EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

Singular spectrum analysis-based image sub-band decomposition filter banks

Julia Evers[1*], Florian Evers[2], Florian Goppelt[1] and Ronald Schmidt-Vollus[1]

*Correspondence:
yulia.evers@gmx.de
[1]Nueremberg Campus of Technology, Technische Hochschule Nueremberg Georg Simon Ohm, Fuerther Str. 246 b, 90429 Nueremberg, Germany
Full list of author information is available at the end of the article

Abstract

This paper presents a novel type of a filter bank for image decomposition that incorporates "singular spectrum analysis" (SSA). SSA is based on "singular value decomposition" (SVD).

The presented filter banks provide good directional selectivity. Sub-band images highlight exposed directional features such as 0°, 90°, + 45°, and − 45° of the input image. The proposed method can be seen as a tool to separate an image into layers or to present an image as a sum of its high-frequency directional components and a low-frequency approximation.

Keywords: Singular spectrum analysis, Singular value decomposition, Directional selectivity, Filter bank, Wavelet transform, Sub-band decomposition

1 Introduction

Filter banks play an important role in signal processing and are used in many areas. In general, the main purpose of filter banks is to divide an input signal into sub-bands containing distinct frequency domains. They are known for their applications in the area of image processing, such as image compression and denoising, and many more. Different filter bank designs can be implemented depending on the application. This process is famous under different names: sub-band decomposition, sub-band transform, sub-band filtering, and multiscale transform.

In general, a filter bank consists of an analysis and a synthesis stage. During the analysis stage, the input signal is divided into sub-bands depending on the requirements of the application. The respective synthesis stage is the inverse transform to reconstruct the input signal from the sub-bands. Depending on the use case, there is a processing stage between the analysis and the synthesis stage. The processing stage is tailored to one of many possible applications, such as extraction of edges, removal of details, image compression, and compressed sensing.

A comprehensive book [1] of 2018 surveys widely used types of multiscale (sub-band) transform methods according to the historical development process: fourier analysis and fourier transform, wavelets together with different types of wavelet transform (discrete, continuous, undecimated, biorthogonal, complex, dual-tree complex, and quaternion), and "New Multiscale Constructions" (ridgelet, curvelets, contourlet, and shearlet transform). In the field of image processing, filter banks are traditionally based on different types of wavelets, highlighted in many old and worldwide famous books [2–5].

Wavelet-based filter banks have a complex structure, require a huge amount of memory, and lack a perfect reconstruction for finite sequences of data. In parallel to the research of wavelet-based filter banks, Kakarala and Ogunbona investigated in new approaches of sub-band decomposition methods based on SVD. In 2001, they published "multiresolution SVD" (MR-SVD [6]) to overcome these issues. This laid the base for hybrid methods combining a wavelet-based core and SVD, such as "hybrid wavelet-SVD" (Ashino et al., 2004 [7]) and "SVD-wavelet" (Arandiga et al., 2005 [8]). Later, subsequent filter banks that solely base on SVD were proposed by Singh and Kumar (2011 [9]) and Bhatnagar et al., (2014 [10]). Their algorithms show good results in image compression, image encryption, lossy image compression, and face recognition. It was also claimed that SVD-based algorithms are able to outperform wavelet-based filter banks.

This work introduces a set of novel filter banks that build on "singular spectrum analysis" (SSA) [11], an algorithm that implements "singular value decomposition" (SVD). These filter banks are referred to as "*singular spectrum analysis-based image sub-band decomposition filter banks*" ("*SSA-iSBD filter banks*").

To adapt one-dimensional Basic SSA for a processing of two-dimensional image data, the two-dimensional image data has to be transformed into a one-dimensional vector first. The choice of a vectorization scheme (scanning order) determines the properties of the filter bank. Four such transformation schemes are realized and form the core of four "*SSA-iSBD 2-channel filter banks*." These are used as building blocks to create "*SSA-iSBD multi (4 and 6)-channel filter banks*" capable of extracting directional features.

Good directional selectivity is achieved on the basis of "*SSA-iSBD 6-channel filter banks*" that decompose an input image into an approximation and high-frequency components. The approximation contains the low frequency components and the latter highlight specific directional features as well as uncaught remains. The distinct high-frequency components contain the horizontal (0°), vertical (90°), and diagonal (+ 45° and − 45°) directional features, respectively. Directional features regarding the directions + 45° and − 45° are clearly separated from each other. This transform is invertible, and the reverse transform is implemented as a trivial addition of the created sub-band images.

The property of good directional selectivity is useful for image processing applications such as edge and object detection, face and gesture recognition, and feature extraction in machine learning.

To demonstrate the mode of operation of filter banks, the well-known synthetic "Octagon" image shown in Fig. 1a is used as the input. It was chosen for all experiments. It is a black-and-white PNG image with 600 × 600 pixels with a shape providing horizontal, vertical, and diagonal directional features.

Additionally, the natural "Cameraman" image, a black and white PNG image with 256 × 256 pixels shown on Fig. 1b, is used to demonstrate the performance of the developed algorithms. All experiments were conducted with the MATLAB toolkit.

The remainder of this paper is structured as follows: Section 2 observes the image sub-band decomposition methods with directional selectivity properties, regarding the directions 0°, 90°, + 45°, and − 45°. Section 3.1 provides the algorithm of SSA. Section 3.2 reviews how SSA is applied to vectors (one-dimensional data) and to two-dimensional

(a) “Octagon” (b) “Cameraman”

Fig. 1 The input test images

image data. Four schemes of how to apply one-dimensional Basic SSA to two-dimensional image data are introduced in Section 3.3. In Section 3.4, a novel scheme of creating “*SSA-iSBD 2-channel filter banks*” is presented. The outputs of the respective filter banks are visualized and discussed in Section 4, where “*SSA-iSBD multi (4 and 6)-channel filter banks*” are introduced, offering good directional selectivity. The paper concludes with Section 5.

2 Related work

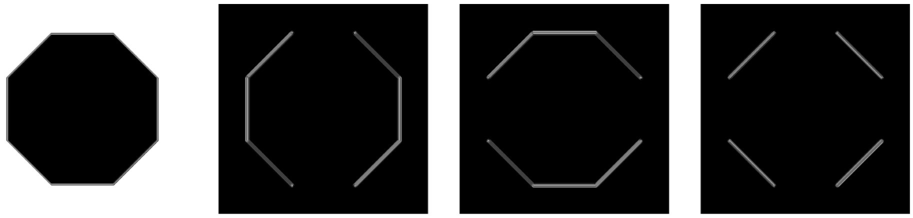
Today, the classical “discrete wavelet transform” (DWT) is still a commonly used scheme for sub-band decomposition of signals (1989, Mallat’s dyadic filter tree [2, 12]). For image processing, 2D-DWT is an extension of DWT for two-dimensional input signals. Typically used mother wavelets (classes of wavelet bases) are Haar-, Daubechies-, Biorthogonal-, Coiflet-, and Symlet-wavelets [12]. 2D-DWT is suited well for image compression in JPEG2000 [12, 13], denoising, and the creation of sparse images for compressed sensing [12, 14]. However, 2D-DWT is not good enough for some specific applications such as pattern recognition. The reasons for that are discussed in the following.

In Fig. 2, the “Octagon” input image was decomposed using 2D-DWT [15] with “Daubechies D4” as the mother wavelet, showing the first level of 2D-DWT decomposition. Every sub-band image has a size of 300 × 300 pixels as a result of down-sampling in the filter banks (decimation).

The two major drawbacks of 2D-DWT in image processing are:

- *Shift sensitivity:* Shift sensitivity or the lack of shift invariance or shift variance causes unpredictable changes in the output coefficients if the input image is shifted slightly [16, 17].
- *Lack of directionality:* A lack of directionality or a poor directional selectivity of a filter bank leads to a non-ideal separation of directional features. Two aspects of poor directional selectivity are distinguished:
  - Horizontal and vertical directional components also contain diagonal features as depicted in Figs. 2b, c. A cleaner extraction of those features is a goal.
  - The filter bank can not separate diagonal features of -45° and $+45^\circ$. Such a compound of both orientations is shown in Fig. 2d. It is preferred to get distinct outputs for -45° and $+45^\circ$ diagonal features.

