

Multidimensional Filter Banks and Multiscale Geometric Representations

By Minh N. Do and Yue M. Lu

Contents

1	Introduction	159
2	Preliminaries: Multidimensional Signals, Transforms, and Filtering	164
2.1	Multidimensional Fourier and z -Transforms	164
2.2	Multidimensional Filters	166
3	Multidimensional Sampling	170
3.1	Sampling on Lattices	170
3.2	The Effect of Sampling in the Fourier Domain	172
3.3	Downsampling and Upsampling of Discrete Signals	175
3.4	Key Properties of Sampling Lattices	178
4	Multidimensional Filter Banks	181
4.1	Filter Banks: from Frequency Decomposition to Signal Representation	181
4.2	Basic Tools: Multirate Identities and Polyphase Representations	184
4.3	Perfect Reconstruction Filter Banks	190
4.4	Example: Two-Channel Filter Banks in 2D	193

5	Characterization and Design of Multidimensional Filter Banks	199
5.1	Characterizing MD Filter Banks Using Gröbner Bases	199
5.2	The Mapping-Based Design for MD Filter Banks	204
5.3	Designing Filter Banks in the Frequency Domain	213
6	Iterated and Directional Filter Banks	216
6.1	Directional Filter Banks	216
6.2	Directional Filter Banks in Higher Dimensions	220
7	Multiscale Geometric Representations	227
7.1	The Contourlet and Surfacelet Transforms	227
7.2	Multiresolution Directional Analysis	233
7.3	Other Multiscale Geometric Representations	237
8	Example Applications	242
8.1	Signal Decomposition and Nonlinear Approximation	242
8.2	Image and Video Denoising	245
8.3	Edge-Preserving Image Interpolation	249
8.4	Compressed Sensing	252
	Conclusions	255
	Acknowledgments	257
	References	258

Multidimensional Filter Banks and Multiscale Geometric Representations

Minh N. Do¹ and Yue M. Lu²

¹ *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA, minhdo@illinois.edu*

² *School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA, yuelu@seas.harvard.edu*

Abstract

Thanks to the explosive growth of sensing devices and capabilities, multidimensional (MD) signals — such as images, videos, multispectral images, light fields, and biomedical data volumes — have become ubiquitous. Multidimensional filter banks and the associated constructions provide a unified framework and an efficient computational tool in the formation, representation, and processing of these multidimensional data sets. In this survey we aim to provide a systematic development of the theory and constructions of multidimensional filter banks. We thoroughly review several tools that have been shown to be particularly effective in the design and analysis of multidimensional filter banks, including sampling lattices, multidimensional bases and frames, polyphase representations, Gröbner bases, mapping methods, frequency domain constructions, ladder structures and lifting schemes. We then focus on the construction of filter banks and signal representations that can capture directional and geometric features, which are unique and key properties of many multidimensional signals. Next,

we study the connection between iterated multidimensional filter banks in the discrete domain and the associated multiscale signal representations in the continuous domain through a directional multiresolution analysis framework. Finally, we show several examples to demonstrate the power of multidimensional filter banks and geometric signal representations in applications.

1

Introduction

Multidimensional (MD) signals are information-carrying physical quantities that depend on several variables, each representing a unique dimension. For example, a video is a three-dimensional (3D) signal with two spatial dimensions (horizontal and vertical) and one temporal dimension. A particularly important and common class of MD signals contains *visual information*, ranging from general images and videos on the Web to special medical images (such as MRI and CT scans) for diagnostics, and from very small scales (molecular images) to very large scales (astronomical images).

Efficient representation of visual information lies at the heart of many image processing tasks such as reconstruction, denoising, compression, and feature extraction. For example, a 512 by 512 color image can be considered as a vector in a $512 \times 512 \times 3$ dimensional space (each pixel is represented by a triplet of color components). However, as we can see in Figure 1.1, a randomly sampled image from this space is far from being a *natural image*. In other words, natural images occupy a very small fraction of the huge space of all possible images. Effectively exploring this fact allows us to efficiently compress an image or to separate a clean image from noise.

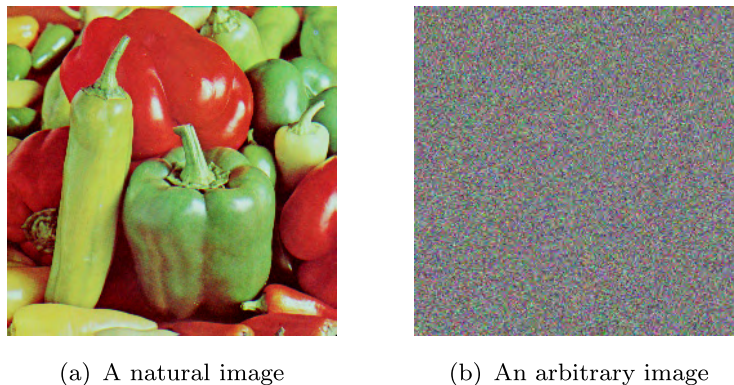


Fig. 1.1 Example of a natural image (a) compared with an arbitrary image (b) that is sampled from the same image space.

As can be seen from Figure 1.1, a key distinguishing feature of natural images is that they have *intrinsic geometric structures*. In particular, visual information is mainly contained in the geometry of object boundaries or edges. For this reason, *wavelets and filter banks* [21, 65, 89, 95, 99] — a breakthrough resulting from the convergence of ideas from several fields — have been found to be particularly well-suited for representing images. In particular, wavelets are good at isolating the discontinuities at *edge points*. However, as a result of their construction by separable extension from 1D bases, wavelets in 2D cannot “see” the *smoothness along the contours*. In addition, separable wavelets can capture only limited *directional* information, which is an important and unique feature of MD signals.

To see how one can improve the 2D separable wavelet transform in representing images with smooth contours, consider the following scenario. Imagine that there are two painters, one with a wavelet-style and the other with a new style, both wishing to paint a natural scene. Both painters apply a refinement technique to increase the resolution from coarse to fine. We consider efficiency as measured by how quickly, that is with how few brush strokes, each painter can faithfully reproduce the scene. In other words, an *efficient* painting style is associated with a *sparse* image representation scheme.

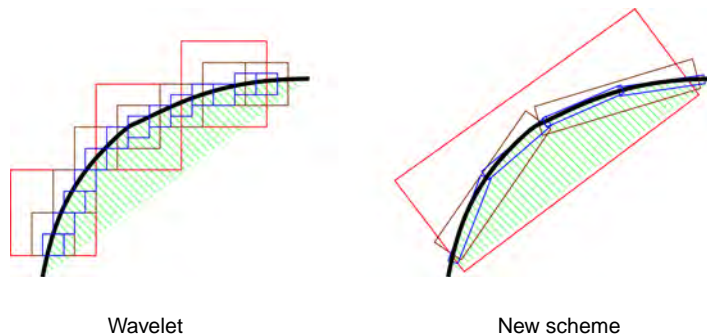


Fig. 1.2 Wavelet versus the new scheme: illustrations of different successive refinement styles by the two systems near a smooth contour, which is shown as a thick curve separating two smooth regions.

Consider the situation when a smooth contour is being painted, as shown in Figure 1.2. Because 2D wavelets are constructed from tensor products of 1D wavelets, the wavelet-style painter is limited to using square-shaped brush strokes along the contour, using different sizes corresponding to the multiresolution structure of wavelets. As the resolution becomes finer, we can clearly see the limitation of the wavelet-style painter who needs to use many fine “dots” to capture the contour.¹ The new style painter, on the other hand, effectively exploits the smoothness of the contour by making brush strokes with different elongated shapes and in a variety of directions following the contour. This intuition was first formalized by Candès and Donoho in the *curvelet* construction [7, 9]. We will also see later an actual realization of the new scheme with the *contourlet* transform in Figure 8.2.

For the human visual system, it is well-known [44] that the receptive fields in the visual cortex are characterized as being *localized*, *oriented*, and *bandpass*. Furthermore, computational experiments in searching for the sparse components of (both still and time-varying) natural images produced basis images that closely resemble the aforementioned characteristics of the visual cortex [72, 73]. These results support the hypothesis that the human visual system has been tuned so as to capture the

¹Or we could consider the wavelet-style painter as a *pointillist*!

essential information of a natural scene using a least number of active visual cells. More importantly, the results suggest that, for a computational image representation to be efficient, it should be based on a *local*, *directional*, and *multiresolution* expansion.

Over the past decade, a number of concurrent studies in applied mathematics, computer vision, and statistical learning theory have independently developed theories and tools to explore and make use of the geometric structures in multidimensional data. In signal processing, the challenges as well as great research opportunities come from the discrete nature of the data, together with the issues of robustness, efficiency, and speed. For example, directions other than horizontal and vertical can look very different on a rectangular grid typically used to sample images. Because of pixelation, the notion of smooth contours on sampled images is not obvious. Moreover, for practical applications, efficient representation has to be obtained by structured transforms and fast algorithms.

Thus, we are particularly interested in a *discrete-space* framework for the construction of multiscale geometric transforms that can be applied to sampled images and MD signals. Following the success of wavelets and filter banks in 1D, we will focus on the constructions using multidimensional filter banks. However, as mentioned above, the commonly used wavelets and filter banks in MD are simply constructed from separable extensions of their 1D counterparts. Here, we want to exploit the full flexibility of true (non-separable) MD constructions in order to achieve the desired multiscale directional and geometric transforms and representations.

