University of Alberta

Directional Tight Framelet Filter Bank Design

By

Menglu Che

A thesis submitted in partial fulfillment of the

requirements for the degree of

Master of Science

in

Applied Mathematics

Department of Mathematical and Statistical Sciences

Fall,2013

Edmonton, Alberta

Abstract

It is well known that most information of an image is contained in its edges. Therefore capturing edges in an image is of fundamental importance in image processing. Tensor product real-valued wavelets only capture edges along the horizontal and vertical directions. Hence they are only suboptimal for handling high-dimensional problems. In this thesis we use a framelet-based approach to enhance the performance of tensor product real-valued wavelets. By employing an additional high-pass filter, we construct finitely supported complex-valued tight framelet filter banks $\{a; b_1, b_2\}$ such that their tensor products in dimension two offer four directions along 0° (horizontal), 45°, 90° (vertical) and 135°. We propose a simple and effective algorithm to construct such directional tight framelet filter banks, and provide a necessary and sufficient condition for their existence. Finally, several concrete examples of such directional tight framelet filter banks are given.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Bin Han, without whose guidance and support this thesis would have not been possible. I deeply appreciate his inspiration on my research work, useful comments and remarks for my thesis and suggestions for my career. I thank my co-supervisor, Dr. Yau Shu Wong for the support though my graduate study.

I would like to thank the examining committee, Dr. John. C. Bowman, Dr. Bin Han, Dr. Linglong Kong, and Dr. Yau Shu Wong.

Last but never the least, I would thank my loved ones throughout my life, my family, my boyfriend, my colleagues and my friends. I am grateful forever for your love.

Contents

List of Figures

List of Tables

Chapter 1

Background and Motivation

Since the introduction of wavelets by Croisier et. al. [4] in 1976, wavelet-based methods have been widely used in various fields such as image processing, scientific computing, differential equations and so on. Benefiting from the multi-resolution structure, wavelet-based tools for processing 2-dimensional signals, such as images, have been considerably developed in the last two decades. Several algorithms were developed for image denoising, feature extraction, segmentation, etc. These processings require us to capture certain features in the time domain, such as edges in images. In this thesis we deal with the tight framelet filter bank design problem with particular interest in directional singularities. To be more specific, we are aiming at the construction of finitely supported directional tight framelet filter banks through tensor product. In this chapter, we give a short introduction to wavelet analysis and existing approaches for the detection of directional singularities. The motivation for our research work is summarized in the last section.

1.1 The Problem: Directional Singularities

Usually, a digital image is represented by a 2-dimensional array, or to say, a matrix with its entries indicating the level of brightness. By a contrast of different levels of brightness, the edges or dots in images are then indicated by a large jump or drop of coefficients within a certain small area. In the image itself, which is usually referred as the "time domain", they are often called singularities since the derivative (difference) will be quite large within these areas. The shapes of these singularities vary but

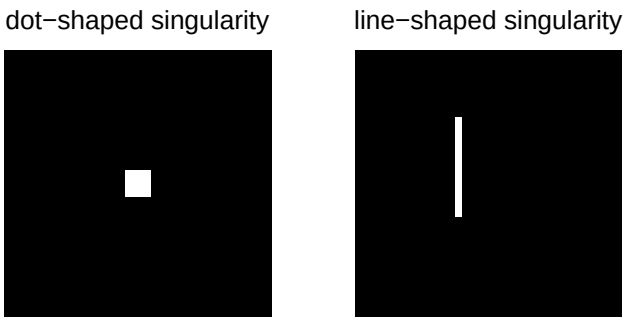


Figure 1: A trivial example of dot-shaped and line-shaped singularities

can be generally classified as dot-shaped singularities and line-shaped singularities, as shown in Fig 1. Detecting these singularities is of extreme importance. For example, denoising is usually performed within the relatively smooth areas. If an edge is located, one naturally wants to preserve it because denoising in this area will make the edge blurred or smoothed. In medical imaging, the edges show the boundaries of different organs, and help to find irregular shapes, such as tumors.

Classic 2-D wavelet transform obtained from tensor product of 1-D wavelets handles point-shape singularities well, because the multi-resolution property will approximate the local jumps very well on fine scales. However, due to the symmetry property of wavelet filter banks, it hardly has any directions other than the vertical and horizontal ones. This makes it difficult to detect directional line-shape singularities in images by classic wavelets. To remedy the missing directions, mathematicians and engineers came up with diverse tools. With different characteristics, they are named "X-lets" uniformly. Let us first review the classic wavelet analysis, the origin of all these "X-lets" methods.

1.2 Wavelet Transform-Continuous Setting

1.2.1 Basics in Wavelets

The most classic wavelet transform is based on a wavelet function and a Multi-Resolution Analysis (MRA) associated with it. With this MRA system any signal in $L_2(\mathbb{R})$ can be represented.

A mother wavelet, or simply a wavelet $\psi(t)$ is a square (Lebesgue-)integrable function which has a zero mean:

$$\int_{\mathbb{R}} \psi(t)dt = 0.$$

This wavelet function can be dilated with a scale parameter a^{-1}, translated with a location parameter b as

$$\psi_{a,b}(t) = a^{-1/2}\psi(\frac{t-ab}{a}),$$

which gives a wavelet transform of any L_2 function f at the scale a and position b as follows

$$Wf(a,b) = \int_{\mathbb{R}} f(t)a^{-1/2}\psi(\frac{t-ab}{a})dt.$$

We are particularly interested in such wavelet functions: when we take a countable subset $\{(a,b) : a = 2^j,\ b = k, (j,k) \in \mathbb{Z}^2\}$, the collection of all $\psi_{j,k} = 2^{-j/2}\psi(\frac{t-2^j k}{2^j})$ generates a basis of $L_2(\mathbb{R})$, i.e, the wavelet basis. The existence of such wavelet functions has been proved by Meyer in 1985 [12]. Any function $f \in L_2(\mathbb{R})$ can be expressed as a linear combination of such basis functions in L_2 sense. As $L_2(\mathbb{R})$ is an infinite-dimensional space, the linear combination can be an infinite sum.

If the basis is an orthonormal one, the linear combination can be directly written as

$$f = \sum_{j,k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k} \tag{1.2.1}$$

in the $L_2(\mathbb{R})$ sense, where the inner product $\langle \cdot, \cdot \rangle$ is that usually defined in $L_2(\mathbb{R})$ space:

$$\langle f, g \rangle = \int_{\mathbb{R}} f(t)\overline{g(t)}dt, \qquad f, g \in L_2(\mathbb{R}).$$

But when the basis is not orthogonal, the coefficients in the linear combination are not so easy to compute. We make a concession to ask for a biorthogonal pair $(\{\psi_{j,k}\}, \{\tilde{\psi}_{j,k}\})$ where $\{\psi_{j,k}\}$ and $\{\tilde{\psi}_{j,k}\}$ are both bases of $L_2(\mathbb{R})$ (not necessarily orthogonal), and satisfy

$$\langle \psi_{j_1,k_1}, \tilde{\psi}_{j_2,k_2} \rangle = \begin{cases} 1, & \text{when } (j_1, k_1) = (j_2, k_2); \\ 0, & \text{otherwise.} \end{cases} \tag{1.2.2}$$

Then any $f \in L_2(\mathbb{R})$ has expressions:

$$f = \sum_{j,k \in \mathbb{Z}} \langle f, \tilde{\psi}_{j,k} \rangle \psi_{j,k} = \sum_{j,k \in \mathbb{Z}} \tilde{W} f(j,k) \psi_{j,k}, \tag{1.2.3}$$

and

$$f = \sum_{j,k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \tilde{\psi}_{j,k} = \sum_{j,k \in \mathbb{Z}} W f(j,k) \tilde{\psi}_{j,k}, \tag{1.2.4}$$

In particular, if the basis is orthonormal, then $\psi_{j,k} = \tilde{\psi}_{j,k}$.

All the expressions above are referred as *homogeneous representation*s of f. The index j represents different scales (resolutions), and the index k is a shift (translation) index. Then this wavelet basis usually (but not always) leads to a "multi-resolution approximation" of a certain function $f \in L_2(\mathbb{R})$. The partial sum

$$f_j = \sum_{i=-\infty}^{j} \sum_{k=-\infty}^{\infty} \langle f, \psi_{i,k} \rangle \psi_{i,k} \tag{1.2.5}$$

is the projection of f onto the subspace $V_j := \overline{\text{span}\{\psi_{i,k} : i \leq j-1, k \in \mathbb{Z}\}}$. As the index j goes larger, the space will be expanded larger, too. The partial sum in (1.2.5) can be interpreted as the approximation of f at the resolution 2^j, and V_j can be

interpreted as the set of all possible approximations at resolution 2^j. Obviously, as j goes to ∞, f_j can be as close to f as we want.

The projection motivates the definition MRA. The concept of MRA was introduced by S. Mallat in 1989 [12]. MRA gives the approximation of a function to any resolution. An approximation at a lower resolution is contained in a high resolution, and as the resolution goes finer and finer, we can approximate the original function as close as we want. The following definition is adapted from [13].

Definition 1.2.1. *A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $L_2(\mathbb{R})$ is a multi-resolution analysis if the following 6 properties are satisfied:*

(1) $f(t) \in V_j \Leftrightarrow f(t-2^j k) \in V_j$ for any $j,k \in \mathbb{Z}$;

(2) $V_{j+1} \subseteq V_j$ for any $j \in \mathbb{Z}$;

(3) $f(t) \in V_j \Leftrightarrow f(t/2) \in V_j$ for any $j \in \mathbb{Z}$;

(4) $\lim_{j\to-\infty} V_j = \cap_{j=-\infty}^{+\infty} V_j = \{0\}$;

(5) $\lim_{j\to+\infty} V_j = \overline{\cup_{j=-\infty}^{+\infty} V_j} = L_2(\mathbb{R})$;

(6) $\exists\ \theta$ such that $\{\theta(t-k)\}_{k\in\mathbb{Z}}$ is a Riesz basis of V_0.

By this nested sequence, we can project any L_2 function onto a space V_j, which leads to an approximation at a certain *resolution* or *scale*. To compute the projection, we need an orthonormal basis for each V_j. Orthogonalizing the Riesz basis $\{\theta(t-k)\}_{k\in\mathbb{Z}}$ of V_0, we get an orthonormal basis for V_0. Then an orthogonal basis of each V_j is constructed by dilating and translating a single function $\phi(t) \in V_0$ in [13]. The family

$$\left\{\phi_{j,k}(t) = 2^{-j/2}\phi\left(\frac{t-2^j k}{2^j}\right)\right\}_{k\in\mathbb{Z}} \tag{1.2.6}$$

forms an orthonormal basis of V_j for any $j \in \mathbb{Z}$. Larger j is corresponding to smaller scaling factor 2^{-j} in its basis function $\phi_{j,k} = 2^{-j/2}\phi(2^{-j}t-k)$, and thus represents a finer scale. Vice versa, smaller j represents a coarser scale.

Now, the difference between two approximations of f at the resolution 2^{j+1} and 2^j

is the orthogonal complement of V_j in V_{j+1}:

$$W_j = V_{j+1} \ominus V_j, \tag{1.2.7}$$

hence we have

$$V_j \oplus W_j = V_{j+1}, \tag{1.2.8}$$

$$\begin{aligned} L_2(\mathbb{R}) &= \ldots \oplus W_{-j} \oplus \ldots \oplus W_0 \oplus \ldots \oplus W_j \oplus \ldots & (1.2.9)\\ &= V_0 \oplus W_0 \oplus \ldots \oplus W_j \oplus \ldots. & (1.2.10) \end{aligned}$$

We can actually find a function $\psi(t) \in W_0$ such that

$$\left\{ \psi_{j,k}(t) = 2^{-j/2} \psi \left(\frac{t - 2^j k}{2^j} \right) \right\}_{k \in \mathbb{Z}} \tag{1.2.11}$$

is an orthonormal basis of W_j for any $j \in \mathbb{Z}$ [13]. So MRA can yield a wavelet function.

The scaling function, which sometimes is also referred as *father wavelet*. From both mother and father wavelet, we get the decomposition of $L_2(\mathbb{R})$:

$$\begin{aligned} L_2(\mathbb{R}) &= V_J \oplus W_J \oplus W_{J+1} \oplus \cdots \oplus W_{J+j} \oplus \cdots & (1.2.12)\\ &= \overline{\text{span}\{\phi_{J,k}\}}_{k \in \mathbb{Z}} \oplus \overline{\text{span}\{\psi_{j,k}\}}_{j \geq J, k \in \mathbb{Z}}. & (1.2.13) \end{aligned}$$

Usually we take $J = 0$, then any signal $f(t)$ with finite energy ($f \in L_2(\mathbb{R})$) can be written in $L_2(\mathbb{R})$ as

$$f(t) = \sum_{k \in \mathbb{Z}} \langle f, \phi_{0,k} \rangle \phi_{0,k}(t) + \sum_{j \geq 0, n \in \mathbb{Z}} \langle f, \psi_{j,n} \rangle \psi_{j,n}(t). \tag{1.2.14}$$

It is often called the nonhomogeneous wavelet representation of $f(t)$. The first portion $\sum_{k \in \mathbb{Z}} \langle f, \phi_{0,k} \rangle \phi_{0,k}(t)$ is the projection of $f(t)$ onto V_0, which is the part in the coarsest scale. Vice versa, the second part $\sum_{j \geq 0, n} \langle f, \psi_{j,n} \rangle \psi_{j,n}(t)$ is the projection on to $\oplus_{j=1}^{\infty} W_j$,

which is the finer scales. So that we can see the low-pass nature of the scaling function (father wavelet), and the high-pass nature of the wavelet function (mother wavelet).

The wavelet function is also called an MRA wavelet associated with ϕ. Most but not all wavelet functions are associated with the MRA structure. (J.L. Journé first gave a wavelet example that has no associated MRA [11].) By the homogeneous and nonhomogeneous representation, we further denote the set of all the basis vectors as *affine systems* in $L_2(\mathbb{R})$. For every integer $J \in \mathbb{Z}$, a (nonhomogeneous) affine system $\mathsf{AS}_J(\phi;\psi)$ is defined to be

$$\mathsf{AS}_J(\phi;\psi) := \{\phi_{J;k} : k \in \mathbb{Z}\} \cup \{\psi_{j,k} : j \geq J, k \in \mathbb{Z}\}, \tag{1.2.15}$$

and the homogeneous affine system $\mathsf{AS}(\psi)$ is

$$\mathsf{AS}(\psi) := \{\psi_{j;k} : j, k \in \mathbb{Z}\}. \tag{1.2.16}$$

In higher dimensions (usually 2 dimensions, which is the typical case for image processing), we need to generalize the wavelet theory to $L_2(\mathbb{R}^2)$. We generalize the wavelet to 2-D by tensor product. The tensor product of two 1-D functions f, g is a 2-D function

$$(f \otimes g)(t_1, t_2) = f(t_1)g(t_2). \tag{1.2.17}$$

There are two routes to construct 2-D wavelets by tensor product [14]. We can either construct a 2-D wavelet basis from 1-D wavelets by choosing multi-resolution approximations given by tensor products, or start with an arbitrary 2-D MRA and apply the same analogue as we did in 1-D case. From the 1-D scaling function ϕ and wavelet function ψ, we have in total 4 combinations: $\phi \otimes \phi$, $\phi \otimes \psi$, $\psi \otimes \phi$ and $\psi \otimes \psi$. Usually the space generated by the translations and dilations of these four functions are

respectively denoted as $V_0 \otimes V_0$, $W^{0,1}$, $W^{1,0}$, $W^{1,1}$, and it is clear that

$$V_1 \otimes V_1 = V_0 \otimes V_0 \oplus W^{0,1} \oplus W^{1,0} \oplus W^{1,1}. \tag{1.2.18}$$

Utilizing the wavelet coefficients, namely, processing in the frequency domain, we can conduct various signal processings on different scales in 2-D. We will now discuss the advantages of wavelet transform.

1.2.2 Localization of Wavelet Transform

As mentioned in the previous section, a singularity in an image is a big jump or drop of coefficients' values within a relatively small area. To obtain wavelet coefficients indicating such singularities requires the wavelet function to be localized with respect to the shape of the singularity.

Fortunately, the wavelet transform is a localized transform. It enables us to do local image processing. By the 2 parameters j and k indicating the scale and location respectively, we have 2 degrees of freedoms which make it possible to give specific processing to different scales, and different segments of the image. When conducting inner product with compactly supported wavelets, a large increment in the time domain results in a large coefficient in the wavelet domain, and any singularity can be approximated, as long as we reach a sufficiently fine scale. This is one of the essential advantages of wavelets when compared with the more classic Fourier transform. It is difficult to use Fourier transform to detect the location of any singularities.

If it is a dot-shaped singularity, a compactly supported wavelet always achieves this by dilation and translation to match the position of the singularity. But when it comes to line-shaped singularities, the strategy of tensor product becomes incapable to tackle with them. Let us take a look at a classical example of wavelet system, the

Haar wavelet:

$$\phi(t) = \begin{cases} 1, & \text{for } t \in [0,1]; \\ 0, & \text{otherwise;} \end{cases} \tag{1.2.19}$$

$$\psi(t) = \begin{cases} 1, & \text{for } t \in [0,1/2); \\ -1, & \text{for } t \in [1/2,1); \\ 0, & \text{otherwise.} \end{cases} \tag{1.2.20}$$

After tensor product, on the unit square $[0,1) \times [0,1)$,

$$\phi \otimes \phi(t_1, t_2) = 1; \tag{1.2.21}$$

$$\phi \otimes \psi(t_1, t_2) = \begin{cases} 1, & \text{for } t_2 \in [0,1/2); \\ -1, & \text{for } t_2 \in [1/2,1); \end{cases} \tag{1.2.22}$$

$$\psi \otimes \psi(t_1, t_2) = \begin{cases} 1, & \text{for } (t_1,t_2) \in [0,1/2)^2 \cup [1/2,1)^2; \\ -1, & \text{otherwise;} \end{cases} \tag{1.2.23}$$

which has only block-shaped singularities. It is difficult to detect line-shaped singularities from the wavelet transform.

But line-shaped singularities have a special property: after Fourier transform, they are somewhat preserved in the frequency domain, just with a different direction. We give a simple example in Fig 2. The image on the left is the original image (time domain) with a diagonal singularity. Its Fourier transform (frequency domain) presents also a diagonal singularity, but with a rotation of $\pi/2$.

This property indicates that, if we want to catch the line-shaped singularities as well as we can, we must find a wavelet function which has good localization in both the time and frequency domain. This is the motivation for most research work and upcoming methods for directional image processing. It also inspires a natural idea to make a frequency design: when we focus on the directionality in the frequency domain, the filter bank still preserves the directionality when we get back to the time domain.

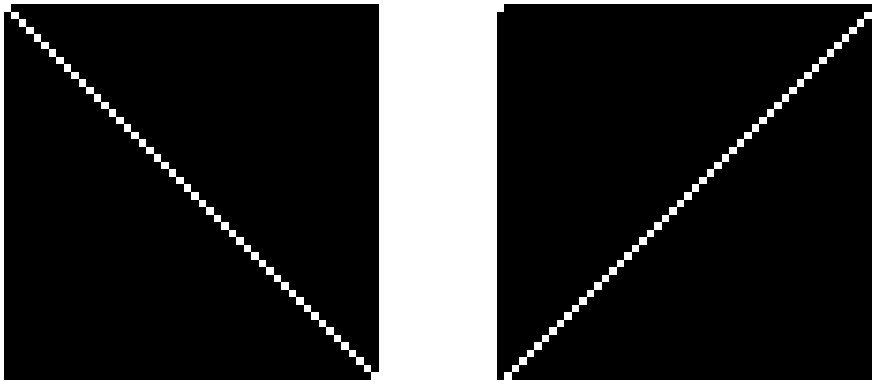


Figure 2: An example of directional singularity and its Fourier transform

Curvelets is one of the most successful frequency-design methods, which we are going to talk about in the coming section.

1.3 Curvelets Transform

Among all the "X-lets" methods, the curvelets method is a transform which can be performed in any direction, and thus makes a good extraction of directional singularities. Based on multi-scale ridgelets and a spatial bandpass filtering operation to isolate different scales, curvelets occur at all scales, locations, and directions. The 2-D continuous ridgelets are defined as

$$\psi_{a,b,\theta}(t) = a^{-1/2}\psi((t_1\cos\theta + t_2\sin\theta - b)/a).$$

This function is constant along lines $t_1 cos\theta + t_2\sin\theta = \text{const}$. And it has 3 parameters a, b, θ corresponding to the scale, location and direction(angle) respectively, which makes the directional processing possible. The first-generation curvelets transform was introduced by Starck et al. in 2000 [16]. Its limitation is that the geometry of the ridgelet is unclear itself, because they are not true ridge functions in digital images. Therefore Candés and Donoho proposed a simpler version based on the frequency partition techniques, which is usually referred to as the second-generation curvelet transform [1, 2]. It has been shown to be very effective in image processing [10].

The curvelet construction is based on polar coordinates in the frequency domain. We partition the frequency domain into circles centered at the origin, and then they are cut into wedges as shown in Fig 3. With frequency functions almost disjointly supported on each wedges, theoretically we can get as many orientations as we want.

Using the ansatz

$$\hat{\phi}_{j,0,0}(r,\omega) := 2^{-3j/4}W(2^{-j}r)\tilde{V}_{N_j}(\omega),\ r \geq 0, \omega \in [0, 2\pi), j \in N_0,$$

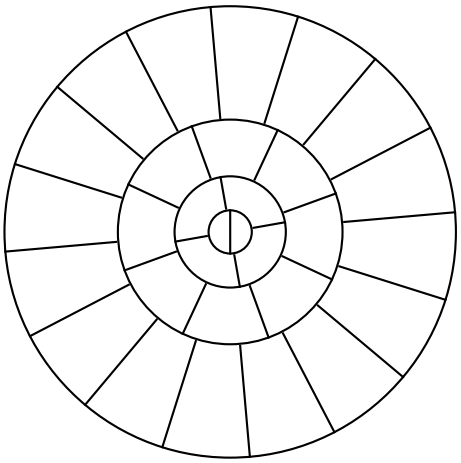


Figure 3: Frequency partition for curvelet transform

where W and $\tilde{V}_{N_j}$ are suitable window functions for the radius and angle respectively, we get the "dilated basic curvelets". This is the fundamental element in the curvelet transform. If we partition the whole frequency plane into circular rings, and continue to partition a particular ring with a number of wedges, we can use a tiling method to cover the frequency domain. We denote the number of wedges in the circular ring at scale 2^j as N_j. Then under polar coordinates, we just need to construct curvelet elements near the wedges with compact support inside $[-2\pi/N_j, 2\pi/N_j]$. By rotations, it can cover the whole ring of a certain scale.

This polar coordinate system, however, is not possible to implement exactly in time-domain. Usually an image is in a rectangle or square domain, with discrete points/pixels, and can be treated as a function defined on a bounded integer grid in $\mathbb{R}^2$. It is natural to consider the signal in Cartesian coordinates, by using Cartesian arrays. Based on concentric squares and shears,one makes a discrete curvelet transform. The wedges are replaced with trapezoids. The ansatz is correspondingly

$$\hat{\tilde{\phi}}_{j,0,0} := 2^{-3j/4} W(2^{-j}\xi_1) V\left(\frac{2^{\lfloor j/2 \rfloor}\xi_2}{\xi_1}\right),$$

which plays the same role as the dilated basic curvelets in continuous setting. Then

the curvelet transform is given as

$$\tilde{c}_{j,k,l}(f) = \langle f, \tilde{\phi}_{j,k,l} \rangle = \int_{\mathbb{R}^2} \hat{f}(\mathrm{S}_{\theta_{j,l}}\xi)\hat{\tilde{\phi}}_{j,0,0}(\xi)e^{i\langle k_j,\xi\rangle}d\xi,$$

with $k_j = (k_1 2^{-j}, k_2 2^{-\lfloor j/2 \rfloor})^T, (k_1, k_2)^T \in \mathbb{Z}^2$. And $\mathrm{S}_{\theta_{j,l}}$is the shear matrix

$$\mathrm{S}_{\theta_{j,l}} = \begin{pmatrix} 1 & 0 \\ -\tan(\theta) & 1 \end{pmatrix}.$$

This is the forward discrete curvelet transform with a computational cost of $\mathcal{O}(N^2 \log N)$ for an $N \times N$ image.

Curvlets method is a very natural and nice attempt to obtain directionality. It can give as many directions as we want, and is very successful in image processing. Especially the second-generation curvelets, since proposed in 2004, have been extensively used in denoising, classification, and feature extraction. However, there are several major drawbacks which make it imperfect: the first one is the design in continuous setting makes it difficult to implement with practical discrete signals. In practice, we can only perform discrete Fourier transform on a discrete image. The frequency domain is then still discrete. But the segmentation designed in curvelets can result in a non-integer index, which makes it difficult to implement. The second one is the high redundancy for discrete curvelet transform. Although the theory about it has been very successful, discrete curvelet transform is not appropriate for image compression. As mentioned above, the computational cost is $\mathcal{O}(N^2 \log N)$ to perform the forward curvelet transform. Compared with common wavelet transform with finite support, (computational complexity $\mathcal{O}(N \times N)$) for an $N \times N$ image, this is rather low efficiency. Curvelets are also restricted to cope with only C^2-singularities. When the singularity is beyond C^2, it will be difficult to extract the feature by curvelets.

From curvelets, we can see that a continuous design is not optimal for signal processing. We need an exact and efficient transform on discrete signals (images) to conduct

time-frequency analysis. In this setting,the Discrete Wavelet Transform(DWT) was introduced. We will not go into the detail of this subject here, as decomposition and reconstruction through filter banks are much more commonly used. The concept of filter banks was introduced by Croisier et. al. [6] in 1976. Through a collection of filters consisted of just number arrays and the discrete convolution operation, a filter bank decomposes a signal $f[n]$ into two signals of half its size by a filtering and subsampling procedure. It provides a convenient, fast approach to decompose and reconstruct discrete signals. Closely related to continuous wavelets, they asymptotically approach continuous functions. This makes it possible to obtain stability of the algorithm, and enable us to give the multi-scale analysis in discrete settings. Details are given in the next section.

1.4 Filter Bank Perspective-Discrete Setting

1.4.1 Decomposition and Reconstruction

When we deal with a discrete sequence $\{v(n)\}_{n\in\mathbb{Z}}$ instead of a continuous function, the decomposition of this sequence will be different from the decomposition of functions. Using a dyadic decomposition, we need to map the sequence into a coarse half-resolution approximation. This can be done by filtering with a halfband low-pass filter followed by a downsampling [17].

Let us denote the set of all complex-valued sequences on the integer grid as $l(\mathbb{Z})$, then in wavelet analysis or signal processing, a filter is a sequence $\{u(k)\}_{k\in\mathbb{Z}} \in l(\mathbb{Z})$. It can be finite or infinite. If the support of the filter, defined as $\text{fsupp}(u) := \{k \in \mathbb{Z} : u(k) \neq 0\}$ is a finite set, the filter is said to be finitely or compactly supported.

In MRA, the integer translations of the scaling function ϕ generates V_0, and by

definition, $V_0 \subseteq V_1 = \overline{\text{span}\{\phi(2t-k)\}_{k\in\mathbb{Z}}}$, so we can see that

$$\phi(t) = 2\sum_{k\in\mathbb{Z}} a(k)\phi(2t-k) \qquad \text{in } L_2(\mathbb{R}), \tag{1.4.1}$$

where $a \in l(\mathbb{Z})$ is a sequence. (1.4.1) called the refinement equation, which is one of the most important topics in MRA. Any scaling function should satisfy this equation. By this equation, we can define a low-pass filter sequence $\{a(k)\}_{k\in\mathbb{Z}}$.

Cavaretta et al. showed in [3] that for a normalized mask a such that $\hat{a}(0) = 1$, there is a unique solution ϕ up to a multiplicative factor.

Similarly, by $W_0 \subseteq V_1 = \text{span}\{\phi(2t-k)\}_{k\in\mathbb{Z}}$, we can express $\psi \in W_0$ as

$$\psi(t) = 2\sum_{k\in\mathbb{Z}} b(k)\phi(2t-k) \tag{1.4.2}$$

and b is the high-pass filter for ψ. The two filters a and b together constitute a wavelet filter bank. We will discuss further about relation between the filters and the scaling and wavelet functions.