These drawbacks motivated scientists and researches to develop extensions of 2D-DWT as well as other sophisticated wavelet-based filter banks.



(a) Approximation–low resolution sub-image (b) Vertical orientation sub-image (c) Horizontal orientation sub-image (d) Diagonal orientation sub-image

Fig. 2 2D-DWT (Daubechies D4) of the "Octagon"

Many other types of image sub-band decomposition schemes were developed, being beneficial in specific applications each. Only a few of them focus on directional selectivity, such as a clear separation of horizontal and vertical directional features, or even diagonal and antidiagonal directional features. Such algorithms are:

- One of DWTs extensions, providing shift invariance, referred to as both "undecimated discrete wavelet transform" (UDWT) and "stationary wavelet transform." Its extension to two dimensions (2D-UDWT) is implemented most efficiently by the "algorithme a’trous" [12]. It was introduced by Holschneider et al., (1989 [18]). Fowler [19] provides an overview of different names for this transform. Without downsampling and aliasing, this filter is shift invariant by the cost of redundancy in the output information. The first level of 2D-UDWT applied to the "Octagon" image is similar to the result of processing the "Octagon" image with 2D-DWT (Fig. 2), with the difference that every sub-band image created with 2D-UDWT has the same size as the decomposed image: 600 × 600 pixels. 2D-UDWT still suffers from poor directional selectivity, but due to its shift invariance, it was successfully implemented in applications such as feature extraction [20] and image classification [21]. Furthermore, 2D-UDWT offers a substantial performance improvement in denoising [22, 23].
- "Dual-tree complex wavelet transform" (DT-CWT) is an extension to DWT capable of approaching both shift invariance and good directional selectivity, proposed by Kingsbury (1998 [24–30]). This complex wavelet-based method offers a combination of properties that was not available in earlier approaches: a principally perfect reconstruction and good directional selectivity together with near shift invariance. According to Kingsbury [29], DT-CWT involves six complex wavelets oriented along the directions ± 15°, ± 45°, and ± 75°. DT-CWT is suited well for numerous applications in image [16] and video [31] processing. A scheme based on DT-CWT in combination with a complex spline, developed by Chaundhury and Unser (2010 [32]), is able to detect the directions 0°, 90°, − 45°, and + 45°. An "Octagon" image decomposed with DT-CWT, based on a complex spline, was compared to an "Octagon" image decomposed with 2D-DWT [33].
- The Gabor-like transform proposed by Chaundhury and Unser (2009 [34]) also provides good directional selectivity. They presented examples of transformed "Octagon" and "Cameraman" images. The Gabor-like transform is based on analytic B-spline wavelets, referred to as Gabor-like wavelets, that are oriented along the four principal directions: 0°, 90°, − 45°, and + 45°. The added redundancy along the horizontal and vertical directions yields shift invariance along these directions. However, the

Gabor-like transform is not invertible. An example of Gabor-like transform implementation in the industry is the identification of damaged elements made of polymer and polymeric composites [35].

- "Multiresolution SVD" (MR-SVD), presented by Kakarala and Ogunbona (2001 [6]), is an SVD-based method, demonstrating properties of directionality. Sub-images contain the mix of 0°, 90°, − 45°, and + 45° directional features. A comparison of the directional selectivity of MR-SVD with that of 2D-DWT, incorporating the "Octagon" image, is presented in [7].

3 Methods

3.1 Singular spectrum analysis

"Singular spectrum analysis" (SSA) is a technique initially designed for the analysis of time series. It is a signal processing technique based on "singular value decomposition" (SVD) that decomposes an input signal into several components. These components can be grouped and merged to compose subsequent components. These results can be interpreted as for example a trend, as an oscillation within the input signal, or as noise. SSA can be used for the extraction of trends, for smoothing, for the extraction of periodic components, or for finding structures in time series. It has been applied in physics, signal analysis, and mathematics, but it is also a valuable tool for market research, economics, and meteorology [36].

SSA was initiated by Broomhead and King (1986 [37]), but it was independently introduced by the researchers Danilov and Zhigljavsky in Russia as the "Caterpillar" method [38]. Today, comprehensive descriptions of the theoretical and practical concepts of SSA are available in [11].

Basic SSA (or SSA) is tailored to one-dimensional input signals referred to as a 1D-array [39]. Images, however, are two-dimensional signals. For such 2D-arrays, 2D-SSA is available. Basic SSA and 2D-SSA are widely used [40]. 2D-SSA has been suggested by Danilov and Zhigljavsky(1997 [38]) and was further discussed in [40–43].

In this paper, a preprocessing step is introduced that allows analyzing images with Basic SSA, by transferring the input image into a 1D-array first. This scheme of transforming 2D-arrays into 1D-arrays and vice versa is discussed later.

Both Basic SSA and 2D-SSA rely on the same four processing steps, as described in [39] and [41]. They only differ in step 1 and 4. This difference will also be discussed afterwards.

1. Compute the trajectory matrix X from the input array
    - *The case of an input 1D-array*

      1D-array is represented as a vector F of length N:

$$F = \left(f_1\ f_2\ \cdots\ f_\mathrm{N} \right)^\mathrm{T} \tag{1}$$

Here, the trajectory matrix $X^{1\mathrm{D}}$ is a Hankel matrix constructed from the input 1D-array F. The dimensions of this Hankel matrix are defined only by the window length L that must confirm to $1 \leq L \leq \mathrm{N}$. Let $K = (N - L + 1)$ to get a Hankel matrix of dimensions $L \times K$. The elements of the input 1D-array F are arranged into the Hankel matrix $X^{1\mathrm{D}}$ according to the following scheme [39]:

$$X^{\mathrm{1D}} = \begin{pmatrix} f_1 & f_2 & \cdots & f_K \\ f_2 & f_3 & \cdots & f_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_L & f_{L+1} & \cdots & f_{\mathrm{N}} \end{pmatrix} \tag{2}$$

- *The case of an input 2D-array*

  Computing the trajectory matrix regarding a 2D-array, we have a matrix I of size $\mathrm{N_x} \times \mathrm{N_y}$:

  $$I = \begin{pmatrix} i_{1,1} & i_{1,2} & \cdots & i_{1,\mathrm{N_y}} \\ i_{2,1} & i_{2,2} & \cdots & i_{2,\mathrm{N_y}} \\ \vdots & \vdots & \ddots & \vdots \\ i_{\mathrm{N_x},1} & i_{\mathrm{N_x},2} & \cdots & i_{\mathrm{N_x},\mathrm{N_y}} \end{pmatrix} \tag{3}$$

  The trajectory matrix is the "Hankel-block-Hankel" (HbH) matrix X^{2D} constructed from the input 2D-array I. The dimensions of the HbH matrix are defined by two window sizes $\mathrm{L_x}$ and $\mathrm{L_y}$, chosen that $1 \leq \mathrm{L_x} \leq \mathrm{N_x}$, $1 \leq \mathrm{L_y} \leq \mathrm{N_y}$, and $1 < \mathrm{L_x L_y} < \mathrm{N_x N_y}$. Let again be $\mathrm{K_x} = (\mathrm{N_x} - \mathrm{L_x} + 1)$ and $\mathrm{K_y} = (\mathrm{N_y} - \mathrm{L_y} + 1)$ for convenience.