Toward this goal, we first provide a thorough review of the theory and design of multidimensional filter banks in this survey. While there are already several excellent papers and reviews on MD filter banks (see, for example, [14, 49, 57, 100]), our review emphasizes MD filter banks as basis and frame expansions for signal representations, in addition to the traditional view of achieving good frequency partitions. Moreover, we will highlight some modern and effective tools for designing MD filter banks such as Gröbner bases, mapping methods, frequency domain constructions, and ladder structures and lifting schemes. We believe that this MD filter bank review will be useful in its own right. Building

upon this background, we then present constructions of iterated and directional filter banks leading to multiscale geometric representations for MD signals, both in discrete and continuous domains. The effectiveness of these constructions will be demonstrated through applications and numerical results.

The outline of this survey is as follows. In Section 2, we define our notation and study the first building block of multidimensional filter bank, namely, MD filtering. In Section 3, we study the other building block: MD sampling. The generalization of sampling from 1D to MD using lattices provides a rich set of new possibilities that will be exploited in later constructions of directional and geometric representations. Section 4 combines these two building blocks into a systematic study of MD filter banks. In particular, we focus on those filter banks that satisfy the perfect reconstruction condition, which lead to bases or frames for MD signal representations. Section 5 presents some of the most effective tools for characterizing and designing MD filter banks. In Section 6, we study the iterated and directional filter banks that are obtained by well-designed combinations of the building blocks for MD filter banks. Based on this directional construction, we present multiscale geometric transforms in Section 7. Moreover, we establish a precise connection between iterated MD filter banks in the discrete domain and the associated multiscale signal representations in the continuous domain through a directional multiresolution analysis framework. Finally, Section 8 illustrates some applications in image and MD signal processing, demonstrating the power of the constructed MD filter banks and signal representations.

2

Preliminaries: Multidimensional Signals, Transforms, and Filtering

In this section, we establish our notation and introduce multidimensional (MD) filtering, which is the first building block of MD filter banks.

2.1 Multidimensional Fourier and z -Transforms

A d -dimensional *continuous-domain* signal $x(\mathbf{t})$ has values defined on \mathbb{R}^d ; that is, $\mathbf{t} = [t_1, t_2, \dots, t_d]^T \in \mathbb{R}^d$. All such signals having finite energy belong to the *Hilbert space* $L_2(\mathbb{R}^d)$. The inner product of any two signals $x(\mathbf{t})$ and $y(\mathbf{t})$ is defined as

$$\langle x, y \rangle_{L_2} \stackrel{\text{def}}{=} \int_{\mathbf{t} \in \mathbb{R}^d} x(\mathbf{t}) y^*(\mathbf{t}) d\mathbf{t}, \quad (2.1)$$

where $y^*(\mathbf{t})$ is the complex conjugate of $y(\mathbf{t})$. The induced L_2 -norm is then $\|x\|_{L_2} = \sqrt{\langle x, x \rangle} = (\int_{\mathbf{t} \in \mathbb{R}^d} |x(\mathbf{t})|^2 d\mathbf{t})^{1/2}$.

The (continuous-domain) *Fourier transform* of $x(\mathbf{t})$ is defined as

$$X(\boldsymbol{\xi}) \stackrel{\text{def}}{=} \int_{\mathbb{R}^d} x(\mathbf{t}) e^{-j2\pi(\xi_1 t_1 + \xi_2 t_2 + \dots + \xi_d t_d)} d\mathbf{t} = \int_{\mathbb{R}^d} x(\mathbf{t}) e^{-j2\pi \boldsymbol{\xi}^T \mathbf{t}} d\mathbf{t}. \quad (2.2)$$

Correspondingly, the inverse Fourier transform is

$$x(\mathbf{t}) = \int_{\mathbb{R}^d} X(\boldsymbol{\xi}) e^{j2\pi \boldsymbol{\xi}^T \mathbf{t}} d\boldsymbol{\xi}. \quad (2.3)$$

With this definition, the frequency variable ξ_i has the reciprocal unit of the corresponding spatial variable t_i . For example, if t_i is measured in [cm] then ξ_i is measured in [cm^{-1}].

The Fourier transform pairs defined in (2.2) and (2.3) are *unitary transformations* on $L^2(\mathbb{R}^d)$. Consequently, we have Parseval's identity

$$\int_{\mathbf{t} \in \mathbb{R}^d} x(\mathbf{t}) y^*(\mathbf{t}) d\mathbf{t} = \int_{\boldsymbol{\xi} \in \mathbb{R}^d} X(\boldsymbol{\xi}) Y^*(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2.4)$$

In particular, when $y(\mathbf{t}) = x(\mathbf{t})$, the above equality becomes:

$$\|x(\mathbf{t})\|_{L_2}^2 = \|X(\boldsymbol{\xi})\|_{L_2}^2,$$

that is, the Fourier transform preserves the energy of the original signal.

The above discussions carry over to *discrete-domain* signals. Let $x[\mathbf{n}]$, $\mathbf{n} = [n_1, n_2, \dots, n_d]^T \in \mathbb{Z}^d$, denote a d -dimensional discrete-domain signal. All such signals having finite energy form the Hilbert space $l_2(\mathbb{Z}^d)$, with inner product

$$\langle x, y \rangle_{l_2} \stackrel{\text{def}}{=} \sum_{\mathbf{n} \in \mathbb{Z}^d} x[\mathbf{n}] y^*[\mathbf{n}], \quad (2.5)$$

and l_2 -norm $\|x\|_{l_2} = \sqrt{\langle x, x \rangle} = (\sum_{\mathbf{n} \in \mathbb{Z}^d} |x[\mathbf{n}]|^2)^{1/2}$.

We can define the z -transform of $x[\mathbf{n}]$ as

$$X(\mathbf{z}) = \mathcal{Z}\{x[\mathbf{n}]\} \stackrel{\text{def}}{=} \sum_{\mathbf{n} \in \mathbb{Z}^d} x[\mathbf{n}] \mathbf{z}^{-\mathbf{n}},$$

where, by following the convenient MD vector notation introduced in [100], raising a d -dimensional complex vector $\mathbf{z} = [z_1, \dots, z_d]^T$ to a d -dimensional integer vector $\mathbf{n} = [n_1, \dots, n_d]^T$ yields

$$\mathbf{z}^{\mathbf{n}} \stackrel{\text{def}}{=} \prod_{i=1}^d z_i^{n_i}. \quad (2.6)$$

On the unit hyper-sphere, $\mathbf{z} = e^{j2\pi \boldsymbol{\xi}} \stackrel{\text{def}}{=} [e^{j2\pi \xi_1}, \dots, e^{j2\pi \xi_d}]^T$, and the z -transform becomes the (discrete-domain) Fourier transform:

$$X(e^{j2\pi \boldsymbol{\xi}}) = \sum_{\mathbf{n} \in \mathbb{Z}^d} x[\mathbf{n}] e^{-j2\pi \boldsymbol{\xi}^T \mathbf{n}}. \quad (2.7)$$

In this survey, for simplicity and by a slight abuse of notation, we write $X(\boldsymbol{\xi})$ instead of $X(e^{j2\pi\boldsymbol{\xi}})$ for the Fourier transform of $x[\mathbf{n}]$. In other words, depending on whether the variable \mathbf{z} or $\boldsymbol{\xi}$ is used, X denotes either the z transform or the Fourier transform. The inverse discrete-domain Fourier transform is

$$x[\mathbf{n}] = \int_{[-\frac{1}{2}, \frac{1}{2}]^d} X(\boldsymbol{\xi}) e^{j2\pi\boldsymbol{\xi}^T \mathbf{n}} d\boldsymbol{\xi}. \quad (2.8)$$

Similar to its continuous-domain counterpart, the discrete-domain Fourier transform has Parseval's identity in the form of

$$\sum_{\mathbf{n} \in \mathbb{Z}^d} x[\mathbf{n}] y^*[\mathbf{n}] = \int_{[-\frac{1}{2}, \frac{1}{2}]^d} X(\boldsymbol{\xi}) Y^*(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2.9)$$

2.2 Multidimensional Filters

A (discrete-domain) *filter* is represented by its impulse response $h[\mathbf{n}]$ or transfer function $H(\mathbf{z}) = \mathcal{Z}\{h[\mathbf{n}]\}$. The output of the filter is the *convolution* of the input signal with the impulse response $h[\mathbf{n}]$, that is,

$$y[\mathbf{n}] = (x * h)[\mathbf{n}] \stackrel{\text{def}}{=} \sum_{\mathbf{k} \in \mathbb{Z}^d} x[\mathbf{k}] h[\mathbf{n} - \mathbf{k}]. \quad (2.10)$$

Equivalently, in the z and frequency domains, respectively, we have

$$Y(\mathbf{z}) = X(\mathbf{z})H(\mathbf{z}) \quad \text{and} \quad Y(\boldsymbol{\xi}) = X(\boldsymbol{\xi})H(\boldsymbol{\xi}). \quad (2.11)$$

In the Hilbert space $l_2(\mathbb{Z}^d)$, filtering can be viewed as either an analysis or a synthesis operator. In the *analysis* view, we can rewrite (2.10) as

$$y[\mathbf{n}] = \sum_{\mathbf{k} \in \mathbb{Z}^d} x[\mathbf{k}] \bar{h}^*[\mathbf{k} - \mathbf{n}] = \langle x, \bar{h}[\cdot - \mathbf{n}] \rangle, \quad (2.12)$$

where $\bar{h}[\mathbf{n}] \stackrel{\text{def}}{=} h^*[-\mathbf{n}]$. That is, each sample of the filtering output is the inner product of the input signal and a translated copy of the time-flipped and complex-conjugated filter impulse response. Alternatively, in the *synthesis* view, we can rewrite (2.10) as

$$y = \sum_{\mathbf{k} \in \mathbb{Z}^d} x[\mathbf{k}] h[\cdot - \mathbf{k}]. \quad (2.13)$$

That is, the output signal of the filtering operation is a linear combination of the impulse response h and its translated copies.

