: $\mathbf{d}_{\pm[m]}^k = \mathbf{B}_{W[m]}^{\lambda}\,\tilde{\mathbf{Z}}_{\pm[m]}^k$, $\quad \mathbf{d}_{[m]}^k = \mathbf{d}_{+[m]}^k \bigcup \mathbf{d}_{-[m]}^k$, $\quad m = 3, 4, 5$;

6. Update the arrays $\mathbf{b}_{\pm[m]}^{k-1}$ by $\mathbf{b}_{\pm[m]}^k = \mathbf{b}_{\pm[m]}^{k-1} + \left(\tilde{\mathbf{Z}}_{\pm[m]}^k - \mathbf{d}_{\pm[m]}^k\right)$, $\quad \mathbf{b}_{[m]}^k = \mathbf{b}_{+[m]}^k \bigcup \mathbf{b}_{-[m]}^k$, $\quad m = 3, 4, 5$;

7. $k = k + 1$. Go to item 1.

4 Experimental results

For the experiments, we use a standard set of benchmark images, which was used in [9]: "Lena", "Boat", "Barbara", and "Fingerprint" with two additional images "Hill" and "Mandrill". The "clean" images are displayed in Fig. 4.1. The performances of **M1** and **M2** are compared with the performances of the state-of-the-art inpainting algorithms that use the SET-4 filter banks. The inpainting results by the SET-4 algorithms applied to the first four images from Fig. 4.1, which are degraded by the application of four masks displayed in Fig. 4.1 and additive Gaussian noise with various intensities (STD $\sigma = 0, 5, 10, 30, 50$ dB), are presented in [9]. The Matlab codes, which produce the results in [9], are available at
http://staffweb1.cityu.edu.hk/xzhuang7/softs/index.html#bdTPCTF. We used these codes to restore the additional images "Hill" and "Mandrill". For each triple *image-mask-σ*, the inpainting

[6] The coefficients $\mathbf{d}_{[5]}^k$, $\mathbf{b}_{[5]}^k$ from the fifth decomposition level are used in the BSA operators.

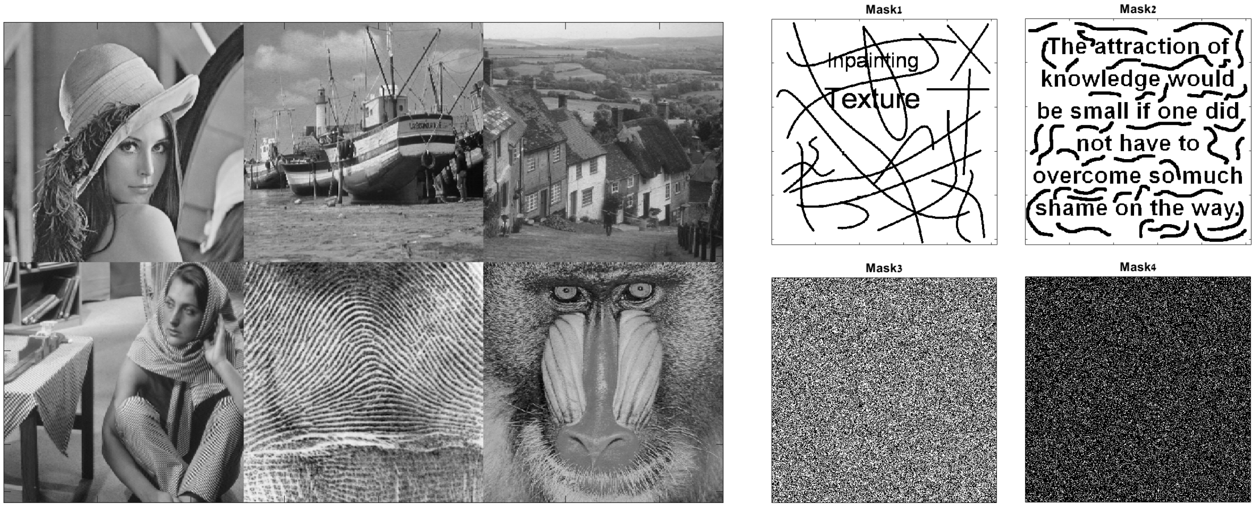
Mask1
Inpainting
Texture
Mask2
The attraction of knowledge would be small if one did not have to overcome so much shame on the way.
Mask3
Mask4

Figure 4.1: Left: Clean images: “Lena”, “Boat”, “Hill”, “Barbara”, “Fingerprint” and “Mandrill”. Right: Four types of masks: Mask1 and Mask2 – text and curves; Mask3 and Mask4– 50% and 80% of random pixels missing, respectively

results from the application of our **M1** and **M2** are compared with the best result produced by the application of the SET-4 algorithms. The results are compared according to PSNR and SSIM values and by visual inspection.

Remark 4.1 *SSIM values for the restored images are computed by using the Matlab function* `ssim`*. It should be noted that the* `ssim` *function in the Matlab 2020a version produces higher SSIM values than Matlab 2019b. The SSIM results listed below are computed using Matlab 2019b.*

4.1 Experiments with “Barbara” and “Fingerprint” images

The evaluated images have complicated structure which comprises smooth regions, multiple edges (in “Barbara”) and oscillatory patterns (in both images). The restoration results in PSNR and SSIM measurements are given in Table 4.1.

Figure 4.2 illustrates the restoration of the “Barbara” image, which was degraded by the application of mask2 and by adding a strong Gaussian noise with $\sigma = 50$ dB, by **M2** and DAS-2. Both algorithms successfully inpainted the mask and suppressed the noise but, in doing so, DAS-2 loses fine texture details and adds some artifacts into the image. Figure 4.2 demonstrates it.

Figure 4.3 illustrates the restoration of the “Fingerprint” image, which was severely degraded by the application of mask3 (50% of the pixels are missing) and strong additive Gaussian noise with $\sigma = 50$ dB, by **M2** and DAS-2. In this case, the PSNR and SSIM for restoration by **M2** are higher than those for the restoration by DAS-2. Both algorithms successfully inpainte the mask and suppress the noise, but DAS-2 over-smooths some parts of the image. It is seen in Fig. 4.3.

4.2 Experiments with “Lena” and “Boat” images

These images are characterized by a number of edges separating smooth areas and some fine details such as hair and the hat decoration in “Lena” and the ground in “Boat”. The PSNR and SSIM results are given in Table 4.2.