  The input 2D-array I is arranged into the HbH matrix X^{2D} of size $(\mathrm{L_x L_y} \times \mathrm{K_x K_y})$ according to the following rule [41]:

  $$X^{\mathrm{2D}} = \begin{pmatrix} H_1 & H_2 & \cdots & H_{\mathrm{K_y}} \\ H_2 & H_3 & \cdots & H_{\mathrm{K_y}+1} \\ \vdots & \vdots & \ddots & \vdots \\ H_{\mathrm{L_y}} & H_{\mathrm{L_y}+1} & \cdots & H_{\mathrm{N_y}} \end{pmatrix} \tag{4}$$

  Each H_j is a Hankel matrix of size $\mathrm{L_x} \times \mathrm{K_x}$ consisting of elements from the input 2D-array I:

  $$H_j = \begin{pmatrix} i_{1,j} & i_{2,j} & \cdots & i_{\mathrm{K_x},j} \\ i_{2,j} & i_{3,j} & \cdots & i_{\mathrm{K_x}+1,j} \\ \vdots & \vdots & \ddots & \vdots \\ i_{\mathrm{L_x},j} & i_{\mathrm{L_x}+1,j} & \cdots & i_{\mathrm{N_x},j} \end{pmatrix} \tag{5}$$

2. Singular value decomposition (SVD) of the trajectory matrix X.

   Denote $\lambda_1, \ldots, \lambda_L$ as eigenvalues of XX^{T} taken in the decreasing order ($\lambda_1 \geq \cdots \geq \lambda_L \geq 0$) and $U_1, \ldots, U_L$ denote as corresponding eigenvectors. Set $d = rank(X)$, for simplicity is considered $d = L$.

   $$X_i = \sqrt{\lambda_i} U_i V_i^{\mathrm{T}} \tag{6}$$

   where V_i is defined as

   $$V_i = X^{\mathrm{T}} U_i / \sqrt{\lambda_i}. \tag{7}$$

   SVD of the trajectory matrix X is represented by

   $$X = X_1 + +X_i + \cdots + X_d. \tag{8}$$

   $\sqrt{\lambda_i}$ – is the ith singular value of X; U_i and V_i – are the associated left and right singular vectors of X, for $i = 1, \ldots, d$.

   The collection $(\sqrt{\lambda_i}, U_i, V_i)$ is defined as the ith *eigentriple* of the SVD.

3. Eigentriple grouping

   The grouping procedure splits the set of indices $\{1, \ldots, d\}$ into m subsets $I_1, \ldots, I_m$. Let $I = \{i_1, \ldots, i_p\}$. Then, corresponding to group I, the resultant matrix X_I can be written as $X_I = X_{i_1} + \cdots + X_{i_p}$. The resultant matrices are calculated for each group $I_1, \ldots, I_m$ and the expansion(8) provides the decomposition

   $$X = X_{I_1} + \cdots + X_{I_m} \tag{9}$$

   The choosing procedure of the sets $I_1, \ldots, I_m$ is defined as *eigentriple grouping.*

   - *The case of an input 1D-array*: $X^{1\mathrm{D}} = X^{1\mathrm{D}}_{I_1} + \cdots + X^{1\mathrm{D}}_{I_m}$
   - *The case of an input 2D-array*: $X^{2\mathrm{D}} = X^{2\mathrm{D}}_{I_1} + \cdots + X^{2\mathrm{D}}_{I_m}$

4. Reconstruction of the signal, regarding each group.

   - *The case of an input 1D-array*

     Opposed to the trajectory matrix $X^{1\mathrm{D}}$, the approximated matrix $X^{1\mathrm{D}}_{I_m}$ is not necessarily a Hankel matrix anymore. To “hankelize” it, all elements of each antidiagonal are replaced with their respective averaged value. Then, the reconstructed 1D-array $\tilde{F}$ can be assembled by simply arranging elements from this created Hankel matrix. Finally, the original 1D-array F can be reconstructed by:

     $$F = \tilde{F}_1 + \tilde{F}_2 + \cdots + \tilde{F}_M \tag{10}$$

   - *The case of an input 2D-array*

     The matrices $X^{2\mathrm{D}}_{I_m}$ is not necessarily a HbH matrix anymore. To allow reprojection, which is again a rearrangement of elements to create the reconstructed 2D-array $\tilde{I}$, $X^{2\mathrm{D}}_{I_m}$ must be transferred to a HbH matrix first. This is performed in two steps. First, each sub-matrix of $X^{2\mathrm{D}}_{I_m}$ (see Eq. (4)) is “hankelized” individually by averaging the elements of their antidiagonals. At second, all hankelized sub-matrices with the same index are averaged element-wise. $\tilde{I}_M$ is the 2D-array projected from group $X^{2\mathrm{D}}_{I_m}$.

     $$I = \tilde{I}_1 + \tilde{I}_2 + \cdots + \tilde{I}_M \tag{11}$$

3.2 Grouping of eigentriples

In Section 3.1, the step of grouping the eigentriples was not specified yet. The algorithm proposed in this paper separates the set of eigentriples into two groups. Each of both groups is used to reconstruct an output signal, resulting in two reconstructed output signals for each given input signal.

The first group of eigentriples contains only the first eigentriple related to the most significant singular value $\sqrt{\lambda_i}$ of the trajectory matrix X. The reconstructed signal is referred to as *main vector* $\tilde{F}_{\mathrm{main}}$ for the 1D-array and *main component* $\tilde{I}_{\mathrm{main}}$ for the 2D-array, respectively.

The second group of eigentriples contains all eigentriples except the first one. The reconstructed signal is referred to as *residual vector* $\tilde{F}_{\mathrm{res}}$ (1D-array) and *residual component* $\tilde{I}_{\mathrm{res}}$ (2D-array).

An interesting feature of this grouping scheme is that $F = \tilde{F}_{\mathrm{main}} + \tilde{F}_{\mathrm{res}}$ and $I = \tilde{I}_{\mathrm{main}} + \tilde{I}_{\mathrm{res}}$. In consequence, this scheme implements a two-channel filter bank with a simple reverse transform. This property can be exploited to lower the computational efforts: if only $\tilde{F}_{\text{main}}$ or $\tilde{I}_{\text{main}}$ is calculated, then $\tilde{F}_{\text{res}}$ or $\tilde{I}_{\text{res}}$ can be calculated through a substraction: $\tilde{F}_{\text{res}} = F - \tilde{F}_{\text{main}}$ and $\tilde{I}_{\text{res}} = I - \tilde{I}_{\text{main}}$.

3.3 Processing images with Basic SSA

An input image I, being a 2D-array, can directly be filtered using 2D-SSA. Thus, it can be presented as a sum of a *main* and a *residual component*. In this section, four schemes are proposed that allow processing 2D-arrays with Basic SSA.

To achieve that, the 2D-array I is transformed to a 1D-array F first to be able to perform Basic SSA. Then, after grouping the eigentriples into two groups as described in Section 3.2, the resulting 1D-arrays $\tilde{F}_{\text{main}}$ and $\tilde{F}_{\text{res}}$ are rearranged element-wise into the 2D-arrays $\tilde{I}_{\text{main}}$ and $\tilde{I}_{\text{res}}$, respectively. Thus, Basic SSA is used to represent an input image I as a sum of a *main* ($\tilde{I}_{\text{main}}$) and a *residual component* ($\tilde{I}_{\text{res}}$).

As there is no unique scheme defining how to convert a 2D-array into a 1D-array and vice-versa, four possible ways of creating F from I are introduced now. They allow processing an input 2D-array with Basic SSA. They also offer the respective reverse transform to derive the output images from the resulting 1D-arrays:

1. f_i are the elements of all columns of I, transposed and arranged one after the other as depicted in Fig. 3a. This modification of SSA is referred to as "*Vertical-SSA*."
2. f_i are the elements of all rows of I, one after the other, as demonstrated in Fig. 3b. This modification of SSA is referred to as "*Horizontal-SSA*."
3. f_i is the ith element of I regarding the path shown on Fig. 3c. This modification of SSA is referred to as "*Diagonal-SSA*."
4. f_i is the ith element of I regarding the path shown on Fig. 3d. This modification of SSA is referred to as "*Antidiagonal-SSA*."