A multidimensional filter is called *separable*, if it can be expanded as the tensor product of several 1D filters, that is,

$$h[\mathbf{n}] = \prod_{i=1}^d h_i[n_i],$$

or equivalently, in the transform domain,

$$H(\mathbf{z}) = \prod_{i=1}^d H_i(z_i) \quad \text{and} \quad H(\boldsymbol{\xi}) = \prod_{i=1}^d H_i(\xi_i). \quad (2.14)$$

As an example, we show in Figure 2.1 the frequency domain supports of two ideal filters in 2D. From the definition in (2.14), we can easily verify that the filter shown in Figure 2.1(a) is separable, while the one in Figure 2.1(b) is nonseparable. In the general d -dimensional case, a well-known example of separable filters is the multidimensional Gaussian kernel, defined as

$$g[\mathbf{n}] = ce^{\|\mathbf{n}\|^2/\sigma^2} = ce^{(n_1^2 + n_2^2 + \dots + n_d^2)/\sigma^2},$$

for some constants c and σ^2 .

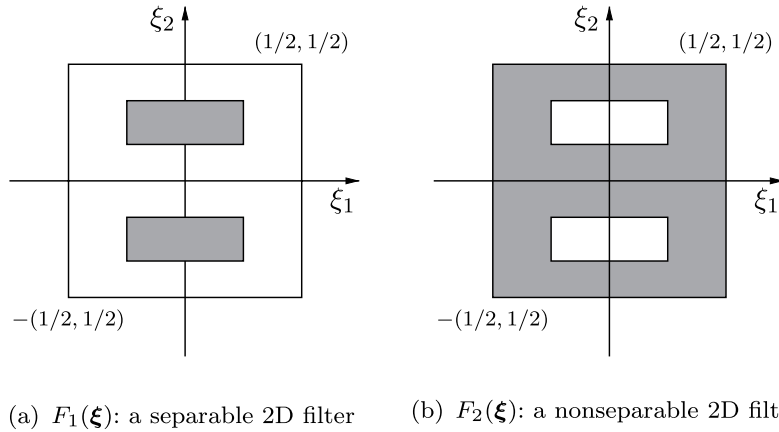


Fig. 2.1 The frequency domain supports (dark gray regions) of two ideal filters $f_1[\mathbf{n}]$ and $f_2[\mathbf{n}]$. Their Fourier transforms, $F_1(\boldsymbol{\xi})$ and $F_2(\boldsymbol{\xi})$, are indicator functions defined on their respective spectral supports.

The merit of separable filters mainly lies in their computational advantages. Consider a d -D filter $h[\mathbf{n}]$ containing $N \times N \times \cdots \times N$ nonzero coefficients. A direct spatial domain implementation of filtering by $h[\mathbf{n}]$ requires N^d multiplications (and a similar number of additions) per output sample. But if $h[\mathbf{n}]$ is separable, it follows from (2.14) that the filtering operation can be efficiently achieved by d consecutive 1D convolutions along each of the dimensions. Correspondingly, the number of required multiplications is reduced to dN per output sample. For large N or d , this reduction in computational complexity can be substantial.

Despite its simplicity in implementation, separability imposes too much restrictions on the filter. The majority of multidimensional filters used in practice are nonseparable. In fact, the greater flexibility offered by nonseparable filters is one of the unique features¹ that distinguish MD signal processing from classical 1D signal processing. Fortunately, it is often still possible to retain the computational advantages of separable filters even if the filter is nonseparable. To that end, we generalize the definition of separability, and call a filter $h[\mathbf{n}]$ K th-order separable, for $K \geq 1$, if

$$h[\mathbf{n}] = \sum_{k=1}^K \left(\prod_{i=1}^d h_{i,k}[n_i] \right), \quad (2.15)$$

for some 1D filters $\{h_{i,k}[n]\}_{i,k}$. For example, the nonseparable filter $F_2(\boldsymbol{\xi})$ shown in Figure 2.1(b) is in fact 2nd-order separable, since it can be written as

$$F_2(\boldsymbol{\xi}) = 1 - F_1(\boldsymbol{\xi}),$$

where $F_1(\boldsymbol{\xi})$ is the separable filter in Figure 2.1(a).

In many cases, K th order separable filters still allow for very efficient filtering operations. To see that, consider again a d -D filter $h[\mathbf{n}]$ with $N \times N \times \cdots \times N$ nonzero coefficients. If $h[\mathbf{n}]$ is K th-order separable

¹ The other unique feature of multidimensional signal processing is the possibility of applying nonseparable sampling operations. See the next section for details.

as in (2.15), then filtering by $h[\mathbf{n}]$ can be achieved through a total of dK convolutions in 1D. Consequently, the total number of required multiplications is about dKN per output sample. For relatively small K , this cost can still be much smaller than N^d , the cost of a direct convolution by $h[\mathbf{n}]$.

3

Multidimensional Sampling

3.1 Sampling on Lattices

Sampling serves as a bridge between the continuous and the discrete domains. In 1D, the uniform sampling operation from a continuous-domain signal $x_c(t)$ to a discrete-domain signal $x_d[n]$ can be represented by a sampling interval T as

$$x_d[n] = x_c(Tn), \quad \text{for } n \in \mathbb{Z}.$$

As a straightforward extension to the MD case, a continuous-domain signal can be sampled *separately* along each dimension, possibly with different sampling intervals (for example, T_1, T_2, \dots, T_d). However, for MD sampling, the choices are much greater. In general, we can represent the sampling operation in d -D by a $d \times d$ nonsingular matrix \mathbf{S} as

$$x_d[\mathbf{n}] = x_c(\mathbf{S}\mathbf{n}), \quad \text{for } \mathbf{n} \in \mathbb{Z}^d. \quad (3.1)$$

Note that the special cases of separable sampling are covered if we choose \mathbf{S} to be a diagonal matrix $\text{diag}(T_1, T_2, \dots, T_d)$.

In (3.1), the locations where samples are taken form a *lattice* [12]

$$\begin{aligned} \Lambda(\mathbf{S}) &\stackrel{\text{def}}{=} \left\{ \mathbf{m} : \mathbf{m} = \mathbf{S}\mathbf{n}, \mathbf{n} \in \mathbb{Z}^d \right\} \\ &= \text{all integer linear combinations of the columns of } \mathbf{S}, \end{aligned} \quad (3.2)$$

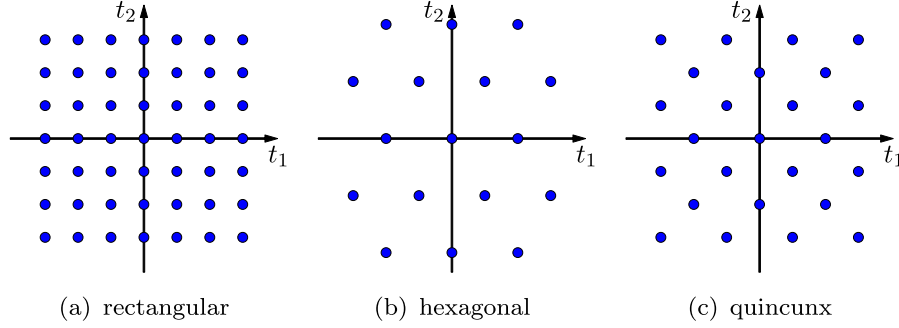


Fig. 3.1 Examples of 2D sampling lattices in the continuous-domain.

and \mathbf{S} is called a generating matrix of $\Lambda(\mathbf{S})$. For example, we show in Figure 3.1 three commonly used sampling lattices in 2D, with corresponding generating matrices

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{H}_0 = \begin{bmatrix} 1 & 1 \\ \sqrt{3} & -\sqrt{3} \end{bmatrix}, \quad \text{and} \quad \mathbf{Q}_0 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad (3.3)$$

respectively.

The generating matrices for a given sampling lattice are *not* unique. For instance, the *hexagonal lattice* in Figure 3.1(b) and the *quincunx* lattice [98] in Figure 3.1(c) can be alternatively generated by

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 2 \\ \sqrt{3} & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_1 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad (3.4)$$

respectively. We can easily verify that

$$\mathbf{H}_0 = \mathbf{H}_1 \mathbf{U}_1 \quad \text{and} \quad \mathbf{Q}_0 = \mathbf{Q}_1 \mathbf{U}_2, \quad (3.5)$$

where

$$\mathbf{U}_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{U}_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Definition 3.1. A matrix \mathbf{U} is called *unimodular* if it is a square integer matrix with determinant equal to 1 or -1 .

From the above definition, we see that both pairs of generating matrices in (3.5) are linked through a unimodular matrix. Rather than

a mere coincidence, this is in fact a special case of the following more general result [95, pp. 558–560].

Theorem 3.2. Two matrices \mathbf{A} and \mathbf{B} generate the same lattice, that is, $\Lambda(\mathbf{A}) = \Lambda(\mathbf{B})$ if and only if $\mathbf{A} = \mathbf{B}\mathbf{U}$ where \mathbf{U} is a unimodular matrix.