Now associated with the filter, we need to decompose the signal for processing and then reconstruct it. We define the subdivision and transition operator $\mathcal{S}_u$ and $\mathcal{T}_u$ based on the filter u to process a signal $v \in l(\mathbb{Z})$:

$$[\mathcal{S}_u v](n) := 2\sum_{k\in\mathbb{Z}} v(k)u(n-2k),\ n \in \mathbb{Z}, \tag{1.4.3}$$

$$[\mathcal{T}_u v](n) := 2\sum_{k\in\mathbb{Z}} v(k)\overline{u(k-2n)},\ n \in \mathbb{Z}. \tag{1.4.4}$$

It is quite similar to the convolution widely used in signal processing:

$$[u * v](n) := \sum_{k\in\mathbb{Z}} u(k)v(n-k), \quad n \in \mathbb{Z}. \tag{1.4.5}$$

But with the dyadic dilation factor 2, the subdivision operator makes the support

extended, whereas the transition operator will shorten the support of the resulted sequence, corresponding to the reconstruction and decomposition process. We want to write these 2 operators using the notation of convolution, so the upsampling operator $\uparrow d$ and downsampling operator $\downarrow d$ are needed to account for the factor 2, and another filter to account for the complex conjugate. The upsampling operator $\uparrow d$ and downsampling operator $\downarrow d$ are

$$v \uparrow d(n) := \begin{cases} v(n/d), & \text{when d divides n;} \\ 0, & \text{otherwise;} \end{cases} \tag{1.4.6}$$

$$v \downarrow d(n) := v(dn), \tag{1.4.7}$$

and for a filter u, its conjugate $u^\star$ is defined as

$$u^\star(k) = \overline{u(-k)}. \tag{1.4.8}$$

Then, by these notations we can conveniently write the subdivision and transition operator as

$$\mathcal{S}_u v = 2u * (v \uparrow 2) \text{ and } \mathcal{T}_u v = 2(u^\star * v) \downarrow 2. \tag{1.4.9}$$

For time-frequency analysis, we shall define the Fourier transform of a filter $u(k)$, at least formally:

$$\hat{u}(\xi) := \sum_{k \in \mathbb{Z}} u(k) e^{-ik\xi}. \tag{1.4.10}$$

Here i is the imaginary unit satisfying $i^2 = -1$. The Fourier transform of a filter is a 2π-periodic trigonometric polynomial. From this formal definition, we can equivalently write all the operations above in the frequency domain as

$$\widehat{u * v}(\xi) = \hat{u}(\xi)\hat{v}(\xi); \tag{1.4.11}$$

$$\widehat{u^\star}(\xi) = \overline{\hat{u}(\xi)}; \tag{1.4.12}$$

$$\widehat{v \uparrow d}(\xi) = \widehat{v}(d\xi); \tag{1.4.13}$$

$$\widehat{v \downarrow d}(\xi) = d^{-1}\left[\widehat{v}\left(\frac{\xi}{d}\right) + \widehat{v}\left(\frac{\xi + 2\pi}{d}\right) + \ldots + \widehat{v}\left(\frac{\xi + 2\pi(d-1)}{d}\right)\right]. \tag{1.4.14}$$

From (1.4.6) and (1.4.7) we can see that, the downsampling operator extracts a sub-sequence of the signal, whereas the upsampling operator is interpolatory. So the transition operator embedded with a downsampling operator can be used as a decomposition, with different filter u's, and the corresponding "$\mathcal{T}_u$" operators cut the signal into different subbands, hence make a decomposition operator. Then utilizing the property, the decomposition operator with respect to a wavelet filter bank $\{a; b\}$ is defined as

$$\mathcal{W}v := \frac{\sqrt{2}}{2}(\mathcal{T}_a v, \mathcal{T}_b v) =: (\omega_0, \omega_1),\ v \in l(\mathbb{Z}). \tag{1.4.15}$$

Note that $\mathcal{W}$ is a mapping from $l(\mathbb{Z})$ to $l(\mathbb{Z})^2$. The reconstruction operator $\mathcal{V}$ is dually defined from $l(\mathbb{Z})^2$ to $l(\mathbb{Z})$ by usage of the subdivision operator with a upsampling:

$$\mathcal{V}(\omega_0, \omega_1) := \frac{\sqrt{2}}{2}\sum_{l=0,1} \mathcal{S}_a \omega_l, \quad \omega_0, \omega_1 \in l(\mathbb{Z}). \tag{1.4.16}$$

The coefficient $\frac{\sqrt{2}}{2}$ is for the preservation of energy.

For the convenient implementation of these operations, in practice, we are particularly interested of those finitely supported filters

$$u = \{u(k)\}_{k\in\mathbb{Z}} : \mathbb{Z} \to \mathbb{C}$$

such that $\{k \in \mathbb{Z} : u(k) \neq 0\}$ is finite set, which make the subdivision and transition operator well defined and easy to compute.

Corresponding to the orthogonal wavelet basis in (1.2.1), where we compute the wavelet coefficients and express the function using the same basis, the filter banks for decomposition and reconstruction are the same. But they may be different if we use a biorthogonal wavelet basis as in (1.2.3). Whichever the case, a natural desire is that

we want a signal to be identically the same after a decomposition and reconstruction procedure. If we use $\{\tilde{a}, \tilde{b}\}$ for decomposition and $\{a, b\}$ for reconstruction, the filter banks are said to have the *perfect reconstruction (PR)* property if for all signals $v \in l(\mathbb{Z})$,

$$v = \mathcal{V}\tilde{\mathcal{W}}v = \frac{1}{2}\sum_{l=0}^{1} \mathcal{S}_{u_l}\mathcal{T}_{\tilde{u}_l}v \text{ or } \mathcal{W}\tilde{\mathcal{V}} = \mathrm{Id}(l(\mathbb{Z})). \tag{1.4.17}$$

where u_0 stands for the low-pass filter a, and u_1 stands for the high-pass filter b, "Id" is the identity map on $l(\mathbb{Z})$. This pair of filter banks $\{\tilde{a}, \tilde{b}\}$ and $\{a, b\}$, is called a pair of biorthogonal wavelet filter bank. In particular, if $a = \tilde{a}, b = \tilde{b}$, the filter bank is called an orthogonal wavelet filter bank.

In the frequency domain, (1.4.17) can be written as

$$\widehat{v}(\xi) = \frac{1}{2}\sum_{l=1}^{2}[\widehat{\mathcal{S}_{u_l}\mathcal{T}_{\tilde{u}_l}v}] = \sum_{l=1}^{2}\widehat{\mathcal{T}_{\tilde{u}_l}v}(2\xi)\widehat{u}_l(\xi) = \sum_{l=1}^{2}\widehat{v}(\xi)\overline{\widehat{\tilde{u}}_l(\xi)}\widehat{u}_l(\xi) + \sum_{l=1}^{2}\widehat{v}(\xi+\pi)\overline{\widehat{\tilde{u}}_l(\xi+\pi)}\widehat{u}_l(\xi). \tag{1.4.18}$$

Comparing the two sides, it is equivalent to

$$\sum_{l=1}^{2}\overline{\widehat{\tilde{u}}_l(\xi)}\widehat{u}_l(\xi) = 1, \tag{1.4.19}$$

$$\sum_{l=1}^{2}\overline{\widehat{\tilde{u}}_l(\xi+\pi)}\widehat{u}_l(\xi) = 0. \tag{1.4.20}$$

We can also write it in a matrix form

$$\begin{bmatrix} \widehat{\tilde{a}}(\xi) & \widehat{\tilde{b}}(\xi) \\ \widehat{\tilde{a}}(\xi+\pi) & \widehat{\tilde{b}}(\xi+\pi) \end{bmatrix} \begin{bmatrix} \widehat{a}(\xi) & \widehat{b}(\xi) \\ \widehat{a}(\xi+\pi) & \widehat{b}(\xi+\pi) \end{bmatrix}^{\star} = I_2. \tag{1.4.21}$$

where $M^{\star}$ is the Hermite conjugate of the matrix M, and I_2 is the 2×2 identity matrix.

With these 2 filter banks, one can define a scheme of decomposition and reconstruction through the subdivision and transition operators. For a certain signal v, after the 1-level decomposition, the signal is filtered into 2 bands $(\omega_{1,0},\ \omega_{1,1})$ by the low-pass

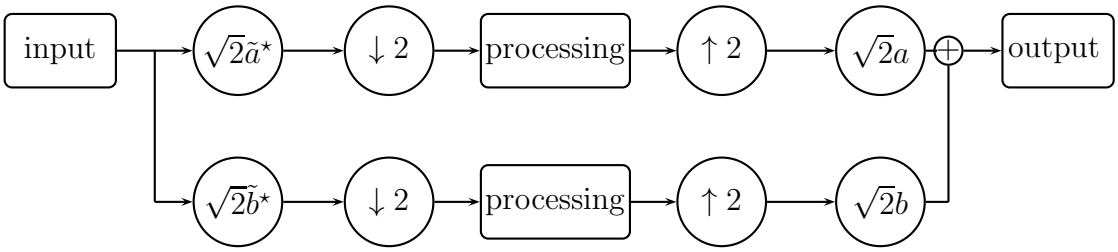


Figure 4: Diagram for 1-level wavelet transform

and high-pass filter respectively. Then after processing in different bands, the reconstruction follows dually: using the transition operator combining $(\omega_{N,0},\ \omega_{N,1})$ to get $\omega_{N-1,0}$, then get $\omega_{N-2,0}$ with $\omega_{N-1,1}$, and finally reconstructs to the original level v_0. The procedure can be conveniently understood in Fig 4 adapted from [6].

For the generalization to 2 dimensions, we still utilize the tensor product. The tensor products of filters are defined as

$$a \otimes a(k,l) = A_1(k,l) = a(k)a(l); \quad a \otimes b(k,l) = A_2(k,l) = a(k)b(l). \tag{1.4.22}$$

From the 2-D tensor product of scaling and wavelet functions, we can see that

$$\begin{aligned} \phi(t) \otimes \phi(s) &= \sum_{k\in\mathbb{Z}} a(k)\phi(2t-k) \sum_{l\in\mathbb{Z}} a(l)\phi(2s-l) && (1.4.23)\\ &= \sum_{(k,l)\in\mathbb{Z}^2} a(k)a(l)\phi(2t-k)\phi(2s-l) && (1.4.24)\\ &= \sum_{(k,l)\in\mathbb{Z}^2} [a\otimes a](k,l)[\phi\otimes\phi](2(t,s)-(k,l)). && (1.4.25) \end{aligned}$$

So the tensor products of functions also satisfy the refinement equations.

The subdivision and transition operator in 2-D is similarly defined, which can be implemented either row by row or column by column.

1.4.2 A Summary of Traditional Wavelet Method

As mentioned before, traditional wavelets only do well in detecting point-shape singularities. Classic wavelet filters are symmetric in both the time and frequency domains, i.e, they have symmetric refinable functions as well as symmetric fourier transforms. The 2-D tensor products of these functions are symmetric in 2 dimensions. If we denote the refinable function of the high-pass filter as ψ, and of the low-pass as ϕ, then in the time domain, the separable implementation of the 2-D DWT is characterized by four wavelets:

$$\phi_1(t_1, t_2) = \phi(t_1)\phi(t_2), \quad \text{LL wavelet}, \tag{1.4.26}$$

$$\psi_2(t_1, t_2) = \psi(t_1)\phi(t_2), \quad \text{HL wavelet}, \tag{1.4.27}$$

$$\psi_3(t_1, t_2) = \phi(t_1)\psi(t_2), \quad \text{LH wavelet}, \tag{1.4.28}$$

$$\psi_4(t_1, t_2) = \psi(t_1)\psi(t_2), \quad \text{HH wavelet}. \tag{1.4.29}$$

Any real function $f(t), t \in \mathbb{R}$ always has a 2-sided spectrum , that is,

$$|\widehat{f}(\xi)| = |\widehat{f}(-\xi)|.$$

Due to the fact, any 2-D tensor product is symmetric in the four quadrants. Thus they only provide two directions-the horizontal and vertical, to help detect the edges in the images. But edges occur in all directions. It is not ideal to restrict our processing only in these two basic directions.

The favorable property of wavelets is that it has a filter bank implementation. This makes it attractive when dealing with discrete signals in practice. This motivates the development of many other wavelet-based image processing methods, utilizing a filter bank in the time domain. Dual tree complex wavelet transform is an excellent example. It is easy to implement, but also with orientations in both the time and frequency domain.

1.4.3 Dual Tree Complex Wavelet Transform

Traditional wavelets are fast and convenient enough to implement, but it is not possible to perform oriental processing with them. Curvelets can cope with any orientation, nonetheless are highly redundant for practical applications. First introduced by Kingsbury et al. in 1998 [9], the Dual-tree Complex Wavelet Transform (dual tree $\mathbb{C}$WT) is a smart compromise approach which achieved great success. It contains several directions to perform directional transform, yet is still computationally economic.

The dual tree complex wavelet transform starts with a tree structure with two discrete wavelet decomposition and reconstruction processes. The two branches are respectively independent wavelet decomposition with two different wavelet filter banks. So the matrix of dual tree complex wavelet transform is given as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F_h} \\ \mathbf{F_g} \end{bmatrix},$$

where $\mathbf{F_h}$ and $\mathbf{F_g}$ are the DWT matrices.

For a real signal x, we denote $w_h := \mathbf{F_h}x$, $w_g := \mathbf{F_g}x$, and define the complex coefficients of dual tree $\mathbb{C}$WT as $\frac{1}{\sqrt{2}}(w_h + iw_g)$. If we want the corresponding wavelet function (denoting as $\psi(x) = \psi_h(x) + i\psi_g(x)$) to be analytic, we need

$$\psi_g(x) = \mathcal{H}\psi_h(x),$$

where $\mathcal{H}$ is the Hilbert transform [15]. This relation may hold only approximately for practice.

Based on 1-D dual tree $\mathbb{C}WT$, by tensor product we obtain 2-D dual tree $\mathbb{C}$WT which produces oriented wavelets. Consider the 2-D wavelet $\psi(x_1, x_2) = \psi(x_1)\psi(x_2)$, where ψ is a complex (approximately analytic) wavelet just given in the form $\psi(x) =$

$\psi_h(x) + i\psi_g(x)$. We obtain

$$\psi_1(x_1, x_2) = [\psi_h(x_1) + i\psi_g(x_1)][\psi_h(x_1) + i\psi_g(x_2)].$$

The spectrum of complex analytic wavelet is supported on only one side, hence the tensor product can be supported in one quadrant of the frequency plane. And the real/complex part of the complex wavelet above has directionality at +45 degree. In the time domain, it is $+135/-45$ degree. The other 2-D complex wavelet is given as

$$\psi_1(x_1, x_2) = [\psi_h(x_1) + i\psi_g(x_1)]\overline{[\psi_h(x_1) + i\psi_g(x_2)]}.$$

This one has a -45 degree directionality in the frequency domain and a +45 degree directionality in the time domain. In total, the combination of LL, LH, HL, HH will result in six directions.

Dual tree complex wavelet transform has some nice properties, too, such as near shift-invariance, near rotation invariance. Its advantages to select directions make it very powerful for image processing, like image rotation, estimating geometric structures, and denoising, etc. Its computational cost, is only double of traditional DWT, which is much less than curvelets, shearlets. Since both branches of the dual tree structure are just traditional wavelet transform, it benefits from the vast resources of traditional wavelets in both theory and computation [15].

There is still a problem for dual tree $\mathbb{C}$WT. The 1-sided support of the wavelets in the frequency domain makes it impossible to have any vertical or horizontal directions. But these are the two most natural directions for images. It is also restricted to 2 branches. This lack of flexibility makes it difficult to have more directions.

1.5 Framelet Approach

1.5.1 Framelets and Affine Systems in $L_2(\mathbb{R})$

In Section 1.2, for an integer $J \in \mathbb{Z}$, we define the nonhomogeneous and homogeneous affine systems with respect to the scaling function ϕ and wavelet function ψ as

$$\mathsf{AS}_J(\phi;\psi) := \{\psi_{J;k} : k \in \mathbb{Z}\} \cup \{\psi_{j,k} : j \geq J, k \in \mathbb{Z}\}, \tag{1.5.1}$$

and the homogeneous affine system $\mathsf{AS}(\phi;\psi)$ is

$$\mathsf{AS}(\psi) := \{\psi_{j;k} : j,k \in \mathbb{Z}\}. \tag{1.5.2}$$

If we generalize the generator(s) of the affine systems from two functions ϕ and ψ to two subsets (allow multiplicity) Φ and Ψ of $L_2(\mathbb{R})$, we can define the affine systems as below:

$$\mathsf{AS}_J(\Phi;\Psi) := \{\phi_{J;k} : k \in \mathbb{Z}, \phi \in \Phi\} \cup \{\psi_{j,k} : j \geq J, k \in \mathbb{Z}, \psi \in \Psi\}, \tag{1.5.3}$$

$$\mathsf{AS}(\Psi) := \{\psi_{j;k} : j,k \in \mathbb{Z}, \psi \in \Psi\}. \tag{1.5.4}$$

When $\mathsf{AS}_0(\Phi;\Psi)$ is a frame for $L_2(\mathbb{R})$ we say that $\{\Phi;\Psi\}$ is a framelet in $L_2(\mathbb{R})$. If another affine system $\mathsf{AS}_0(\widetilde{\Phi};\widetilde{\Psi})$ is the dual frame of it, the pair $(\{\Phi;\Psi\};\{\widetilde{\Phi};\widetilde{\Psi}\})$ is called a dual framelet in $L_2(\mathbb{R})$. In particular, if it is a (normalized) tight frame, meaning its dual frame is just itself, $\{\Phi;\Psi\}$ is a tight framelet. More explicitly, $\{\Phi;\Psi\}$ is a tight framelet in $L_2(\mathbb{R})$ if

$$\|f\|^2_{L_2(\mathbb{R})} = \sum_{h \in \mathsf{AS}(\Phi;\Psi)} |\langle f,h\rangle|^2. \tag{1.5.5}$$

1.5.2 Framelet Filter Banks and Discrete Framelet Transform (DFrT)

For a framelet $\{\Phi; \Psi\}$ if we consider in the frequency domain,

$$\widehat{\Phi} = \{\widehat{\phi} : \phi \in \Phi\}, \widehat{\Psi} = \{\widehat{\psi} : \psi \in \Psi\}, \tag{1.5.6}$$

then $\widehat{\mathsf{AS}_J(\{\Phi; \Psi\})}$ is called a frequency-based affine system in $L_2(\mathbb{R})$, and $(\{\widehat{\Phi}; \widehat{\Psi}\})$ is a frequency-based framelet.

Note the refinement equations (1.4.1) and (1.4.2) can be written equivalently in the frequency domain as

$$\widehat{\phi}(\xi) = \widehat{a}(\xi/2)\widehat{\phi}(\xi/2); \tag{1.5.7}$$

$$\widehat{\psi}(\xi) = \widehat{b}(\xi/2)\widehat{\phi}(\xi/2). \tag{1.5.8}$$

We generalize these refinement relation to framelets. For convenience, we shall consider Φ to be a singleton $\{\phi\}$ and Ψ to be a finite set $\{\psi^1, \ldots, \psi^s\}$ for some positive integer s. Then the filter bank associated to this framelet can be similarly defined. It is written as $\{a; b_1, \ldots, b_s\}$, where $a, b_1, \ldots, b_s$ are sequences in $l_2(\mathbb{Z})$ with the Fourier transform defined in (1.4.10) satisfying (1.5.7) and

$$\widehat{\psi^l}(\xi) = \widehat{b_l}(\xi/2)\widehat{\phi}(\xi/2), \text{ for } l = 1, \ldots, s. \tag{1.5.9}$$

There can be a dual framelet filter bank $\{\widetilde{a}; \widetilde{b_1}, \ldots, \widetilde{b_s}\}$ satisfying refinement equations for the dual frame $\{\widetilde{\Phi}; \widetilde{\Psi}\}$. The two filter banks are called a pair of dual framelet filter banks.

Similar to wavelet filter banks, we use this filter bank to find a representation of

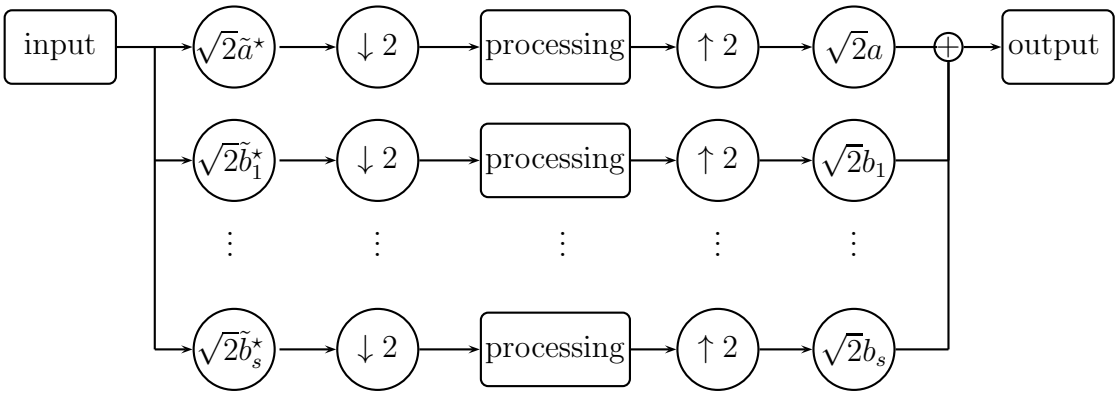


Figure 5: Diagram for 1-level discrete framelet transform

discrete signals. The decomposition and reconstruction procedure and perfect reconstruction property are very much like defined above in (1.4.15) and (1.4.16):

$$\mathcal{W}v := \frac{\sqrt{2}}{2}(\mathcal{T}_{u_0}v, \mathcal{T}_{u_1}v, ..., \mathcal{T}_{u_s}v) =: (\omega_0, \omega_1, ..., \omega_s),\ v \in l(\mathbb{Z}); \tag{1.5.10}$$

$$\mathcal{V}(\omega_0, \omega_1, ..., \omega_s) := \frac{\sqrt{2}}{2}\sum_{l=1}^{s}\mathcal{T}_{u_l}\omega_l. \tag{1.5.11}$$

This is called discrete framelet transform (DFrT). It can be regarded as increasing the number of high-pass filters in a wavelet filter bank. The diagram in Fig 5 (adapted from [6]) shows the procedure.

The perfect reconstruction(PR) property is desirable for the DFrT, too. We say the filter bank ($\{u_0, ..., u_s\}$,
$\{\tilde{u}_0, ..., \tilde{u}_s\}$) has the PR property if for all signals $v \in l(\mathbb{Z})$,

$$v = \mathcal{V}\mathcal{W}v = \frac{1}{2}\sum_{l=0}^{s}\mathcal{S}_{u_l}\mathcal{T}_{\tilde{u}_l}v. \tag{1.5.12}$$

The PR condition can be equivalently expressed in the matrix form

$$\begin{bmatrix} \widehat{\widetilde{a}}(\xi) & \widehat{\widetilde{b}}_1(\xi) & \dots & \widehat{\widetilde{b}}_s(\xi) \\ \widehat{\widetilde{a}}(\xi+\pi) & \widehat{\widetilde{b}}_1(\xi+\pi) & \dots & \widehat{\widetilde{b}}_s(\xi+\pi) \end{bmatrix} \begin{bmatrix} \widehat{a}(\xi) & \widehat{b}_1(\xi) & \dots & \widehat{b}_s(\xi) \\ \widehat{a}(\xi+\pi) & \widehat{b}_1(\xi+\pi) & \dots & \widehat{b}_s(\xi+\pi) \end{bmatrix}^\star = I_2. \tag{1.5.13}$$

Note that a dual framelet filter bank with $s = 1$ is a biorthogonal wavelet filter bank, and a tight framelet filter bank with $s = 1$ is an orthogonal wavelet filter bank. If $s > 1$, the framelet is actually redundant for representing a signal: the length of each subsignal is half of the original one due to the dyadic downsampling, but we have $s+1 > 2$ subsignals, whose sum exceeds the size of the input. This redundancy makes DFrT not appropriate for data compression, however, it provides possibility to obtain more features by designing different high-pass filters.

For the 2-D case (image processing), we still use the trick of tensor product. More details about the properties of the filter bank's tensor product are given in Chapter 3. Provided that the refinable functions of the high-pass filters are concentrated on a single side, we can get a directional refinable function in 2-D by tensor product. We deal with the tight framelet filter banks with 2 generators, that is, 2 high-pass filters. Inheriting the notations as in (1.4.26) to (1.4.29), we denote the tensor products of filters as L-L, L-H1, L-H2, H1-L, H1-H1, H1-H2, H2-L, H2-H1, H2-H2. Provided that the 1-D filter has an asymmetric refinable function, the tensor products will obtain orientations. It can be shown that tight framelet filter banks with all real coefficients are always symmetric (i.e., 2-sided) in the frequency domain. So we are interested in complex-valued filter banks. By the tensor product of complex values, the 2-D filter bank can have orientations in both time and frequency domain when we take the real or imaginary part. In 2-D the real and imaginary parts of L-H1, L-H2, H1-L, H1-H1, H1-H2, H2-L, H2-H1, H2-H2 will have 4 directions in total: horizontal, vertical, $+45°$, $-45°$.

1.5.3 Vanishing Moments and Sum Rules of Filters

Besides the PR condition, vanishing moment or sum rule of a filter is also desirable property. They are relevant to the sparse representation of a signal after DFrT. Naturally, when a smooth signal is input, we want most of the framelet coefficients to be zero so that we can use only a few coefficients to represent the signal. It actually requires the smoothness of the filter in the frequency domain. Any analytic functions can be approximated by polynomials (Taylor expansion) so we use polynomials as a model for smooth signals. But polynomials are defined in the continuous setting. How can a discrete signal be described as a polynomial? Notice that sampling a polynomial p on integers will give us a sequence $\{p_k = \mathsf{p}(k)\}_{k\in\mathbb{Z}}$, conversely, given such a sequence, a polynomial p is uniquely determined. So we use the sequence, which is just sampling the polynomial on integers, to represent the smooth polynomial signal.

Denote the set of all polynomial sequences with highest degree no greater than p as Π_p, and the set of all polynomials sequences as Π . A filter u (or its Fourier transform $\widehat{u}$) is said to have m vanishing moments if $\widehat{u}(\xi) = (1-e^{-i\xi})^m Q(\xi)$ for some 2π-periodic trigonometric polynomial $Q(\xi)$. We denote the vanishing moment of a filter u as $vm(u)$. Obviously this is a property for high-pass filters as $\widehat{u}(0) = 0$. For low-pass filters we have a corresponding concept, sum rule, which is defined as $\widehat{u}(\xi) = (1+e^{-i\xi})^m Q(\xi)$ for some $2\pi-$periodic trigonometric polynomial $Q(\xi)$. The sum rule of a filter u is denoted as $sr(u)$.

With these two properties, a framelet filter bank has a sparse representation for smooth input signals, thus makes the processing for smooth signals quite convenient and fast. Moreover, this gives information of the smoothness of input in coefficients. Back to our ultimate aim, the singularities can be approximated by only high order polynomials thus made the coefficients large at the corresponding area. This is very important for directional image processing.

DFrT is a filter bank-based method. It is easy and economic to implement, with compactly supported filters the time complexity to perform a 1-level DFrT is $\mathcal{O}(N)$.

When it has certain vanishing moments and sum rules, coefficients reflect the existence of singularities with large framelet coefficients. Besides the 4 directions given above, it is also flexible to extend, just by increasing the number of high-pass filters, we can theoretically obtain more combinations of tensor products and of course provide more directions. All of these are the reasons that we choose this approach.

1.6 Motivation

To capture directional singularities, we need a filter bank which is directional in the time domain. Complex-valued 2-D filter banks with 4 directions obtained by tensor product have been initially introduced in [7]. However the construction is done in the frequency domain, and the tight framelet filter banks obtained there have infinite supports. To minimized the computational cost, we want the transform associated with this filter bank to be easy to perform. Usually, this means a compactly supported filter bank. Therefore, the major task of this thesis is to construct finitely supported complex tight framelet filter banks with 4 directions in 2-D.

An image with a directional singularity in the time domain has a similar directional singularity in the frequency domain, just with an angle of $\pi/2$. Fig 2 shows the relation between the directional singularities. So we base our construction on a frequency separation. We want to find 1-D filters that concentrate on one side. Then the tensor product is concentrated in one certain quadrant in the frequency domain, so when we go back to the time domain, they have diagonal directions. The low-pass nature of the mask makes it peak at the origin in the frequency domain, and thus it is difficult to have a frequency separation. So we turn to working on the high-pass filters. To be more specific, we are given a symmetric low-pass filter, how can we construct 2 high-pass filters so that they form a tight framelet filter bank, and the tensor products of these 2 high-pass filters have directions in 2-D.