3.4 Presentation of "*SSA-iSBD 2-channel filter bank*"

As mentioned in Section 3.2, the proposed SSA-based algorithms decompose an image into a *main* and a *residual component*, configurable by selecting one of the introduced eigentriple grouping schemes. The decomposition is considered a "*SSA-iSBD 2-channel filter bank* as depicted in Fig. 4. Here, and in all following diagrams of this type, the *main* and *residual component* are reflected by a dashed and a solid line, respectively. Important to mention, the theory of SSA does not propose an optimal value for the window size/length. Instead, the window size/length depends on the application, and all window sizes/lengths used in this paper were chosen empirically to achieve a good visual presentation.

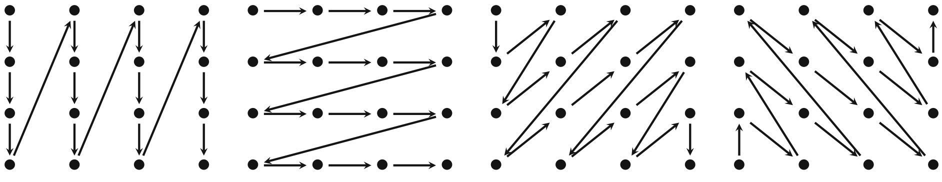

(a) for "*Vertical-SSA*" (b) for "*Horizontal-SSA*" (c) for "*Diagonal-SSA*" (d) for "*Antidiagonal-SSA*"

Fig. 3 These walking paths of "*Vertical-SSA*," "*Horizontal-SSA*," "*Diagonal-SSA*," and "*Antidiagonal-SSA*" convert the pixels of an input image *I* into a vector *F* to allow processing of images with Basic SSA

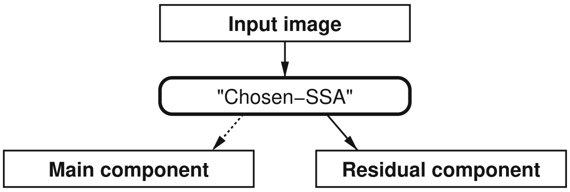


Fig. 4 The structure of the "*SSA-iSBD 2-channel filter bank*"

To demonstrate, the "Octagon" input image was processed with the "*SSA-iSBD 2-channel filter bank*," where 2D-SSA is used as "*chosen-SSA*," which can be also called "*2D-SSA-iSBD 2-channel filter bank*," with the window sizes $(\mathrm{L_x}, \mathrm{L_y}) = (10, 10)$ and (30, 30). The resulting *main* and *residual component* are shown in Fig. 5.

Note that the smoothness of *main components* and the thickness of the lines highlighting the edges of the "Octagon" in the *residual components* relate to the window size. The "*2D-SSA-iSBD 2-channel filter bank*" operates as a filter bank that separates the input image into a low-frequency and a high-frequency part. The cutoff frequency relates to the chosen window size.

To present alternatives to incorporating 2D-SSA as the "*chosen-SSA*" building block, four alternatives are presented here to create a "*SSA-iSBD 2-channel filter bank*." These are "*Vertical-SSA*," "*Horizontal-SSA*," "*Diagonal-SSA*," and "*Antidiagonal-SSA*." The resulting *main* and *residual components* of the "Octagon" input image are shown on Fig. 6. The window length was $L = 30$ in all cases. A comparison of only the *residual components* highlights the differences of these filter banks with the bank incorporating 2D-SSA, Fig. 5d:

- "*Vertical-SSA*" (Fig. 6b): all vertical directional features are removed,
- "*Horizontal-SSA*" (Fig. 6d): all horizontal directional features are removed,
- "*Diagonal-SSA*" (Fig. 6f): all diagonal directional features are removed, and
- "*Antidiagonal-SSA*" (Fig. 6h): all antidiagonal directional features are removed.

Note that the "*SSA-iSBD 2-channel filter bank*" provides a "redundant form" of image decomposition: every *component* (*main* and *residual*) contains the same amount of pixels as the input image ($N = \mathrm{N_x N_y}$). Thus, N input pixels are transformed to a total of $2N$ output pixels. Independent from the type of SSA referred to by "*chosen-SSA*," the input image can always be reconstructed by just adding the *main* and the *residual component*.

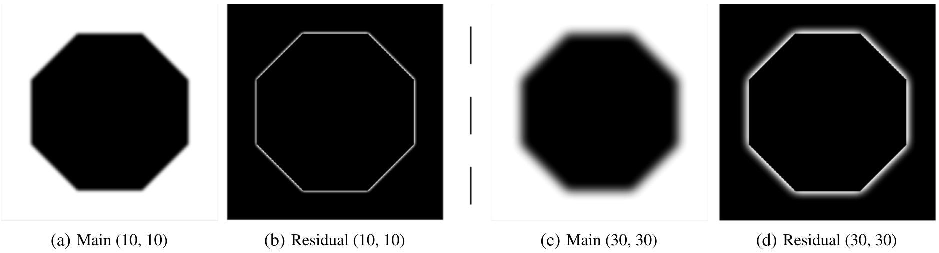


Fig. 5 These *main* and *residual components* of the "Octagon" image were created with the "*SSA-iSBD 2-channel filter bank*" using 2D-SSA with window sizes (10, 10) and (30, 30), respectively

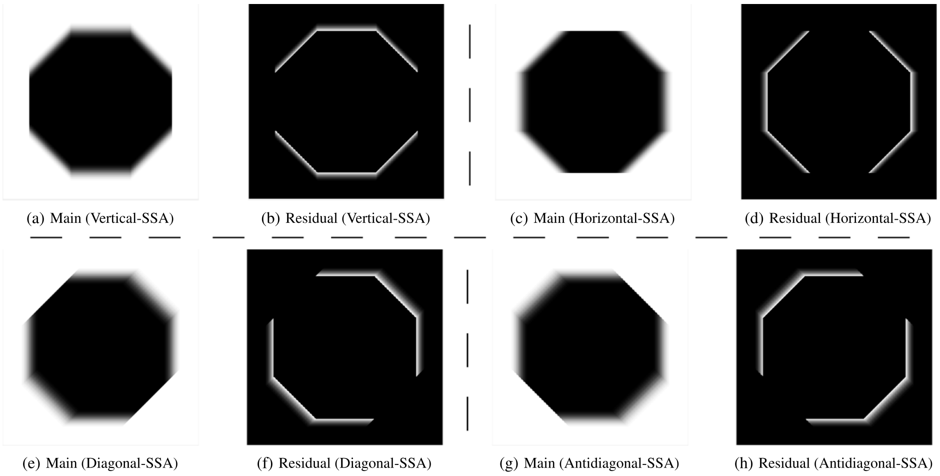

(a) Main (Vertical-SSA)
(b) Residual (Vertical-SSA)
(c) Main (Horizontal-SSA)
(d) Residual (Horizontal-SSA)
(e) Main (Diagonal-SSA)
(f) Residual (Diagonal-SSA)
(g) Main (Antidiagonal-SSA)
(h) Residual (Antidiagonal-SSA)

Fig. 6 These *main* and *residual components* show the "Octagon" image processed with "*SSA-iSBD 2-channel filter bank*" with following types of "*chosen-SSA*": "*Vertical-SSA*," "*Horizontal-SSA*," "*Diagonal-SSA*, "and "*Antidiagonal-SSA*"

4 Discussion and results

In Section 3.4, "*SSA-iSBD 2-channel filter banks*" were presented. It was shown that in the case of using "*Vertical-SSA*," "*Horizontal-SSA*," "*Diagonal-SSA*," or "*Antidiagonal-SSA*," the *residual component* lacks one of the directional features. As such a removal is not the goal, this section introduces that a chaining of multiple "*SSA-iSBD 2-channel filter banks*" perform an *extraction* of directional features. These filter banks implement a multi-channel decomposition that deconstruct a given input image into a sum of multiple *sub-images*.

4.1 "SSA-iSBD 4-channel filter bank"

As a first step to realize such filter banks, the proposed SSA types are organized into *orthogonal pairs*. These *pairs* correspond to the properties of the residual components they create, whereupon the removed directional features are orthogonal to each other. The following *orthogonal pairs* are introduced:

- "*Vertical-SSA*" and "*Horizontal-SSA*" with the same window length L and
- "*Diagonal-SSA*" and "*Antidiagonal-SSA*" with the same window length L.