Barbara				Fingerprint		
σ	**M1**	**M2**	Best from SET-4	**M1**	**M2**	Best from SET-4
mask1						
0	38.13/0.958	**38.42/0.967**	36.68/0.958 (4)	31.93/0.978	**32.64/0.980**	31.72/0.976 (4)
5	34.93/0.8256	**34.94/0.831**	34.05/0.8138(1)	30.69/0.963	**30.86/0.964**	30.36/0.960 (1)
10	**32.57**/0.748	32.56/**0.753**	32.02/0.739 (1)	28.96/0.939	**29.09/0.940**	28.77/0.935 (1)
30	**28.09**/0.601	28.01/**0.603**	27.74/0.592 (1)	25.15/0.843	**25.25/0.852**	25.04/0.839(1)
50	**25.85**/ 0.501	25.77/**0.504**	25.51/0.489 (1)	23.26/0.767	**23.42/0.778**	23.21/0.766 (1)
mask2						
0	34.26/0.922	**34.70/0.932**	33.66/0.923 (1)	28.76/0.949	**28.94/0.950**	28.31/0.942 (4)
5	**32.78**/0.80	32.75 **/0.806**	32.24/0.787 (1)	28.02/**0.938**	**28.31**/0.936	27.68/0.926(1)
10	**31.13**/0.730	31.11 **/0.734**	30.71/0.718 (1)	27.01/0.908	**27.08/0.911**	26,76/0.903 (1)
30	**27.22**/0.579	27.1/**0.581**	26.95/0.570 (1)	24.16/0.812	**24.35/0.822**	24.15/0.809 (1)
50	**25.21**/0.487	25.13/**0.487**	24.96/0.471 (1)	22.58/0.739	**22.77/0.753**	22.65/0.738 (1)
mask3						
0	37.30/0.901	**37.48/0.913**	35.72/0.9051 (1)	34.53/0.979	**34.54/0.979**	34.19/0.977 (4)
5	**34.24**/0.789	34.21/ **0.793**	33.29/0.778 (1)	31.72/0.958	**31.92/0.961**	31.54/0.957 (4)
10	31.82/0.721	**31.85/0.723**	31.12/0.707 (1)	29.14/0.926	**29.37/0.932**	29.09/0.927 (4)
30	27.1/0.571	**27.17/0.572**	26.77/0.554 (1)	24.51/0.809	**24.73/0.825**	24.43/0.806 (4)
50	24.84/0.460	**24.9/0.467**	24.60/0.441 (1)	22.48/0.721	**22.69/0.748**	22.53/0.722 (1)
mask4						
0	30.17/0.7609	**30.34/0.779**	29.12/0.753 (1)	**27.03**/0.899	27.02/0.9	26.77/0.894 (4)
5	29.28/0.687	**29.44/0.696**	28.41/0.673 (1)	26.27/0.878	**26.29/0.881**	25.87/0.872(4)
10	28.01/0.626	**28.19/0.633**	27.25/0.609 (1)	25.11/0.843	**25.11/0.845**	24.60/0.834 (4)
30	**24.59**/0.452	24.45/**0.461**	24.23/0.434(1)	22.09/0.712	**22.15/0.731**	22.11/0.7115 (1)
50	**22.67**/0.347	22.58/**0.365**	22.35/325 (1)	20.32/0.613	20.31/**0.651**	**20.61**/0.614 (1)

Table 4.1: PSNR/SSIM values for the restoration of "Barbara" (columns 2–4) and "Fingerprint" (columns 5–7) images by **M1**, **M2** and the best algorithm from SET-4. Boldface highlights the best results. σ – noise STD. Numbers in parentheses indicate which is the best algorithm from SET-4 that produces the best result: (1) means DAS-2, (2)–DAS-1, (3)–TP-$\mathbb{C}$TF$_6$, (4)–TP-$\mathbb{C}$TF$_6^{\downarrow}$

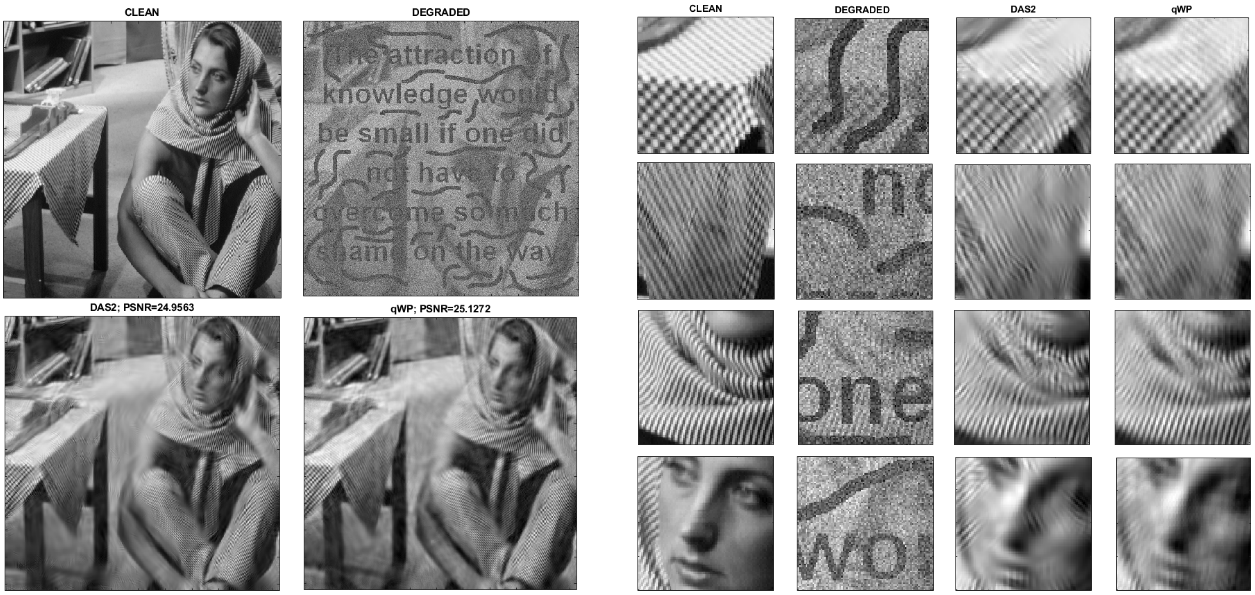

Figure 4.2: Restoration of “Barbara” image. **Left:** Top left: clean image. Top right: image degraded by application to it mask2 and Gaussian noise with $\sigma = 50$ dB. Bottom: Restoration by **M2** (right), PSNR=25.13 dB, SSIM=0.4868 and by DAS-2 (left), PSNR=24.96 dB, SSIM=0.4709. **Right:** Fragments of images. Columns: First - clean fragments; Second – degraded; Third– restored by DAS-2; Fourth – restored by **M2**