In Chapter 2, we give a quite trivial proof to show that, a filter bank with real

coefficients is always symmetric. There is no possibility of having any directionality in 2-D using the tensor product. So the problem is therefore about complex-valued framelet filter banks.

We are focused to study compactly supported tight framelet filter banks with 2 generators, i.e., 2 high-pass filters with compact support in the time domain. This makes us benefit from a series of existing theoretical results of construction. On taking the tensor product, we can obtain 4 directions in total. These four directions are much more natural than the directions given by Dual-Tree $\mathbb{C}$WT. And the work flow follows the tree-structure of framelet transforms. With the compact support of the filters in the time domain, we can get rid of heavy computational burden.

We shall first theoretically solve this problem in Chapter 2, where a necessary and sufficient condition for the existence of such tight framelet filter banks is given. Since the nonuniqueness of these filter banks, we derive its optimization explicitly. In Chapter 3, we talk about the relation between the wavelet function and discrete systems in 2-D, as it is the foundation for the search of an optimized filter bank. Chapter 4 follows with the specific algorithm and several interesting examples with clear directionality. At last, concluding remarks will be given in Chapter 5.

Chapter 2

Existence and Optimization of Directional Framelet Filter Banks

2.1 Background

As mentioned in Chapter 1, to obtain directions in the time domain, we focus on a related aspect, the orientations in the frequency domain. We wish to construct such a 2-D filter bank: it is a tensor product of a 1-D filter bank $\{a; b_1, b_2\}$ with itself, and has more directional features other than the vertical and horizontal ones. This requires the 1-D filter bank $\{a; b_1, b_2\}$ to be asymmetric, or with concentration (unbalance) in frequency domain. In terms of the 2π-periodic function $\widehat{b_l}(\xi)$, we are aiming to make the energy on one side $\int_0^\pi |\widehat{b_l}(\xi)|^2 d\xi$ or $\int_{-\pi}^0 |\widehat{b_l}(\xi)|^2 d\xi$ maximized (or equivalently, minimized).

Considering the magnitude of the Fourier transform, it can be easily shown that a filter u with real coefficients will always have a 2-sided spectrum, or to be symmetric in the frequency domain. That is, $|\widehat{u}(\xi)| = |\widehat{u}(-\xi)|$ for any $\xi \in \mathbb{R}$:

$$\begin{aligned} |\widehat{u}(\xi)| &= |\sum_{k\in\mathbb{Z}} u(k)e^{-ik\xi}| = |\overline{\sum_{k\in\mathbb{Z}} u(k)e^{-ik\xi}}| && (2.1.1)\\ &= |\sum \overline{u(k)}e^{ik\xi}| = |\sum u(k)e^{ik\xi}| && (2.1.2)\\ &= |\widehat{u}(-\xi)|. && (2.1.3) \end{aligned}$$

This filter is extended to 2 dimensions by tensor product. But due to its symmetry, it cannot have any orientations other than the horizontal and vertical ones. Hence we need

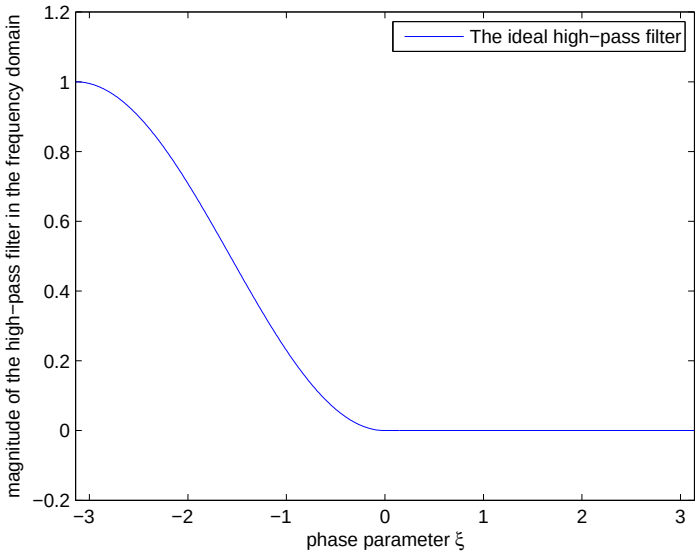


Figure 6: An ideal high-pass filter in the frequency domain (magnitude).

complex-valued coefficients, leading to asymmetry in the frequency domain. Actually, we want a frequency separation. The better the frequency domain is separated, the better directionality we can possibly get in 2-D. An ideal high-pass filter which is supported entirely on $[-\pi, \pi]$ is shown in Fig 6.

When the low-pass filter a has symmetry in the frequency domain, we naturally want to make the magnitude of the 2 high-pass filters in frequency domain mutually symmetric, say, $|\widehat{b}_1(\xi)| = |\widehat{b}_2(-\xi)|, \xi \in [-\pi, \pi]$. That is why we can benefit from the results known for construction of symmetric framelet filter banks.

The construction of directional tight framelet filter banks is based on the asymmetry of the high-pass filters. To put it ideally and simply, we consider the situation of a high-pass filter b with spectrum $\widehat{b}$ completely supported on one side, $[0, \pi]$, (as in Fig 6). The tensor product $b \otimes b$ will then have a spectrum $\widehat{b} \otimes \widehat{b}$ supported inside $[0, \pi] \times [0, \pi]$, which is the first quadrant (as in Fig 2.1). Then since a real function always has a spectrum symmetric about the origin, the real part of the wavelet function in the frequency domain is supported within the first and third quadrant, making an angle of 45°. By the property of the Fourier transform, in the time domain we will get a

Figure 7: Tensor product of the ideal high-pass filter with itself in frequency domain (magnitude)

Figure 8: Tensor product of the ideal high-pass filter with itself in frequency domain (real part)

direction along the angle of $-45°$.

In this chapter, we shall first introduce some definitions and notations for the sake of convenience for theoretical study. Then a necessary and sufficient condition is given for the existence of a mutually symmetric tight framelet filter bank with 2 generators. This theory guarantees an infinite number of such filter banks, so we shall also talk about the optimization of these filters in the last section.

2.1.1 Laurent Polynomials and Matrix Splitting

Let us begin with several important notations about Laurent polynomials, which are significant in our theoretical study. It is actually the so-called Z-transform of the filter, and can be regarded as a generalization of the Fourier transform. The Laurent polynomial notations can be used towards any description of the Fourier transform, and giving convenient definition of the symmetry of filters. More importantly, we then transfer the construction problem of filter bank into a matrix splitting problem, making us benefit from known results such as Fejér-Riesz lemma.

For a filter $u \in l(\mathbb{Z})$, its corresponding Laurent polynomial is defined as a function, indeed, a polynomial from $\mathbb{C}\backslash\{0\}$ to $\mathbb{C}$:

$$\mathsf{u}(z) = \sum_{k\in\mathbb{Z}} u(k) z^k. \tag{2.1.4}$$

And the complex conjugate polynomial follows right away:

$$\mathsf{u}^\star(z) = \sum_{k\in\mathbb{Z}} \overline{u(-k)} z^k = \sum_{k\in\mathbb{Z}} \overline{u(-k)} z^{-k} = \overline{\mathsf{u}(\overline{z}^{-1})}. \tag{2.1.5}$$

From this definition we can directly see that

$$\widehat{u}(\xi) = \mathsf{u}(e^{-i\xi}), \tag{2.1.6}$$

$$\widehat{u}(\xi+\pi) = \mathsf{u}(-e^{-i\xi}), \tag{2.1.7}$$

so that the perfect reconstruction property of tight framelet filter banks can be written as

$$\begin{bmatrix} \tilde{\mathsf{u}}_0(z) & \dots & \tilde{\mathsf{u}}_s(z) \\ \tilde{\mathsf{u}}_0(-z) & \dots & \tilde{\mathsf{u}}_s(-z) \end{bmatrix} \begin{bmatrix} \mathsf{u}_0(z) & \dots & \mathsf{u}_s(z) \\ \mathsf{u}_0(-z) & \dots & \mathsf{u}_s(-z) \end{bmatrix}^\star = I_2,\ z \in \mathbb{C}\backslash\{0\}. \tag{2.1.8}$$

A filter with m sum rules has a Laurent polynomial $\mathsf{u}(z)$ satisifying

$$\mathsf{u}(z) = (1+z)^m \mathsf{u}_0(z) \tag{2.1.9}$$

for some Laurent polynomial u_0; a filter with m vanishing moments has a Laurent polynomial $\mathsf{u}(z)$ with

$$\mathsf{u}(z) = (1-z)^m \mathsf{u}_1(z) \tag{2.1.10}$$

for some Laurent polynomial u_1.

Now for a specific tight framelet filter bank $\{a; b_1, b_2\}$ with 1 low-pass filter and 2 high-pass filters, the PR condition is

$$\begin{bmatrix} \mathsf{a}(z) & \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{a}(-z) & \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix} \begin{bmatrix} \mathsf{a}(z) & \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{a}(-z) & \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix}^\star = I_2,\ z \in \mathbb{C}\backslash\{0\}. \tag{2.1.11}$$

If we are given the low-pass filter a, the condition in (2.1.11) can be equivalently written as

$$\begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix} \begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix}^\star = \mathcal{M}_a(z) \tag{2.1.12}$$

where $\mathcal{M}_a$ is

$$\mathcal{M}_a(z) := \begin{bmatrix} 1 - \mathsf{a}(z)\mathsf{a}^\star(z) & \mathsf{a}(z)\mathsf{a}^\star(-z) \\ \mathsf{a}(-z)\mathsf{a}^\star(z) & \mathsf{a}(-z)\mathsf{a}^\star(-z) \end{bmatrix}. \tag{2.1.13}$$

The form of (2.1.12) is almost what we desire, but $\mathsf{b}_l(-z)$ $(l = 1, 2)$ is completely determined by $\mathsf{b}_l(z)$. The relation between the entries of the same matrix makes the

splitting difficult. It gets much simpler if we assume the entries within the matrix are independent. A convenient way to get rid of the negative sign is through the coset sequence.

For any integer γ we define the coset polynomial $\mathsf{u}^{[\gamma]}$ of a polynomial u as

$$\mathsf{u}^{[\gamma]} = \sum_{k\in\mathbb{Z}} u(\gamma+2k)z^k, \tag{2.1.14}$$

then we have $\mathsf{u}(z) = \mathsf{u}^{[0]}(z^2) + z\mathsf{u}^{[1]}(z^2)$, and more specific,

$$\begin{bmatrix} \mathsf{u}(z) \\ \mathsf{u}(-z) \end{bmatrix} = \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \begin{bmatrix} \mathsf{u}^{[0]}(z^2) \\ \mathsf{u}^{[1]}(z^2) \end{bmatrix}, \tag{2.1.15}$$

$$\begin{bmatrix} \mathsf{u}^{[0]}(z^2) \\ \mathsf{u}^{[1]}(z^2) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ z^{-1} & -z^{-1} \end{bmatrix} \begin{bmatrix} \mathsf{u}(z) \\ \mathsf{u}(-z) \end{bmatrix}. \tag{2.1.16}$$

Then the identity in (2.1.12) can be further written as

$$\begin{bmatrix} \mathsf{b}_1^{[0]}(z) & \mathsf{b}_2^{[0]}(z) \\ \mathsf{b}_1^{[1]}(z) & \mathsf{b}_2^{[1]}(z) \end{bmatrix} \begin{bmatrix} \mathsf{b}_1^{[0]}(z) & \mathsf{b}_2^{[0]}(z) \\ \mathsf{b}_1^{[1]}(z) & \mathsf{b}_2^{[1]}(z) \end{bmatrix}^\star = \mathcal{N}_a(z), \tag{2.1.17}$$

where the right hand size

$$\mathcal{N}_a = \frac{1}{2}I_2 - \begin{bmatrix} \tilde{\mathsf{a}}^{[0]}(z) \\ \tilde{\mathsf{a}}^{[1]}(z) \end{bmatrix} \begin{bmatrix} \mathsf{a}^{[0]}(z)^\star & \mathsf{a}^{[1]}(z)^\star \end{bmatrix} \tag{2.1.18}$$

is uniquely determined by a.

Denoting

$$\begin{bmatrix} \mathsf{b}_1^{[0]}(z) & \mathsf{b}_2^{[0]}(z) \\ \mathsf{b}_1^{[1]}(z) & \mathsf{b}_2^{[1]}(z) \end{bmatrix} = \mathcal{U}(z), \tag{2.1.19}$$

if we are given such a matrix of polynomials, then the filter b_1 and b_2 are completely determined. Because a filter u is uniquely determined if and only if its cosets $b^{[0]}$ and

$b^{[1]}$ are given, and the matrix $\mathcal{U}(z)$ contains all information about the coset sequences. This matrix is called the polyphase matrix. Therefore, with a low-pass filter a, the construction of a tight framelet filter bank is transferred into a matrix splitting problem. That is, finding a polyphase matrix $\mathcal{U}$ such that $\mathcal{U}\mathcal{U}^\star = \mathcal{N}_a$.

2.1.2 Symmetry of Filters

Since we are aiming to find high-pass filters mutually symmetric in the frequency domain, we naturally think of the symmetry in the time domain. As mentioned above, Laurent polynomials provide a convenient definition for the symmetry of filters.

We say that a filter u has (real) symmetry if its corresponding Laurent polynomial $\mathsf{u}(z)$ satisfies

$$\mathsf{S}\mathsf{u}(z) := \frac{\mathsf{u}(z)}{\mathsf{u}(z^{-1})} = \epsilon z^c, \forall z \in \mathbb{C}\backslash\{0\} \text{ with } \epsilon \in \{-1, 1\}, c \in \mathbb{Z}, \tag{2.1.20}$$

and complex symmetry if

$$\mathbb{S}\mathsf{u}(z) := \frac{\mathsf{u}(z)}{\mathsf{u}^\star(z)} = \epsilon z^c \quad \forall z \in \mathbb{C}\backslash\{0\} \text{ with } \epsilon \in \{-1, 1\}, c \in \mathbb{Z}, \tag{2.1.21}$$

and we call ϵz^c the (complex) symmetry type of the Laurent polynomial, or simply of the filter. S or $\mathbb{S}$ is called the symmetry operator.

From (2.1.16), given the symmetry type of a filter u as $\mathsf{S}\mathsf{u} = \epsilon z^c$ the symmetry type of its coset sequences is automatically determined:

(1) If c is even,

$$\mathsf{u}^{[0]}(z) = \epsilon z^{c/2}\mathsf{u}^{[0]}(z^{-1}), \quad \mathsf{u}^{[1]}(z) = \epsilon z^{c/2-1}\mathsf{u}^{[1]}(z^{-1}); \tag{2.1.22}$$

(2) If c is odd,

$$\mathsf{u}^{[1]}(z) = \epsilon z^{(c-1)/2}\mathsf{u}^{[0]}(z^{-1}). \tag{2.1.23}$$

Within a tight framelet filter bank, the symmetry types, in particular, the symmetry center obey certain rules [6].

Lemma 2.1.1. *Let $\{a; b_1, b_2\}$ be a tight framelet filter bank with each filter not identically zero and having symmetry:*

$$\mathsf{Sa}(z) = \epsilon z^c, \quad \mathsf{Sb}_1(z) = \epsilon_1 z^{c_1}, \quad \mathsf{Sb}_2(z) = \epsilon_2 z^{c_2}, \tag{2.1.24}$$

for $\epsilon, \epsilon_1, \epsilon_2 \in \{1, -1\}$ and $c, c_1, c_2 \in \mathbb{Z}$. Then

$$c_1 - c, \quad c_2 - c \in 2\mathbb{Z}. \tag{2.1.25}$$

Proof. As part of the PR condition, we have

$$\mathsf{a}(z)\mathsf{a}^\star(-z) + \mathsf{b}_1(z)\mathsf{b}_1^\star(-z) + \mathsf{b}_2(z)\mathsf{b}_2^\star(-z) = 0. \tag{2.1.26}$$

Consider the symmetry types:

$$\mathsf{Sa}(z)\mathsf{a}^\star(-z) = \mathsf{Sa}(z)\mathsf{Sa}^\star(-z) = z^c(-1)^c z^c z^{-c} = (-1)^c; \tag{2.1.27}$$

$$\mathsf{Sb}_1(z)\mathsf{b}_1^\star(-z) = (-1)^{c_1} \tag{2.1.28}$$

$$\mathsf{Sb}_2(z)\mathsf{b}_2^\star(-z) = (-1)^{c_2} \tag{2.1.29}$$

we have c and c_1, c_2 are all integers so the symmetry types can only be 1 or -1, so 2 of the three must be the same. If another one is not identical to them, then the sum cannot be zero. So

$$(-1)^c = (-1)^{c_1} = (-1)^{c_2}, \tag{2.1.30}$$

which implies

$$c - c_1 \in 2\mathbb{Z}, \quad c - c_2 \in 2\mathbb{Z}. \tag{2.1.31}$$

□

2.2 Construction of Tight Framelet Filter Banks with Mutual Symmetry Between High-pass Filters

Due to the desire of the interesting features of framelet filter banks with symmetry, B. Han and Q. Mo proposed in [8] a necessary and sufficient condition for the existence of such framelet filter banks with 2 generators. Further description of the construction of tight framelet filter banks with or without symmetry in [6].

Based on the construction of symmetric filter banks, we use a simple but smart transform to transform the symmetric ones into mutually symmetric. Of course, not all symmetric filter banks can be transformed into such mutually symmetric ones. The lemma below specifies the transform.

Lemma 2.2.1. *For a low-pass filter a, there exists a tight framelet filter bank $\{a; b_1, b_2\}$ for some $b_1, b_2 \in l_0(\mathbb{Z})$ with mutual symmetry, namely, the corresponding Laurent polynomials satisfying $\mathsf{b}_2(z) = z^k \mathsf{b}_1(z^{-1})$ for some integer k if and only if there exists a tight framelet filter bank $\{a; \widetilde{b_1}, \widetilde{b_2}\}$ such that $\widetilde{b_1}, \widetilde{b_2}$ both have symmetry, with symmetry type respectively $\epsilon_1 z^{c_1}$ and $\epsilon_2 z^{c_2}$, satisfying*

$$\epsilon_1 \epsilon_2 = -1; \ c_1 - c_2 \in 4\mathbb{Z}. \tag{2.2.1}$$

Proof. $\Rightarrow$: Define $\widetilde{b_1} = \frac{1}{\sqrt{2}}(b_1 + b_2)$ and $\widetilde{b_2} = \frac{1}{\sqrt{2}}(b_1 - b_2)$. In terms of Laurent polynomials, the transform can be written as

$$\begin{bmatrix} \widetilde{\mathsf{b}_1}(z) \\ \widetilde{\mathsf{b}_2}(z) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathsf{b}_1(z) \\ \mathsf{b}_2(z) \end{bmatrix}. \tag{2.2.2}$$

Therefore we have

$$\begin{bmatrix} \widetilde{\mathsf{b}_1}(z) & \widetilde{\mathsf{b}_2}(z) \\ \widetilde{\mathsf{b}_1}(-z) & \widetilde{\mathsf{b}_2}(-z) \end{bmatrix} = \begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.2.3}$$

and

$$\begin{aligned} & \begin{bmatrix} \widetilde{\mathsf{b}_1}(z) & \widetilde{\mathsf{b}_2}(z) \\ \widetilde{\mathsf{b}_1}(-z) & \widetilde{\mathsf{b}_2}(-z) \end{bmatrix} \begin{bmatrix} \widetilde{\mathsf{b}_1}(z) & \widetilde{\mathsf{b}_2}(z) \\ \widetilde{\mathsf{b}_1}(-z) & \widetilde{\mathsf{b}_2}(-z) \end{bmatrix}^\star \\ = & \begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^2 \begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix}^\star \qquad (2.2.4) \\ = & \begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix} \begin{bmatrix} \mathsf{b}_1(z) & \mathsf{b}_2(z) \\ \mathsf{b}_1(-z) & \mathsf{b}_2(-z) \end{bmatrix}^\star = \mathcal{M}_a(z). \qquad (2.2.5) \end{aligned}$$

So that $\{a; \widetilde{b_1}, \widetilde{b_2}\}$ is still a tight framelet filter bank and we can calculate

$$\epsilon_1 z^{c_1} = \mathsf{S}\widetilde{\mathsf{b}_1}(z) = \frac{\tilde{\mathsf{b}}_1(z)}{\tilde{\mathsf{b}}_1(z^{-1})} = \frac{\mathsf{b}_1(z) + \mathsf{b}_2(z)}{\mathsf{b}_1(z^{-1}) + \mathsf{b}_2(z^{-1})} = \frac{\mathsf{b}_1(z) + z^k \mathsf{b}_1(z^{-1})}{\mathsf{b}_1(z^{-1}) + z^{-k}\mathsf{b}_1(z)} = z^k \tag{2.2.6}$$

$$\epsilon_2 z^{c_2} = \mathsf{S}\widetilde{\mathsf{b}_2}(z) = \frac{\tilde{\mathsf{b}}_2(z)}{\tilde{\mathsf{b}}_2(z^{-1})} = \frac{\mathsf{b}_1(z) - \mathsf{b}_2(z)}{\mathsf{b}_1(z^{-1}) - \mathsf{b}_2(z^{-1})} = \frac{\mathsf{b}_1(z) - z^k \mathsf{b}_1(z^{-1})}{\mathsf{b}_1(z^{-1}) - z^{-k}\mathsf{b}_1(z)} = -z^k \tag{2.2.7}$$

hence $\epsilon_1 = -1 = -\epsilon_2$ and $c_1 - c_2 = k - k = 0 \in 4\mathbb{Z}$.

$\Leftarrow$: Similarly if we have $\tilde{b_1}$ and $\tilde{b_2}$ with symmetry type respectively $\epsilon_1 z_1^c$ and $\epsilon_2 z_2^c$, satisfying

$$\epsilon_1 = -\epsilon_2;\ c_1 - c_2 \in 4\mathbb{Z},$$

first we shift $\widetilde{b}_2(z)$ by $2k$ and get $\widetilde{\mathsf{b}}_2' = z^{2k}\widetilde{\mathsf{b}}_2$ for some integer k to make $\widetilde{b}_1$ and $\widetilde{b}_2'$ have the same symmetry center. Then $\{a; \widetilde{b}_1, \widetilde{b}_2'\}$ remains as a tight framelet filter bank.

Then we define $b_1 = \frac{1}{\sqrt{2}}(\widetilde{b}_1 + \widetilde{b}_2')$ and $b_2 = \frac{1}{\sqrt{2}}(\widetilde{b}_1 - \widetilde{b}_2')$, and it's easy to see that $\{a; b_1, b_2\}$ satisfies

$$\mathsf{b}_2(z) = z^{c_1}\mathsf{b}_1(z^{-1}).$$

□

Now the construction of a tight framelet filter bank $\{a; b_1, b_2\}$ with b_1, b_2 having mutual symmetry can be done by constructing a tight framelet filter bank $\{a; b_1, b_2\}$ with symmetry. In particular, we need the symmetry type of two high-pass filters having different signs and their symmetry center difference can be divided by 4. And we can thereon derive the main theorem for the existence. For the sake of convenience we define the odd indicator function

$$\mathrm{odd}(k) := \begin{cases} 1, & \text{when } k \text{ is odd;} \\ 0, & \text{otherwise;} \end{cases} \tag{2.2.8}$$

for any integer k.

Theorem 2.2.2. *Let $a \in l_0(\mathbb{Z})$ be a filter with real coefficients and have real symmetry such that $\mathsf{Sa} = \epsilon z^c$. Let n_b be a nonnegative integer satisfying $0 \leqslant n_b \leqslant min(sr(a), \frac{1}{2}vm(-\mathsf{a}(z)\mathsf{a}^\star(z))$. Then there exists a tight framelet filter bank $\{a; b_1, b_2\}$ for some $b_1, b_2 \in l_0(\mathbb{Z})$ such that the corresponding Laurent polynomials satisfy*

$$\mathsf{b}_2(z) = z^k \mathsf{b}_1(1/z) \tag{2.2.9}$$

if and only if
(i) $\mathcal{N}_{a|n_b}(z) \geq 0$ for any $z \in \mathbb{T} := \{\xi : |\xi| = 1\}$, where $\mathcal{N}_{a|n_b}$ is defined as

$$\mathcal{N}_{a|n_b}(z) = \frac{1}{2} \begin{bmatrix} \mathsf{A}^{[0]}(z) + \mathsf{B}^{[0]}(z) & z(\mathsf{A}^{[1]}(z) + \mathsf{B}^{[1]}(z)) \\ \mathsf{A}^{[1]}(z) - \mathsf{B}^{[1]}(z) & \mathsf{A}^{[0]}(z) - \mathsf{B}^{[0]}(z) \end{bmatrix} \tag{2.2.10}$$

with

$$\mathsf{A}(z) := \frac{\Theta(z) - \Theta(z^2)\mathsf{a}(z)\mathsf{a}^\star(z)}{(1-z)^{n_b}(1-z^{-1})^{n_b}}, \qquad \mathsf{B}(z) := -\Theta(z^2)\frac{\mathsf{a}(-z)\mathsf{a}^\star(z)}{(1-z)^{n_b}(1+z^{-1})^{n_b}}; \tag{2.2.11}$$

(ii) there exists a Laurent polynomial $\mathsf{d}(z)$ *with symmetry type* $\epsilon_d z^{c_d}$ *such that*

$$\mathsf{d}(z)\mathsf{d}^\star(z) = det(\mathcal{N}_{a|n_b}(z)), \tag{2.2.12}$$

and

$$\epsilon_d = (-1)^{\mathrm{odd}(c+n_b)+1}; \tag{2.2.13}$$

$$\mathrm{odd}(c_d) = 1 - \mathrm{odd}(c + n_b). \tag{2.2.14}$$

Proof. $\Rightarrow$: If there exist b_1, b_2 satisfying (2.2.9), with vanishing moment n_b, i.e., we can write them as

$$\mathsf{b}_1(z) = (1 - z^{-1})^{n_b} \mathring{\mathsf{b}}_1(z), \tag{2.2.15}$$

$$\mathsf{b}_2(z) = (1 - z^{-1})^{n_b} \mathring{\mathsf{b}}_2(z) \tag{2.2.16}$$

for some Laurent polynomials $\mathring{\mathsf{b}}_2$ and $\mathring{\mathsf{b}}_1$. By Lemma 2.2.1, there exist $\tilde{\mathsf{b}}_1$ and $\tilde{\mathsf{b}}_2$ with

$$\epsilon_1 z^{c_1} = \mathsf{S}\tilde{\mathsf{b}}_1(z) = z^k, \tag{2.2.17}$$

$$\epsilon_2 z^{c_2} = \mathsf{S}\tilde{\mathsf{b}}_2(z) = -z^k \tag{2.2.18}$$

such that $\{a; \tilde{b}_1, \tilde{b}_2\}$ is a tight framelet filterbank. By Lemma 2.1.1, the symmetry centers satisfy $c - k \in 2\mathbb{Z}$. The transform keeps the polynomial factors, so $\tilde{b}_1, \tilde{b}_2$ have also vanishing moments at least n_b, hence can be written as

$$\tilde{\mathsf{b}}_1(z) = (1 - z^{-1})^{n_b} \mathring{\tilde{\mathsf{b}}}_1, \tag{2.2.19}$$

$$\tilde{\mathsf{b}}_2(z) = (1 - z^{-1})^{n_b} \mathring{\tilde{\mathsf{b}}}_2. \tag{2.2.20}$$

Then we have

$$\mathsf{S}\mathring{\tilde{\mathsf{b}}}_1(z) = z^{k-n_b},$$

$$\mathring{\tilde{\mathsf{Sb}}}_2(z) = -z^{k-n_b}.$$

Define

$$\mathcal{U} = \begin{bmatrix} \mathring{\tilde{b}}_1^{[0]} & \mathring{\tilde{b}}_2^{[0]} \\ \mathring{\tilde{b}}_1^{[1]} & \mathring{\tilde{b}}_2^{[1]} \end{bmatrix},$$

then $\mathcal{U}$ satisfies $\mathcal{U}\mathcal{U}^\star = \mathcal{N}_{a|n_b}$.

Then define

$$\mathcal{N} = \begin{cases} \mathcal{N}_{a|n_b} & \text{if } c+n_b \text{ is even;} \\ P_{k_a}\mathcal{N}_{a|n_b}P_{k_a}^\star & \text{if } c+n_b \text{ is odd,} \end{cases} \tag{2.2.21}$$

where

$$P = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & z^{k_a} \\ 1 & -z^{k_a} \end{bmatrix}, \tag{2.2.22}$$

with k_a=0. It is easy to see that $\det(\mathcal{N}(z)) = \det(\mathcal{N}_{a|n_b}(z)) = \mathsf{d}(z)\mathsf{d}^\star(z)$.