We can see from the above theorem that unimodular matrices are essential in characterizing the generating matrices for a given lattice. Since there is an infinite number of different unimodular matrices, it follows that any given sampling lattice allows for an infinite number of generating matrices.

3.2 The Effect of Sampling in the Fourier Domain

The classical Whittaker–Shannon–Kotelnikov sampling theorem [46, 94] states that a 1D bandlimited signal can be exactly reconstructed from its uniform samples if the sampling rate is beyond the Nyquist rate. In general, the effect of the uniform sampling process in the frequency domain is that the spectrum of the original bandlimited signal gets replicated over intervals whose density is inversely proportional to the sampling density. If the sampling density is high enough and thus the shifted copies of the spectrum do not overlap with the baseband, then we have an *alias-free sampling*; consequently, the original signal can be reconstructed from its sampled version by applying an ideal interpolation filter whose passband is supported on the baseband.

The situation is similar in multidimensional cases [79, 95]. To make our arguments more precise, we first need to define the notion of *density* for a MD sampling lattices $\Lambda(\mathbf{S})$. To that end, we define the *fundamental lattice cell* of a lattice $\Lambda(\mathbf{S})$ as the region

$$\mathcal{F}_{\Lambda(\mathbf{S})} \stackrel{\text{def}}{=} \left\{ \mathbf{S}\mathbf{t} : \mathbf{t} \in [0, 1)^d \right\}. \quad (3.6)$$

Note that the shape of $\mathcal{F}_{\Lambda(\mathbf{S})}$ depends on the choice of the generating matrix \mathbf{S} . For example, consider the lattice shown in Figure 3.2. If we choose $\mathbf{S} = \mathbf{H}_0$ as in (3.3), then its fundamental cell $\mathcal{F}_{\Lambda(\mathbf{S})}$ is the gray region depicted in the figure. We observe that $\mathcal{F}_{\Lambda(\mathbf{S})}$ and its shifted

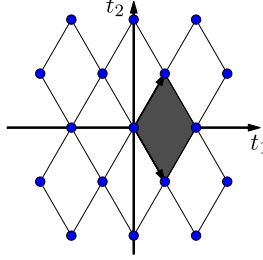


Fig. 3.2 The gray region shows the fundamental lattice cell of $\Lambda(\mathbf{S})$, as defined in (3.6). The fundamental lattice cell and its shifted copies on the lattice form a partition of \mathbb{R}^2 .

copies on the lattice form a partition of the entire space. Since all such lattice cells have the same volume (which is equal to $|\det(\mathbf{S})|$) and each cell contains exactly one sampling point in the lattice, we can thus define the *sampling density* of the lattice to be the inverse of the cell volume, that is,

$$\text{density}(\Lambda(\mathbf{S})) \stackrel{\text{def}}{=} \frac{1}{|\det(\mathbf{S})|}.$$

As a consequence of Theorem 3.2, we can verify that the sampling density defined above is a property of the underlying lattice and that it does not depend on the specific generating matrix we use.

For every sampling lattice $\Lambda(\mathbf{S})$, we define its *reciprocal lattice* (a.k.a. dual lattice) to be $\Lambda(\mathbf{S}^{-T})$, that is, a lattice generated by

$$\mathbf{S}^{-T} \stackrel{\text{def}}{=} (\mathbf{S}^T)^{-1}.$$

The concept of reciprocal lattices plays an important role in the following theorem, which characterizes the effect of MD lattice sampling in the Fourier domain.

Theorem 3.3. The Fourier transforms of the continuous-domain and discrete-domain signals in the sampling operation (3.1) are related by

$$\begin{aligned} X_d(\boldsymbol{\xi}) &= \frac{1}{|\det(\mathbf{S})|} \sum_{\boldsymbol{\ell} \in \mathbb{Z}^d} X_c(\mathbf{S}^{-T}(\boldsymbol{\xi} + \boldsymbol{\ell})) \\ &= \frac{1}{|\det(\mathbf{S})|} \sum_{\mathbf{k} \in \Lambda(\mathbf{S}^{-T})} X_c(\mathbf{S}^{-T}\boldsymbol{\xi} + \mathbf{k}). \end{aligned} \quad (3.7)$$

The above sampling formula (see, for example, [95] for a proof) has a simple geometric interpretation: Suppose that the continuous signal $x_c(\mathbf{t})$ is bandlimited, and its frequency region of support is a bounded open set $\mathcal{D} \subset \mathbb{R}^d$. It follows from (3.7) that the discrete-time Fourier transform of the discrete-domain signal $x_d[\mathbf{n}]$ is supported in

$$\mathbf{S}^T \left(\bigcup_{\mathbf{k} \in \Lambda(\mathbf{S}^{-T})} (\mathcal{D} + \mathbf{k}) \right). \quad (3.8)$$

In words, the frequency support of the discrete samples can be obtained by first taking the union of the baseband \mathcal{D} and all of its shifted copies (that is, aliasing components), and then applying a linear mapping \mathbf{S}^T .

For sampling lattices with sufficiently high densities, the aliasing components in (3.8) do not overlap with the baseband frequency support \mathcal{D} . In this important case, we can fully recover the original continuous signal $x_c(\mathbf{t})$ by applying an ideal interpolation filter spectrally supported on \mathcal{D} to the discrete samples $x_d[\mathbf{n}]$.

Definition 3.4. We say a frequency support \mathcal{D} allows an *alias-free* S -fold sampling, if different shifted copies of \mathcal{D} in (3.8) are disjoint, that is,

$$\mathcal{D} \cap (\mathcal{D} + \mathbf{k}) = \emptyset \text{ for all } \mathbf{k} \in \Lambda(\mathbf{S}^{-T}) \setminus \{\mathbf{0}\}. \quad (3.9)$$

Furthermore, we say \mathcal{D} can be *critically sampled* by \mathbf{S} , if in addition to the alias-free condition in (3.9), the union of the shifted copies also covers the entire spectrum, that is,

$$\bigcup_{\mathbf{k} \in \Lambda(\mathbf{S}^{-T})} (\overline{\mathcal{D}} + \mathbf{k}) = \mathbb{R}^d, \quad (3.10)$$

where $\overline{\mathcal{D}}$ is the closure of the open set \mathcal{D} .

We illustrate in Figure 3.3 examples of alias-free sampling and critical sampling in the frequency domain. While we can always achieve alias-free sampling for any finitely supported frequency shape \mathcal{D} (as long as the sampling density is high enough), critical sampling is more restrictive and imposes stringent constraints on \mathcal{D} .

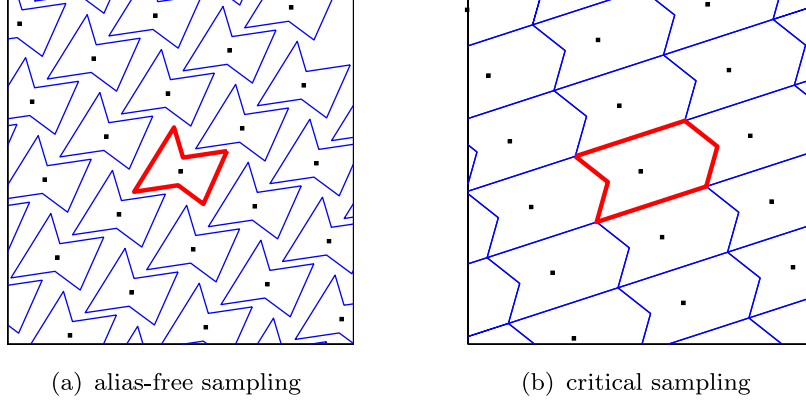


Fig. 3.3 Examples of alias-free sampling and critical sampling. The basebands (drawn in thick lines) are shown together with their aliasing copies (drawn in thin lines).

Clearly not all frequency support shapes can be critically sampled (for example, consider a disc-shaped region in 2D). Let $\mathbf{1}_{\mathcal{D}}(\boldsymbol{\xi})$ denote the indicator function defined on \mathcal{D} , and $\widehat{\mathbf{1}}_{\mathcal{D}}(t)$ its inverse Fourier transform. Then \mathcal{D} can be critically sampled by a matrix \mathbf{S} *if and only if*

$$\widehat{\mathbf{1}}_{\mathcal{D}}(\mathbf{S}\mathbf{n}) = \frac{\delta[\mathbf{n}]}{|\det(\mathbf{S})|}, \quad (3.11)$$

where $\delta[\mathbf{n}]$ is the Kronecker delta function (see [63] for a proof). In particular, when $\mathbf{n} = \mathbf{0}$, the above equality implies that

$$\widehat{\mathbf{1}}_{\mathcal{D}}(\mathbf{0}) = 1/|\det(\mathbf{S})|. \quad (3.12)$$

Since

$$\widehat{\mathbf{1}}_{\mathcal{D}}[\mathbf{0}] \stackrel{\text{def}}{=} \int_{\mathbb{R}^d} \mathbf{1}_{\mathcal{D}}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int_{\mathcal{D}} 1 d\boldsymbol{\xi},$$

it follows from (3.12) that, for critical sampling, the frequency region \mathcal{D} and the fundamental lattice cell of the reciprocal lattice $\Lambda(\mathbf{S}^{-T})$ must have the same volume, which is equal to $1/|\det(\mathbf{S})|$.