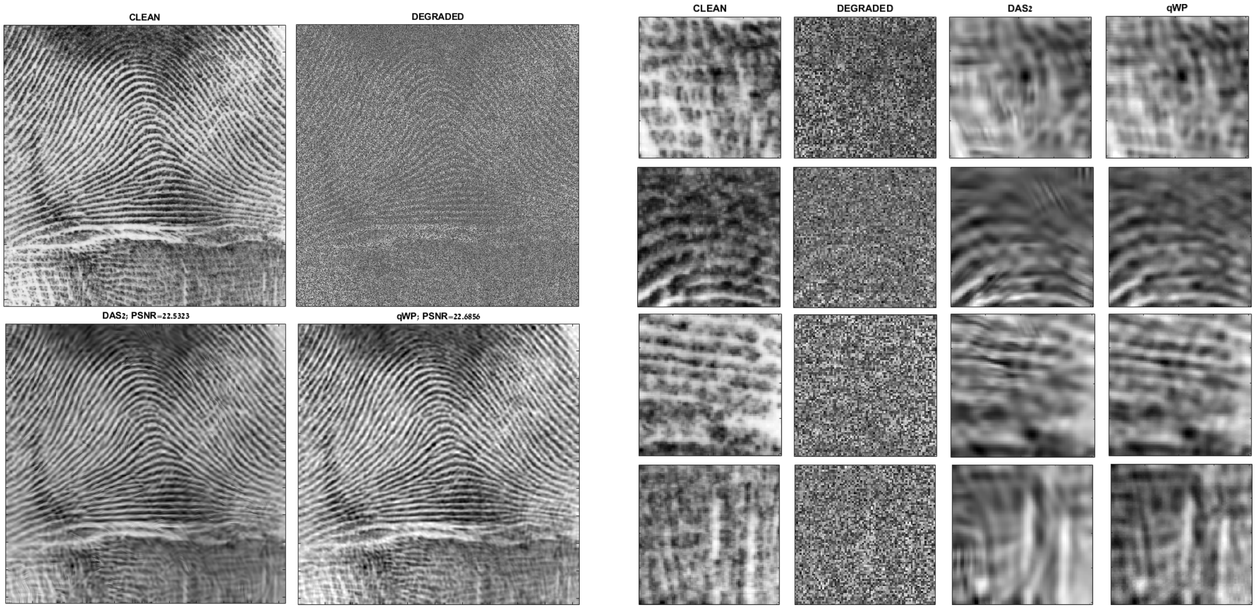

Figure 4.3: **Left:** Restoration of “Fingerprint” image. Top left: clean image. Top right: image degraded by application to it mask3 and Gaussian noise with $\sigma = 50$ dB. Bottom: Restoration by **M2** (right), PSNR=22.69 dB, SSIM=0.748 and by DAS-2 (left), PSNR=22.53 dB, SSIM=0.7224. **Right:** Fragments of images. Columns: First - clean fragments; Second – degraded; Third– restored by DAS-2; Fourth – restored by **M2**

	Lena			Boat		
σ	**M1**	**M2**	Best from SET-4	**M1**	**M2**	Best from SET-4
mask1						
0	37.50/0.9371	37.84/**0.9488**	**38.02**/0.9487 (3)	34.78/0.936	34.95/**0.94072**	**34.96**/0.94071 (3)
5	34.94/0.702	34.95/**0.724**	**35.19**/0.672 (3)	32.68/0.778	32.68/**0.792**	**32.81**/0.752 (3)
10	33.18/0.605	33.22/**0.614**	**33.42**/0.586 (3)	30.79/0.659	30.75/**0.674**	**31.04**/0.630 (3)
30	29.34/0.449	29.43/**0.454**	**29.81**/0.449 (3)	27.03/0.454	27.02/**0.463**	**27.41**/0.433 (3)
50	27.31/0.370	27.40/**0.377**	**27.85**/0.377 (3)	25.22/0.348	25.15/**0.360**	**25.6**/0.334 (2)
mask2						
0	33.72/0.884	33.96/**0.9005**	**34.3**/0.9001 (3)	30.23/0.874	30.57/0.880	**30.80/0.881** (3)
5	32.48/0.661	32.43/**0.692**	**32.97**/0.647 (3)	29.30/0.721	29.29/**0.741**	**29.79**/0.718 (1)
10	31.39/0.582	31.37/**0.593**	**31.78**/0.567 (3)	28.18/0.611	28.25/**0.630**	**28.85**/0.59 (2)
30	28.39/0.432	28.51/**0.440**	**28.89**/0.434 (3)	25.78/0.417	25.83/**0.430**	**26.34**/0.407 (2)
50	26.68/0.356	26.65/**0.364**	**27.22**/0.362 (3)	24.33/0.319	24.37/**0.334**	**24.84**/0.312 (2)
mask3						
0	37.70/0.813	37.95/**0.8528**	38.0/0.852 (3)	34.39/0.8518	**34.47/0.865**	34.42/0.858 (3)
5	35.14/0.647	35.17/**0.660**	**35.40**/0.632 (3)	32.26/0.700	32.29/**0.731**	**32.50**/0.684 (3)
10	33.03/0.568	33.04/**0.577**	**33.40**/0.558 (3)	30.3/0.598	30.41/**0.623**	**30.65**/0.586 (3)
30	28.67/0.418	28.81/**0.426**	**29.18**/0.421 (3)	26.33/0.405	26.32/**0.428**	**26.66**/0.380 (3)
50	26.51/0.341	26.65/**0.349**	**27.06**/0.346 (3)	24.43/ 0.299	24.49/**0.335**	**24.75**/0.278 (3)
mask4						
0	32.09/0.6311	32.16/**0.6756**	32.33/0.6740 (3)	28.39/0.629	28.25/**0.644**	**28.58**/0.643 (3)
5	31.18/0.552	31.15/ **0.553**	**31.44**/0.551 (3)	27.7/0.544	27.71/**0.558**	**27.98**/0.546 (3)
10	29.86/0.484	29.84/**0.4874**	**30.25** /0.487(3)	26.77/0.465	26.73/**0.480**	**27.08**/0.465 (3)
30	26.40/0.3406	26.52/**0.349**	**26.95**/0.347 (3)	24.01/0.278	24.0/**0.308**	**24.46**/0.277 (3)
50	24.25/0.260	24.47/**0.2753**	**25.15**/0.2746 (3)	22.50/0.196	22.22/**0.233**	**22.96**/0.193 (3)