The first SSA type of such an *orthogonal pair* is referred to as "*chosen-SSA*" and the remaining is referred to as "*complement-SSA*." Both members of the *pair* are equivalent, so each can be selected to become the "*chosen-SSA*."

Figure 7 shows the "*SSA-iSBD 4-channel filter bank*." It performs a 2-stage decomposition that decomposes an input image into a sum of four *sub-images*. Each output *sub-image* is the result of having the input image passing two consecutive "*SSA-iSBD 2-channel filter banks*" implementing an *orthogonal pair*.

The first filter bank implements "*Vertical-SSA*" and "*Horizontal-SSA*" as its *orthogonal pair*. This filter bank is referred to as "*SSA-iSBD 4-channel filter bank*," type "*vertical/horizontal*," with "*Vertical-SSA*" picked as the "*chosen-SSA*." Figure 8 shows the four generated *sub-images*, whereas a window length of $L = 30$ was used.

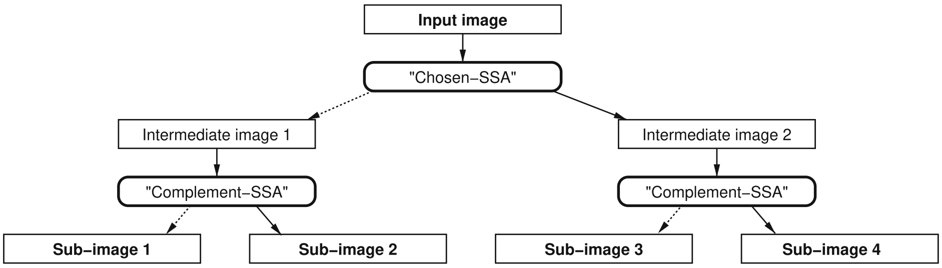


Fig. 7 The structure of the *"SSA-iSBD 4-channel filter bank"*

- *Sub-image 1* is the approximation, containing the low-frequency components (Fig. 8a),
- *Sub-image 2* provides all vertical (90°) directional features (Fig. 8b),
- *Sub-image 3* contains all horizontal (0°) directional features with orientation (Fig. 8c), and
- *Sub-image 4* consists of the remains. They contain all diagonal (+ 45° and − 45°) directional features (Fig. 8d).

Sub-image 2 in Fig. 8b and *sub-image 3* in Fig. 8c contain the extracted 0° and 90° directional features, respectively. Noticeably, *sub-image 1* in Fig. 8a contains only low-frequency components. This demonstrates that both "*Vertical-SSA*" (stage 1) and "*Horizontal-SSA*" (stage 2) extract not only the desired directional features, but they also let the low-frequency components pass. Furthermore, *sub-image 4* in Fig. 8d contains all diagonal directional components. This highlights that the directional selectivity of all stages of the filter bank are good: if the low-frequency components, the vertical, and the horizontal directional features are removed, then the "remains" must contain the omitted diagonal directional features.

The second filter bank incorporates "*Diagonal-SSA*" and "*Antidiagonal-SSA*" as its *orthogonal pair*, with "*Diagonal-SSA*" as the "*chosen-SSA*." The resulting filter bank, referred to as "*SSA-iSBD 4-channel filter bank*," type "*diagonal/antidiagonal*," extracts + 45° and − 45° directional features. The four resulting *sub-images* created with this filter bank are as shown in Fig. 9.

- *Sub-image 1* is the approximation containing the low-frequency components (Fig. 9a),
- *Sub-image 2* provides all diagonal (+ 45°) directional features (Fig. 9b),
- *Sub-image 3* contains all antidiagonal (− 45°) directional features (Fig. 9c), and
- *Sub-image 4* consists of the remains, in this case vertical (0°) and horizontal (90°) directional features (Fig. 9d).

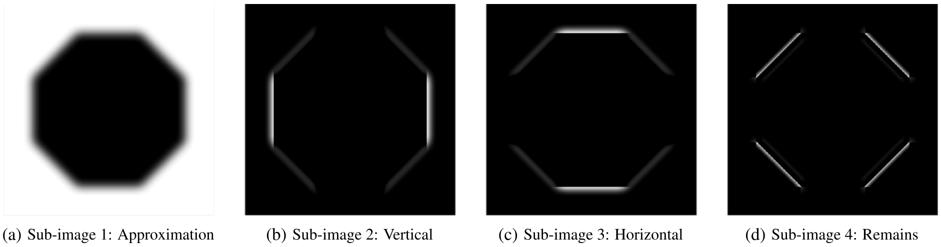

(a) Sub-image 1: Approximation (b) Sub-image 2: Vertical (c) Sub-image 3: Horizontal (d) Sub-image 4: Remains

Fig. 8 *Sub-images 1–4* of the "Octagon" image processed with the *"SSA-iSBD 4-channel filter bank,"* type *"vertical/horizontal"* with L = 30

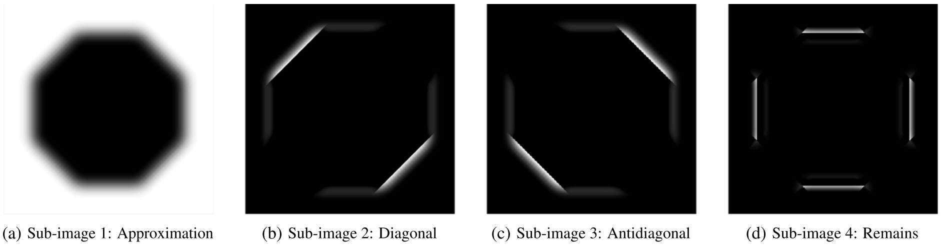

(a) Sub-image 1: Approximation (b) Sub-image 2: Diagonal (c) Sub-image 3: Antidiagonal (d) Sub-image 4: Remains

Fig. 9 *Sub-images 1–4* of the "Octagon" image processed with the *"SSA-iSBD 4-channel filter bank,"* type *"diagonal/antidiagonal"* with L = 30

Noticeably, the *sub-images 2* and *sub-images 3* created with both exemplary filter banks highlight not only the desired directional components, but they also provide slightly visible traces of the directional components contained in *sub-images 4*.

Regarding each of the "*SSA-iSBD 4-channel filter banks*," N input pixels are transformed to a total of $4N$ pixels. The sum of all four *sub-images* is again equal to the input image.

4.2 "SSA-iSBD 6-channel filter bank"

The next target is a clear separation 0°, 90°, + 45°, and − 45° directional features with a single multi-stage filter bank. The "*SSA-iSBD 4-channel filter bank*," type "*vertical/horizontal*," ispresented in Section 4.1, is already able to separate 0° and 90° directional components and to provide a compound of all diagonal directional components (see Fig. 8). However, there are still two issues: the directional features with + 45° and − 45° are still not separated from each other, and the respective *sub-image 4* does not explicitly contain the diagonal features, but the remains. Thus, if there were subsequent omitted components, they also would have been mixed into the remains. However, with subsequent filtering steps, it is possible to extract and to separate all diagonal directional features into distinct sub-images.

Here, "*SSA-iSBD 6-channel filter banks*" are created as an extension to a "*SSA-iSBD 4-channel filter bank*." The filter bank structure presented on the Fig. 10 shows such a filter bank that is based on a "*SSA-iSBD 4-channel filter bank*," type "*vertical/horizontal*." The six resulting *sub-images* presented in Fig. 11 provide the following "Octagon" decomposition:

- *Sub-image 1* is the approximation containing the low-frequency components (Fig. 11a),
- *Sub-image 2* provides all vertical (90°) directional features (Fig. 11b),
- *Sub-image 3* contains all horizontal (0°) directional features (Fig. 11c),
- *Sub-image 4* highlights all diagonal (+ 45°) directional features (Fig. 11d),
- *Sub-image 5* shows all antidiagonal (− 45°) directional features (Fig. 11e), and
- *Sub-image 6* consists of the remains (Fig. 11f).