1) When $c+n_b$ is even, by Lemma 2.1.1, $\mathcal{U}\mathcal{U}^\star = \mathcal{N}$, $k-n_b = k-c+(c-n_b)$ is also even. Then by (2.1.22),

$$\mathring{\tilde{\mathsf{Sb}}}_1^{[0]}(z) = z^{\frac{k-n_b}{2}},$$

$$\mathring{\tilde{\mathsf{Sb}}}_2^{[1]}(z) = -z^{\frac{k-n_b}{2}-1}.$$

Hence

$$\mathsf{Sd}(z) = \mathring{\tilde{\mathsf{Sb}}}_1^{[0]}(z)\mathring{\tilde{\mathsf{Sb}}}_2^{[1]}(z) = -z^{k-n_b-1} = (-1)^{\mathrm{odd}(c+n_b+1)}z^{c+n_b+2j+1}. \tag{2.2.23}$$

for some integer j.

2) When $c+n_b$ is odd, set

$$\mathcal{U}' := P_{k_a}\mathcal{U},$$

then

$$\mathcal{U}'\mathcal{U}'^\star = \mathcal{N}, \text{ and } \det(\mathcal{U}) = \det(\mathcal{U}'),$$

and we can verify that all entries in $\mathcal{U}'$ has symmetry. In particular,

$\mathsf{S}\mathcal{U}'_{11}(z) = (-1)^{n_b} z^{\frac{k+n_b-1}{2}}, \mathsf{S}\mathcal{U}'_{22}(z) = -(-1)^{n_b+1} z^{\frac{k+n_b-1}{2}}$, therefore

$$\mathsf{Sd}(z) = \mathsf{S}(z)(\det(\mathcal{N}(z))) = \mathsf{S}\mathcal{U}'_{11}(z)\mathsf{S}\mathcal{U}'_{22}(z) = z^{k+n_b-1} = (-1)^{\mathrm{odd}(c+n_b+1)} z^{2j+c+n_b+1} \tag{2.2.24}$$

for some integer j.

$\Leftarrow$: We just claim that if $\mathsf{d}(z)$ satisfies the condition we stated above, then the filter bank obtained through the algorithm in Chapter 4 (also see [6]) is just the type in Lemma 2.2.1. That is, denoting the symmetry types as $\mathsf{Sb}_1(z) = \epsilon_1 z^{c_1}$ and $\mathsf{Sb}_2(z) = \epsilon_2 z^{c_2}$, then $\epsilon_1\epsilon_2 = -1$, $c_1 - c_2 \in 4\mathbb{Z}$.

Let $\mathcal{N}$ be defined as in (2.2.21). By the algorithm, we can construct

$$\begin{bmatrix} \mathring{\mathsf{b}}_1^{[0]} & \mathring{\mathsf{b}}_2^{[0]} \\ \mathring{\mathsf{b}}_1^{[1]} & \mathring{\mathsf{b}}_2^{[1]} \end{bmatrix} = \mathcal{U} = \begin{bmatrix} \mathsf{q}(z) & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathring{\mathcal{U}}_{11} & \mathring{\mathcal{U}}_{12} \\ \mathring{\mathcal{U}}_{21} & \mathring{\mathcal{U}}_{22} \end{bmatrix},$$

where q is such that $\mathsf{q}\mathsf{q}^\star = \gcd(\mathcal{N}_{ij})$.

1) When $c + n_b$ is even, we have $\mathsf{Sd}(z) = -z^{2k+1}$ for some integer k.
The symmetry types of each entry in $\mathcal{N} = \mathcal{N}_{a|n_b}$ are $\mathsf{S}\mathcal{N}_{11}(z) = \mathsf{S}\mathcal{N}_{22}(z) = 1$, $\mathsf{S}\mathcal{N}_{21}(z) = z^{-1}$.

$\mathsf{b}_1, \mathsf{b}_2$ are defined as

$$\mathsf{b}_i(z) = (1 - z^{-1})^{n_b} \mathring{\mathsf{b}}_i(z), \quad i = 1, 2, \tag{2.2.25}$$

therefore $\mathsf{S}\mathring{\mathsf{b}}_i(z) = \frac{\mathsf{Sb}_i(z)}{(-z)^{n_b}} = (-1)^{n_b}\epsilon_i z^{c_i - n_b}$. Since $c + n_b$ is even, $c_i - n_b (i = 1, 2)$ is also even. By (2.1.23),

$$\mathsf{S}\mathring{\mathsf{b}}_i^{[0]} = (-1)^{n_b}\epsilon_i z^{\frac{c_i - n_b}{2}},$$

$$\frac{\mathsf{Sb}_1}{\mathsf{Sb}_2} = \frac{\mathsf{S}\mathring{\mathsf{b}}_1}{\mathsf{S}\mathring{\mathsf{b}}_2} = \frac{\mathsf{S}\mathring{\mathsf{b}}_1^{[0]}}{\mathsf{S}\mathring{\mathsf{b}}_2^{[0]}} = \frac{\mathsf{S}\mathcal{U}_{11}}{\mathsf{S}\mathcal{U}_{22}} = \frac{\mathsf{Sq}\mathsf{S}\mathring{\mathcal{U}}_{11}}{\mathsf{Sq}\mathsf{S}\mathring{\mathcal{U}}_{12}} = \frac{\mathsf{S}\mathring{\mathcal{U}}_{11}}{\mathsf{S}\mathring{\mathcal{U}}_{12}}.$$

In the algorithm we have the symmetry types

$$\mathring{\mathsf{S}\mathcal{U}}_{11}(z) = \epsilon\epsilon_{\mathring{\mathcal{N}}_{21}} z^{n-c_{\text{odd}}}, \tag{2.2.26}$$

$$\mathring{\mathsf{S}\mathcal{U}}_{11}(z) = \epsilon\mathring{\epsilon} z^{n},$$

hence

$$\frac{\mathring{\mathsf{S}\mathcal{U}}_{11}(z)}{\mathring{\mathsf{S}\mathcal{U}}_{12}(z)} = \frac{\epsilon_{\mathring{\mathcal{N}}_{21}}}{\mathring{\epsilon}} z^{c_{\text{odd}}}.$$

While

$$\mathring{\epsilon} z^{\mathring{c}} = \tilde{\mathsf{Sd}}(z) = \frac{\mathsf{Sd}(z)}{\mathsf{Sq}(z)} = \frac{-z^{2j+1}}{\mathsf{Sq}(z)}, \tag{2.2.27}$$

$$\epsilon_{\mathring{\mathcal{N}}_{21}} z^{c_{\mathring{\mathcal{N}}_{21}}} = \mathring{\mathsf{S}\mathcal{N}}_{21}(z) = \frac{\mathsf{S}\mathcal{N}_{21}(z)}{\mathsf{Sq}(z)} = \frac{z^{-1}}{\mathsf{Sq}(z)}, \tag{2.2.28}$$

$$\tag{2.2.29}$$

so we get

$$\epsilon_{\mathring{\mathcal{N}}_{21}}/\mathring{\epsilon} = -1, \tag{2.2.30}$$

$$c_{\text{odd}} = \text{odd}(2j + 1 + 1 + 2c_q) = 0, \tag{2.2.31}$$

where c_q is the symmetry center of q. Finally we have

$$\frac{\mathsf{Sb}_1}{\mathsf{Sb}_2} = -1.$$

2) When $c + n_b$ is odd, we have $\mathsf{Sd}(z) = z^{2k}$ for some integer k. The matrix is defined as $\mathcal{N} = P_{k_a} \mathcal{N}_{a|n_b} P_{k_a}^{\star}$.

$$\mathsf{b}_1(z) := \frac{(1-z^{-1})^{n_b}}{\sqrt{2}}[(1+z^{1-2k_a})\mathcal{U}_{11}(z^2) + (1-z^{1-2k_a})\mathcal{U}_{21}(z^2)], \tag{2.2.32}$$

$$\mathsf{b}_2(z) := \frac{(1-z^{-1})^{n_b}}{\sqrt{2}}[(1+z^{1-2k_a})\mathcal{U}_{12}(z^2) + (1-z^{1-2k_a})\mathcal{U}_{22}(z^2)]. \tag{2.2.33}$$

So we can calculate

$$\frac{\mathsf{Sb}_1(z)}{\mathsf{Sb}_2(z)} = \frac{\mathsf{S}(1+z^{1-2k_a})}{\mathsf{S}(1-z^{1-2k_a})}\frac{\mathsf{S}\mathcal{U}_{11}(z^2)}{\mathsf{S}\mathcal{U}_{22}(z^2)} = (-1)^{1-2k_a}\frac{\mathsf{S}\mathcal{U}_{11}(z^2)}{\mathsf{S}\mathcal{U}_{22}(z^2)} = -\frac{\mathsf{S}\mathcal{U}_{11}(z^2)}{\mathsf{S}\mathcal{U}_{22}(z^2)} \tag{2.2.34}$$

Denoting $\mathsf{S}\mathcal{U}_{11}(z) = \epsilon_{11}z^{c_{11}}, \mathsf{S}\mathcal{U}_{22}(z) = \epsilon_{22}z^{c_{22}}$, we have $\mathsf{Sd}(z) = \epsilon_{11}\epsilon_{22}z^{c_{11}+c_{22}} = z^{2k}$ for some integer k.

Then

$$\frac{\mathsf{Sb}_1(z)}{\mathsf{Sb}_2(z)} = -\frac{\mathsf{S}\mathcal{U}_{11}(z^2)}{\mathsf{S}\mathcal{U}_{22}(z^2)} = \frac{\epsilon_{11}z^{2c_{11}}}{\epsilon_{22}z^{2c_{22}}} = -\epsilon_{11}\epsilon_{22}z^{2(c_{11}+c_{22})} = -z^{4k}$$

for some integer k.

To conclude, both cases will construct a tight framelet filterbank with

$$\frac{\mathsf{Sb}_1(z)}{\mathsf{Sb}_2(z)} = -z^{4k},$$

and by Lemma 2.2.1, this is equivalent to a tight framelet filterbank with 2 mutually symmetric high-pass filters. □

2.3 Directional Tight Framelet Filter Bank

We now go back to the Lemma 2.2.1. It can be further generalized. Actually, in the transform

$$\begin{bmatrix} \widetilde{b}_1(z) \\ \widetilde{b}_2(z) \end{bmatrix} = A \begin{bmatrix} b_1(z) \\ b_2(z) \end{bmatrix}, \tag{2.3.1}$$

any 2×2 unitary matrix A with make the image $\{\widetilde{b}_1, \widetilde{b}_2\}$ form a tight framelet filter bank with the original mask a .

In general, a series of complex unitary matrices are in the form of

$$A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{i\theta} \\ 1 & e^{i\theta} \end{bmatrix} \tag{2.3.2}$$

up to a scaling factor e^{iu}, where θ, u are both real parameters. But this factor does not influence the magnitude in the frequency domain. Therefore, we can get the optimized fiter bank by optimizing on the parameter $\theta \in [-\pi, \pi]$.

There are several different strategies to evaluate the directionality. One is to measure the energy concentration in the frequency domain, to be more specific, we measure the high-pass filters b_1 and b_2 using

$$E_i = \int_0^\pi |\hat{b}_i(\xi)|^2 d\xi, \quad i = 1, 2. \tag{2.3.3}$$

When E_1 is large, we conclude that the filter $\widehat{b}_1$ is concentrated on $[0, \pi]$ thus makes a good separation in the frequency domain. Actually, in this sense, we can prove that when the parameter θ in the unitary matrix A is equal to $\pm\frac{\pi}{2}$, E_1 can reach its extreme values (maximum or minimum).

Proof. Denoting the original real-valued filter as $\mathring{b}_1$ and $\mathring{b}_2$, then the new filters are

$$b_1 = \frac{1}{\sqrt{2}}(\mathring{b}_1 + e^{i\theta}\mathring{b}_2), b_2 = \frac{1}{\sqrt{2}}(\mathring{b}_1 - e^{i\theta}\mathring{b}_2).$$

The energy can be therefore calculated:

$$\begin{aligned} E_1 &= \int_0^\pi (\widehat{b}_1(\xi)\overline{\widehat{b}_1}(\xi))d\xi = \int_0^\pi (\widehat{\mathring{b}}_1(\xi) + e^{i\theta}\widehat{\mathring{b}}_2(\xi))\overline{(\widehat{\mathring{b}}_1(\xi) + e^{i\theta}\widehat{\mathring{b}}_2(\xi))}d\xi \\ &= \int_0^\pi (|\widehat{\mathring{b}}_1(\xi)|^2 + |\widehat{\mathring{b}}_2(\xi)|^2 + e^{i\theta}\overline{\widehat{\mathring{b}}_1(\xi)}\widehat{\mathring{b}}_2(\xi) + \overline{e^{i\theta}\widehat{\mathring{b}}_1(\xi)\widehat{\mathring{b}}_2(\xi)})d\xi. \end{aligned}$$

$$\frac{\partial E_1}{\partial \theta} = ie^{i\theta}\int_0^\pi (\overline{\hat{\mathring{b}}_1(\xi)}\hat{\mathring{b}}_2(\xi))d\xi - ie^{-i\theta}\int_0^\pi (\overline{\hat{\mathring{b}}_2(\xi)}\hat{\mathring{b}}_1(\xi))d\xi.$$

Since the original $\mathring{b}_1$ and $\mathring{b}_2$ are both real-valued filters, so

$$\overline{\hat{\mathring{b}}_i(\xi)} = \sum_k \overline{\mathring{b}_i(k)e^{-ik\xi}} = \sum_k \mathring{b}_i(k)e^{ik\xi} = \hat{\mathring{b}}_i(-\xi), \quad i = 1, 2. \tag{2.3.4}$$

Their symmetry types are respectively $\epsilon_1 z^{c_1}$ and $\epsilon_2 z^{c_2}$ with $\epsilon_1 = -\epsilon_2$, $c_1 = c_2$ so we have

$$\widehat{b}_i(-\xi) = \epsilon_i e^{ic_i\xi}\widehat{b}_i(\xi), \ i = 1, 2. \tag{2.3.5}$$

Hence

$$\int_0^\pi \overline{\hat{\mathring{b}}_1(\xi)}\hat{\mathring{b}}_2(\xi)d\xi = \int_0^\pi \hat{\mathring{b}}_1(-\xi)\hat{\mathring{b}}_2(\xi)d\xi \tag{2.3.6}$$

$$= \int_0^\pi \epsilon_1 e^{ic_1\xi}\hat{\mathring{b}}_1(\xi)\hat{\mathring{b}}_2(\xi)d\xi = -\int_0^\pi \epsilon_2 e^{ic_2\xi}\hat{\mathring{b}}_1(\xi)\hat{\mathring{b}}_2(\xi)d\xi \tag{2.3.7}$$

$$= -\int_0^\pi \overline{\hat{\mathring{b}}_2(\xi)}\hat{\mathring{b}}_1(\xi)d\xi, \tag{2.3.8}$$

and then

$$\frac{\partial E_1}{\partial \theta} = (ie^{i\theta} + ie^{-i\theta})\int_0^\pi \overline{\hat{\mathring{b}}_1(\xi)}\hat{\mathring{b}}_2(\xi)d\xi. \tag{2.3.9}$$

It equals 0 when $\theta = \pm\pi/2$. □

But only considering the concentration in the frequency domain is not the best approximation in the time domain. In contrast, maximizing

$$F_i = \int_0^\pi |\hat{b}_i(\xi)\hat{a}(\xi/2)|^2 d\xi, i = 1, 2 \tag{2.3.10}$$

may be a better choice, since it is more close to the wavelet function ψ^i in the time domain whose Fourier transform satisfies

$$\widehat{\psi^i}(t) = \hat{b}_i(t/2)\prod_{j=1}^{\infty}\hat{a}(2^{-1-j}t).$$

And this will be our topic in the next chapter, the wavelet functions and the discrete affine system (DAS). Through tensor product we shall talk about the 2-D DAS.

Chapter 3

Wavelet Functions and 2-D Discrete Affine Systems

Following the topic that we proposed in Chapter 2, when optimizing the filter bank, we need to evaluate and visualize its directionality in a more direct way besides the directions in the frequency domain. In image processing practice, after the 1-level Discrete Framelet Transform, we usually continue the decomposition to the low-pass sub-band and the transform is performed over and over again on a signal. The signal is then decomposed into numerous bands for processing. The directions actually lie in these sub-bands. Nonetheless, we have not investigated the behavior of the multi-level DFrT, nor do we know the information contained in each sub-band. It is actually an asymptotic process, closely related to the wavelet function in continuous setting. More importantly, for an input image, the first 1 or 2 level DFrT actually can hardly catch the directional features. The evaluation of good directionality of a filter bank therefore locates at the higher levels. In this chapter, we shall investigate the relation between a framelet filter bank and its corresponding wavelet functions. To achieve this, we start from the continuous wavelets again, and use the concept of discrete affine systems to explain.

3.1 Discrete Affine System in 1-D

In DFrT, the signal is decomposed through a filter bank and a downsampling process. In the usual dyadic decomposition, each sub-band contains half information of the

input. If we continue the decomposition (usually just for the low-pass band), the subsequence we get is much related to the Discrete Affine System (DAS). In this section we first investigate the DAS in 1-dimension, then generalize it to the 2-D setting in the next section. The key is that, after multi-level DFrT, the sequence in each sub-band can be actually obtained by a single transition operator. So that we can just use more filters to illustrate the information contained in each band.

3.1.1 Multilevel Discrete Framelet Transform

We have introduced in Chapter 1 the 1-level discrete framelet transform with respect to a dual framelet filter bank $(\{\tilde{a};\tilde{b}_1,...,\tilde{b}_s\},\{a;b_1,...,b_s\})$:

$$\mathcal{W}v=(\omega_0,\omega_1,...,\omega_s) = \frac{\sqrt{2}}{2}(\mathcal{T}_{\tilde{a}}v,\mathcal{T}_{\tilde{b}_1}v,...,\mathcal{T}_{\tilde{b}_s}v), \tag{3.1.1}$$

$$\mathcal{V}(\omega_0,\omega_1,...,\omega_s) = \frac{\sqrt{2}}{2}\sum_{l=0}^{s}\mathcal{S}_{u_l}\omega_l. \tag{3.1.2}$$

If we go further to decompose ω_0, we can get the multi-level framelet decomposition. For example, the 2-level DFrT is illustrated in Fig 9 (adapted form [6]). Using the same dual framelet filter bank $(\{\tilde{a};\tilde{b}_1,...,\tilde{b}_s\},\{a;b_1,...,b_s\})$, a *J-level discrete framelet decomposition* is defined as

$$v_{j-1}:=\frac{\sqrt{2}}{2}\mathcal{T}_a v_j, \tag{3.1.3}$$

$$\omega_{j-1;l}:=\frac{\sqrt{2}}{2}\mathcal{T}_{b_l}v_j,\ l=1,...s;j=J,...,1. \tag{3.1.4}$$

The corresponding *J-level discrete framelet reconstruction* is given by

$$\mathring{v}_j:=\frac{\sqrt{2}}{2}\mathcal{S}_{\tilde{a}}\mathring{v}_{j-1}+\frac{\sqrt{2}}{2}\sum_{l=1}^{s}\mathcal{S}_{\tilde{b}_l}\mathring{\omega},\ j=1,...,J. \tag{3.1.5}$$

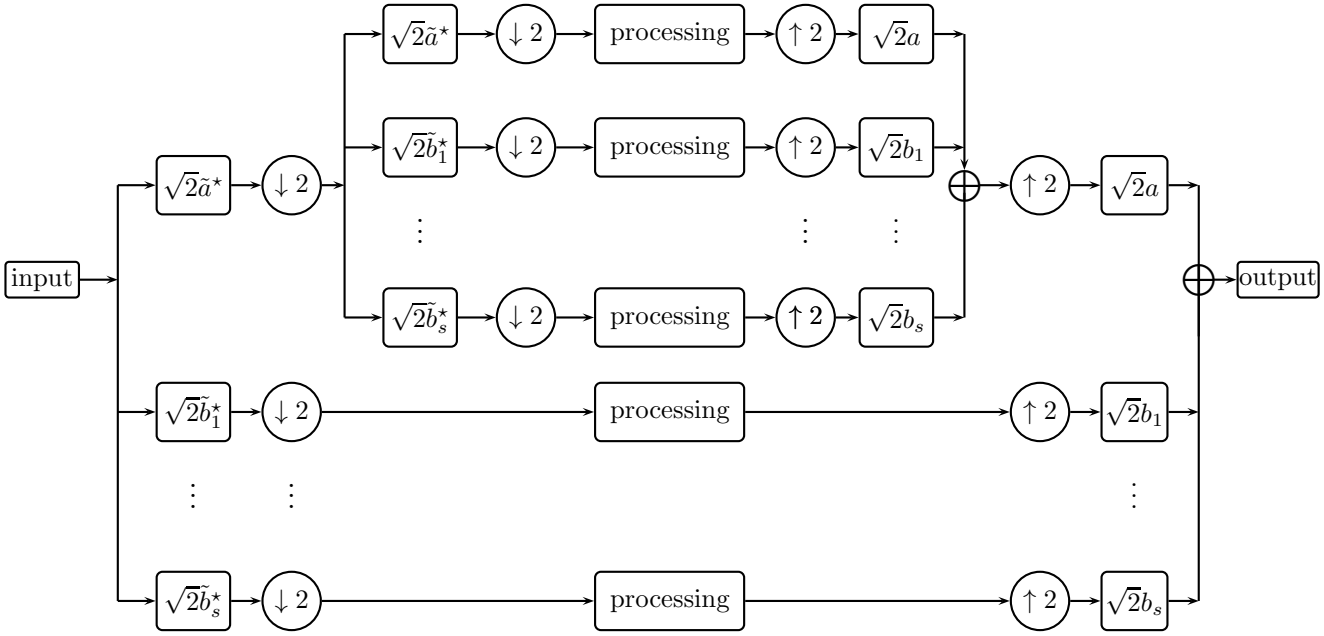


Figure 9: Diagram for 2-level DFrT

For convenience, we write the *J-level analysis operator* and *J-level synthesis operator* as

$$\mathcal{W}_J v_j := (\omega_{J-1;1}, ..., \omega_{J-1;s}, ..., \omega_{0;1}, ..., \omega_{0;s}, v_0), \tag{3.1.6}$$

$$\mathcal{V}_J(\omega_{J-1;1}, ..., \omega_{J-1;s}, ..., \omega_{0;1}, ..., \omega_{0;s}, v_0) = v_j. \tag{3.1.7}$$

To study the behavior of multi-level DFrT, let us first take a look at 2-level DFrT. For a signal $v \in l(\mathbb{Z})$, the subdivision operator associated with the a filter u will map it to

$$v' = [\mathcal{S}_u v](n) = 2\sum_k v(k)u(n-2k) = 2u * (v \uparrow 2), \tag{3.1.8}$$

which is just another sequence in $l(\mathbb{Z})$, and in frequency domain it satisfies

$$\widehat{v'} = \widehat{\mathcal{S}_u}(\xi) = 2\widehat{u}(\xi)\widehat{v}(2\xi). \tag{3.1.9}$$

If the subdivision is done twice, the signal is mapped to

$$v'' = \mathcal{S}_u[\mathcal{S}_u v_0] \tag{3.1.10}$$

which is

$$\widehat{v''} = 4\widehat{u}(\xi)\widehat{u}(2\xi)\widehat{v}(4\xi) \tag{3.1.11}$$

in frequency domain. From (3.1.11), if we define another filter $u_1 := u * (u \uparrow 2)$ so that $\widehat{u_1}(\xi) = \widehat{u}(\xi)\widehat{u}(2\xi)$, in the frequency and time domain we can alternatively write

$$\widehat{v''}(\xi) = 4\widehat{u_1}(\xi)\widehat{v}(4\xi), \tag{3.1.12}$$

$$v''(n) = 4u_1 * (v \uparrow 4). \tag{3.1.13}$$

This motivates us to generalize the dyadic subdivision operator to a more general one $\mathcal{S}_{u,d} : l(\mathbb{Z}) \to l(\mathbb{Z})$:

$$[\mathcal{S}_{u,d} v](n) := |d| \sum_k v(k) u(n - dk) = |d| u * (v \uparrow d), \quad d \in \mathbb{Z}. \tag{3.1.14}$$

Correspondingly the transition operator is defined as

$$[\mathcal{T}_{u,d} v](n) := |d| \sum_k v(k) \overline{u(k - dn)} = |d| (u^\star * v) \downarrow d. \tag{3.1.15}$$

From this definition we can write

$$v'' = \mathcal{S}_{u_1,4} v = \mathcal{S}_{u*(u\uparrow 2),4} v, \tag{3.1.16}$$

so that the iterative subdivision or transition operations will still be subdivision or transition, but generalized ones. In particular, a 2-level dyadic DFrT with respect to tight framelet filter bank $\{a; b_1, ..., b_s\}$ is

Analysis operator:

$$\mathcal{W}_2 v_2 = (\omega_{1,1},,...,\omega_{1,s},\omega_{0,1},...,\omega_{0,s},v_0) \tag{3.1.17}$$

where

$$\omega_{1,l} = \frac{\sqrt{2}}{2}\mathcal{T}_{b_l,2}v_2, \qquad \text{for } l=1,...,s; \tag{3.1.18}$$

$$\omega_{0,l} = \left(\frac{\sqrt{2}}{2}\right)^2 \mathcal{T}_{b_l*(a\uparrow 2),4}v_2, \qquad \text{for } l=1,...,s; \tag{3.1.19}$$

$$v_0 = \left(\frac{\sqrt{2}}{2}\right)^2 \mathcal{T}_{a*(a\uparrow 2),4}v_2; \tag{3.1.20}$$

Synthesis operator:

$$\begin{aligned}
&\mathcal{V}_2(\omega_{1,1},,...,\omega_{1,s},\omega_{0,1},...,\omega_{0,s},v_0)\\
=\ & \frac{\sqrt{2}}{2}\sum_{l=1}^{s}\mathcal{S}_{b_l,2}\omega_{1,l} + \frac{1}{2}(\sum_{l=1}^{s}\mathcal{S}_{b_l*(a\uparrow 2),4}\omega_{0,l} + \mathcal{S}_{a*(a\uparrow 2),4}v_0) \qquad (3.1.21)\\
=\ & \frac{\sqrt{2}}{2}\sum_{l=1}^{s}(2\sum_k \omega_{1,l}(k)b_l(\cdot-2k)) \qquad (3.1.22)\\
&+\frac{1}{2}\sum_{l=1}^{s}(4\sum_k \omega_{0,l}(k)(b_l*(a\uparrow 2))(\cdot-4k))\\
&+2\sum_k v_0(k)(a*(a\uparrow 2))(\cdot-4k).
\end{aligned}$$

The sequences $\omega_{j,l}$ are generally called high-pass sub-bands, and v_0 is the low-pass sub-band. If the filter bank we employ satisfies PR condition, then the reconstructed $\mathcal{V}_2(\omega_{1,1},,...,\omega_{1,s},\omega_{0,1},...,\omega_{0,s},v_0)$ is exactly the original input, and we can express the input as linear combinations of $(a*(a\uparrow 2))(\cdot-4k)$, $(b_l*(a\uparrow 2))(\cdot-4k)$ and $b_l(\cdot-2k)$, which inspires us to define an affine system as (1.5.1), but in a discrete setting.

3.1.2 Discrete Affine System

From (3.1.19) and (3.1.20) we can see that after J-level framelet decomposition, the sequence in each high-pass subband is just the transition operator

$$\left(\frac{\sqrt{2}}{2}\right)^{J-j} \mathcal{T}_{b_l*(a\uparrow 2)*\ldots*(a\uparrow 2^j),2^{J-j}},, 0 < j \leq J \tag{3.1.23}$$

acting on the input v_J. So they are closely related to the filters $a*(a\uparrow 2)*...*(b_l\uparrow 2^j)$ and $a*(a\uparrow 2)*...*(a\uparrow 2^J)$.