Table 4.2: PSNR/SSIM values for the restoration of "Lena" (columns 2–4) and "Boat" (columns 5–7) images by **M1**, **M2** and the best from SET-4 algorithms. Boldface highlights the best results. σ – noise STD. Numbers in parentheses indicate which algorithm in SET-4 produces the best result: (1) means DAS-2, (2)–DAS-1, (3)–TP-$\mathbb{C}$TF$_6$, (4)–TP-$\mathbb{C}$TF$_6^{\downarrow}$

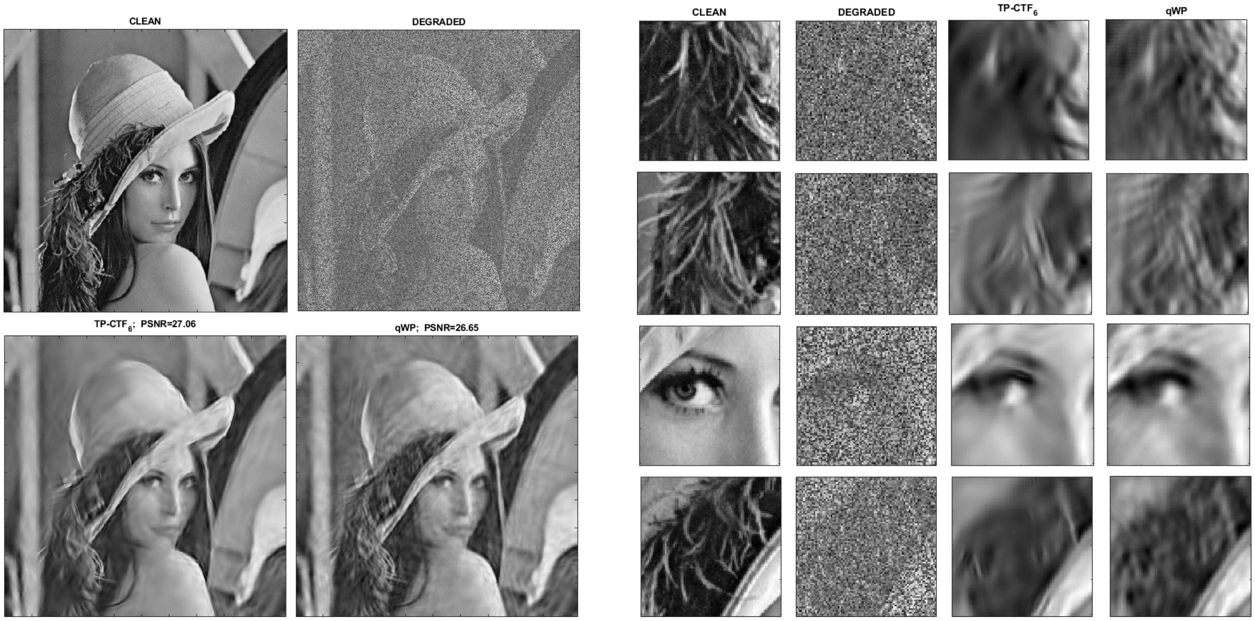

Figure 4.4: Restoration of the “Lena” image. **Left:**Top left: clean image. Top right: image degraded by application of mask3 and Gaussian noise with $\sigma = 50$ dB. Bottom: Restoration by **M2** (right), PSNR=26.65 dB, SSIM=0.3494 and by TP-$\mathbb{C}$TF$_6$ (left), PSNR=27.05 dB, SSIM=0.3457. **Right:** Fragments of images. Columns: First - clean fragments; Second – degraded; Third– restored by DAS-2; Fourth – restored by **M2**

Figure 4.4 displays the restoration of the “Lena” image, which was severely degraded by the application of mask3 (50% of pixels are missing), and strong Gaussian noise with $\sigma = 50$ dB, by **M2** and TP-$\mathbb{C}$TF$_6$, which was the best from the SET-4 algorithms. The PSNR of the restoration by TP-$\mathbb{C}$TF$_6$ is higher than what **M2** has produced. On the other hand, the SSIM values are close to each other while some fine details, which are lost by TP-$\mathbb{C}$TF$_6$, are retained by **M2**.

Figure 4.5 displays the results of the “Boat” restoration by **M2** and TP-$\mathbb{C}$TF$_6$, which produced the highest PSNR from the SET-4 algorithms. The image was degraded by the application of mask3 (50% of pixels are missing) and strong additive Gaussian noise with $\sigma = 50$ dB. Although the PSNR for restoration by TP-$\mathbb{C}$TF$_6$ is higher than what **M2** has produced, the **M2**-SSIM value is much higher compared to TP-$\mathbb{C}$TF$_6$. The image restored by TP-$\mathbb{C}$TF$_6$ looks over-smoothed compared to restoration by **M2**. Some edges and almost the entire texture are lost by the application of TP-$\mathbb{C}$TF$_6$ but are retained by **M2**.

4.3 Additional examples: “Hill” and “Mandrill”

We conducted experiments with the images “Hill” and “Mandrill” that were not used in [9]. The former image is smooth up to the texture on the roofs and ground, whereas the most part of the latter one consists of very fine texture. The performance of **M2** is compared with the performances of algorithms that use the SET-4 filter banks. The inpainting results by the SET-4 algorithms are derived using the Matlab codes at the web site
http://staffweb1.cityu.edu.hk/xzhuang7/softs/index.html#bdTPCTF. The PSNR and SSIM re-