Obviously, other similar structures of "*SSA-iSBD 6-channel filter banks*" can be created, for example, a filter bank that is based on a "*SSA-iSBD 4-channel filter bank*," type "*diagonal/antidiagonal*." Such a filter bank will differ in the order of extraction of the directional features.

The presented "*SSA-iSBD 6-channel filter bank*" decomposes an image into a sum of six *sub-images* and provides a clear separation of 0°, 90°, − 45°, and + 45° directional

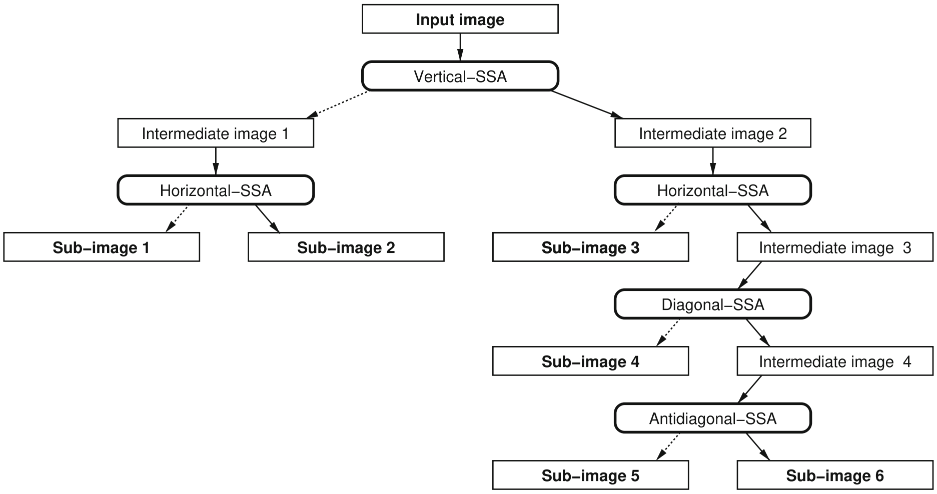


Fig. 10 The structure of the "*SSA-iSBD 6-channel filter bank*" implemented as an extension of the "*SSA-iSBD 4-channel filter bank*," type "*vertical/horizontal*"

features. To compare, a related transform consisting of a combination of DT-CWT and a complex spline [32] is also able to detect the directions 0°, 90°, − 45°, and + 45°. DT-CWT is famous for its good directional selectivity. Using that transform, a decomposed "Octagon" image also provides clearly separated 0° and 90° directional components with slightly visible traces of the − 45° and + 45° directional features [33]. Thus, the "*SSA-iSBD 6-channel filter banks*" presented in this paper shares this property of providing good directional selectivity.

As in the previous examples, all window lengths are $L = 30$. N input pixels are transformed to a total of $6N$ pixels. For the inverse transform to get a reconstruction, all six *sub-images* have to be added.

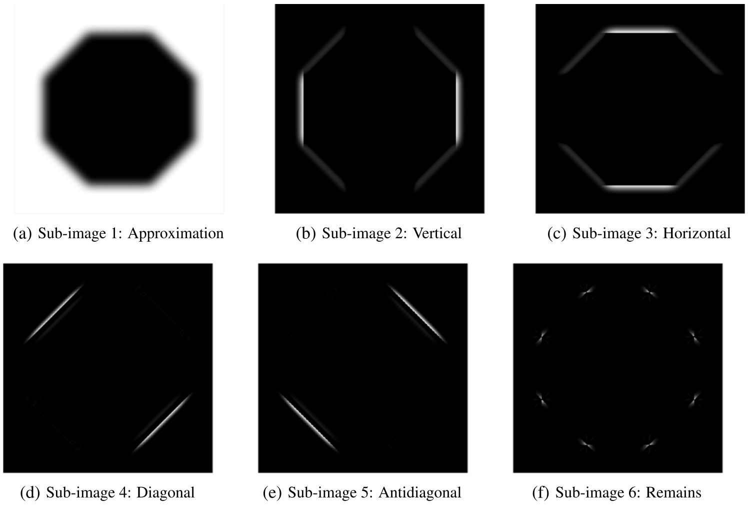

(a) Sub-image 1: Approximation
(b) Sub-image 2: Vertical
(c) Sub-image 3: Horizontal
(d) Sub-image 4: Diagonal
(e) Sub-image 5: Antidiagonal
(f) Sub-image 6: Remains

Fig. 11 The sub-images 1–6 of the "Octagon" image produced by the "*SSA-iSBD 6-channel filter bank*" according to the filter bank structure shown in Fig. 10

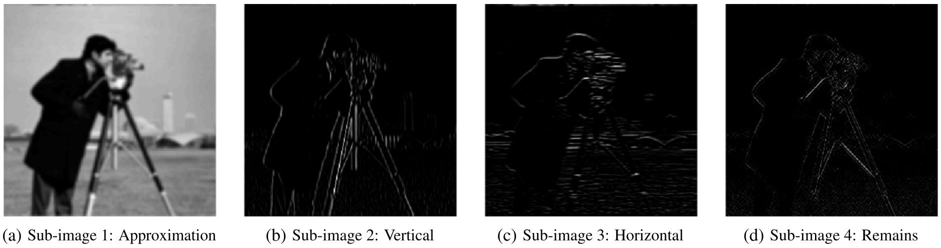

(a) Sub-image 1: Approximation (b) Sub-image 2: Vertical (c) Sub-image 3: Horizontal (d) Sub-image 4: Remains

Fig. 12 *Sub-images 1–4* of the "Cameraman" image processed with the "*SSA-iSBD 4-channel filter bank*," type "*vertical/horizontal*," with $L = 4$

4.3 Demonstration of "*SSA-iSBD multi-channel filter banks*" using a natural image

So far, all provided examples involved the synthetic "Octagon" test image. In this section, the performance of the "*SSA-iSBD multi-channel filter bank*" is rated using the natural "Cameraman" image (see Fig. 1b).

At first, Fig. 12 presents the results of processing the "Cameraman" image with the "*SSA-iSBD 4-channel filter bank*," type "*vertical/horizontal*" (see Fig. 7).

Figure 13 presents the results of processing the "Cameraman" image with the "*SSA-iSBD 4-channel filter bank*," type "*diagonal/antidiagonal*."

In the next experiment, the "Cameraman" image was decomposed with the "*SSA-iSBD 6-channel filter bank*." Figure 14 presents the calculated *sub-images 1–6* according to the filter bank structure shown on Fig. 10 with a window length $L = 12$.

As mentioned before, the window length L has to be determined empirically depending on the application. Here, the larger window length of $L = 12$ was selected to demonstrate its influence on the extraction of directional features.

A side-by-side comparison of Figs. 12a–c and 14a–c shows the influence of the chosen window size L on the cutoff frequency. The larger the value of L is, the smaller becomes the part of the spectrum contained in the low-frequency approximation.

All sub-images of Figs. 12, 13, and 14 have been contrast-stretched. The value of each pixel fits into the range from 0 to 255, calculated in the same way as used for the publication of MR-SVD [6].

To increase visibility, the same output *sub-images 1–6* of the "*SSA-iSBD 6-channel filter bank*" are presented again on Fig. 15 in its normalized form. To achieve that, the value of each pixel was rescaled as $min_{i,j}\tilde{I}_{i,j} \rightarrow 0$ and $max_{i,j}\tilde{I}_{i,j} \rightarrow 1$. They are printed using the standard grayscale colormap. Such presentation is commonly used in related work [40, 44].