3.3 Downsampling and Upsampling of Discrete Signals

Sampling can also be applied to discrete-domain signals. Similar to sampling in the continuous-domain, *downsampling* an MD discrete-domain signal $x[\mathbf{n}]$ is defined by an *integer* $d \times d$ nonsingular matrix

\mathbf{M} and results in the output [95, 99]

$$x_d[\mathbf{n}] = x[\mathbf{M}\mathbf{n}]. \quad (3.13)$$

Since discrete sampling only involves integer matrices, the corresponding sampling lattices $\Lambda(\mathbf{M})$ are always sub-lattices of \mathbb{Z}^d .

For example, as shown in Figure 3.4, the 2D quincunx lattice generated by \mathbf{Q}_1 in (3.4) is a subset of \mathbb{Z}^2 . It retains one out of every two integer points. Figure 3.5(c) shows an example of a downsampled image by \mathbf{Q}_1 .

In the frequency domain, similar to (3.7), one can show [95, 100] that the input and output of the downsampling operation (3.13) are

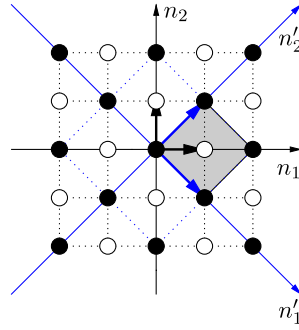


Fig. 3.4 The quincunx sampling lattice (black dots) as a sub-lattice of \mathbb{Z}^2 . A generating matrix for the quincunx sampling lattice is \mathbf{Q}_1 in (3.4). The gray area is the fundamental lattice cell $\mathcal{F}_{\Lambda(\mathbf{Q}_1)}$.

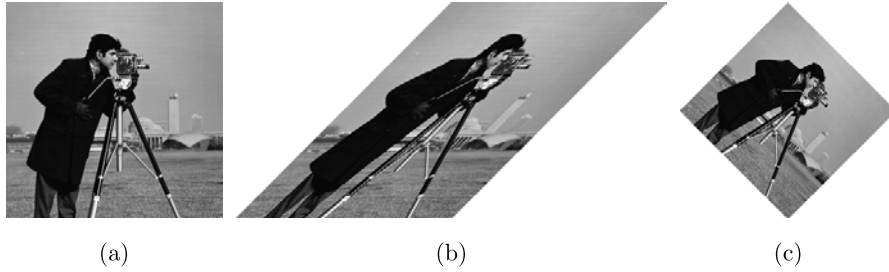


Fig. 3.5 Examples of 2D sampling. (a) The *Cameraman* image. (b) The image after downsampling by \mathbf{R}_0 . (c) The image after downsampling by \mathbf{Q}_1 .

related by

$$X_d(\boldsymbol{\xi}) = \frac{1}{|\det(\mathbf{M})|} \sum_{\mathbf{k} \in \mathcal{N}(\mathbf{M}^T)} X(\mathbf{M}^{-T}(\boldsymbol{\xi} + \mathbf{k})). \quad (3.14)$$

Here $\mathcal{N}(\mathbf{M})$ is defined as the set of integer vectors that can be written in the form of $\mathbf{M}\mathbf{t}$, for some $\mathbf{t} \in [0, 1)^d$. In other words, we have

$$\mathcal{N}(\mathbf{M}) = \mathcal{F}_{\Lambda(\mathbf{M})} \cap \mathbb{Z}^d,$$

where $\mathcal{F}_{\Lambda(\mathbf{M})}$ is the fundamental lattice cell as defined in (3.6). Moreover, one can show [95] that the number of elements in $\mathcal{N}(\mathbf{M})$ is always equal to $|\det(\mathbf{M})|$.

Example 3.5. Consider again the quincunx sampling matrix defined in (3.4) with the generated lattice illustrated in Figure 3.4

$$\det \mathbf{Q}_1 = \det \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = 2.$$

The corresponding set $\mathcal{N}(\mathbf{Q}_1)$ contains exactly two vectors:

$$\mathbf{Q}_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_1 \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

For discrete signals, we can also define the *upsampling* operation. For upsampling by \mathbf{M} , the input $x[\mathbf{n}]$ and the output $x_u[\mathbf{n}]$ are related in the spatial domain by

$$x_u[\mathbf{n}] = \begin{cases} x[\mathbf{k}] & \text{if } \mathbf{n} = \mathbf{M}\mathbf{k}, \mathbf{k} \in \mathbb{Z}^d \\ 0 & \text{otherwise.} \end{cases} \quad (3.15)$$

In the frequency domain, the relation (3.15) becomes [95, 100]

$$X_u(\boldsymbol{\xi}) = X(\mathbf{M}^T \boldsymbol{\xi}). \quad (3.16)$$

To obtain the z -transform of $x_u[\mathbf{n}]$ from that of $x[\mathbf{n}]$, we first extend the vector notation in (2.6) to the matrix case, and write

$$\mathbf{z}^{\mathbf{M}} \stackrel{\text{def}}{=} (z^{\mathbf{M}_1}, z^{\mathbf{M}_2}, \dots, z^{\mathbf{M}_d}),$$

where \mathbf{M}_i , $1 \leq i \leq d$, denotes the i th column vector of \mathbf{M} . We can then verify that

$$X_u(\mathbf{z}) = X(\mathbf{z}^{\mathbf{M}}).$$

There are special cases when the sampling operations use unimodular matrices (recall that they are integer matrices with determinant equal to ± 1). Sampling by a unimodular matrix does not change the data rate but only rearranges the input samples; thus, it is often referred to as a *resampling* operation.

Lemma 3.6. Let \mathbf{R} be a unimodular matrix. Upsampling by \mathbf{R} is equivalent to downsampling by \mathbf{R}^{-1} , and vice versa.

Note that discrete-domain downsampling and upsampling by \mathbf{R}^{-1} are well-defined, since the inverse of a unimodular matrix is still a unimodular, and thus, integer matrix. The equivalence in Lemma 3.6 can be verified either directly in the spatial domain or indirectly through the frequency domain formulas on downsampling in (3.14) and upsampling in (3.16).

In later sections, we will use the following four basic unimodular matrices:

$$\begin{aligned} \mathbf{R}_0 &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, & \mathbf{R}_1 &= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \\ \mathbf{R}_2 &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, & \mathbf{R}_3 &= \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \end{aligned} \quad (3.17)$$

where $\mathbf{R}_0\mathbf{R}_1 = \mathbf{R}_2\mathbf{R}_3 = \mathbf{I}_2$ (here \mathbf{I}_2 denotes the 2×2 identity matrix). Figure 3.5(b) shows an example of a resampled image by \mathbf{R}_0 .

3.4 Key Properties of Sampling Lattices

Theorem 3.2 indicates that the generating matrices for a given sampling lattice are not unique. Among the infinite number of possibilities, we are particularly interested in the *Hermite normal form* for the generating matrices.

Theorem 3.7. (Hermite Normal Form [63, 70]) Given a $d \times d$ non-singular integer-valued matrix \mathbf{M} , there exists a unique unimodular integer matrix \mathbf{U} such that $\mathbf{MU} = \mathbf{H}$ and that \mathbf{H} is an integer matrix in the *Hermite normal form*, which means:

- (1) \mathbf{H} is upper triangular; and
- (2) $h_{i,i} > 0$ and $0 \leq h_{i,j} < h_{i,i}$, for $1 \leq i < j \leq d$.

Furthermore, the Hermite normal form \mathbf{H} of \mathbf{M} is unique.

Combining the above result with Theorem 3.2, we see that the Hermite normal form provides a *complete* and *nonredundant* enumeration of all possible integer sampling lattices. It follows that, to efficiently go through all possible integer sampling lattices of density $1/\delta$, we just need to examine all Hermite matrices of determinant δ .

Example 3.8. In 2D there are exactly three 2×2 Hermite normal matrices whose determinant is equal to 2:

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}.$$

We can see that these matrices generate the vertical, the horizontal, and the quincunx lattices, as shown in Figure 3.6 respectively. Moreover, from Theorem 3.7, these are the complete set of discrete-domain lattices with sampling density of $1/2$.

In general, we can conclude from the definition in Theorem 3.7 that all 2×2 Hermite matrices of determinant δ have the simple form

$$\begin{bmatrix} a & c \\ 0 & b \end{bmatrix},$$

where a, b are positive integers with $ab = \delta$ and $0 \leq c < a$. That is, both a and b are divisors of δ . Given the divisor a , we determine b from $b = \delta/a$, and observe that the number of choices for c equals exactly a . Summing over all possible divisors a gives the total number of Hermite

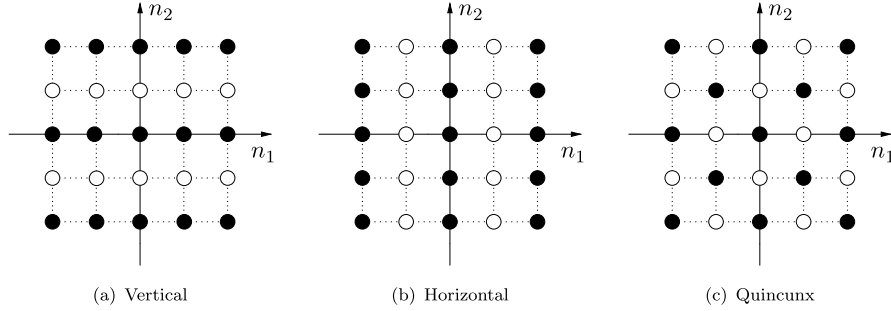


Fig. 3.6 The complete set of three lattices with sampling density $1/2$ in the 2D discrete-domain. The points in the lattices are shown as black dots in the figures.

matrices as

$$\sigma(\delta) \stackrel{\text{def}}{=} \sum_{a|\delta} a.$$

For large δ , it can be shown that the growth rate of $\sigma(\delta)$ (that is, the total number of 2×2 Hermite matrices of determinant δ) is asymptotic to $e^\gamma \delta \ln \ln \delta$ [41], where $\gamma \approx 0.577$ is the Euler constant. For the general d -D case, one can show that the total number of such matrices is essentially equal to $\mathcal{O}(\delta^{d-1})$ [63].