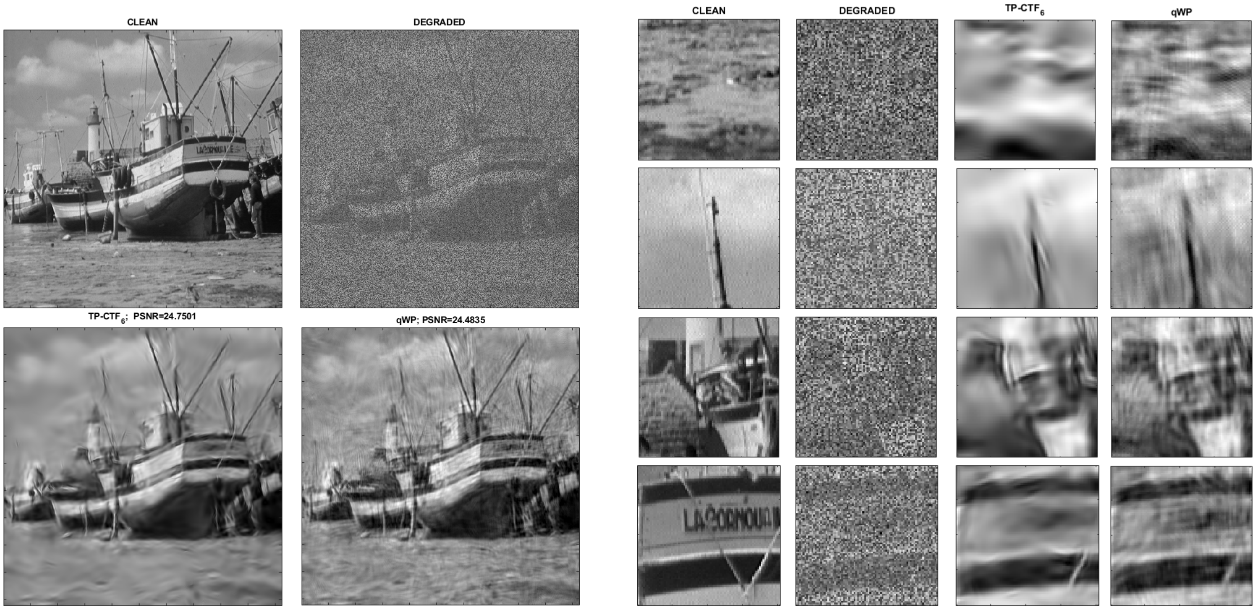

Figure 4.5: Restoration of "Boat" image. **Left:**Top left: clean image. Top right: image degraded by application of mask3 and Gaussian noise with $\sigma = 50$ dB. Bottom: Restoration by **M2** (right), PSNR=24.48 dB, SSIM=0.3347 and by TP-$\mathbb{C}$TF$_6$ (left), PSNR=24.75 dB, SSIM=0.2777. **Right:** Fragments of images. Columns: First - clean fragments; Second – degraded; Third– restored by DAS-2; Fourth – restored by **M2**

sults are given in Table 4.3. In most experiments with the "Hill" image, the PSNR values produced by SET-4 algorithms are higher than the values produced by **M2** but for the SSIM values the situation is quite opposite. In many experiments, the **M2**-SSIM values are much higher compared to those from SET-4. For the "Mandrill" image, all the SSIM and almost all the PSNR values produced by the SET-4 algorithms are inferior to those produced by **M2**.

Figure 4.6 displays the restoration results of the "Hill" image, which was degraded by the application of mask3 (50% of pixels are missing), and moderate Gaussian noise with $\sigma = 10$ dB, by **M2** and TP-$\mathbb{C}$TF$_6$, which has produced the highest PSNR out of the SET-4 algorithms. The PSNR of the restoration by TP-$\mathbb{C}$TF$_6$ is higher than that from **M2**. On the other hand, the SSIM value from the **M2** restoration significantly exceeds that from TP-$\mathbb{C}$TF$_6$. Consequently, fine details, which are lost by TP-$\mathbb{C}$TF$_6$, are retained by **M2**.

Figure 4.7 displays the restoration results of the "Mandrill" image, which was degraded by the application of mask3 (50% of pixels are missing) and strong additive Gaussian noise with $\sigma = 50$ dB, by **M2** and DAS-2, which has produced the highest PSNR from the SET-4 algorithms. Both the PSNR and SSIM values for restoration by **M2** are higher than the values produced by DAS-2. The image restored by DAS-2 looks over-smoothed compared to restoration by **M2**. Much of the fine texture details, which are lost by the application of DAS-2, are retained by **M2**.

4.4 A special example

All the above experiments were conducted with the inpainting masks shown in Fig. 4.1 where missing pixels are either arranged as texts and curves or are randomly distributed. However, our

	Hill		Mandrill	
σ	**M2**	Best from SET-4	**M2**	Best from SET-4
mask1				
0	**36.2/0.9419**	35.97/0.9403 (3)	**30.13/0.9291**	30.02/0.9278 (1)
5	**33.55/0.8082**	33.49/0.7851 (3)	**28.99/0.8678**	28.89/0.8609 (1)
10	**31.53/0.7033**	31.41/0.6816 (1)	**27.32/0.7864**	27.22/0.7667
30	27.81/**0.4774**	**27.89**/0.4556 (1)	**23.57/0.5634**	23.51/0.5333
50	25.99/**0.3683**	**26.19**/0.3473 (1)	**21.89/0.412**	21.86/0.3875
mask2				
0	32.44/**0.8884**	**32.74**/0.8876 (3)	**27.43/0.8652**	27.37/0.8638 (1)
5	31.11/**0.765**	**31.37**/0.7465 (1)	**26.76/0.805**	26.74/0.8(1)
10	29.79/**0.6618**	**30.03**/0.6434 (1)	**25.76/0.7325**	25.71/0.7106 (1)
30	26.78/**0.4404**	**27.15**/0.4245 (1)	**22.97/0.5238**	22.96/0.4897 (1)
50	25.38/**0.3427**	**25.69**/0.3229 (1)	**21.5601/0.3843**	21.558/0.3495(2)
mask3				
0	**34.69/0.8674**	34.52.0/0.8667 (3)	**26.44/ 0.8276**	25.92/0.8127 (1)
5	32.57/**0.7529**	**32.59**/0.7326 (3)	**25.88/0.7768**	25.17/0.7413 (1)
10	30.64/**0.6522**	**30.7**/0.6077 (3)	**25/0.7028**	24.42/0.6653 (1)
30	26.87/**0.4369**	27.14/0.3948 (1)	**22.39/0.5017**	22.2/0.441 (1)
50	25.2/**0.3245**	**25.4**/0.26 (1)	**21.05/ 0.355**	20.99/0.2887 (1)
mask4				
0	29.34/**0.6756**	29.59/0.6659 (3)	**21.84/0.5644**	21.48/0.5431 (1)
5	28.58/ **0.5846**	**28.87**/0.5736 (3)	**21.54/0.5328**	20.93/0.4951 (1)
10	27.59/**0.4978**	**17.91** /0.4731(3)	**21.17/0.4773**	20.67/0.4455 (1)
30	24.67/**0.3103**	**25.35**/0.2935 (1)	19.8/**0.3264**	**20.23**/0.2977 (1)
50	23.33/**0.2753**	**24.06**/0.2094 (1)	18.93/**0.2323**	**19.81**/0.1414 (1)

Table 4.3: PSNR/SSIM values for the restoration of "Hill" (columns 2–3) and "Mandrill" (columns 4–5) images by **M2** and the best from SET-4 algorithms. Boldface highlights the best results. σ – noise STD. Numbers in parentheses indicate which algorithm in SET-4 produces the best result: (1) means DAS-2, (3)–TP-$\mathbb{C}$TF$_6$