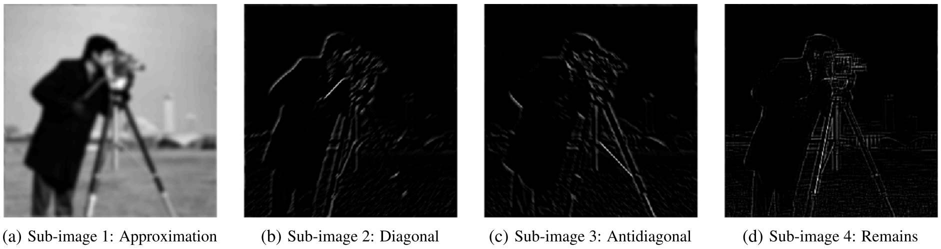

(a) Sub-image 1: Approximation (b) Sub-image 2: Diagonal (c) Sub-image 3: Antidiagonal (d) Sub-image 4: Remains

Fig. 13 *Sub-images 1–4* of the "Cameraman" image processed with the "*SSA-iSBD 4-channel filter bank*," type "*diagonal/antidiagonal*," with $L = 4$

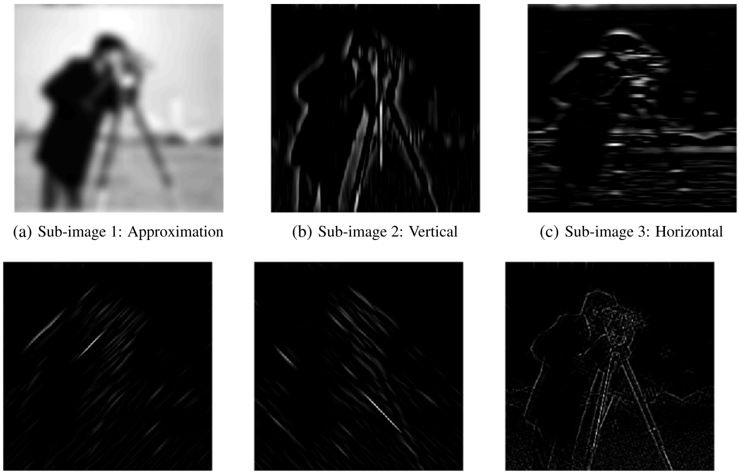

(a) Sub-image 1: Approximation
(b) Sub-image 2: Vertical
(c) Sub-image 3: Horizontal
(d) Sub-image 4: Diagonal
(e) Sub-image 5: Antidiagonal
(f) Sub-image 6: Remains

Fig. 14 The *sub-images 1–6* of the "Cameraman" image produced by the "*SSA-iSBD 6-channel filter bank*" with $L = 12$

A comparable decomposition of the "Cameraman" image was achieved with the first level of DT-CWT decomposition [44]. In that publication, Fig. 5 contains sub-band images that contain separated directional features for ± 15°, ± 45°, and ± 75° using the same normalization scheme as used here for the images on Fig. 15. The quality of the performance of both filter banks is *visually similar*: the + 45° and − 45° directional features were extracted successfully by both. DT-CWT divides an input image into its low and high frequency components, with a cutoff frequency depending on the chosen mother wavelet (type and order). Regarding the "*SSA-iSBD filter banks*," the window length L provides the means for such adjustments.

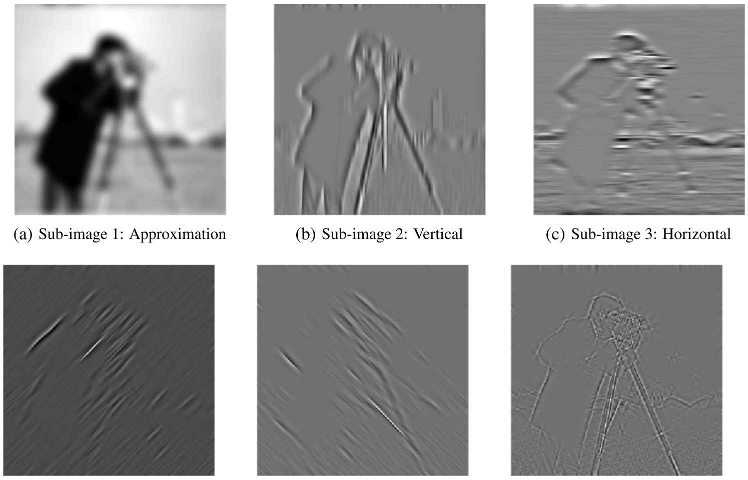

(a) Sub-image 1: Approximation
(b) Sub-image 2: Vertical
(c) Sub-image 3: Horizontal
(d) Sub-image 4: Diagonal
(e) Sub-image 5: Antidiagonal
(f) Sub-image 6: Remains

Fig. 15 Rescaled *sub-images 1–6* of the "Cameraman" image produced by the "*SSA-iSBD 6-channel filter bank*" with $L = 12$

5 Conclusions

The "*singular spectrum analysis-based image sub-band decomposition filter banks*" ("*SSA-iSBD filter banks*") are introduced. Four types of "*SSA-iSBD 2-channel filter banks*" based on Basic SSA serve as building blocks for "*SSA-iSBD multi-channel filter banks*."

Due to the sub-band separation based on SVD, the proposed filter banks are not affected negatively by shift variance. To remind, shift variance is typical for wavelet-based image sub-band decomposition schemes.

Good directional selectivity is provided by the "*SSA-iSBD 6-channel filter bank*" that represents an input image as a sum of an approximation containing the low-frequency components and, separated from each other, the high-frequency components. These high-frequency components refer to distinct horizontal (0°), vertical (90°), diagonal (+ 45°), and antidiagonal (− 45°) directional features, and the remains. This transform is invertible: a simple addition of all components is sufficient to reproduce the input image. The presented novel filter banks introduce the window length as a parameter that has to be determined empirically depending on the application.

This new method of image sub-band decomposition is presented for the attention of the scientific community. Mathematicians may be interested in proving or explaining this heuristically developed method while engineers may incorporate this algorithm to improve their specific features extraction, edge detection, or face and gesture recognition tasks.

Abbreviations

1-D: One-dimensional; 2-D: Two-dimensional; SSA: Singular spectrum analysis; SVD: Singular value decomposition; DWT: Discrete wavelet transform; 2D-DWT: Two-dimensional-discrete wavelet transform; MR-SVD: Multiresolution singular value decomposition; UDWT: Undecimated discrete wavelet transform; 2D-UDWT: Two-dimensional undecimated discrete wavelet transform; DT-CWT: Dual-tree complex wavelet transform; "*SSA-iSBD filter bank*": Singular spectrum analysis-based image sub-band decomposition filter banks

Acknowledgements

The corresponding author is deeply grateful for the MSCSP program of the Ilmenau University of Technology and her scholarship of the "German Academic Exchange Service" (DAAD) (Grant No. A/08/77322) that made her masters studies possible. The presented work is based on the knowledge gained during the course AASP (Prof. Dr.-Ing. Martin Haardt). We thank the anonymous reviewers for their excellent remarks. They helped to improve this paper.

Authors' contributions

J.E. developed all ideas and algorithms, wrote most of the manuscript together with F.E., and carried out all experiments. F.E. organized focusing discussions and did many corrections according structure of the publication and used terminology. F.G. assisted during the implementation (programming and debugging of some pieces of the code). R.S. supervised and reviewed the work, as a head of research center. The authors read and approved the final manuscript.

Funding

The work presented in this paper was solely financed by the research budget of the professorship for Control Technology at the Nuremberg Campus of Technology in Germany (NCT).

Availability of data and materials

Please contact the corresponding author for data requests.

Ethics approval and consent to participate

Not applicable.

Consent for publication

This manuscript does not contain any individual persons data in any form.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Nueremberg Campus of Technology, Technische Hochschule Nueremberg Georg Simon Ohm, Fuerther Str. 246 b, 90429 Nueremberg, Germany. [2] Independent researcher, Nueremberg, Germany.