Another useful tool in analyzing MD multirate operations is the *Smith normal form* [88]: Any integer matrix \mathbf{M} can be decomposed into a product \mathbf{UDV} , where \mathbf{U} and \mathbf{V} are unimodular integer matrices and \mathbf{D} is an integer diagonal matrix [95]. For example, the quincunx matrices in (3.3) and (3.4) can be expressed in the Smith form as

$$\begin{aligned} \mathbf{Q}_0 &= \mathbf{R}_1 \mathbf{D}_1 \mathbf{R}_2 = \mathbf{R}_2 \mathbf{D}_2 \mathbf{R}_1 \quad \text{and} \\ \mathbf{Q}_1 &= \mathbf{R}_0 \mathbf{D}_1 \mathbf{R}_3 = \mathbf{R}_3 \mathbf{D}_2 \mathbf{R}_0, \end{aligned} \tag{3.18}$$

where

$$\mathbf{D}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{D}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \tag{3.19}$$

are two 2D diagonal matrices that correspond to dyadic sampling in each dimension, and $\{\mathbf{R}_i\}_{0 \leq i \leq 3}$ are unimodular matrices defined in (3.17). We will revisit the Smith form in Section 4.2 where we explore the concept of multirate identities.

4

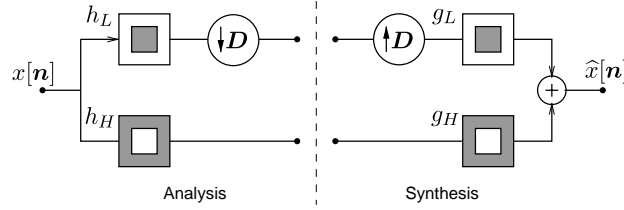
Multidimensional Filter Banks

In the previous sections we have introduced MD filtering, downsampling, and upsampling. These operators form the building blocks of multirate systems and filter banks [95, 99], which are the focus of this section. Key developments and references in MD filter banks include [2, 15, 49, 92, 98, 100].

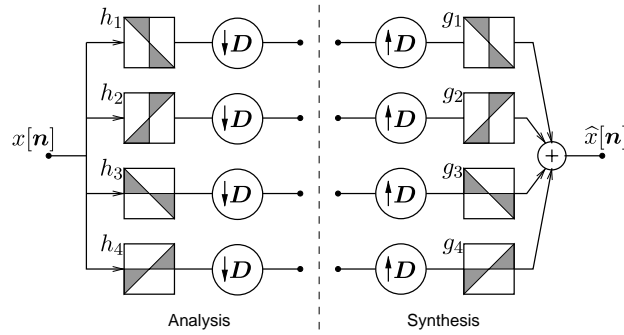
4.1 Filter Banks: from Frequency Decomposition to Signal Representation

Figure 4.1 shows two examples of filter banks in 2-D. The dark regions in the diagrams indicate the idealized frequency domain supports of the corresponding channel filters. From these examples we can see that a complete filter bank consists of two parts: the *analysis* side and the *synthesis* side.

The analysis filter bank decomposes an input signal into different subbands, each corresponding to a different part of the frequency spectrum. For example, the filter bank in Figure 4.1(a) decomposes the input signal $x[\mathbf{n}]$ into a lowpass subband, with frequency domain support $[-1/4, 1/4]^2$, and a complementary highpass subband. Similarly,



(a) A multiresolution filter bank.



(b) A directional filter bank.

Fig. 4.1 Two examples of 2-D filter banks. The dark regions in the diagrams indicate the idealized frequency domain supports of the channel filters. For example, the analysis filter $h_L[n]$ in (a) is a lowpass filter, whose Fourier transform is supported on $[-1/4, 1/4]^2$.

the filter bank in Figure 4.1(b) splits the input into four directional subbands, each occupying one of the wedge-shaped frequency regions. The dual to the analysis part of the filter bank is the synthesis part, which reassembles the subband signals and generates a reconstruction $\hat{x}[n]$. A filter bank is called *perfect reconstruction*, if there is no loss in the decomposition and the subsequent reconstruction, that is, $x[n] = \hat{x}[n]$. As one of the key results of this section, we will discuss in Section 4.3 the conditions that the channel filters have to satisfy to guarantee perfect reconstruction. The design of practical filters satisfying these conditions will be addressed in the following section.

Before presenting more detailed properties of filter banks, we would like to first point out two different perspectives from which filter banks can be understood. As evidenced by the two examples shown in Figure 4.1, it is natural and intuitively appealing to see filter banks from

a frequency domain perspective, in terms of subband decomposition and reconstruction. However, equally important is the following Hilbert space interpretation of filter banks, which plays a key role in our later discussions of geometrical signal representations.

Consider a generic K -channel filter bank, with analysis filters $\{h_k[\mathbf{n}]\}_{k=1}^K$, synthesis filters $\{g_k[\mathbf{n}]\}_{k=1}^K$, and sampling matrices $\{\mathbf{M}_k\}_{k=1}^K$. From the analysis filters, we can define vectors in $\ell^2(\mathbb{Z}^d)$ as

$$\varphi_{k,\mathbf{m}}[\mathbf{n}] \stackrel{\text{def}}{=} h_k^*[\mathbf{M}_k\mathbf{m} - \mathbf{n}], \quad (4.1)$$

each indexed by two parameters: $1 \leq k \leq K$ and $\mathbf{m} \in \mathbb{Z}^2$. We can see that the collection of all such vectors

$$\Phi \stackrel{\text{def}}{=} \{\varphi_{k,\mathbf{m}}[\mathbf{n}] : 1 \leq k \leq K; \mathbf{m} \in \mathbb{Z}^2\} \quad (4.2)$$

consists of the time-flipped and complex-conjugated filters $h_k^*[-\mathbf{n}]$ and all their shifted versions $h_k^*[-(\mathbf{n} - \mathbf{M}_k\mathbf{m})]$ centered on the lattice $\Lambda_{\mathbf{M}_k}$.

Similarly, for the synthesis filters $g_k[\mathbf{n}]$, we can define

$$\psi_{k,\mathbf{m}}[\mathbf{n}] \stackrel{\text{def}}{=} g_k[\mathbf{n} - \mathbf{M}_k\mathbf{m}], \quad (4.3)$$

and write the corresponding collection as

$$\Psi \stackrel{\text{def}}{=} \{\psi_{k,\mathbf{m}}[\mathbf{n}] : 1 \leq k \leq K; \mathbf{m} \in \mathbb{Z}^2\}. \quad (4.4)$$

Applying the analysis/synthesis equalities established in (2.12) and (2.13) for the convolution operator, we can easily verify the following result.

Lemma 4.1. Denote by $c_k[\mathbf{m}]$ the output of the k th channel of the analysis filter bank. Then

$$c_k[\mathbf{m}] = \langle x[\mathbf{n}], \varphi_{k,\mathbf{m}}[\mathbf{n}] \rangle. \quad (4.5)$$

Moreover, the reconstruction from the synthesis filter bank can be written as

$$\hat{x}[\mathbf{n}] = \sum_{1 \leq k \leq K, \mathbf{m} \in \mathbb{Z}^2} c_k[\mathbf{m}] \psi_{k,\mathbf{m}}[\mathbf{n}]. \quad (4.6)$$

In essence, Lemma 4.1 shows that the analysis filter bank computes the inner products of the input signal $x[\mathbf{n}]$ and the vectors $\varphi_{k,\mathbf{m}}$ from the *analysis set* Φ . Meanwhile, the reconstruction $\hat{x}[\mathbf{n}]$ generated by the synthesis filter bank is a linear combination of the vectors $\psi_{k,\mathbf{m}}$ from the *synthesis set* Ψ , and the combination coefficients are just the previously computed inner products.

Combining (4.5) and (4.6), we can write the reconstructed signal as

$$\hat{x}[\mathbf{n}] = \sum_{1 \leq k \leq K, \mathbf{m} \in \mathbb{Z}^2} \langle x[\mathbf{n}], \varphi_{k,\mathbf{m}}[\mathbf{n}] \rangle \psi_{k,\mathbf{m}}[\mathbf{n}].$$

It is well-known from frame theory [21, 65] that the perfect reconstruction condition (that is, $\hat{x}[\mathbf{n}] = x[\mathbf{n}]$ for any input $x[\mathbf{n}]$) holds if the vectors in Φ form a frame of $\ell^2(\mathbb{Z}^2)$ and those in Ψ compose a corresponding dual frame.