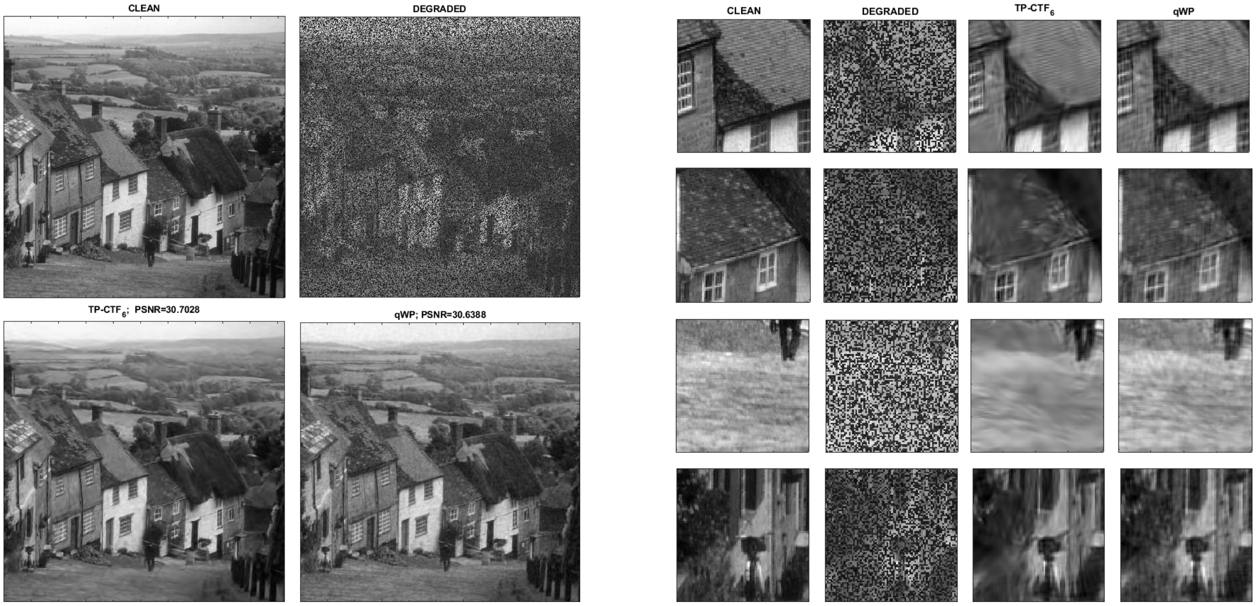

Figure 4.6: Restoration of "Hill" image. **Left:**Top left: clean image. Top right: image degraded by application of mask3 and Gaussian noise with $\sigma = 10$ dB. Bottom: Restoration by **M2** (right), PSNR=30.64 dB, SSIM=0.6522 and by TP-$\mathbb{C}$TF$_6$ (left), PSNR=30.7 dB, SSIM=0.6077. **Right:** Fragments of images. Columns: First - clean fragments; Second – degraded; Third– restored by TP-$\mathbb{C}$TF$_6$; Fourth – restored by **M2**

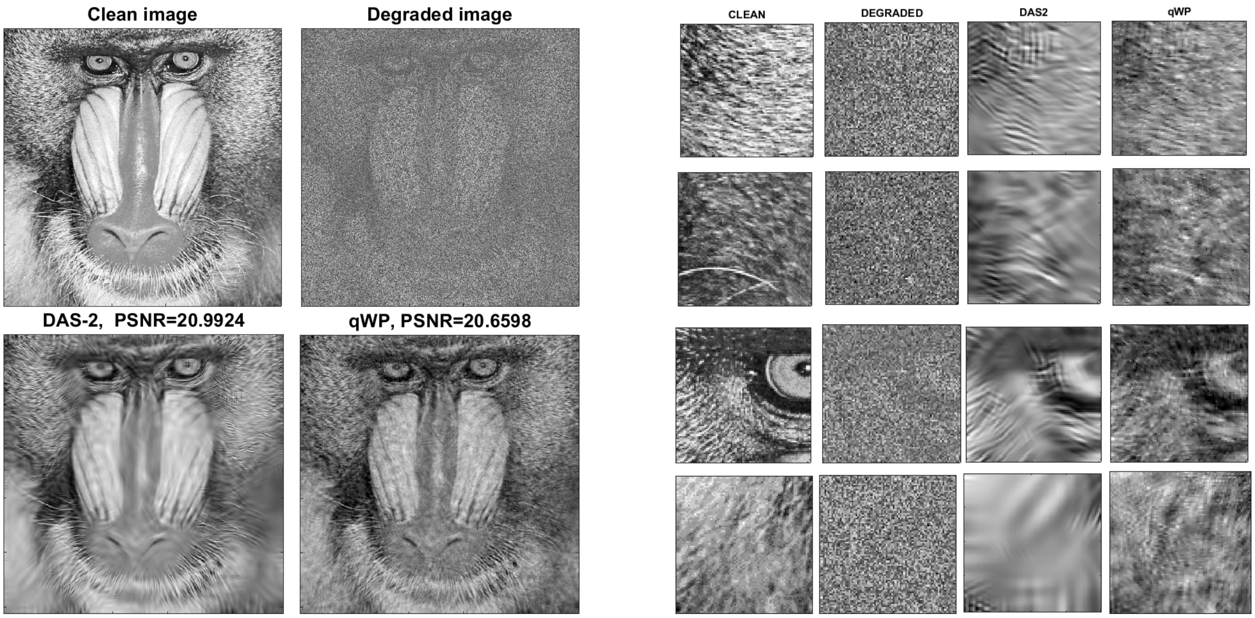

Figure 4.7: Restoration of the "Mandrill" image. **Left:**Top left: clean image. Top right: image degraded by application of mask3 and Gaussian noise with $\sigma = 50$ dB. Bottom: Restoration by **M2** (right), PSNR=21.05 dB, SSIM=0.355 and by DAS-2 (left), PSNR=20.99 dB, SSIM=0.2887. **Right:** Fragments of images. Columns: First - clean fragments; Second – degraded; Third– restored by DAS-2; Fourth – restored by **M2**

qWP-based algorithms are able to inpaint images where large patches are missing. In the following experiment with "Barbara'" image, mask4 (80% of pixels missing) was modified by addition of several 16×16 missing squares and one 32×32 missing square. Noise was not added. The **M2** performance was compared with the performance of SET-4 algorithms, from which DAS-2 appeared the best. Figure 4.8 illustrates results of this experiment. We can see in Fig. 4.8 that, while DAS-2 simply fills the gaps, **M2** restores missing complicated structures.