Received: 4 December 2019 Accepted: 28 April 2020
Published online: 17 June 2020

References

1. A. Vyas, S. Yu, J. Paik, *Multiscale transforms with application to image processing*. (Springer, Singapore, 2018)
2. S. G. Mallat, *A wavelet tour of signal processing: The Sparse Way*. (Elsevier Ltd, USA, 2009)
3. M. Vetterli, J. Kovačević, *Wavelets and subband coding*. (CreateSpace Independent Publishing Platform, USA, 2013)
4. G. Strang, T. Nguyen, *Wavelets and filter banks, 2nd edn.* (Wellesley-Cambridge Press, USA, 1996)
5. A. N. Akansu, M. J. T. Smith, *Subband and wavelet transforms: design and applications*. (Springer, USA, 1995)
6. R. Kakarala, P. O. Ogunbona, Signal analysis using a multiresolution form of the singular value decomposition. IEEE Trans. Image Process. **10**(5), 724–735 (2001). https://doi.org/10.1109/83.918566
7. R. Ashino, A. Morimoto, R. Vaillancourt, Image compression with multiresolution singular value decomposition and other methods. Math. Comput. Model. **41**, 773–790 (2005)
8. A. Francesc, R. Donat, J. C. Trillo, *SVD-wavelet algorithm for image compression*, (2005), pp. 172–177
9. S. K. Singh, S. Kumar, Singular value decomposition based sub-band decomposition and multi-resolution representation of digital colour images. Pertanika J. Sci. Technol. **19**(2), 229–235 (2011)
10. G. Bhatnagar, A. Saha, Q. M. J. Wu, P. K. Atrey, Analysis and extension of multiresolution singular value decomposition. Inf. Sci. **277**, 247–262 (2014). https://doi.org/10.1016/j.ins.2014.02.018
11. N. E. Golyandina, V. Nekrutkin, A. A. Zhigljavsky, *Analysis of time series structure: SSA and related techniques*. (Chapman & Hall/CRC, USA, 2001)
12. S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**, 674–693 (1989). https://doi.org/10.1109/34.192463
13. A. Skodras, C. Christopoulos, T. Ebrahimi, The jpeg 2000 still image compression standard. IEEE Signal Process. Mag. **18**(5), 36–58 (2001). https://doi.org/10.1109/79.952804
14. G. Kutyniok, Theory and applications of compressed sensing. GAMM-Mitteilungen (2013)
15. M. Misiti, Y. Misiti, G. Oppenheim, J.-M. Poggi, *MATLAB: wavelet toolbox user's guide*. (The MathWorks, Inc., USA, 2015)
16. I. W. Selesnick, R. G. Baraniuk, N. C. Kingsbury, The dual-tree complex wavelet transform. IEEE Signal Process. Mag. **22**(6), 123–151 (2005). https://doi.org/10.1109/MSP.2005.1550194
17. A. P. Bradley, in *DICTA*, Shift-invariance in the discrete wavelet transform, (2003)
18. M. Holschneider, R. Kronland-Martinet, J. Morlet, P. Tchamitchian, A real-time algorithm for signal analysis with the help of the wavelet transform. Wavelets Time-Freq. Methods Phase Space (1989). https://doi.org/10.1007/978-3-642-75988-8_28
19. J. E. Fowler, The redundant discrete wavelet transform and additive noise. IEEE Signal Process. Lett. **12**(9), 629–632 (2005)
20. Y. Zhang, Feature extraction of brain mri by stationary wavelet transform and its applications. J. Biol. Syst. **12**(9), 115–132 (2010). https://doi.org/10.1142/S0218339010003652
21. Z. Dong, Magnetic resonance brain image classification via stationary wavelet transform and generalized eigenvalue proximal support vector machine. J. Med. Imaging Health Inform. **5**(7), 1395–1403 (2015). https://doi.org/10.1166/jmihi.2015.1542
22. R. R. Coifman, D. L. Donoho, Translation-invariant de-noising. Wavelet Stat. Lect. Notes Stat. **3**, 125–150 (1995)
23. M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, R. O. Wells, Noise reduction using an undecimated discrete wavelet transform. IEEE Signal Process. Lett. **3**(1), 10–12 (1996). https://doi.org/10.1109/97.475823
24. N. Kingsbury, *The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters*, (1998), pp. 319–322
25. N. Kingsbury, *The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement*, (1998), pp. 1–4
26. N. Kingsbury, J. Magarey, *Wavelet transforms in image processing*, (1997), pp. 24–27
27. N. Kingsbury, Image processing with complex wavelets. Phil. Trans. R. Soc. Lond. A. **357**, 2543–2560 (1997)
28. N. Kingsbury, A dual-tree complex wavelet transform with improved orthogonality and symmetry properties. Proc. 2000 Int. Conf. Image Process. **2**, 375–378 (2000). https://doi.org/10.1109/ICIP.2000.899397
29. N. Kingsbury, Complex wavelets for shift invariant analysis and filtering of signals. J. Appl. Comput. Harmon. Anal., 234–253 (2001). https://doi.org/10.1006/acha.2000.0343
30. N. Kingsbury, in *Proceedings 2003 International Conference on Image Processing*, Design of q-shift complex wavelets for image processing using frequency domain energy minimization, vol. 1, (2003). https://doi.org/10.1109/ICIP.2003.1247137
31. I. W. Selesnick, K. Y. Li, Video denoising using 2D and 3D dual-tree complex wavelet transforms. Wavelets Appl. Signal Image Process. X, 607–618 (2003). https://doi.org/10.1117/12.504896
32. K. N. Chaudhury, M. Unser, On the shiftability of dual-tree complex wavelet transforms. IEEE Trans. Signal Process. **58**(1), 221–232 (2010)
33. G. Cristobal, P. Schelkens, H. Thienpont, *Optical and digital image processing: fundamentals and applications*. (Wiley-VCH Verlag GmbH & Co. KGaA, p.151, Singapore, 2011)
34. K. N. Chaudhury, M. Unser, Construction of Hilbert transform pairs of wavelet bases and Gabor-like transforms. IEEE Trans. Signal Process. **57**(9), 3411–3425 (2009). https://doi.org/10.1109/TSP.2009.2020767
35. A. Katunin, P. Przystałka, Damage assessment in composite plates using fractional wavelet transform of modal shapes with optimized selection of spatial wavelets. Eng. Appl. Artif. Intell. **30**, 73–85 (2014). https://doi.org/10.1016/j.engappai.2014.01.003
36. H. Hassani, Singular spectrum analysis: methodology and comparison. J. Data Sci. **5**, 239–257 (2007)
37. D. S. Broomhead, G. P. King, Extracting qualitative dynamics from experimental data. Phys D Nonlinear Phenom. **20**(2), 217–236 (1986). https://doi.org/10.1016/0167-2789(86)90031-X
38. D. Danilov, A. A. Zhigljavsky, *Principal components of time series: "Caterpillar" method*. (St. Petersburg University (in Russian), Russia, 1997)

39. N. E. Golyandina, A. A. Zhigljavsky, Singular spectrum analysis for time series. Springer Briefs Stat (2013). https://doi.org/10.1007/978-3-642-34913-3_2
40. L. S. Rodriguez-Aragon, A. A. Zhigljavsky, Singular spectrum analysis for image processing. Stat. Interface. **3**(3), 419–426 (2010)
41. N. E. Golyandina, K. D. Usevich, 2D-extension of singular spectrum analysis: algorithm and elements of theory. Matrix Methods Theory Algoritm. Appl., 449–473 (2010). https://doi.org/10.1142/9789812836021_0029
42. N. E. Golyandina, K. D. Usevich, I. V. Florinsky, Filtering of digital terrain models by two-dimensional singular spectrum analysis. Int. J. Ecol. Dev. **8**, 81–94 (2007)
43. N. E. Golyandina, K. D. Usevich, An algebraic view on finite rank in 2D-SSA. 6th St. Petersburg Work. Simul. **1**, 308–313 (2009)
44. A. Mansouri, A. Mahmoudi, T. Azar, SVD-based digital image watermarking using complex wavelet transform. Sadhana. **34**(3), 393–406 (2009). https://doi.org/10.1007/s12046-009-0016-y

Publisher's Note

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