In general, frames are redundant sets of vectors, and hence the corresponding filter bank is oversampled [see, for example, the multiresolution filter bank shown in Figure 4.1(a)]. When the frame Φ is nonredundant, the vectors in Φ and Ψ form a pair of biorthogonal bases. Correspondingly, the filter bank becomes *critically-sampled* [see, for example, the directional filter bank in Figure 4.1(b)]. Even more special is the case when $\Phi = \Psi$ and they form an orthogonal basis. In this case, the filter bank is orthogonal and we have $\varphi_{k,\mathbf{m}}[\mathbf{n}] = \psi_{k,\mathbf{m}}[\mathbf{n}]$. It follows from the constructions in (4.1) and (4.3) that

$$h_k^*[-\mathbf{n}] = g_k[\mathbf{n}], \quad (4.7)$$

that is, in an orthogonal filter bank, each analysis filter is simply a time-flipped and complex-conjugated version of the corresponding synthesis filter.

4.2 Basic Tools: Multirate Identities and Polyphase Representations

As we have seen, the building blocks of filter banks are filtering, down-sampling, and up-sampling. In what follows we examine the effects when these building blocks are cascaded. The basic tools for such analysis are the multirate identities and the polyphase representation [95, 99].

When two operations of the same type — either filtering, downsampling, or upsampling — are cascaded, they can be straightforwardly combined into one operation. For example, we show in Figure 4.2 that the cascading of two sampling operations (either downsampling or upsampling) is equivalent to a single sampling operation, where the new and combined sampling matrix is the product of two individual sampling matrices. In the MD case, the order of \mathbf{M}_1 and \mathbf{M}_2 in forming the product is important: we can easily verify through definition that, for downsampling, the combined matrix should be $\mathbf{M}_1\mathbf{M}_2$, while for upsampling, the order is switched to $\mathbf{M}_2\mathbf{M}_1$.

When a downsampling operation is followed by filtering, or filtering is followed by upsampling, then, equivalently, the order of the two operations can be swapped by replacing the filter with its upsampled version. Figure 4.3 illustrates these *multirate identities*. They can be easily verified by showing that with any given input, the outputs of the two equivalent systems are the same [95, 99].

As an application of these multirate identities, consider a typical MD filter bank channel shown in the first line of Figure 4.4. Recall from the previous section that the Smith form factors *any* integer matrix \mathbf{M}

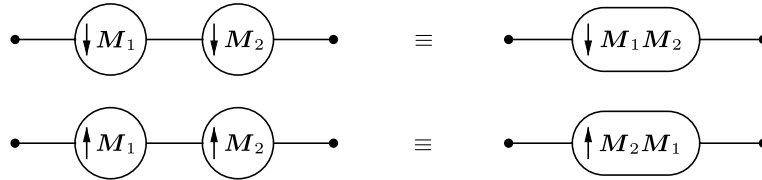


Fig. 4.2 Cascading downsampling or upsampling operators. Note that the order of the matrices is switched in the upsampling case.

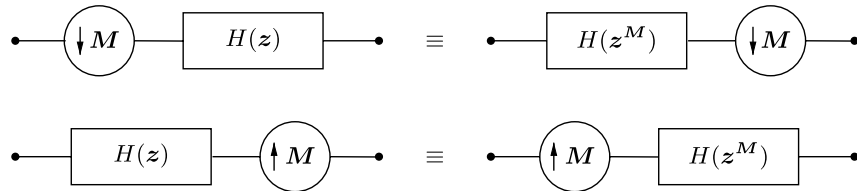


Fig. 4.3 Multidimensional multirate identities. Note that the filter $H(z^M)$ is obtained by upsampling $H(z)$ by \mathbf{M} .

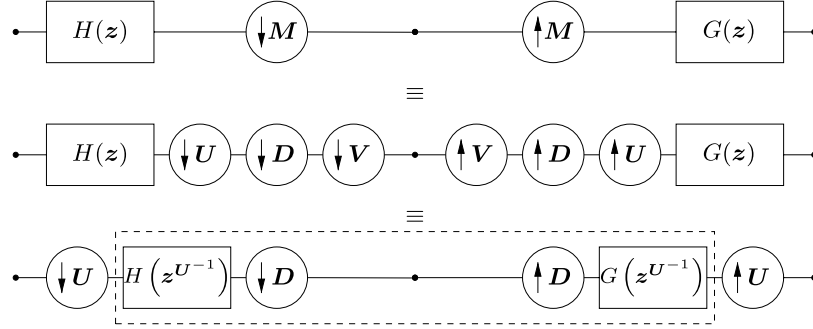


Fig. 4.4 Equivalent forms of a multidimensional filter bank channel obtained through the Smith form and multirate identities.

into a product UDV , where U and V are unimodular integer matrices and D is an integer diagonal matrix. Using this factorization, we get an equivalent implementation of the channel in the second line of Figure 4.4. Since V is an integer unimodular matrix, downsampling by V is perfectly recovered by upsampling by V . (To see this, we note that upsampling by V is equivalent to downsampling by V^{-1} and that the cascading of downsampling by V and V^{-1} reduces to an identity operation.) Consequently, we can omit the sampling steps involving V . Similarly, since U is unimodular, we can rewrite downsampling by U as upsampling by U^{-1} and apply the multirate identities to obtain another equivalent form of the filter bank channel in the third line of Figure 4.4. Note that the part encircled by the dashed lines only uses simple diagonal sampling; that is, the sampling operations are done separately in each dimension.

The equivalent form obtained in Figure 4.4 can simplify the study and characterization of perfect reconstruction filter banks. We note that upsampling by U is the inverse of downsampling by U . Consequently, if we are given a filter bank whose channels involve sampling by a general matrix M , as in the first line of Figure 4.4, then this filter bank has perfect reconstruction *if and only if* a simplified filter bank using corresponding filters with the change of variable $z \mapsto z^{U^{-1}}$ and separate sampling by a diagonal matrix D has perfect reconstruction.

In later sections, we will consider tree-structured filter banks such as the one shown in the left-hand side of Figure 4.5. By applying the

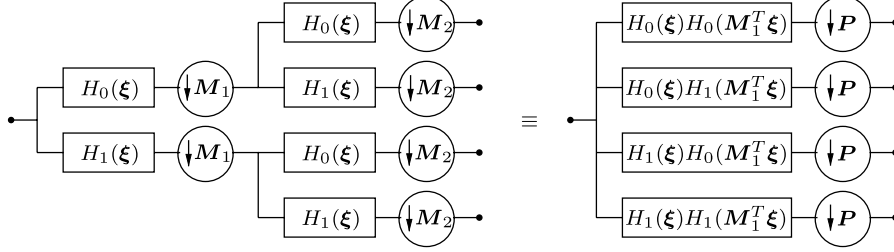


Fig. 4.5 Left: the analysis part of a tree-structured filter bank. Right: the equivalent parallel form of the filter bank. The equivalent downsampling matrix $P = M_1 M_2$.

multirate identities, we can transform the tree-structured filter bank to its equivalent parallel form as shown in the right-hand side of Figure 4.5. In the parallel form, each channel is made of an equivalent filter followed by an equivalent downsampling matrix.

The multirate identities demonstrated in Figure 4.3 do not cover the cases when a generic filtering operation is followed by downsampling, or when upsampling is followed by filtering: the order of these operations cannot be easily swapped. Fortunately, these cases can be effectively treated by a powerful tool called the *polyphase representation* [95, 99]. The polyphase components with respect to the sampling matrix M are defined as

$$x_i[\mathbf{n}] \stackrel{\text{def}}{=} x[\mathbf{M}\mathbf{n} + \ell_i] \quad (4.8)$$

where ℓ_i is drawn from $\mathcal{N}(\mathbf{M})$ — that is, the set of $|\mathbf{M}| \stackrel{\text{def}}{=} |\det(\mathbf{M})|$ integer vectors of the form $\mathbf{M}\mathbf{t}$, $\mathbf{t} \in [0, 1)^d$. Each polyphase component lives on a *coset*

$$\mathcal{C}_i \stackrel{\text{def}}{=} \left\{ \mathbf{n} : \mathbf{n} = \mathbf{M}\mathbf{k} + \ell_i, \mathbf{k} \in \mathbb{Z}^d \right\}, \quad 0 \leq i \leq |\mathbf{M}| - 1. \quad (4.9)$$

We show in Figure 4.6 examples of polyphase components for separable and quincunx lattices. It is easy to see that for each $\mathbf{n} \in \mathbb{Z}^d$, there exist a unique pair of $\ell_i \in \mathcal{N}(\mathbf{M})$ and $\mathbf{k} \in \mathbb{Z}^d$ such that $\mathbf{n} = \mathbf{M}\mathbf{k} + \ell_i$. In other words, the $|\mathbf{M}|$ cosets defined in (4.9) tile the integer lattice \mathbb{Z}^d .