Now we define new filters

$$a_j := a*(a\uparrow 2)*\ldots*(a\uparrow 2^{j-1}); \tag{3.1.24}$$

$$a_{[j;k]} := 2^{j/2} a_j(\cdot - 2^j k); \tag{3.1.25}$$

$$b_{l,[j;k]} := 2^{j/2}[a_j*(b_l\uparrow 2^{j-1})](\cdot - 2^j k). \tag{3.1.26}$$

Then the framelet coefficients in any band are just linear combinations of the sequence after transitions $\mathcal{T}_{a_{[j;k]},2^j}$ and $\mathcal{T}_{b_{l,[j;k]},2^j}$. It is also easy to see that, different indices j represent different scales of the input signal. The DFrT gives a multi-scale structure of a signal. Given a tight framelet filter banks $\{a; b_1, ..., b_s\}$, we define the J-level DAS as follows:

$$\mathsf{DAS}_J(\{a; b_1, ..., b_s\}) := \{a_{[J;k]}, k \in \mathbb{Z}\} \cup \{b_{l,[J-j;k]} : l = 1, ..., s, j = 0, ..., J, k \in \mathbb{Z}\}. \tag{3.1.27}$$

If $\{a; b_1, \ldots, b_2\}$ is a tight framelet filter bank, then this system gives any signal

$v \in l_2(\mathbb{Z})$ a nonhomogeneous representation

$$v = \sum_{k\in\mathbb{Z}} \langle v, a_{[J;k]}\rangle a_{[J;k]} + \sum_{j=1}^{J}\sum_{l=1}^{s}\sum_{k\in\mathbb{Z}} \langle v, b_{l,[j;k]}\rangle b_{l,[j;k]} \tag{3.1.28}$$

$$= \sum_{u\in \mathsf{DAS}_J(\{a;b_1,\ldots,b_s\})} \langle v, u\rangle u, \quad J \in \mathbb{N}. \tag{3.1.29}$$

Similar to the affine systems in continuous setting, the DAS gives a multi-resolution representation of elements in $l_2(\mathbb{Z})$. Actually the filter bank $\{a; b_1, \ldots, b_s\}$ is a tight framelet filter bank if and only if the $\mathsf{DAS}_J(\{a; b_1, \ldots, b_s\})$ is a tight frame of $l_2(\mathbb{Z})$ for any J, and a wavelet filter bank $\{a; b\}$ is orthogonal if and only if $\mathsf{DAS}_J\{a; b\}$ is an orthonormal basis of $l_2(\mathbb{Z})$.

Now it is obvious to see that, for a J-level DFrT, the sequence in each sub band is determined by the generators in DAS_J. By investigating the properties of these generators we can derive the features contained in the framelet coefficients, and hence infer the feature of the input signal. The stability, or asymptotic behavior of multi-level DFrT relies on the asymptotic behavior of a_J.

We now show a simple example of the generators of a DAS. Namely, they are filters defined in (3.1.25) and (3.1.26).

Example 3.1.1. *Let $\{a; b\}$ be a wavelet filter bank obtained by Daubechies [5]:*

$$a = \frac{1}{8}\left\{1+\sqrt{3}, 3+\sqrt{3}, 3-\sqrt{3}, 1-\sqrt{3}\right\}_{[-1,2]};$$

$$b = \frac{1}{8}\left\{1-\sqrt{3}, \sqrt{3}-3, 3+\sqrt{3}, 1+\sqrt{3}\right\}_{[-1,2]}.$$

The generators for the first 3 levels are shown in Fig 10.

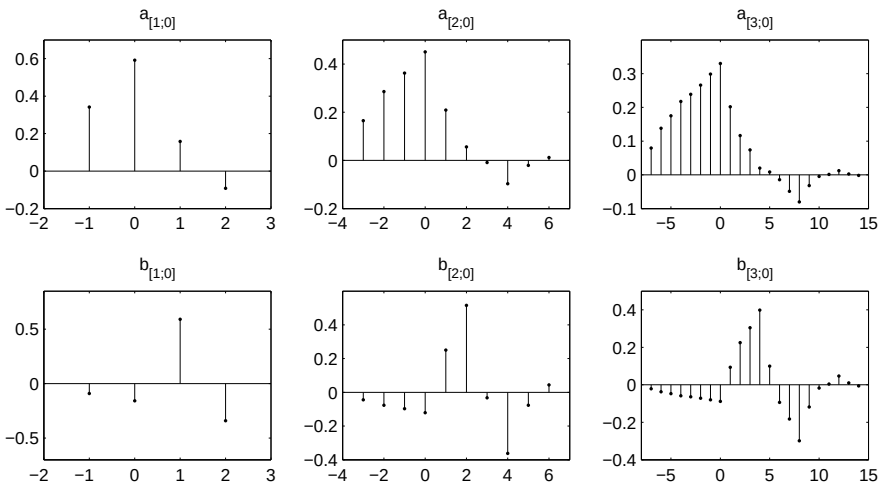


Figure 10: Stem plot of the generators in Example 3.1.1 for the first 3 levels

3.2 Wavelet Functions

We introduced the wavelet and scaling functions in Chapter 1. They are refinement functions corresponding to the high-pass and low-pass filters respectively. From the refinable equation, we establish the relation between a filter (or, a sequence in $l_2(\mathbb{Z})$) and the continuous wavelet or scaling functions. If we know the refinable function explicitly, it is easy to find out the sequence. In this section, we shall investigate their relations in a reversed direction: we show that under a stability condition, the refinable function is actually determined by the sequence uniquely, and can be approximated by a sequence obtained by iterative convolutions.

3.2.1 Frequency Domain Analysis

The refinable equation

$$\phi(t) = 2\sum_{k\in\mathbb{Z}} a(k)\phi(2t-k) \tag{3.2.1}$$

can be equivalently written in the frequency domain as

$$\widehat{\phi}(2\xi) = \sum_{k\in\mathbb{Z}} a(k)e^{-ik\xi}\widehat{\phi}(\xi), \tag{3.2.2}$$

which is just

$$\widehat{\phi}(2\xi) = \widehat{a}(\xi)\widehat{\phi}(\xi). \tag{3.2.3}$$

Based on this relation, we are inspired to define a formal function

$$\varphi^a(\xi) := \prod_{j=1}^{\infty} \widehat{a}(2^{-j}\xi). \tag{3.2.4}$$

$\varphi^a(\xi)$ can be easily checked to satisfy

$$\varphi^a(2\xi) = \prod_{j=1}^{\infty} \widehat{a}(2^{-j+1}\xi) = \prod_{j=0}^{\infty} \widehat{a}(2^{-j}\xi) = \widehat{a}(\xi)\varphi(\xi) \tag{3.2.5}$$

if it is well-defined. It is then called a frequency-based refinable function.

It is well known that, if $\{a; b_1, b_2\}$ is a tight framelet filter bank, then $|\widehat{a}(\xi)|^2 + |\widehat{a}(\xi + \pi)|^2 \leq 1$ for all $\xi \in R$. If a satisfies $\widehat{a}(0) = 1$ and

$$\|\mathcal{W}_J v\|^2_{(l_2(\mathbb{Z})^{1\times(sJ+1)})} \leq C\|v\|^2_{l_2(\mathbb{Z})}, \qquad \forall v \in l_2(\mathbb{Z}), J \in \mathbb{N}, \tag{3.2.6}$$

then it is shown in [6] that φ^a is then a well-defined L_2 function. We define

$$\phi^a(t) = F^{-1}\varphi^a = \frac{1}{2\pi}\int_{\mathbb{R}} \varphi^a(\xi)e^{i\xi t}d\xi, \tag{3.2.7}$$

where F^{-1} is the inverse Fourier transform which is a 1-1, onto mapping in $L_2(\mathbb{R})$. Then $\phi^a(t)$ also belongs to $L_2(\mathbb{R})$ and satisfies the refinement equation in (3.2.1).

3.2.2 Time Domain Approximation

The formal function defined in (3.2.4), even though well-defined, is not easy to compute in practice. However, noticing that for a low-pass filter a, its Fourier transform always satisfies $\widehat{a}(0) = 1$ and as long as it sufficiently smooth at the origin, for sufficiently large J, 2^{-J} is sufficiently small, we have

$$\widehat{a}(2^{-j}\xi) \approx 1, \tag{3.2.8}$$

and hence

$$\varphi^a(\xi) \approx \prod_{j=1}^{J} \widehat{a}(2^{-j}\xi) = \widehat{a_J}(2^{-J}\xi). \tag{3.2.9}$$

This approximation in frequency domain can be transferred into the time domain by Fourier inverse transform:

$$\phi^a(2^{-J}k) = \frac{1}{2\pi}\int_{\mathbb{R}} \varphi^a(\xi) e^{i\xi\cdot 2^{-J}k} d\xi \approx \frac{1}{2\pi}\int_{\mathbb{R}} \widehat{a_J}(2^{-J}\xi) e^{i\xi\cdot 2^{-J}k} d\xi = 2^J a_J(k) \tag{3.2.10}$$

for any integer k.

So the values of ϕ^a on a fine grid $2^{-J}\mathbb{Z}$ can be approximated by the dilation of the filter a_J. For practical use, we only need to address the features in a_J for large J to roughly investigate the properties of ϕ^a in the time domain.

3.3 Discrete Affine System in 2-D

Now that we have defined the DAS in 1-D, the 2-D DAS just consists of the tensor products. We shall first define the tensor product in the continuous framelet setting, then give the DAS in 2-D.

As mentioned above, the framelet filter bank $\{a; b_1, b_2\}$ is corresponding to the

scaling and framelet functions $\{\phi; \psi^1, \psi^2\}$ which satisfies

$$\phi(t) = 2\sum_{k\in\mathbb{Z}} a(k)\phi(2t-k); \tag{3.3.1}$$

$$\psi^1(t) = 2\sum_{k\in\mathbb{Z}} b_1(k)\phi(2t-k); \tag{3.3.2}$$

$$\psi^2(t) = 2\sum_{k\in\mathbb{Z}} b_2(k)\phi(2t-k). \tag{3.3.3}$$

After tensor products of these functions, there are 9 combinations $\phi\otimes\phi, \phi\otimes\psi^1, \phi\otimes\psi^2, \psi^1\otimes\phi, \psi^1\otimes\psi^1, \psi^1\otimes\psi^2, \psi^2\otimes\phi, \psi^2\otimes\psi^1, \psi^2\otimes\psi^2$. Since ϕ corresponds to the low-pass filter and ψ^i corresponds to the high-pass filters, we name them L-L ($\phi\otimes\phi$), L-H1 ($\phi\otimes\psi^1$), L-H2 ($\phi\otimes\psi^2$) respectively. We can see that

$$\phi(t)\otimes\phi(s) = 4\sum_{k\in\mathbb{Z}} a(k)\phi(2t-k)\sum_{l\in\mathbb{Z}} a(l)\phi(2s-l) \tag{3.3.4}$$

$$= 4\sum_{(k,l)\in\mathbb{Z}^2} a(k)a(l)\phi(2t-k)\phi(2s-l) \tag{3.3.5}$$

$$= 4\sum_{(k,l)\in\mathbb{Z}^2} [a\otimes a](k,l)[\phi\otimes\phi](2(t,s)-(k,l)). \tag{3.3.6}$$

So the filter bank in 2-D is just defined as $\{a\otimes a; a\otimes b_1, a\otimes b_2, b_1\otimes a, ..., b_2\otimes b_2\}$, where $a\otimes a$ is the L-L band, $a\otimes b_1$ is the L-H1 band, etc.

Based on the filters

$$[a\otimes a]_j(k,l) = a_j\otimes a_j, \tag{3.3.7}$$

where a_j is defined in (3.1.24),

$$[a\otimes a]_{[j,(k,l)]} := 4^{j/2}[a\otimes a]_j(\cdot - 2^j(k,l)); \tag{3.3.8}$$

$$[a\otimes b_i]_{[j,(k,l)]} := 4^{j/2}\{[a\otimes a]_j * [a\otimes b_i]\uparrow 2^{j-1}\}(\cdot - 2^j(k,l)); \quad i\in\{1,2\}; \tag{3.3.9}$$

$$[b_i\otimes b_j]_{[j,(k,l)]} := 4^{j/2}\{[a\otimes a]_j * [b_i\otimes b_j]\uparrow 2^{j-1}\}(\cdot - 2^j(k,l)); \quad i,j\in\{1,2\}. \tag{3.3.10}$$

The 2-D DAS_J is defined as

$$\begin{aligned}
&\mathsf{DAS}_J(\{a\otimes a; a\otimes b_1, ..., b_s\otimes b_s\})\\
&:= \{[a\otimes a]_{[J;k]}, k\in\mathbb{Z}\}\\
&\cup \{[a\otimes b_i]_{[J-j;k]} | i = 1, ..., s, j = 0, ..., J, k\in\mathbb{Z}\}\\
&\cup \{[b_i\otimes b_j]_{[J-j;k]} | i, j = 1, ..., s, j = 0, ..., J, k\in\mathbb{Z}\}.
\end{aligned} \tag{3.3.11}$$

It gives any signal $v\in(l_2(\mathbb{Z}))^2$ a representation

$$v = \sum_{u\in\mathsf{DAS}_J(\{a\otimes a,...,b_s\otimes b_s\})} \langle v, u\rangle u, \quad J\in\mathbb{N}. \tag{3.3.12}$$

Therefore, similar to 1-D case, the sub-signals in 2-D is also determined completely by the generators of the 2-D DAS. Through similar derivation, in 2-D case the refinable functions can also be approximated by the generators of 2-D DAS_J for large J. Hence to evaluate the directionality of a filter bank, except for the scaling and wavelet functions, we can also investigate the generators in the 2-D DAS.

Because we are interested in the directional features of images, now it comes to the problem to see the directional properties of the generators of the 2-D DAS. In Chapter 4, we will give examples of directional filter banks, and plot the generators to a certain level. It is clear that the filter banks process good directionality.

Chapter 4

Construction Algorithm and Examples

In this chapter, we shall give the algebraic algorithm to construct directional tight framelet filter banks, and provide several examples with relatively short support yet good directionality.

4.1 Algorithm

Our algorithm is based on the construction of symmetric tight framelet filter banks with 2 generators in [8] and [6]. In Chapter 2, we have stated the necessary and sufficient condition to construct a tight framelet filter bank $\{a; b_1, b_2\}$ given the symmetric low-pass filter a with $\mathsf{Sa}(z) = \epsilon z^c$, and a nonnegative integer $n_b \in [0, \min(sr(a), \frac{1}{2}vm(-\mathsf{a}(z)\mathsf{a}^\star(z)))]$:

(i) $\mathcal{N}_{a|n_b}(z) = \begin{pmatrix} \frac{1}{2} - \mathsf{a}^{[0]}(z)(\mathsf{a}^{[0]}(z))^\star & -\mathsf{a}^{[0]}(z)(\mathsf{a}^{[1]}(z))^\star \\ -\mathsf{a}^{[1]}(z)(\mathsf{a}^{[0]}(z))^\star & \frac{1}{2} - \mathsf{a}^{[1]}(z)(\mathsf{a}^{[1]}(z))^\star \end{pmatrix}$ is semi-positive definite for any z;

(ii) there exists a Laurent polynomial d with symmetry type $\epsilon_d z^{c_d}$ such that

$$\mathsf{d}(z)\mathsf{d}^\star(z) = \det(\mathcal{N}_{a|n_b})(z) \tag{4.1.1}$$

and

$$\epsilon_d = (-1)^{\mathrm{odd}(c+n_b)+1}, \tag{4.1.2}$$

$$c_d = 2j + c + n_b + 1 \tag{4.1.3}$$

for some integer j.

Now, based on the algorithm to construct symmetric tight framelet filter banks with 2 generators in [6], we give the detailed algorithm.

(1) Matrix construction

Denote the filter support of $[\mathcal{N}_{a|n_b}]_{1,2}$ as $[n_-, n_+]$. Calculate a 2×2 matrix $\mathcal{N}$ as

$$\mathcal{N} = \begin{bmatrix} \mathcal{N}_{1,1} & \mathcal{N}_{1,2} \\ \mathcal{N}_{2,1} & \mathcal{N}_{2,2} \end{bmatrix} = \begin{cases} \mathcal{N}_{a|n_b} & \text{if } c + n_b \text{ is even;} \\ P_{k_a} \mathcal{N}_{a|n_b} P_{k_a}^\star & \text{if } c + n_b \text{ is odd,} \end{cases} \tag{4.1.4}$$

where

$$P = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & z^{k_a} \\ 1 & -z^{k_a} \end{bmatrix}, \tag{4.1.5}$$

with $k_a = \lfloor \frac{n_- + n_+}{2} \rfloor$.

(2) Define $\mathsf{p}(z) = \gcd(\mathcal{N}_{1,1}, \mathcal{N}_{1,2}, \mathcal{N}_{2,1}, \mathcal{N}_{2,2})$, and

$$\breve{\mathcal{N}}(z) := \frac{1}{\mathsf{p}(z)} \mathcal{N}(z), \tag{4.1.6}$$

then calculate $\breve{\mathsf{p}}(z) = \gcd(\breve{\mathcal{N}}_{1,1}(z), \breve{\mathcal{N}}_{1,2}(z) \breve{\mathcal{N}}_{2,1}(z))$.

(3) Because $\mathcal{N}$ is semi-positive definite and each entry of it is symmetric, so is $\mathsf{p}(z)$ and $\breve{\mathcal{N}}(z)$, and therefore $\breve{\mathsf{p}}(z)$. So we can find a Laurent polynomial $\mathsf{q}(z) \geq 0$ for any $z \in \mathbf{T}$ with symmetry such that

$$\mathsf{q}(z) \mathsf{q}^\star(z) = \breve{\mathsf{p}}(z), \quad \mathsf{q}(z) \mid \breve{\mathcal{N}}_{1,2}(z). \tag{4.1.7}$$

The construction of q is as follow:

(i)Compute $\mathsf{q}_1 = \gcd(\breve{\mathsf{p}}, \breve{\mathcal{N}}_{1,2})$ with symmetry;

(ii) Compute $\tilde{\mathsf{p}} = \frac{\mathsf{q}_1 \mathsf{q}_1^\star}{\breve{\mathsf{p}}}$ then it is a well-defined Laurent polynomial s.t. $\tilde{\mathsf{p}}(z) \geq 0$ for all $z \in \mathbb{T}$; so that roots of $\tilde{\mathsf{p}}$ can be all collected as $\{z, \overline{1/z}$.

(iii) Compute q_2 such that $\mathsf{q}_2 \mathsf{q}_2^\star = \tilde{\mathsf{p}}$ with symmetry, in particular we can pick $\mathsf{q}_2 =$

$\prod_i(z-z_i)$ such that $\{z_i, \overline{1/z_i}$ are distinct sets of $\tilde{\mathsf{p}}$'s roots.

(iv) Define $\mathsf{q}=\frac{\mathsf{q}_1}{\mathsf{q}_2}$. Then q also has symmetry.

(4) Then define

$$\mathring{\mathcal{N}}(z):=\begin{bmatrix} \frac{\breve{\mathcal{N}}_{1,1}(z)}{\mathsf{q}(z)\mathsf{q}^\star(z)} & \frac{\breve{\mathcal{N}}_{1,2}(z)}{\mathsf{q}(z)} \\ \frac{\breve{\mathcal{N}}_{2,1}(z)}{\mathsf{q}^\star(z)} & \breve{\mathcal{N}}_{2,2} \end{bmatrix} \tag{4.1.8}$$

so that every entry has symmetry, and $\mathring{\mathcal{N}}_{1,1}$,$\mathring{\mathcal{N}}_{1,2}$ has no common zeros.

(5) Define $[-n,n]=\mathrm{fsupp}(\mathring{\mathcal{N}}_{1,1}), \epsilon_{\mathring{\mathcal{N}}_{2,1}} z^{c_{\mathring{\mathcal{N}}_{2,1}}}=\mathsf{S}\mathring{\mathcal{N}}_{2,1}, \mathring{\epsilon} z^{\mathring{c}}=\widetilde{\mathsf{Sd}}, c_{\mathrm{odd}}=\mathrm{odd}(c_{\mathring{\mathcal{N}}_{2,1}})$ and

$$\mathring{\mathsf{d}}(z)=z^{n+\frac{c_{\mathring{\mathcal{N}}_{2,1}}-\mathring{c}-c_{\mathrm{odd}}}{2}} \tag{4.1.9}$$

with $\mathring{\mathsf{Sd}}(z)=\mathring{\epsilon} z^{2n+c_{\mathring{\mathcal{N}}_{2,1}}-c_{\mathrm{odd}}}$.

(6) Write $\mathring{\mathcal{U}}_{1,1}(z)=\sum_{j=0}^{n-c_{\mathrm{odd}}} t_j z^j$ and $\mathring{\mathcal{U}}_{1,2}(z)=\sum_{j=0}^{n} \tilde{t}_j z^j$, then solve for $\{t_0,...,t_{n-c_{\mathrm{odd}}},\tilde{t}_0,...,\tilde{t}_n\}$ with the equation

$$\mathring{\mathcal{N}}_{2,1}(z)\mathring{\mathcal{U}}_{1,1}(z)-\mathring{\mathsf{d}}(z)\mathring{\mathcal{U}}^\star_{1,2}(z)=\mathring{\mathcal{N}}_{1,1}(z)\mathring{\mathcal{U}}_{2,1}(z). \tag{4.1.10}$$

This equation has solution space with dimension at least 1, because by Euclidean division, we can always write

$$\mathring{\mathcal{N}}_{2,1}(z)\mathring{\mathcal{U}}_{1,1}(z)-\mathring{\mathsf{d}}(z)\mathring{\mathcal{U}}^\star_{1,2}(z)=\mathring{\mathcal{N}}_{1,1}(z)\mathring{\mathcal{U}}_{2,1}(z)+\mathcal{R} \tag{4.1.11}$$

with $\mathrm{fsupp}(\mathcal{R})\subseteq[-n,n-1]$, set $\mathcal{R}=0$, we have $2n$ equations but at leat $2n+1$ variables.

(7) Update $\mathring{\mathcal{U}}_{1,1}(z)=[\mathring{\mathcal{U}}_{1,1}(z)+\epsilon\epsilon_{\mathring{\mathcal{N}}_{2,1}} z^{n-c_{\mathrm{odd}}}\mathring{\mathcal{U}}_{1,1}(z^{-1})]/2$, and $\mathring{\mathcal{U}}_{1,2}(z)=[\mathring{\mathcal{U}}_{1,2}(z)+\epsilon\mathring{\epsilon} z^n\mathring{\mathcal{U}}_{1,2}(z^{-1})]/2$, choosing $\epsilon=1$ or -1 such that the two are not zero at the same time.

(8) Update $[\mathring{\mathcal{U}}_{1,1}(z),\mathring{\mathcal{U}}_{1,2}(z)]=\lambda[\mathring{\mathcal{U}}_{1,1}(z),\mathring{\mathcal{U}}_{1,2}(z)]$ where $\lambda=\sqrt{\frac{\mathring{\mathcal{N}}_{1,1}(1)}{|\mathring{\mathcal{U}}_{1,1}(1)|^2+|\mathring{\mathcal{U}}_{1,2}(1)|^2}}$, and define

$$\mathring{\mathcal{U}}_{2,1}(z)=\frac{\mathring{\mathcal{N}}_{2,1}(z)\mathring{\mathcal{U}}_{1,1}(z)-\mathring{\mathsf{d}}(z)\mathring{\mathcal{U}}^\star_{1,2}(z)}{\mathring{\mathcal{N}}_{1,1}(z)}, \tag{4.1.12}$$

$$\mathring{\mathcal{U}}_{2,2}(z) = \frac{\mathring{\mathcal{N}}_{2,1}(z)\mathring{\mathcal{U}}_{1,2}(z) + \mathring{\mathsf{d}}(z)\mathring{\mathcal{U}}^{\star}_{1,1}(z)}{\mathring{\mathcal{N}}_{1,1}(z)}. \tag{4.1.13}$$

(9) Set

$$\mathcal{U} = \begin{bmatrix} \mathcal{U}_{1,1} & \mathcal{U}_{1,2} \\ \mathcal{U}_{2,1} & \mathcal{U}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathsf{q}(z) & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathring{\mathcal{U}}_{1,1} & \mathring{\mathcal{U}}_{1,2} \\ \mathring{\mathcal{U}}_{2,1} & \mathring{\mathcal{U}}_{2,2} \end{bmatrix} \tag{4.1.14}$$

where q is such that $\mathsf{q}\mathsf{q}^{\star} = \gcd(\mathcal{N}_{ij})$.

(10) If $c + n_b$ is even,

$$\mathsf{b}_1(z) = (1 - z^{-1})^{n_b}[\mathcal{U}_{1,1}(z^2) + z\mathcal{U}_{2,1}(z^2)], \tag{4.1.15}$$

$$\mathsf{b}_1(z) = (1 - z^{-1})^{n_b}[\mathcal{U}_{1,2}(z^2) + z\mathcal{U}_{2,2}(z^2)]; \tag{4.1.16}$$

If $c + n_b$ is odd,

$$\mathsf{b}_1(z) = (1 - z^{-1})^{n_b}[\frac{1 + z^{1-2k_a}}{\sqrt{2}}\mathcal{U}_{1,1}(z^2) + \frac{1 - z^{1-2k_a}}{\sqrt{2}}\mathcal{U}_{2,1}(z^2)], \tag{4.1.17}$$

$$\mathsf{b}_2(z) = (1 - z^{-1})^{n_b}[\frac{1 + z^{1-2k_a}}{\sqrt{2}}\mathcal{U}_{1,2}(z^2) + \frac{1 - z^{1-2k_a}}{\sqrt{2}}\mathcal{U}_{2,2}(z^2)]. \tag{4.1.18}$$

(11) Replace b_1 and b_2 by $\mathsf{b}_1 + e^{i\theta}\mathsf{b}_2$ and $\mathsf{b}_1 - e^{i\theta}\mathsf{b}_2$ respectively. Optimize with respect to $\theta \in [-\pi, \pi)$, numerically or algebraicly, with the target function

$$\mathrm{argmin}_{\theta} E_i = \int_0^{\pi} |\hat{b}_i(\xi)|^2 d\xi, i = 1, 2; \tag{4.1.19}$$

or

$$\mathrm{argmin}_{\theta} F_i = \int_0^{\pi} |\hat{b}_i(\xi)\hat{a}(\xi/2)|^2 d\xi, i = 1, 2, \tag{4.1.20}$$

where E_i and F_i are defined in (2.3.3) and (2.3.10). Then we finish the construction of a tight framelet filter bank $\{a; b_1, b_2\}$.

4.2 Examples

We now show some calculations and figures of directional tight framelet filter banks.

Example 4.2.1. *The second order B-spline low pass filter is* $a = \{1/4, 1/2, 1/4\}_{[0,2]}$, $c = 2, n_b = 1$, *so* $c + n_b$ *is odd.*

$$\mathcal{N} = P_{k_a}\mathcal{N}_{a|1}P_{k_a}^{\star} = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{8} \end{bmatrix}, \tag{4.2.1}$$

$$\mathcal{U} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2\sqrt{2}} \end{bmatrix}. \tag{4.2.2}$$

$$\mathring{\mathsf{b}}_1 = \frac{\sqrt{2}}{4}(z^2 - 1); \mathring{\mathsf{b}}_2 = -\frac{1}{4}(z^2 - 2z + 1). \tag{4.2.3}$$

Numerically seeking for least E_i, *we find the high-pass filters*

$$\mathsf{b}_1 = -0.1082532 + 0.3040525i + (-0.1804220 - 0.3040525i)z + 0.2886751z^2; \tag{4.2.4}$$

$$\mathsf{b}_2 = 0.2886751 + (-0.1804220 - 0.3040525i)z + (-0.1082532 + 0.3040525i)z^2. \tag{4.2.5}$$

We plot the graphs of the filters' absolute values in the frequency domain in Fig 11. In the time domain, the real and imaginary parts of the scaling function (corresponding to the low-pass filter) and wavelet functions (corresponding to the high-pass filters) are shown in Fig 12. The generators of the 1-D DAS up to level 3 are shown in Fig 13. Since the 2 high-pass filters have a relative symmetry, we shall only give the generators of b_1 in other examples.

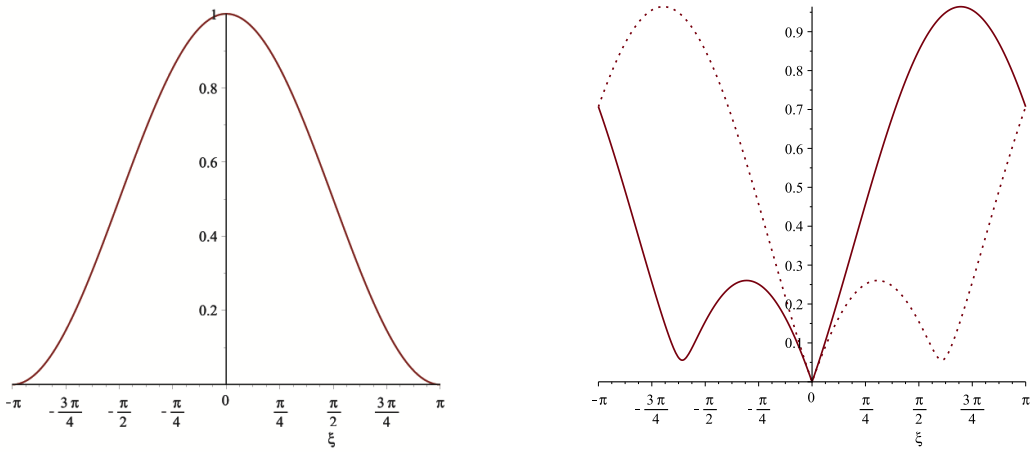


Figure 11: Magnitudes of the B_2 spline low-pass filter (left) and B_2 spline high-pass filter (right) in the frequency domain.