5 Discussion

The paper presents two methods **M1** and **M2** for image impainitng. These methods are based on the directional quasi-analytic WPs (qWPs) originated from polynomial splines of arbitrary order that are designed in [3]. The computational scheme in **M1** is a slight modification of the inpainting **AlgorithmI** introduced in [23]. **M2** couples **M1** with the Split Bregman Iterations (SBI) scheme. In essence, it is the SBI algorithm supplied with decreasing localized thresholds that are determined by BSA.

The successful application of **M1** and **M2** to image inpainting stems from the exclusive properties of qWPs such as:

- The qWP transforms provide a variety of 2D waveforms oriented in multiple directions. For example, fourth-level qWPs are oriented in 62 different directions.

- The waveforms are close to directional cosines with a variety of frequencies that are modulated

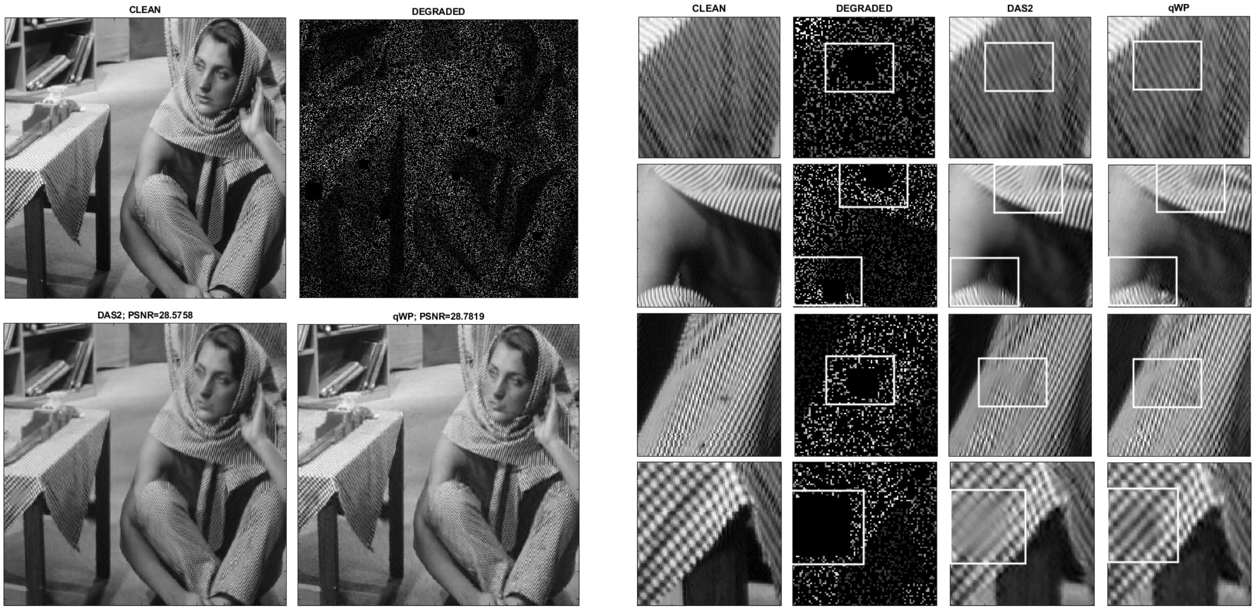

Figure 4.8: Restoration of “Barbara” image. **Left:** Top left: clean image. Top right: image degraded by application of modified mask4 (80% of pixels and square patches are missing). Bottom: Restoration by **M2** (right), PSNR=28.78 dB, SSIM=0.7312 and by DAS-2 (left), PSNR=28.58 dB, SSIM=0.7471. **Right:** Image fragments. Columns: First - clean fragments; Second – degraded; Third– restored by DAS-2; Fourth – restored by **M2**. Bottom row – gap of size 32×32, in the rest of rows – gaps of size 16×16

by spatially localized low-frequency 2D signals.

- The waveforms may have any number of local vanishing moments.

- The DFT spectra of the waveforms produce a refined tiling of the frequency domain.

- The transforms have a number of free parameters that makes it possible a flexible adaptation to objects under processing.

- Fast implementation of the transforms by using FFT that enables us to use the transforms with increased redundancy.

In multiple experiments, the performances of **M1** and **M2** are compared with the performances of the state-of-the-art inpainting algorithms that use the SET-4 filter banks. The inpainting results for 4 images by the SET-4 algorithms, which are degraded by the application of 4 masks (Fig. 4.1) and additive Gaussian noise of various intensities (noise STD $\sigma = 0, 5, 10, 30, 50$ dB), are presented in [9]. We added two images, which were not used in [9]. For each triple *image-mask-*σ, the inpainting results by **M1** and **M2** vs. the SET-4 algorithms are compared through PSNR, SSIM values and visual inspection. In all the experiments, our methods succeeded in inpainting even in severely degraded images and in suppressing noise. In most cases, fine structures were recovered. On images with complicated structures such as "Barbara", "Fingerprint" and "Mandrill", **M1** and especially **M2** significantly outperform the SET-4 algorithms in all three criteria - PSNR, SSIM and visual perception. As for the "Lena", "Boat" and "Hill" images, while in most experiments the SET-4 algorithms produce higher PSNR values compared to **M1** and **M2**, the SSIM values are higher for **M1** and especially for **M2**. This fact indicates that our **M1** and **M2** are superior over the SET-4 algorithms for image structure recovery. Visual observations that confirm this claim are illustrated in Figs. 4.2 – 4.7. Note that **M2** succeeded in the restoration of the image fine structures even in the situations where large patches in the image such as 16×16 and 32×32 squares were missing. It is illustrated in Fig. 4.8.

Generally, we claim that our **M1** and especially **M2**, which are based on directional quasi-analytic wavelet packets originated from polynomial splines, are powerful tools for images' restoration that are severely degraded by loss of the majority of the pixels and have strong additive noise. These methods succeed in capturing fine details in the images in cases where other state-of-the-art algorithms fail. This fact is illustrated in Fig. 5.1, which displays the restoration results of the 'Mandrill" image from the input where 80% of pixels are missing and additive Gaussian noise with $\sigma = 50$ dB is present. The output from DAS-2 algorithm (which was the best for this image in SET-4) has PSNR=19.81 dB compared to 18.93 dB produced by **M2**. On the other hand, the SSIM from the **M2** restoration is 0.2343 compared to 0.1414 produced by DAS-2. We can see that many texture patterns that were blurred by DAS-2, are restored by **M2**. This figure is a good illustration to the fact that SSIM is a much more informative characteristics than PSNR.