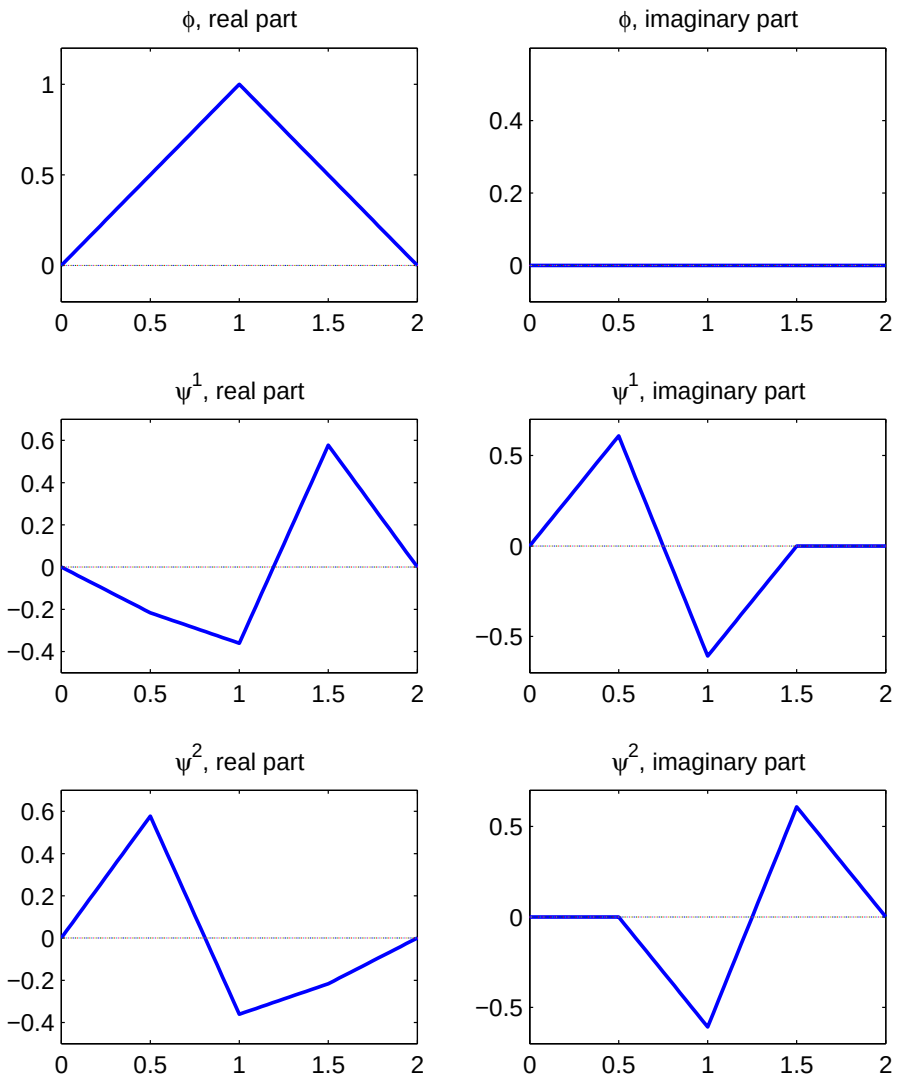


Figure 12: The scaling function ϕ and wavelet functions ψ^1 and ψ^2 of the directional B_2 spline tight framelet filter bank.

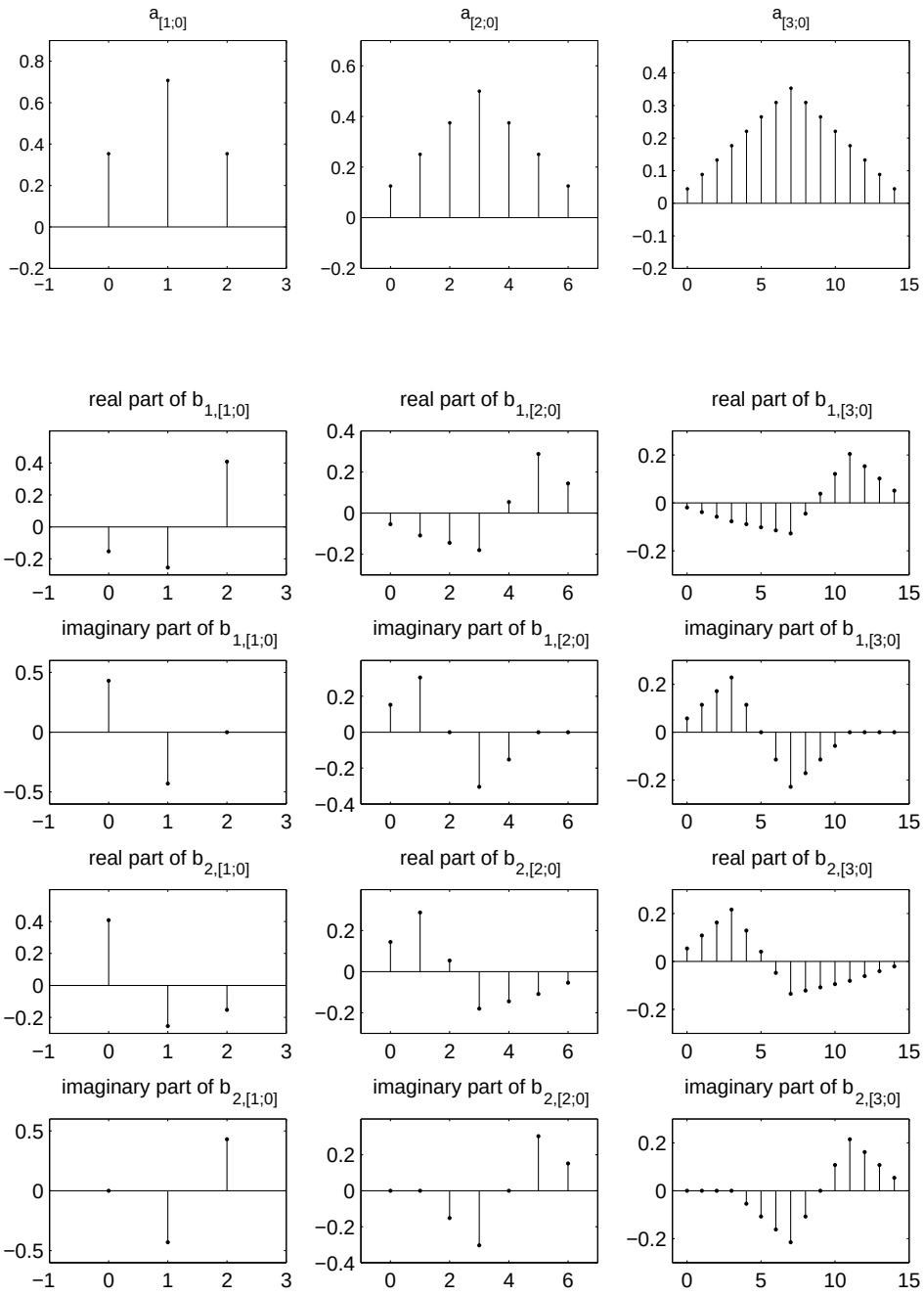
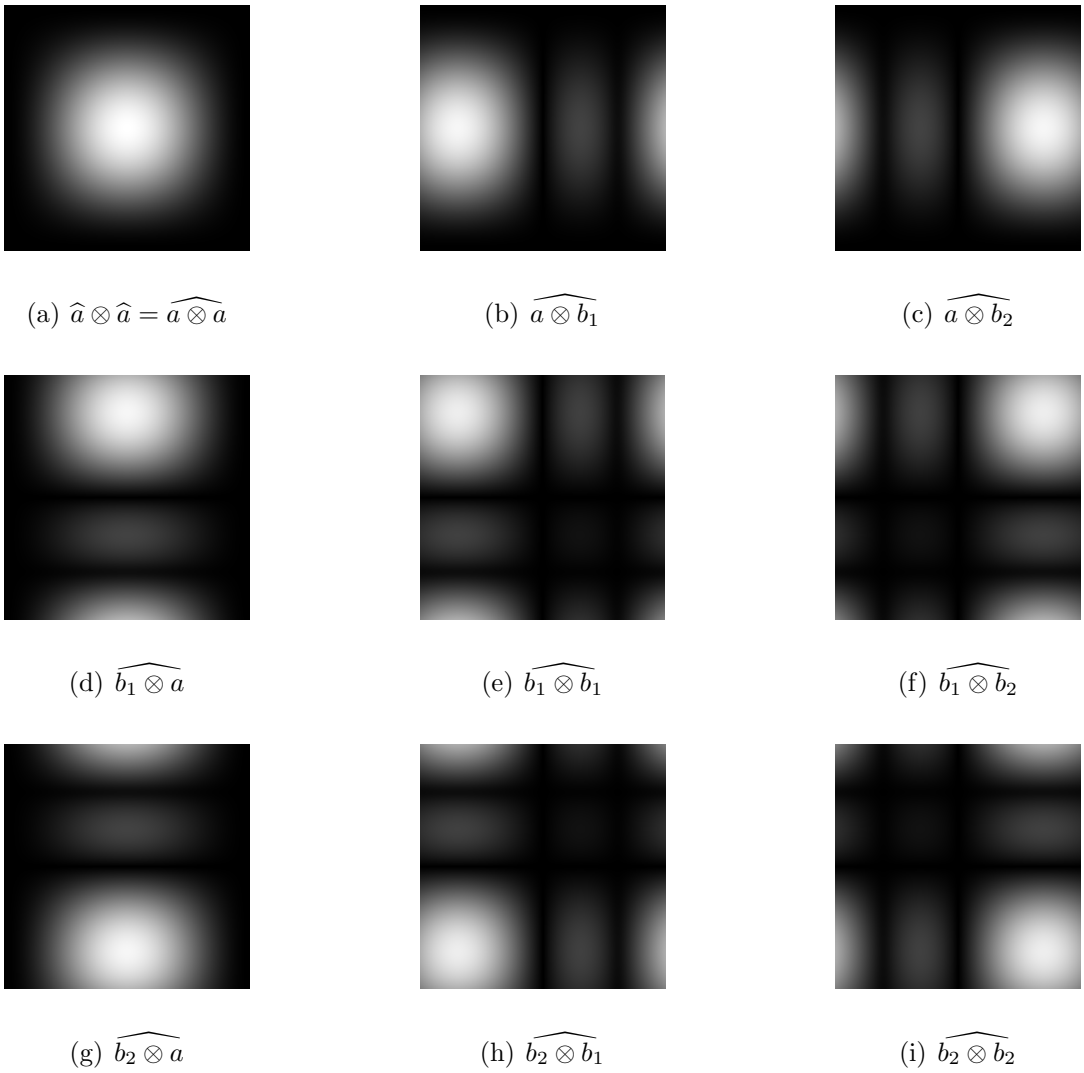


Figure 13: Stem plot of the generators of the 1-D DAS of the directional B_2 spline tight framelet filter bank.



(a) $\widehat{a} \otimes \widehat{a} = \widehat{a \otimes a}$ (b) $\widehat{a \otimes b_1}$ (c) $\widehat{a \otimes b_2}$

(d) $\widehat{b_1 \otimes a}$ (e) $\widehat{b_1 \otimes b_1}$ (f) $\widehat{b_1 \otimes b_2}$

(g) $\widehat{b_2 \otimes a}$ (h) $\widehat{b_2 \otimes b_1}$ (i) $\widehat{b_2 \otimes b_2}$

Figure 14: Grey-scale plot of the 2-D directional filter bank in frequency domain (Example 4.3.1), obtained from tensor product.

In 2 dimensions, both the time domain and frequency domain are just tensor products of 1-D objects, including the bands L-L, L-H1, L-H2, H1-L, H1-H1, H1-H2, H2-L, H2-H1, H2-H2. The plots of the filters in the frequency domain are in Fig 14. Here we plot the absolute value to illustrate the concentration. The generators of 2-D DAS are shown in Fig 15 and Fig 16. From the generators we can see the apparent directionality covering 4 directions in total : the vertical, horizontal, and 2 diagonals.

However, the filter bank has a very short support (the length of each filter's support

equals 3), the time domain functions ϕ, ψ^1 and ψ^2 are not very smooth. the frequency separation is not quite ideal actually. For example, the (h) and (i) in Fig 14 have several lighted areas. This explains that the directionality in the time domain is not always good, like in the $[b_1 \otimes b_2]_{[3,0]}$ band in Fig 15 is relatively difficult to tell the diagonal direction.

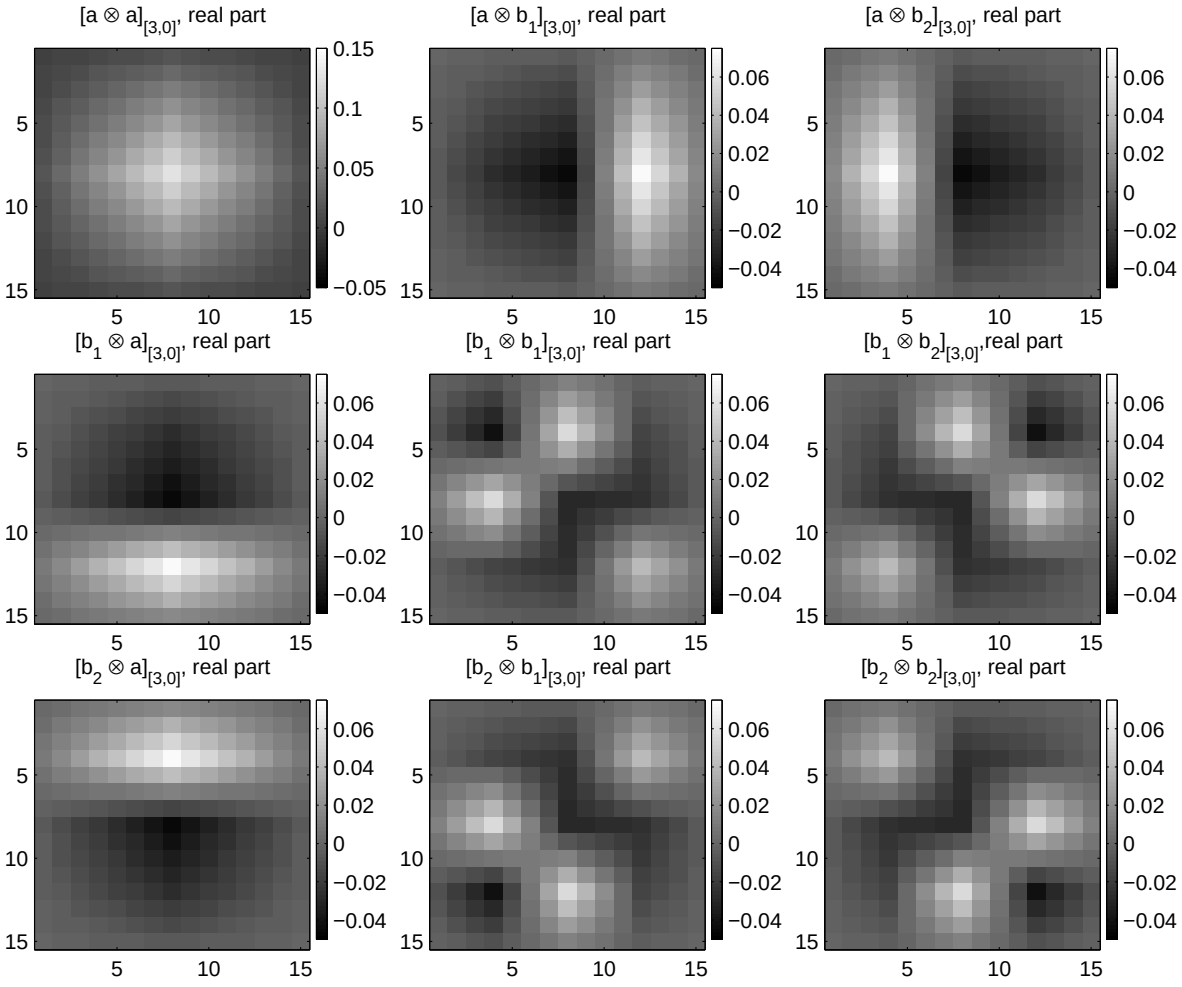


Figure 15: The generators of 2-D DAS for the directional B_2 spline tight framelet filter bank on $\mathbb{Z}^2$, real part (level=3).

Example 4.2.2. *Let* $a = \{-0.051777, 0.301777, 0.25, 0.25, 0.301777, -0.051777\}_{[0,5]}$,

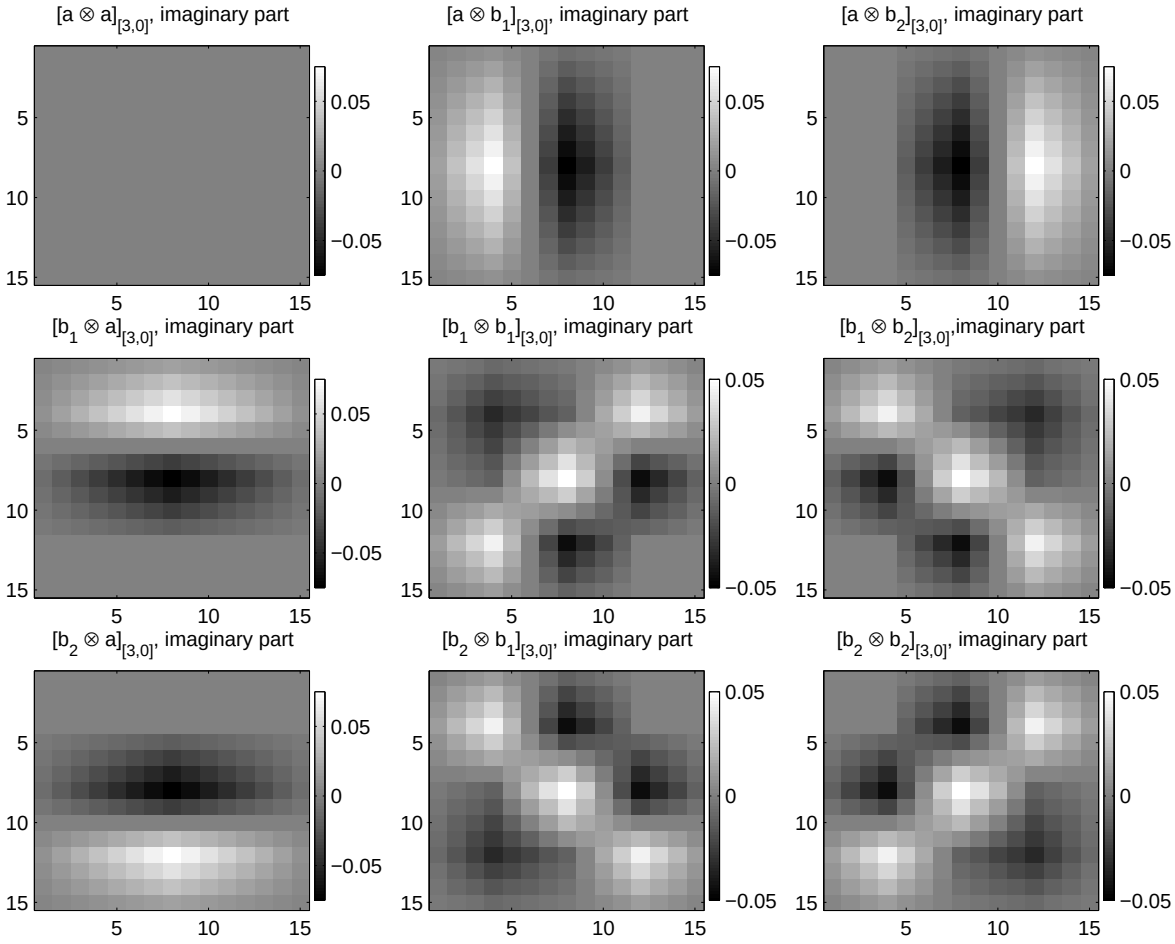


Figure 16: The generators of 2-D DAS for the directional B_2 spline tight framelet filter bank on $\mathbb{Z}^2$, imaginary part (level=3).

k	$a(k)$	$b_1(k)$
0	-0.051777	0.051776479838823 + 0.036611867809496i
1	0.301777	-0.301776203647670 - 0.213388563106133i
2	0.25	0 + 0.176776695296637i
3	0.25	0 + 0.176776695296637i
4	0.301777	0.301776203647670 - 0.213388563106133i
5	-0.051777	-0.051776479838823 + 0.036611867809496i

Table 1: Coefficients of the directional tight framelet filter bank in Example 4.3.2

then $\mathrm{sr}(a) = 1$, $c = 5$. *We take* $n_b = 0$, *then* $\det \mathcal{N}_{a|n_b} = 0.9375 - 0.625(z + z^{-1}) + 0.15625(z^2 + z^{-2})$. *There exists* $\mathsf{d} = 0.25 + 0.125(z + z^{-1})$ *such that* $\mathsf{dd}^\star = \det \mathcal{N}_{a|n_b}$. *The conditions in Theorem 2.2.2.*

The coefficients of the directional tight framelet filter bank are given as in Table 1. and $b_2(k) = b_1(5 - k)$.

The low-pass filter has sum rule $\mathrm{sr}(a) = 1$, *which means relatively less smoothness. The scaling function* ϕ *and wavelet functions* ψ^1 *and* ψ^2 *are given in Fig 4.2. Fig 18 shows the generators of the 1-D* DAS. *For 2-D, the frequency domain of the filters are shown in Fig 19. Generators of the 2-D* DAS *are given in Fig 20 and 21.*

This example has clear diagonal directions in the real part of the 2-D DAS generators. But from the frequency domain we can see the frequency separation is still not good enough, leading to the non-satisfactory directionality in the imaginary part of the 2-D DAS generators (see Fig 21). It also has obvious block-like effect. This can be explained easily by the 1-D generators. From Fig 18 we can see the apparent jumps in the 1-D generators, which leads to this block-like effect.

Example 4.2.3. *Let* a *be the interpolatory low-pass filter*

$$a_6^I = \{\frac{3}{512}, 0, -\frac{25}{512}, 0, \frac{75}{256}, \frac{1}{2}, \frac{75}{256}, 0, -\frac{25}{512}, 0, \frac{3}{512}\}_{[-5,5]},$$

then $\mathrm{sr}\, a = 6, c = 0$. *Take* $n_b = 3$, *then there exists* $\mathsf{d}(z) = \frac{\sqrt{2}}{1024} z^2(3-(16-6\sqrt{15}i)z+3z^2)$ *such that* $\mathsf{dd}^\star = \det \mathcal{N}_{a|n_b}$.

The coefficients of the directional tight filter bank are given in Table 2.

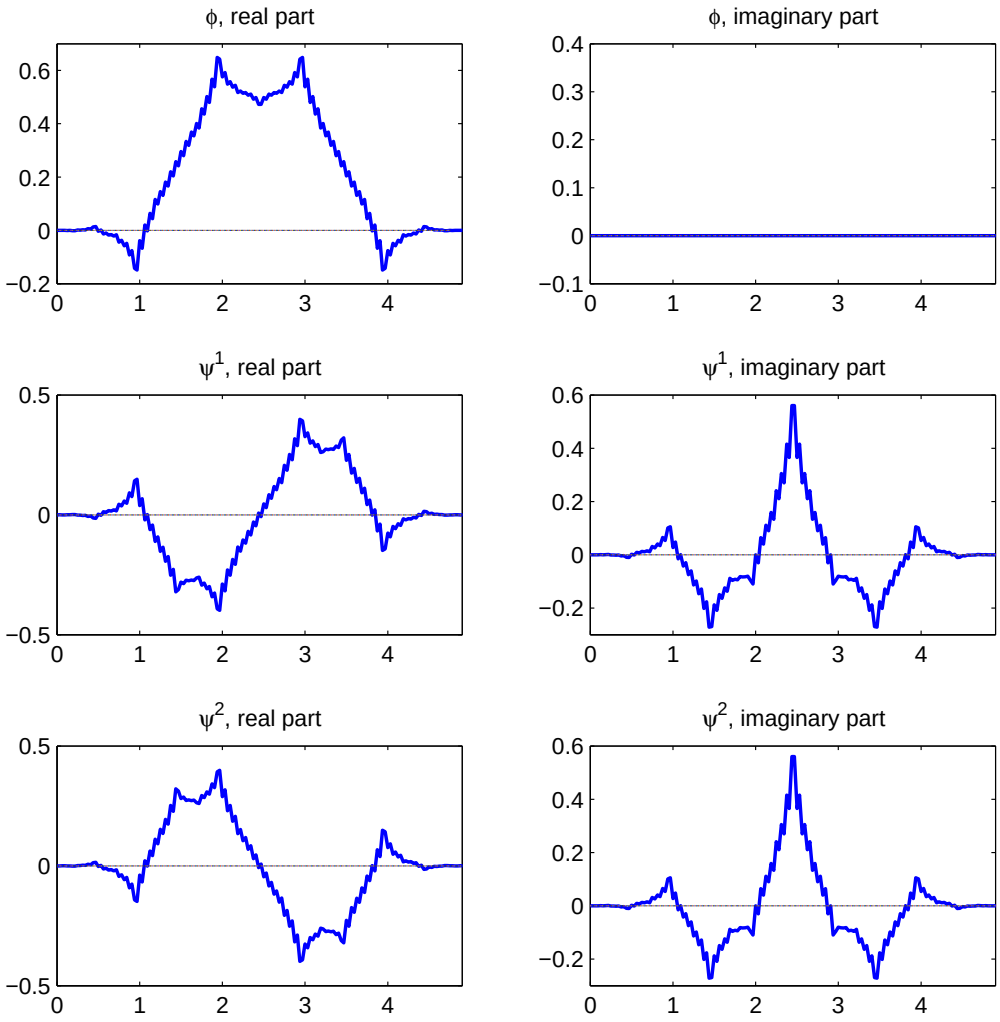


Figure 17: The scaling function ϕ and wavelet functions ψ^1 and ψ^2 of the directional tight framelet filter bank in Example 4.3.2.

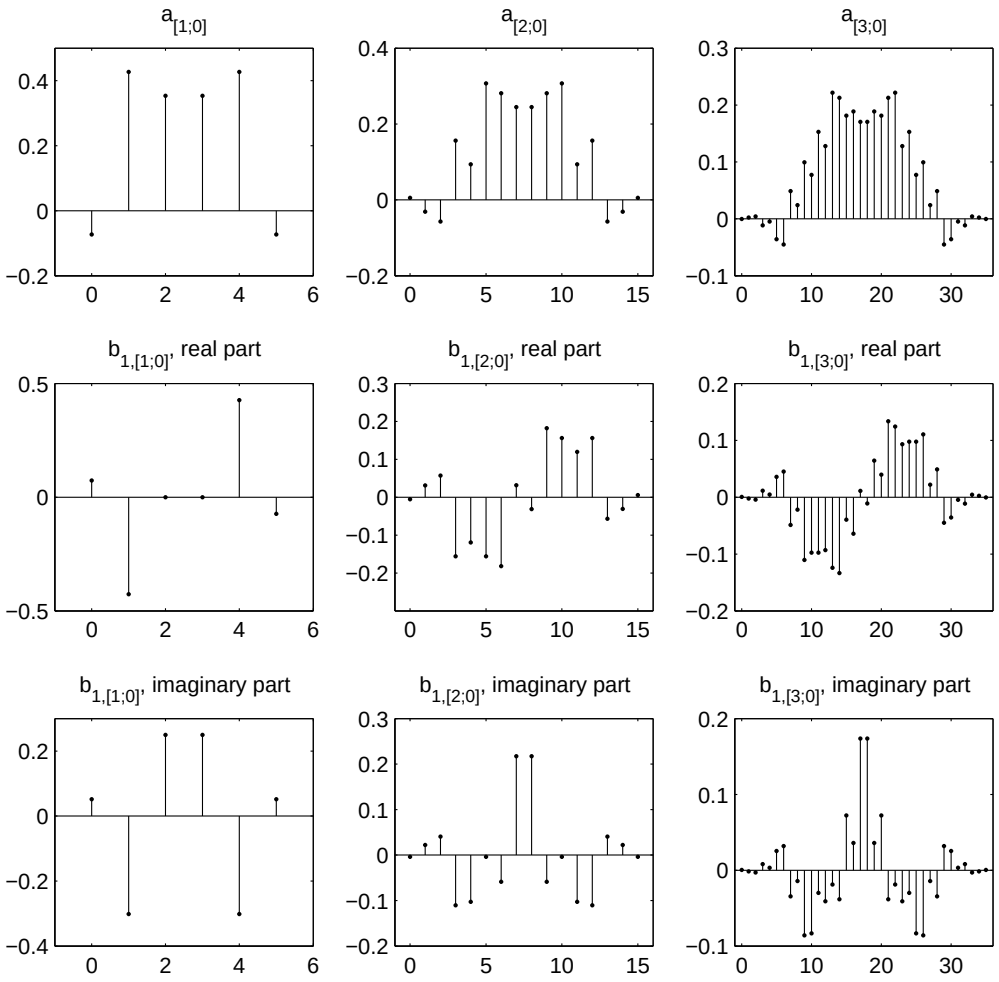
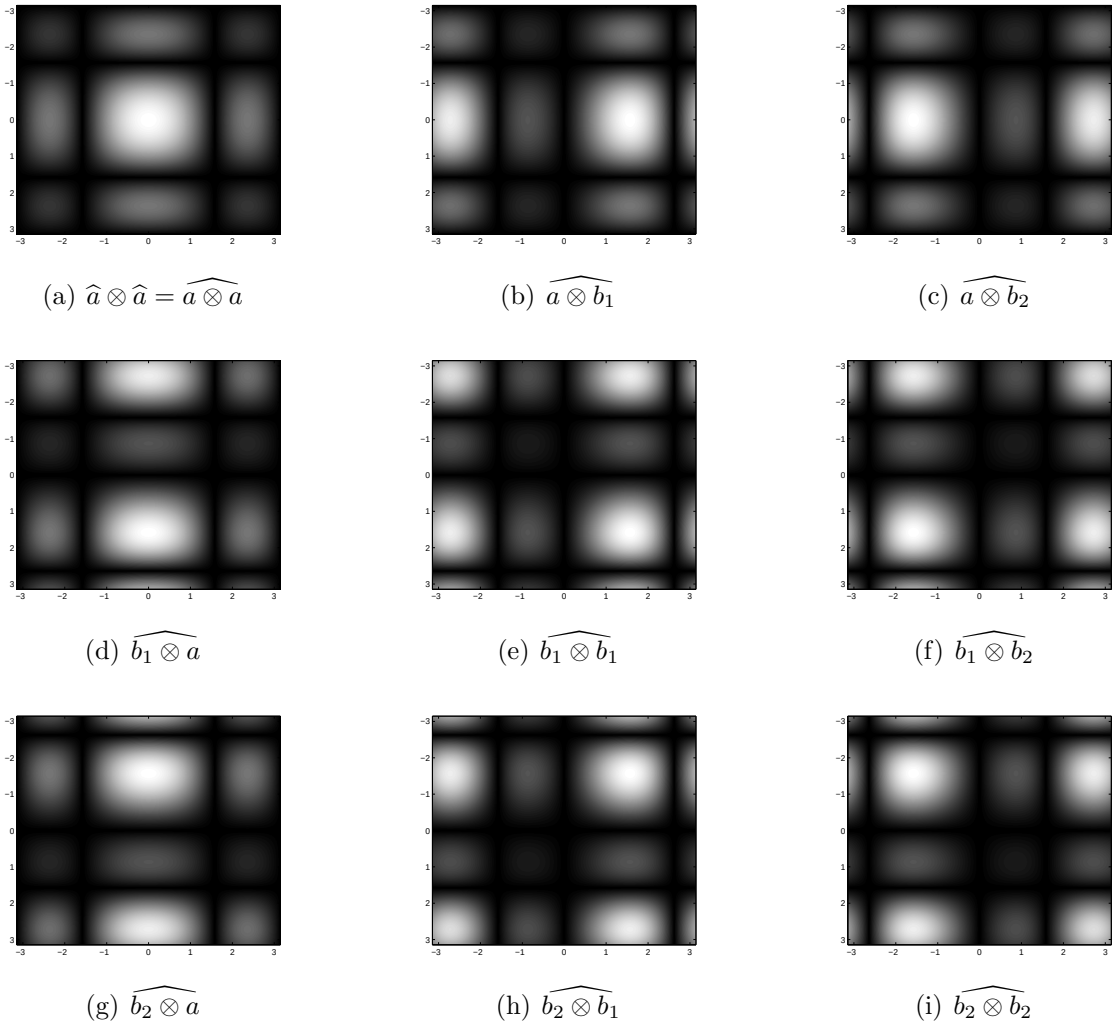


Figure 18: Stem plot of the generators of the 1-D DAS of the directional tight framelet filter bank in Example 4.3.2.



(a) $\widehat{a} \otimes \widehat{a} = \widehat{a \otimes a}$ (b) $\widehat{a \otimes b_1}$ (c) $\widehat{a \otimes b_2}$

(d) $\widehat{b_1 \otimes a}$ (e) $\widehat{b_1 \otimes b_1}$ (f) $\widehat{b_1 \otimes b_2}$

(g) $\widehat{b_2 \otimes a}$ (h) $\widehat{b_2 \otimes b_1}$ (i) $\widehat{b_2 \otimes b_2}$

Figure 19: Grey-scale plot of the 2-D directional filter bank in frequency domain (Example 4.3.2), obtained from tensor product.

and $b_2(k) = b_1(10 - k)$.

Note that $\mathrm{sr}(a) = 6$, *The 1-D functions* ϕ,ψ^1 *and* ψ^2 *are given in Fig 4.2. The stem figures in 1-D are given in Fig 23. In 2-D, the figures in the frequency domain are shown in Fig 24. Generators of 2-D* DAS *are given in Fig 25 and 26.*

This filter bank has a very nice directionality for the real parts of 2-D DAS generators. From the 2-D frequency domain, it has a relatively good separation but still not perfect. In the H1H1, H1H2, H2H1 and H2H2 bands, besides the lighted corners,

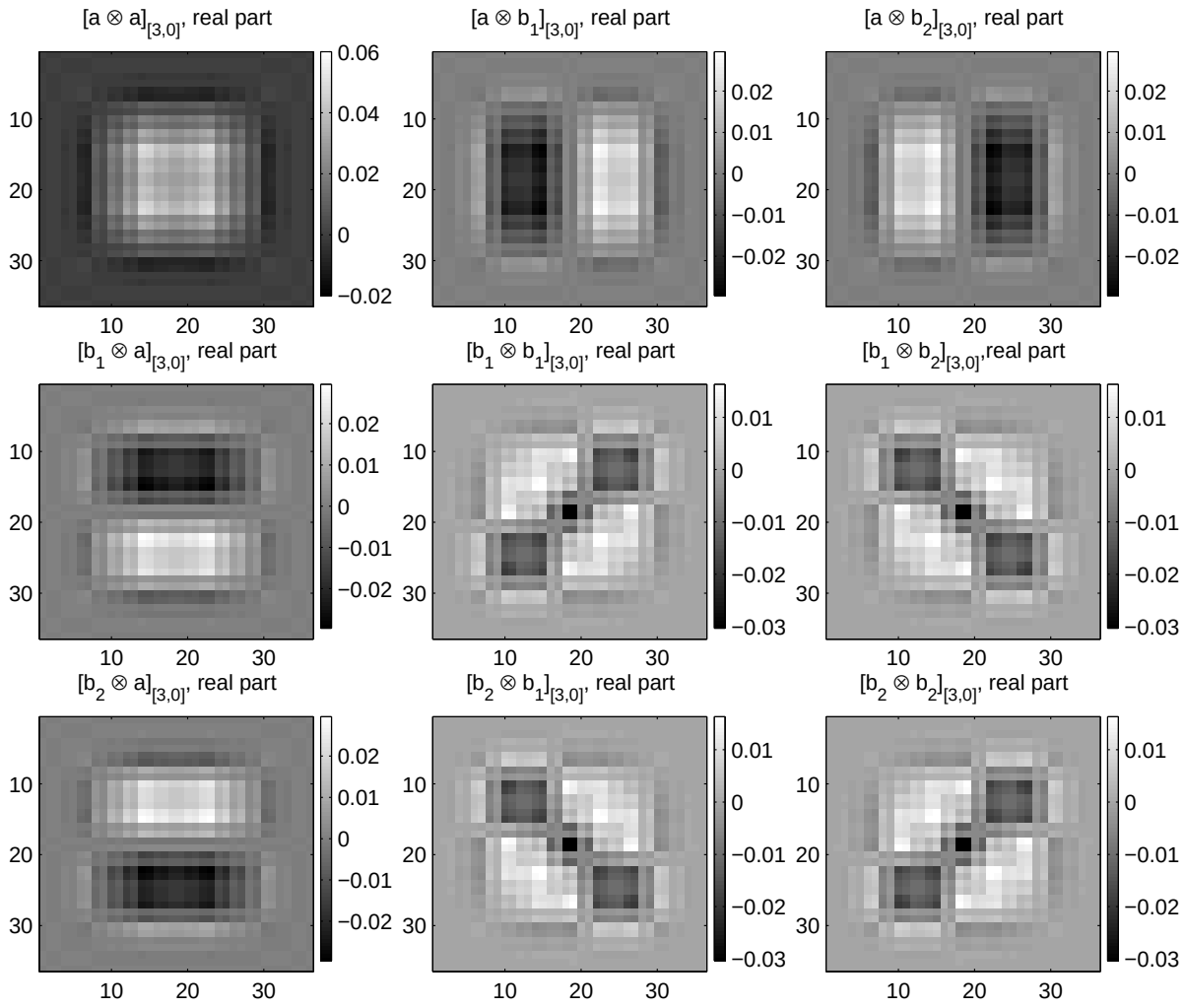


Figure 20: The generators of 2-D DAS for the directional tight framelet filter bank in Example 4.3.2 on $\mathbb{Z}^2$, real part (level=3).

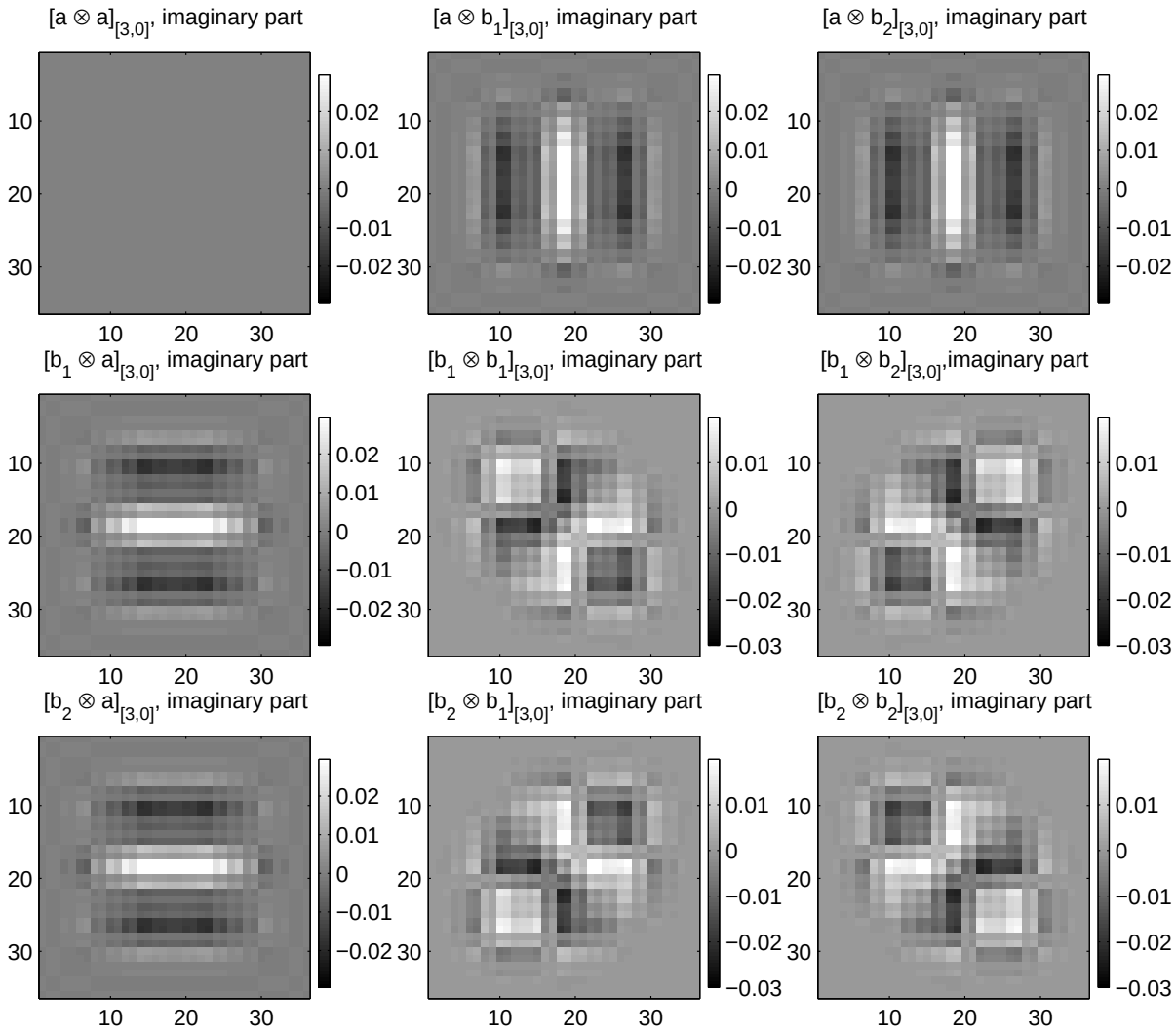


Figure 21: The generators of 2-D DAS for the directional tight framelet filter bank in Example 4.3.2 on $\mathbb{Z}^2$, imaginary part (level=3).

k	$a(k)$	$b_1(k)$
0	0.005859375	0.000634295950037 + 0.002825770851695i
1	0	0
2	-0.048828125	0.100326740979554 - 0.002508343205915i
3	0	0
4	0.29296875	-0.379695998223867 + 0.008967964875470i
5	0.5	0.349385621484342 + 0.054126587736527i
6	0.29296875	-0.029740276953096 - 0.072397559879213i
7	0	0
8	-0.048828125	-0.032087361783393 + 0.013079942373205i
9	0	0
10	0.005859375	$0.000634295950037 + 0.002825770851695i$

Table 2: Coefficients of the directional tight framelet filter bank in Example 4.3.3

there is still energy left elsewhere. This makes its directionality in the imaginary parts worse than real parts.

Example 4.2.4. *Let*

$$\begin{aligned} a =&\{0.00069616789827, -0.02692519074183, -0.04145457368921, 0.19056483888762,\\ &0.58422553883170, 0.58422553883170, 0.19056483888762, -0.04145457368921,\\ &-0.02692519074183, 0.00069616789827\}_{[0,9]}. \end{aligned}$$

We have $\mathrm{sr}(a) = 5, c = 9$. *Take* $n_b = 2$, *then there exists* $\mathsf{d}(z) := -0.00612281849z^4 + 0.1917188309z^3 - 0.00612281849z^2$ *such that* $\mathsf{dd}^\star = \det \mathcal{N}_{a|n_b}$.

The coefficients of the directional tight filter bank are given as in Table 3. and $b_2(k) = b_1(9-k)$.

The 1-D refinable functions (ϕ,ψ^1 *and* ψ^2*) are given in Fig 27. The stem figure in 1-D are given in Fig 28. In 2-D, the frequency domain are shown in Fig 29. Generators of 2-D* DAS *are given in Fig 30 and 31.*

This example has much smoother generators and shows perfectly the directions in both real and imaginary part. This is due to a relatively longer support and a smoother scaling function (as shown in Fig 27).

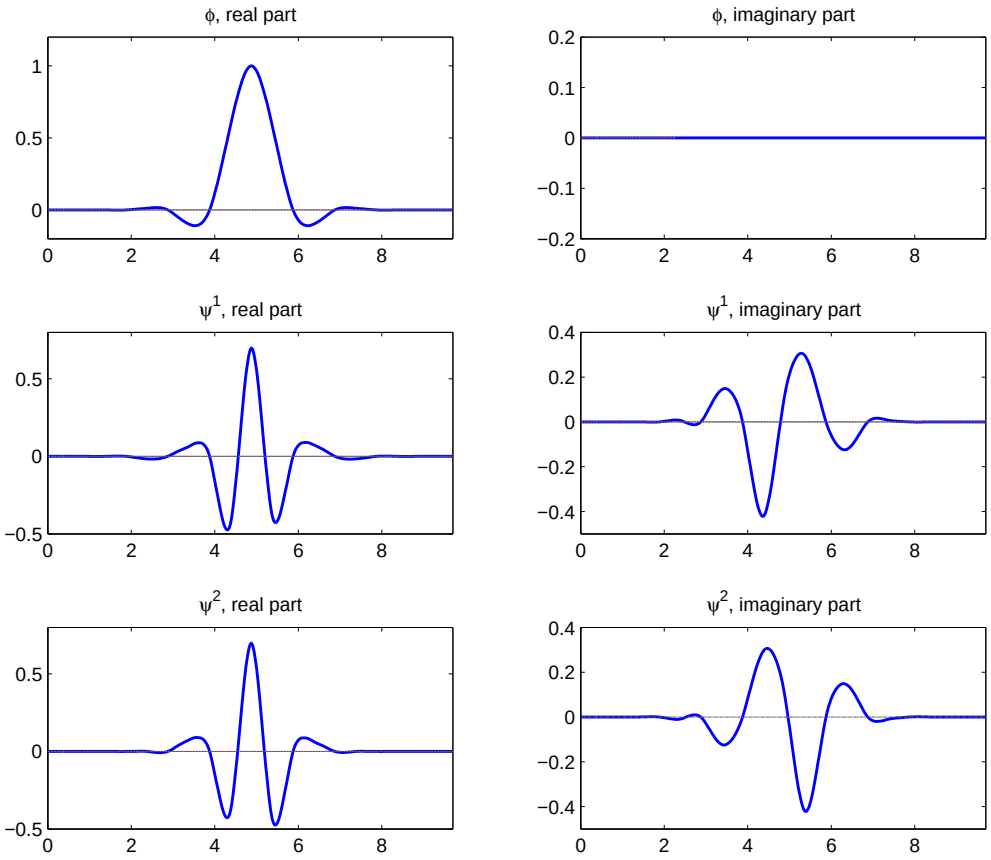


Figure 22: The scaling function ϕ and wavelet functions ψ^1 and ψ^2 of the directional tight framelet filter bank in Example 4.3.3.

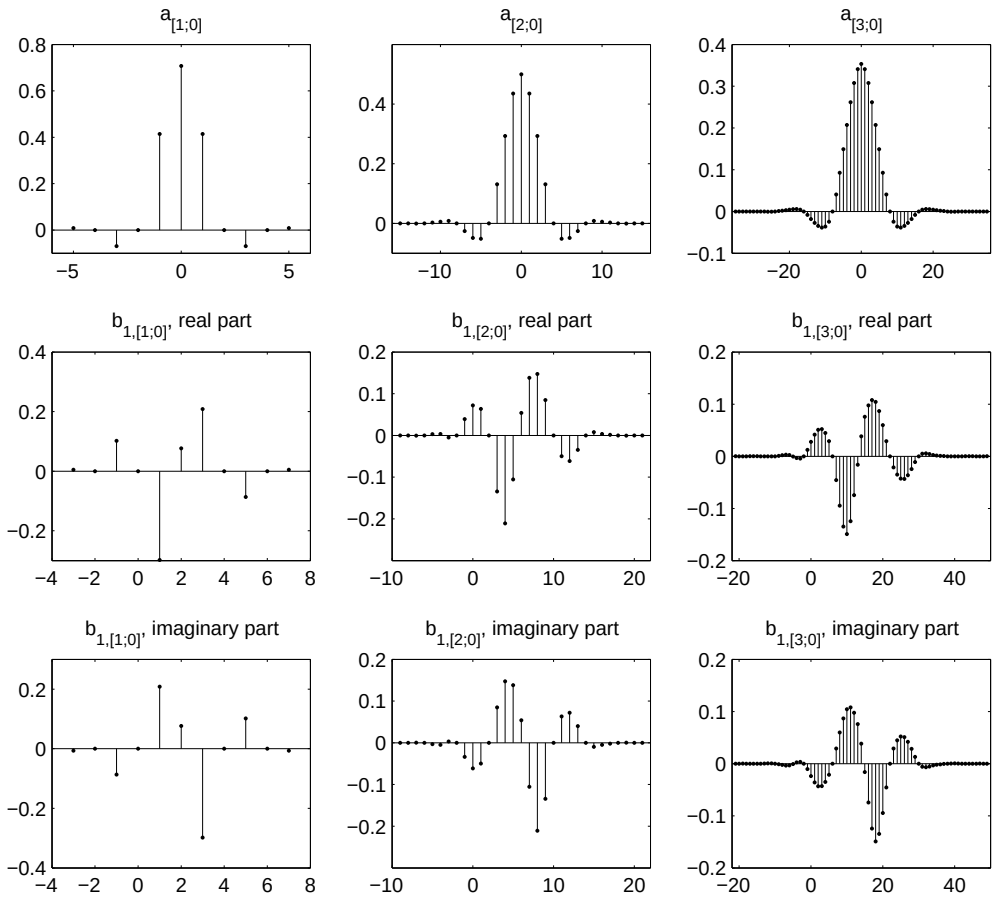
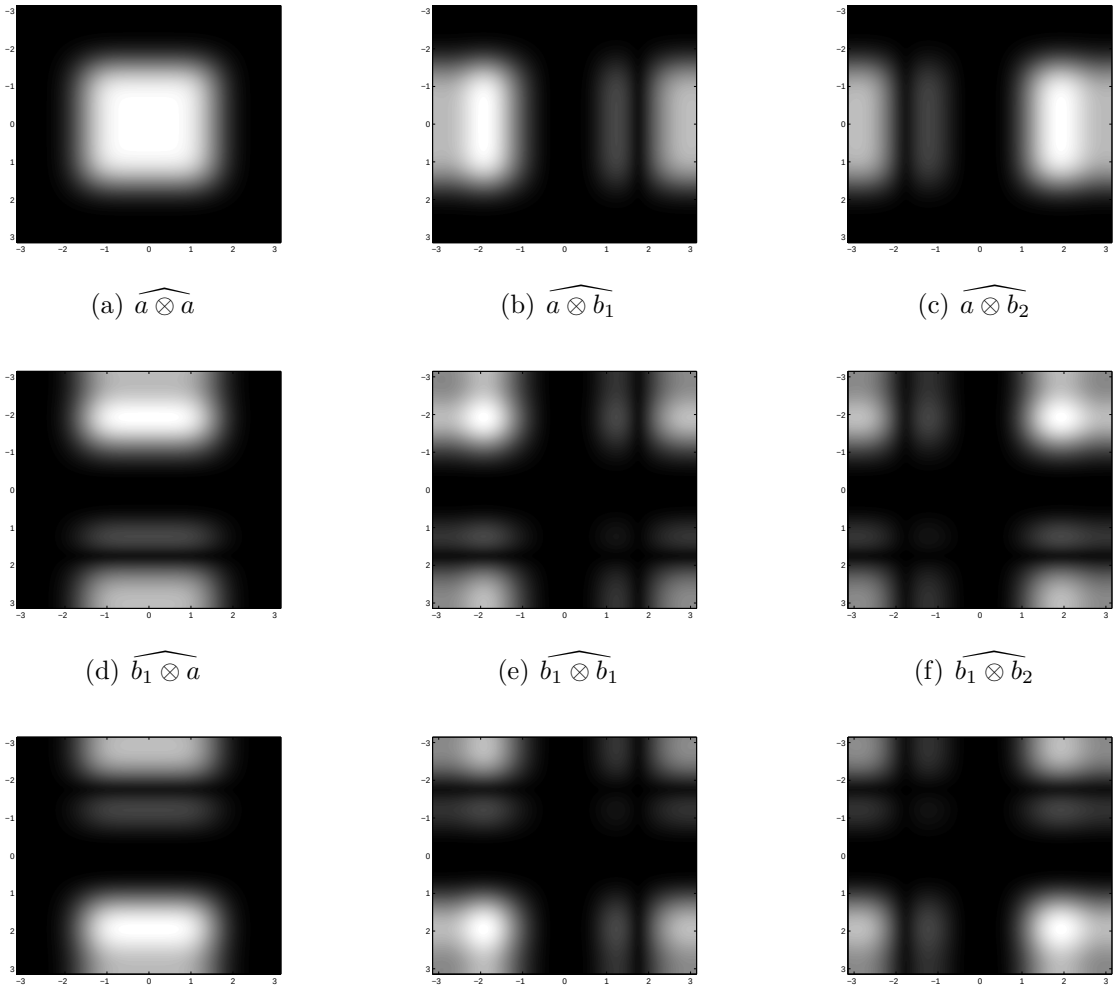


Figure 23: Stem plot of the generators of the 1-D DAS of the directional tight framelet filter bank in Example 4.3.3.



(a) $\widehat{a \otimes a}$ (b) $\widehat{a \otimes b_1}$ (c) $\widehat{a \otimes b_2}$

(d) $\widehat{b_1 \otimes a}$ (e) $\widehat{b_1 \otimes b_1}$ (f) $\widehat{b_1 \otimes b_2}$

(g) $\widehat{b_2 \otimes a}$ (h) $\widehat{b_2 \otimes b_1}$ (i) $\widehat{b_2 \otimes b_2}$

Figure 24: Grey-scale plot of the 2-D directional filter bank in frequency domain (Example 4.3.3), obtained from tensor product.

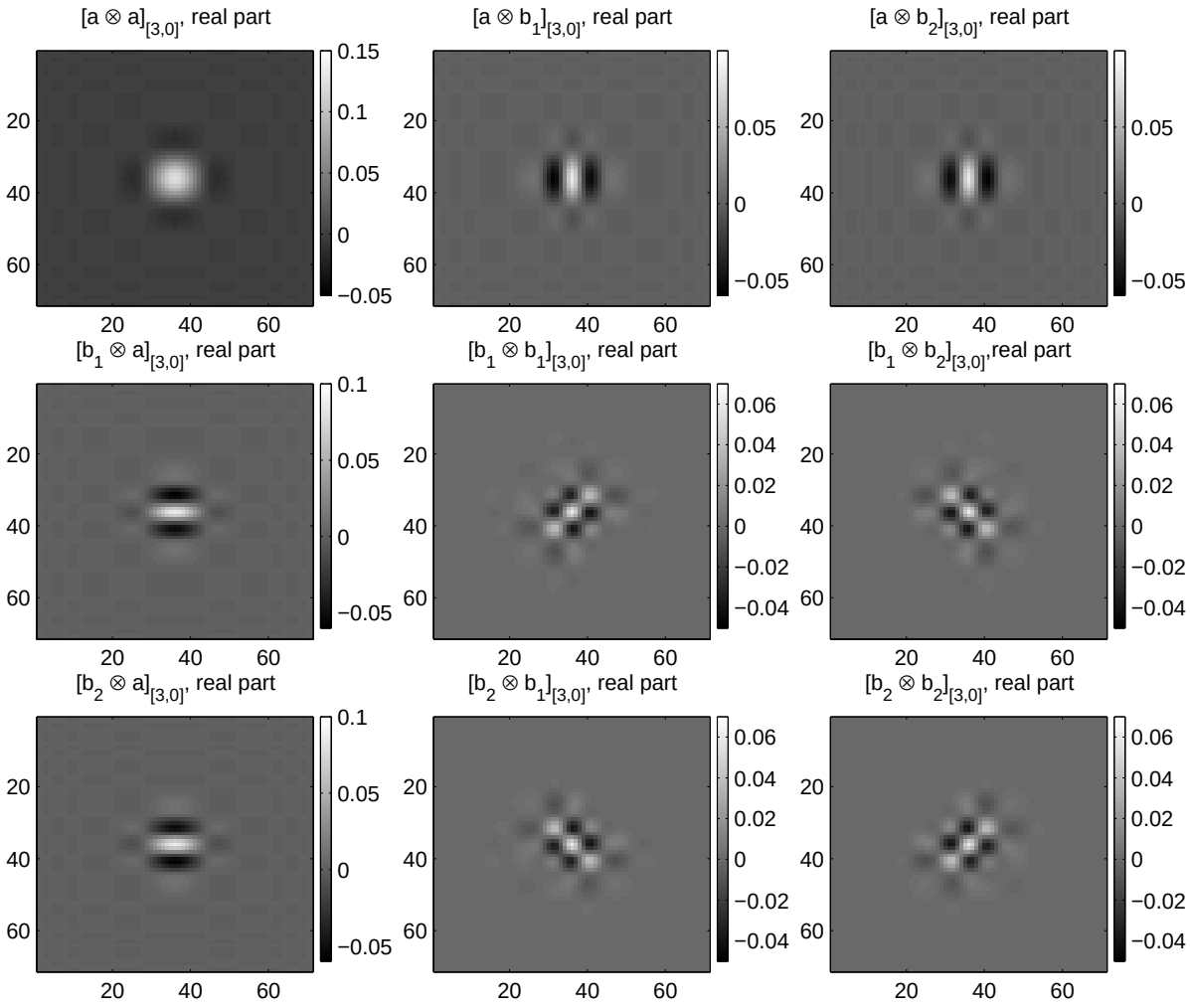


Figure 25: The generators of 2-D DAS for the directional tight framelet filter bank in Example 4.3.3 on $\mathbb{Z}^2$, real part (level=3).