Summarizing, by having such a versatile and flexible tool at hand, we are in a position to address several data processing problems such as image denoising, deblurring, superresolution, segmentation and classification, target detection (here the directionality is of utmost importance). The 3D directional wavelet packets, whose design is underway, may be beneficial for seismic and hyper-spectral processing.

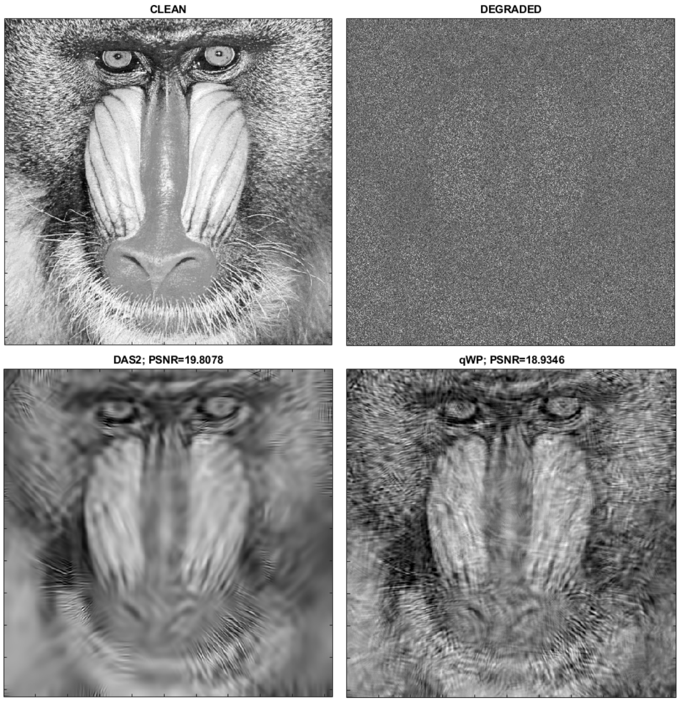

Figure 5.1: Restoration of the "Mandrill" image. Top left: clean image. Top right: image degraded by the application of mask4 with additive Gaussian noise with $\sigma = 50$ dB. Bottom right: **M2** restoration, PSNR=18.93 dB, SSIM=0.2343. Bottom left: DAS-2 restoration, PSNR=19.81 dB, SSIM=0.1414

Acknowledgment This research was partially supported by the Israel Science Foundation (ISF, 1556/17), Supported by Len Blavatnik and the Blavatnik Family Foundation, Israel Ministry of Science Technology and Space 3-16414, 3-14481 and by Academy of Finland (grant 311514).

References

[1] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and Y. Shkolnisky. A framework for discrete integral transformations I – the pseudopolar Fourier transform. *SIAM Journal on Scientific Computing*, 30(2):764–784, 2008.

[2] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, Y. Shkolnisky, and I. Sedelnikov. A framework for discrete integral transformations II – the 2d discrete Radon transform. *SIAM Journal on Scientific Computing*, 30(2):785–803, 2008.

[3] A. Averbuch, P. Neittaanmäki, and V. Zheludev. Directional wavelet packets originating from polynomial splines. http://arxiv.org/abs/2008.05364.

[4] A. Averbuch, P. Neittaanmäki, and V. Zheludev. *Splines and spline wavelet methods with application to signal and image processing, Volume III: Selected topics*. Springer, 2019.

[5] A. Z. Averbuch, P. Neittaanmäki, and V. A. Zheludev. *Spline and spline wavelet methods with applications to signal and image processing, Volume I: Periodic splines*. Springer, 2014.

[6] I. Bayram and I. W. Selesnick. On the dual-tree complex wavelet packet and m-band transforms. *IEEE Trans. Signal Process.*, 56:2298–2310, 2008.

[7] E. Candés, L. Demanet, D. Donoho, and L. X. Ying. Fast discrete curvelet transforms. *Multiscale Model. Simul.*, 5:861–899, 2006.

[8] E. Candés and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise c^2 singularities. *Commun. Pure Appl. Math.*, 57:219–266, 2004.

[9] Z. Che and X. Zhuang. Digital affine shear filter banks with 2-layer structure and their applications in image processing. *IEEE Trans. on Image Processing*, 27(8):3931–3941, 2018.

[10] L. Şendur and I. W. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans. Signal Process.*, 50:2744–2756, 2002.

[11] M. N. Do and M. Vetterli. Contourlets. In *Beyond Wavelets, G. V. Welland, ed.* Academic Press, San Diego, CA, 2008.

[12] W.-Q Lim G. Kutyniok and X. Zhuang. Digital shearlet transforms. In *Shearlets: Multiscale Analysis for Multivariate Data*, pages 239–282. Birkhäuser, Boston, 2012.

[13] T. Goldstein and S. Osher. The split Bregman method for $L1$-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.

[14] B. Han, Q. Mo, Z. Zhao, and X. Zhuang. Directional compactly supported tensor product complex tight framelets with applications to image denoising and inpainting. *SIAM J. Imaging Sci.*, 12(4):1739–1771, 2019.

[15] B. Han and Z. Zhao. Tensor product complex tight framelets with increasing directionality. *SIAM J. Imaging Sci.*, 7(2):997–1034, 2014.

[16] B. Han, Z. Zhao, and X. Zhuang. Directional tensor product complex tight framelets with low redundancy. *Appl. Comput. Harmon. Anal.*, 41(2):603–637, 2016.

[17] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Satellite image deconvolution using complex wavelet packets. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pages 809–812, 2000.

[18] H. Ji, Z. Shen, and Y. Xu. Wavelet based restoration of images with missing or damaged pixels. *East Asian J. Appl. Math.*, 1(2):108–131, 2011.

[19] N.G. Kingsbury. Image processing with complex wavelets. *Philos. Trans. R. Soc. London A, Math. Phys. Sci.*, 357(1760):2543–2560, 1999.

[20] G. Kutyniok and D. Labate. *Shearlets: Multiscale Analysis for Multivariate Data.* Birkhäuser, Boston, 2012.

[21] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing.* Prentice Hall, New York, 3rd edition, 2010.

[22] I.W. Selesnick, R.G. Baraniuk, and N.G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Process. Mag.*, 22(6):123—-151, 2005.

[23] Y. Shen, B. Han, and E. Braverman. Image inpainting using directional tesnor product complex tight framelets. *Preprint, arXiv: 1407.3234*, 2014.

[24] Y. Shen, B. Han, and E. Braverman. Image inpainting from partial noisy data by directional complex tight framelets. *Anziam J.*, 58(3–4):1–9, 2017.

[25] X. Zhuang. Digital affine shear transforms: fast realization and applications in image/video processing. *SIAM J. Imag. Sci.*, 9(3):1437–1466, 2016.