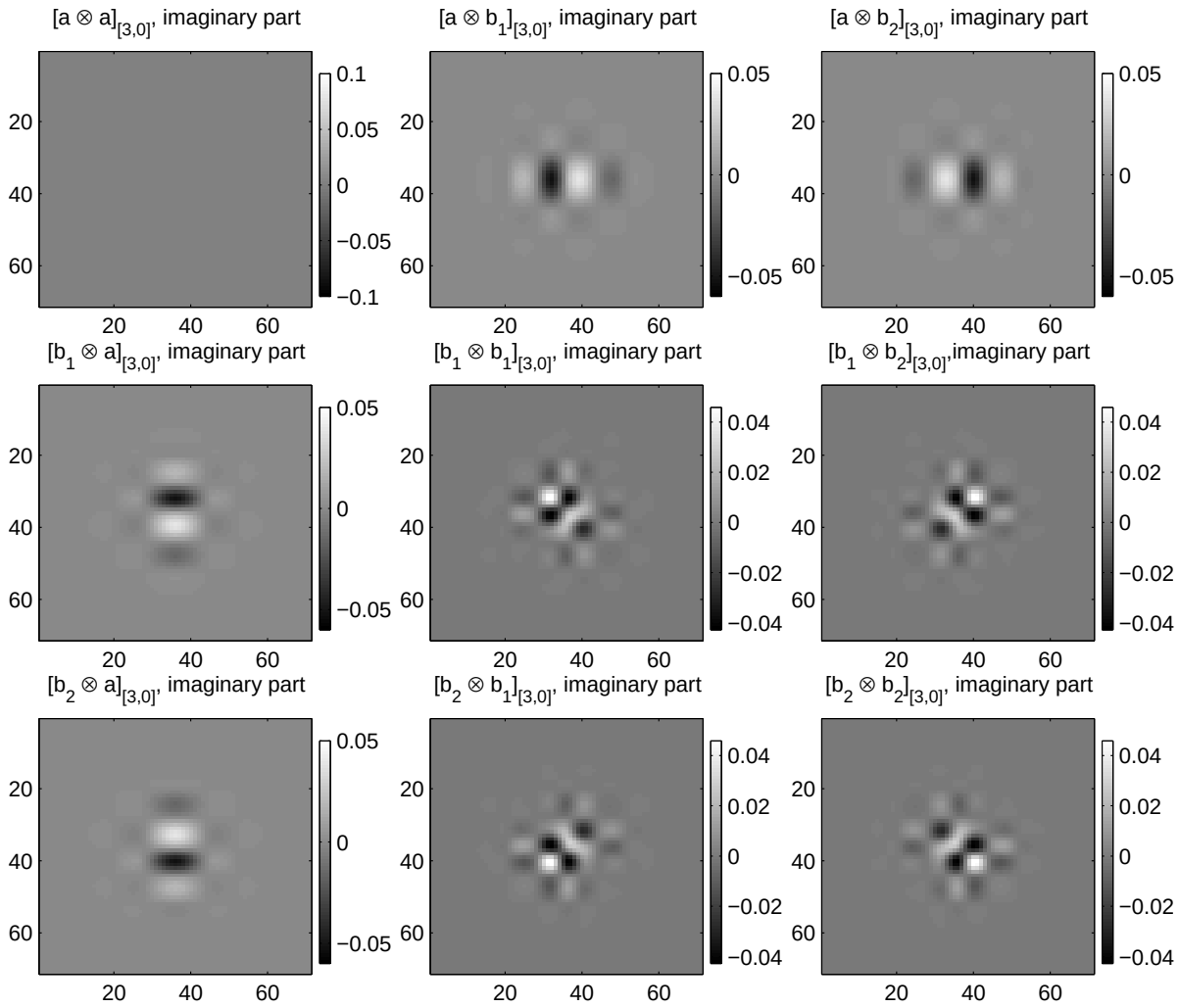


Figure 26: The generators of 2-D DAS for the directional tight framelet filter bank in Example 4.3.3 on $\mathbb{Z}^2$, imaginary part (level=3).

k	$a(k)$	$b_1(k)$
0	0.00069616789827	0.000703645838835 + 0.000502784839885i
1	-0.02692519074183	-0.027214409731390 - 0.019445851711510i
2	-0.04145457368921	-0.035989236107445 - 0.021667328937135i
3	0.19056483888762	-0.035989236107445 - 0.182297139143435i
4	0.58422553883170	0.472081944429775 + 0.222907534952195i
5	0.58422553883170	-0.472081944429775 + 0.222907534952195i
6	0.19056483888762	0.035989236107445 - 0.182297139143435i
7	-0.04145457368921	0.035989236107445 - 0.021667328937135i
8	-0.02692519074183	0.027214409731390 - 0.019445851711510i
9	0.00069616789827	-0.000703645838835 + 0.000502784839885i

Table 3: Coefficients of the directional tight framelet filter bank in Example 4.3.4

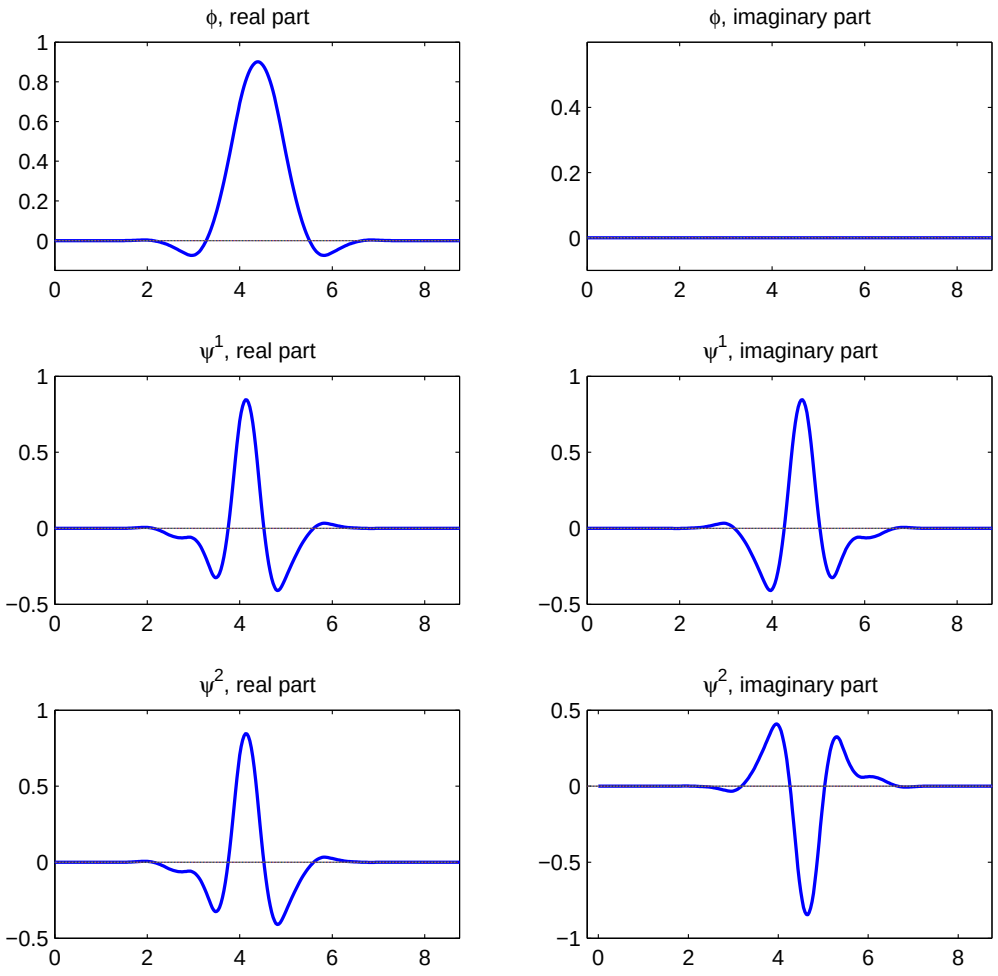


Figure 27: The scaling function ϕ and wavelet functions ψ^1 and ψ^2 of the directional tight framelet filter bank in Example 4.3.4.

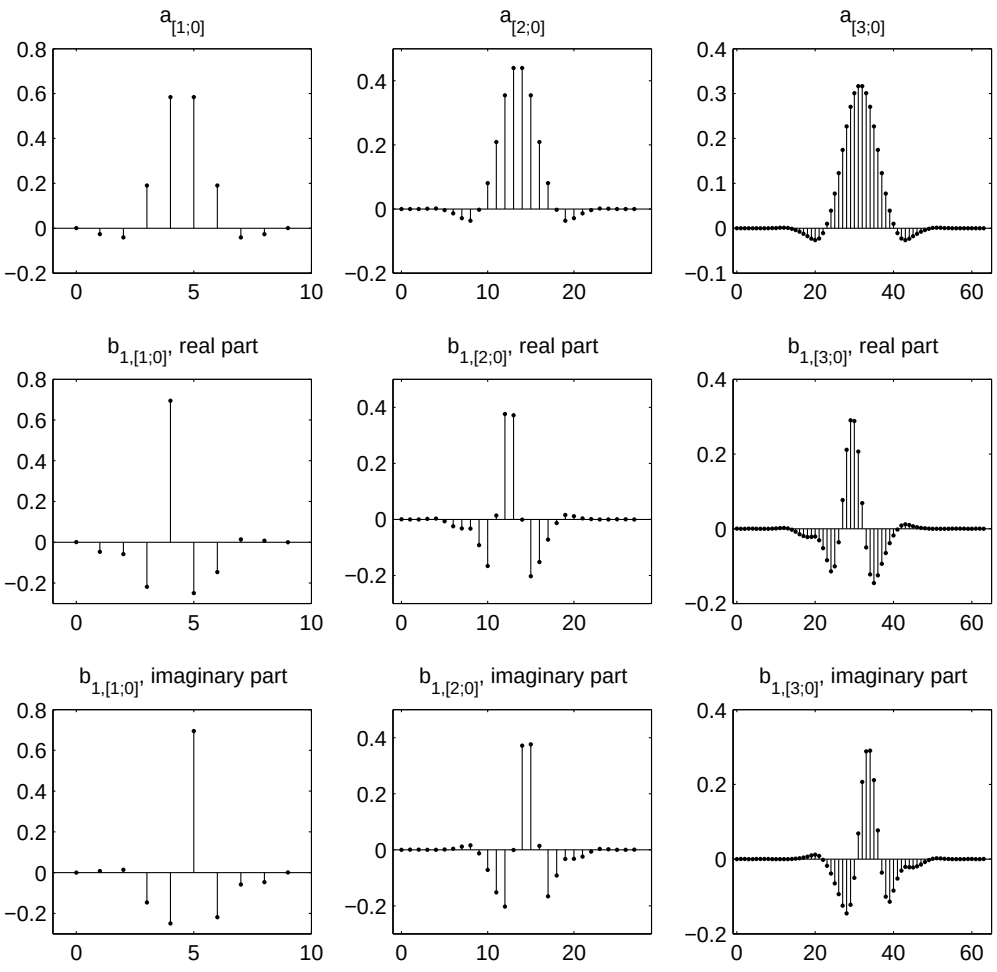
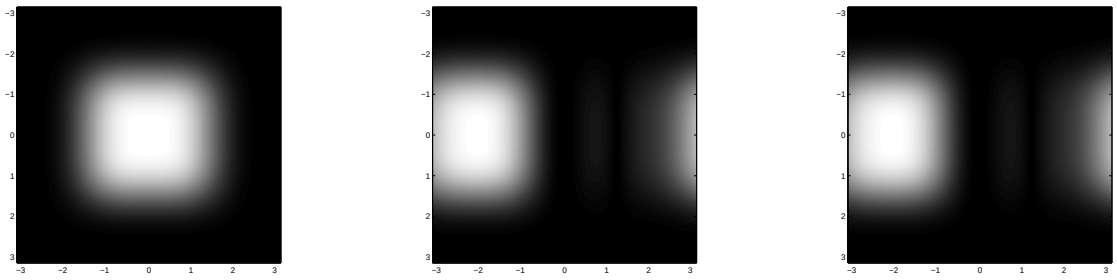
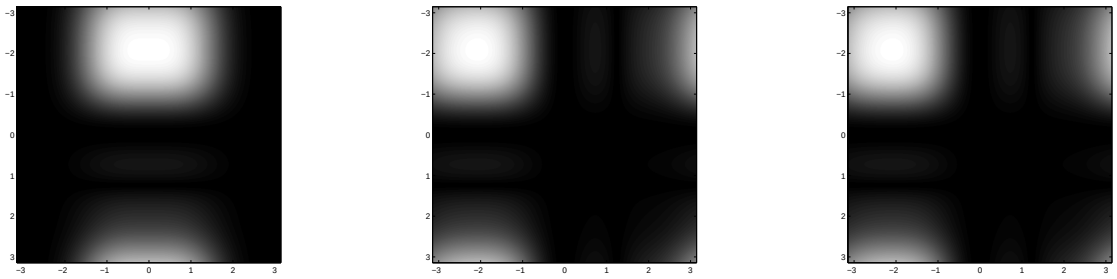


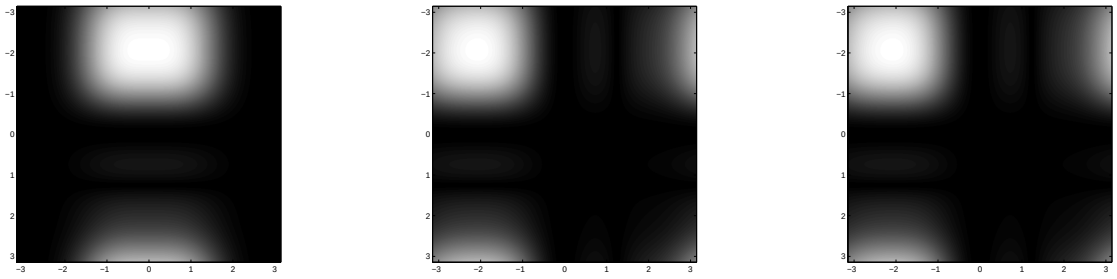
Figure 28: Stem plot of the generators of the 1-D DAS of the directional tight framelet filter bank in Example 4.3.4.



(a) $\widehat{a} \otimes \widehat{a} = \widehat{a \otimes a}$ (b) $\widehat{a \otimes b_1}$ (c) $\widehat{a \otimes b_2}$



(d) $\widehat{b_1 \otimes a}$ (e) $\widehat{b_1 \otimes b_1}$ (f) $\widehat{b_1 \otimes b_2}$



(g) $\widehat{b_2 \otimes a}$ (h) $\widehat{b_2 \otimes b_1}$ (i) $\widehat{b_2 \otimes b_2}$

Figure 29: Grey-scale plot of the 2-D directional filter bank in frequency domain (Example 4.3.4), obtained from tensor product.

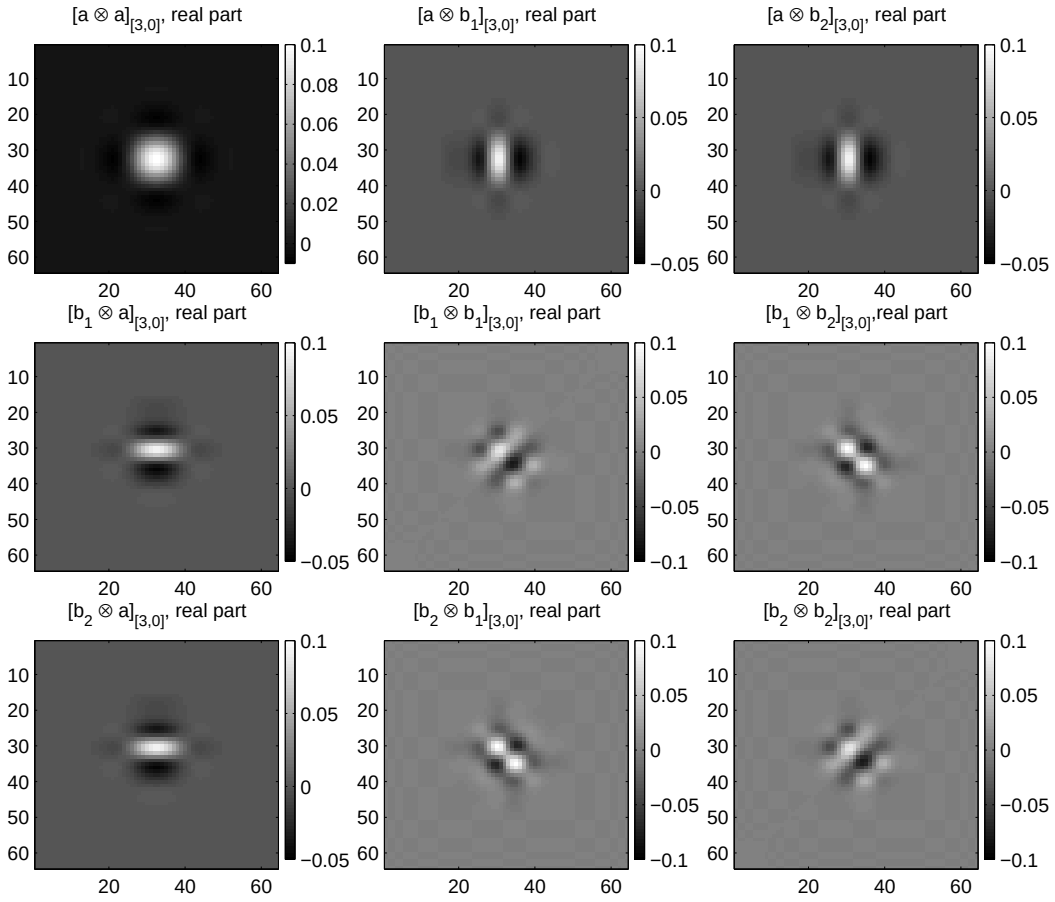


Figure 30: The generators of 2-D DAS for the directional tight framelet filter bank in Example 4.3.4 on $\mathbb{Z}^2$, real part (level=3).

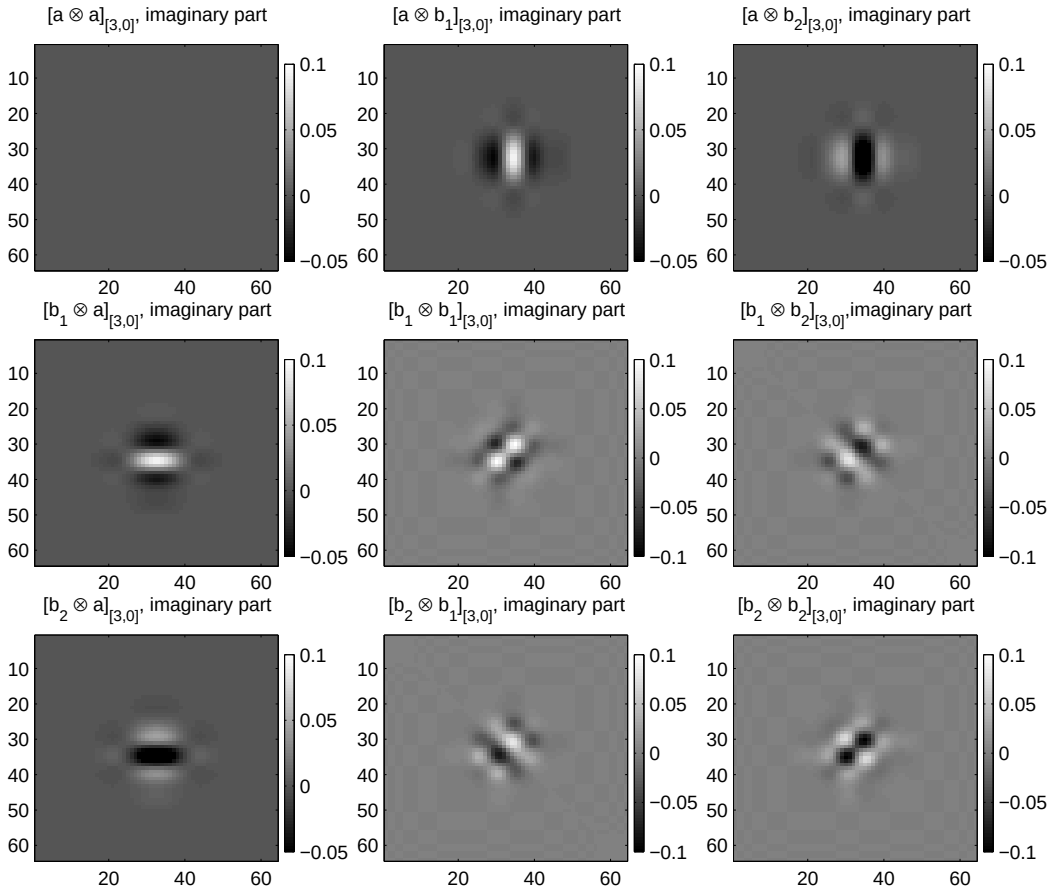


Figure 31: The generators of 2-D DAS for the directional tight framelet filter bank in Example 4.3.4 on $\mathbb{Z}^2$, imaginary part (level=3).

Chapter 5

Summary and Future Work

To capture line-shaped singularities in images, the discrete framelet transform has many preferred properties compared to other wavelet-based image processing tools. It-s filter bank structure makes it easy to implement and also computationally economic. Meanwhile, the redundancy of high-pass filters provides possibility to obtain directionality. In this thesis, we study the construction of finitely supported directional tight framelet filter banks.

We use a frequency-design strategy to construct 2-D directional framelet filter banks via tensor product. Because 2-D Fourier transform has a nice property, that is, a directional singularity is kept through this transform, only rotated by an angle of 90°, we look for a filter bank which has a good frequency separation, so that each of the tensor products is concentrated at a certain quadrant in the frequency domain, thus result in diagonal directions in time domain.

Given a symmetric low-pass filter, we get directional tight framelet filter banks from symmetric ones through a unitary transform. Theoretically, we give a necessary and sufficient condition for the existence of such filter banks. The explicit expression of a constant unitary transform is provided, which makes the best frequency separation and yields good directionality.

We then give a specific algorithm to construct such tight framelet filter banks with finite support. Examples with 4 directions $(0°, 45°, 90°, 135°)$ are provided. Some of them have very clear directionality (Figure 20, Figure 31), while some are not so good.

The main defect of our results is that the directionality of some filter banks is not satisfactory. There are several reasons possibly explaining for this problem. One is

the length of filter support. In our algorithm we seek for filters with shortest support. This reduces the freedom hence even we get the optimal frequency separation, it is still not quite ideal (see Figure 14). Finally the refinable functions in time domain do not have a perfect directionality. The other one is the essence of the low-pass filter. If the low-pass filter itself is not smooth enough, the high-pass filters can never have smoothness (Figure 4.2).

The tensor product strategy has its own advantages as well as drawbacks. Compared to the direct construction of 2-D (tight) framelet filter banks, construction via tensor product reduces the problem into 1-D case, which is the most studied. This enables us benefit from a lot of known theoretical results, for example, [8] and [6]. However its limitation is also clear: it is only feasible for relatively less directions. In this thesis, we deal with the tensor product of framelet filter banks with 2 high-pass filters. The tensor products have 4 different directions in 2-D. If we want more directions for better image processing performance, we naturally need more generators. It is an advantage of tensor product strategy, meaning that we have more flexibility to choose freely how many directions we need. But at the same time, the more generators we have, the higher complexity we will encounter. If we want more than 100 directions, the complexity can be astonishing.

We hereby give some future directions we shall work on.

The first is to extend the support of the filters. In our algorithm we calculate the high-pass filter with the shortest support, but indeed we can have longer supports. In Chapter 2 we used a constant matrix A for the construction. However, if A is not a constant, i.e., a unitary matrix of polynomials then we have more degrees of freedom and could promisingly get better directionality.

The second possibility is to construct "better" low-pass filters. In this thesis we always presume that we have known the low-pass filter. Nonetheless we can see that the low-pass filter plays a fatal role as the properties of high-pass filters highly rely on it. For example, the vanishing moments of high-pass filters are restricted by the sum

rule of a. So we can focus on finding better low-pass filters. It can also be seen from our examples. In Example 2, because the low-pass filter has jumps in the time domain, the high-pass then always have block-like effects. So if we can find a smooth enough symmetric filter, we are more likely to find a filter bank with better directionality.

For a more general topic, the construction of directional tight framelets, we have more strategies to get better directionality. A possible future work is to increase the number of generators. We only deal with filter banks with 2 generators here. But in fact framelet filter banks are quite flexible with the number of high-pass filters. It can be modified to satisfy different demands. When we pursue for less computational cost, we can have less generators to make less redundancy. But when we need better performance and pay more attention to effectiveness, we can have more generators, or redundancy to help get more directions and therefore improve the accuracy.

Bibliography

[1] E. J Candes and D. L Donoho, *Continuous curvelet transform: I. resolution of the wavefront set*, Applied and Computational Harmonic Analysis **19** (2005), no. 2, 162–197.

[2] ———, *Continuous curvelet transform: II. discretization and frames*, Applied and Computational Harmonic Analysis **19** (2005), no. 2, 198–222.

[3] A. S Cavaretta, W. Dahmen, and C. A Micchelli, *Stationary subdivision*, no. 453, Oxford University Press, 1991.

[4] A. Croisier, D. Esteban, and C. Galand, *Perfect channel splitting by use of interpolation-decimation tree decomposition techniques*, International Conference on Information Sciences and Systems, vol. 2, Patras, Greece, 1976, pp. 443–446.

[5] I. Daubechies, B. Han, A. Ron, and Z. Shen, *Framelets: MRA-based constructions of wavelet frames*, Applied and Computational Harmonic Analysis **14** (2003), no. 1, 1–46.

[6] B. Han, *Framelets and wavelets: algorithms, analysis and applications*, Ongoing book manuscript.

[7] ———, *Properties of discrete framelet transforms*, Mathematical Modelling of Natural Phenomena **8** (2013), no. 1, 18–47.

[8] B. Han and Q. Mo, *Splitting a matrix of laurent polynomials with symmetry and its application to symmetric framelet filter banks*, SIAM journal on matrix analysis and applications **26** (2004), no. 1, 97–124.

[9] N. G Kingsbury, *The dual-tree complex wavelet transform: a new technique for*

shift invariance and directional filters, Proc. 8th IEEE DSP Workshop, vol. 8, Citeseer, 1998, p. 86.

[10] J. Ma and G. Plonka, *The curvelet transform*, Signal Processing Magazine, IEEE **27** (2010), no. 2, 118–133.

[11] S. G Mallat, *Multiresolution approximations and wavelet orthonormal bases*, Transactions of the American Mathematical Society **315** (1989), no. 1, 69–87.

[12] ———, *A theory for multiresolution signal decomposition: the wavelet representation*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **11** (1989), no. 7, 674–693.

[13] ———, *A wavelet tour of signal processing*, Academic press, 1999.

[14] Y. Meyer and D. H Salinger, *Wavelets and operators*, vol. 37, Cambridge Univ Press, 1992.

[15] I. W Selesnick, R. G Baraniuk, and N. C Kingsbury, *The dual-tree complex wavelet transform*, Signal Processing Magazine, IEEE **22** (2005), no. 6, 123–151.

[16] J. Starck, E. J Candès, and D. L Donoho, *The curvelet transform for image denoising*, Image Processing, IEEE Transactions on **11** (2002), no. 6, 670–684.

[17] M. Vetterli and C. Herley, *Wavelets and filter banks: Theory and design*, Signal Processing, IEEE Transactions on **40** (1992), no. 9, 2207–2232.