The original signal $x[\mathbf{n}]$ can be formed from its polyphase components $\{x_i[\mathbf{n}]\}_{i=0}^{|\mathbf{M}|-1}$ through *interleaving* samples according to (4.8). Equivalently, we can reconstruct the original signal by upsampling each

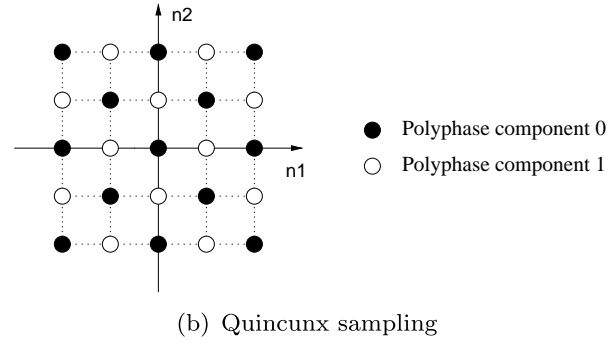
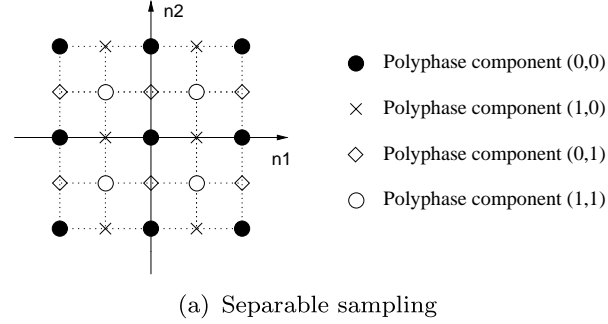


Fig. 4.6 Examples of separable and nonseparable polyphase decompositions in 2-D.

of the polyphase component by \mathbf{M} , shifting the i th component by ℓ_i , and then summing up the resulting signals. In the z -domain, this procedure can be expressed as

$$X(z) = \sum_{i=0}^{|\mathbf{M}|-1} z^{-\ell_i} X_i(z^{\mathbf{M}}). \quad (4.10)$$

Example 4.2 (Separable and quincunx polyphase). For a separable sampling operation in 2-D by matrix

$$\mathbf{M} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$$

we have $\mathcal{N}(\mathbf{M}) = \{(0,0)^T, (1,0)^T, (0,1)^T, (1,1)^T\}$. The four associated polyphase components are depicted in Figure 4.6(a). In the z -domain,

we can specialize (4.10) to

$$\begin{aligned} X(z_1, z_2) &= X_0(z_1^2, z_2^2) + z_1^{-1} X_1(z_1^2, z_2^2) + z_2^{-1} X_2(z_1^2, z_2^2) \\ &\quad + z_1^{-1} z_2^{-1} X_3(z_1^2, z_2^2). \end{aligned}$$

For a nonseparable sampling associated with the quincunx matrix

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},$$

we have $\mathcal{N}(\mathbf{Q}_1) = \{(0, 0)^T, (1, 0)^T\}$. In this case, we have two polyphase components as depicted in Figure 4.6(b). In the z -domain, we have

$$X(z_1, z_2) = X_0(z_1 z_2^{-1}, z_1 z_2) + z_1^{-1} X_1(z_1 z_2^{-1}, z_1 z_2).$$

Consider a cascade of two operations, where we first filter an input signal $x[n]$ by $h[n]$ and then downsample the result by a matrix \mathbf{M} . Using the polyphase representation, we can express the final output $y[n]$ as

$$\begin{aligned} y[\mathbf{n}] &= \sum_{\mathbf{m} \in \mathbb{Z}^d} x[\mathbf{m}] h[\mathbf{M}\mathbf{n} - \mathbf{m}] \\ &= \sum_{i=0}^{|\mathbf{M}|-1} \sum_{\mathbf{k} \in \mathbb{Z}^d} x[\mathbf{M}\mathbf{k} + \boldsymbol{\ell}_i] h[\mathbf{M}\mathbf{n} - \mathbf{M}\mathbf{k} - \boldsymbol{\ell}_i] \\ &= \sum_{i=0}^{|\mathbf{M}|-1} \sum_{\mathbf{k} \in \mathbb{Z}^d} x_i[\mathbf{k}] h_i[\mathbf{n} - \mathbf{k}], \end{aligned}$$

or equivalently, in the z -domain,

$$Y(z) = \sum_{i=0}^{|\mathbf{M}|-1} H_i(z) X_i(z), \quad (4.11)$$

where

$$h_i[\mathbf{k}] \stackrel{\text{def}}{=} h[\mathbf{M}\mathbf{k} - \boldsymbol{\ell}_i], \quad \text{for } i = 0, \dots, |\mathbf{M}| - 1 \quad (4.12)$$

are the polyphase components of the *analysis* filter $h[\mathbf{n}]$. Comparing the above definition with that in (4.8), we note that the

polyphase decomposition of the analysis filters $h[\mathbf{n}]$ has the sign of ℓ_i flipped. In the literature, the two forms in (4.8) and (4.12) are sometimes called Type-1 and Type-2 polyphase decompositions [95], respectively.

The output $r[n]$ of upsampling $y[n]$ by \mathbf{M} followed by filtering by $g[n]$ can be written as

$$r[\mathbf{n}] = \sum_{\mathbf{m} \in \mathbb{Z}^d} y[\mathbf{m}] g[\mathbf{n} - \mathbf{M}\mathbf{m}].$$

It follows that the polyphase components of the output $r[n]$ are

$$\begin{aligned} r_i[\mathbf{k}] &= r[\mathbf{M}\mathbf{k} + \ell_i] = \sum_{\mathbf{m} \in \mathbb{Z}^d} y[\mathbf{m}] g[\mathbf{M}\mathbf{k} + \ell_i - \mathbf{M}\mathbf{m}] \\ &= \sum_{\mathbf{m} \in \mathbb{Z}^d} y[\mathbf{m}] g_i[\mathbf{k} - \mathbf{m}], \end{aligned}$$

or equivalently, in the z -domain,

$$R_i(z) = G_i(z)Y(z), \quad \text{for } i = 0, \dots, |\mathbf{M}| - 1, \quad (4.13)$$

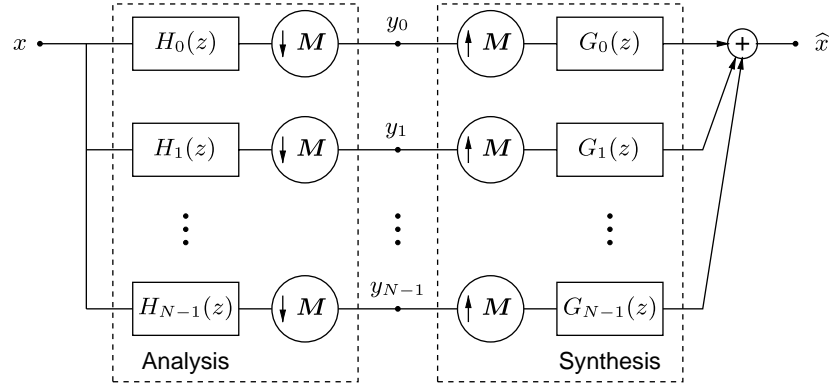
where

$$g_i[\mathbf{k}] \stackrel{\text{def}}{=} g[\mathbf{M}\mathbf{k} + \ell_i], \quad \text{for } i = 0, \dots, |\mathbf{M}| - 1$$

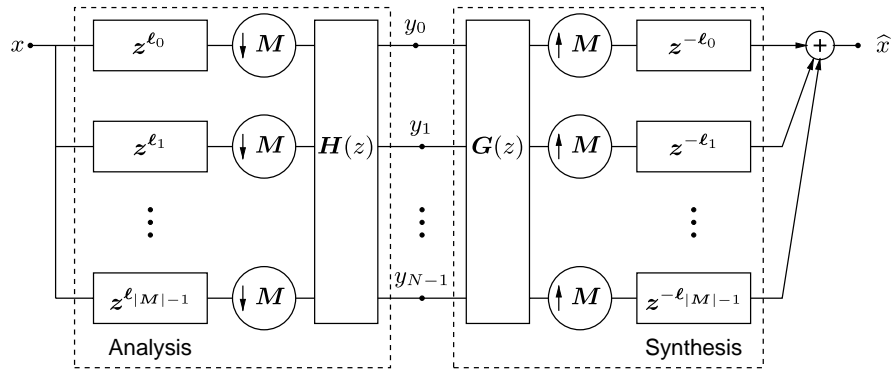
are the polyphase components of the *synthesis* filter $g[\mathbf{n}]$. Note that the polyphase decomposition for synthesis filters has the same form as that in (4.8).

4.3 Perfect Reconstruction Filter Banks

Consider a general multidimensional filter bank as shown in Figure 4.7(a) with N channels and a common sampling matrix \mathbf{M} . The *analysis part* transforms the input signal $x[\mathbf{n}]$ into N filtered and downsampled outputs $y_j[\mathbf{n}]$, $j = 0, 1, \dots, N - 1$. The *synthesis part* attempts to recover the original signal from $y_j[\mathbf{n}]$ by upsampling and filtering. This filter bank setup appears in many applications; notably discrete wavelet transforms [65], subband coding [95, 99], and multi-channel acquisition [52].



(a) A general multidimensional filter bank.



(b) The equivalent filter bank in the polyphase-domain.

Fig. 4.7 A multidimensional filter bank and its equivalent polyphase representation.

Using the polyphase representation, the input signal $x[n]$ can be represented by a vector of its polyphase components $\mathbf{x}(z) \stackrel{\text{def}}{=} (X_0(z), \dots, X_{|M|-1}(z))^T$. Denote $\mathbf{y}(z) \stackrel{\text{def}}{=} (Y_0(z), \dots, Y_{N-1}(z))^T$. Using (4.11) we can express the input–output relationship of the analysis side of the filter bank in Figure 4.7 as a matrix-vector multiplication in the polyphase domain

$$\mathbf{y}(z) = \mathbf{H}(z)\mathbf{x}(z), \quad (4.14)$$

where

$$\mathbf{H}(\mathbf{z}) = \begin{bmatrix} H_{0,0}(\mathbf{z}) & H_{0,1}(\mathbf{z}) & \cdots & H_{0,|\mathbf{M}|-1}(\mathbf{z}) \\ H_{1,0}(\mathbf{z}) & H_{1,1}(\mathbf{z}) & \cdots & H_{1,|\mathbf{M}|-1}(\mathbf{z}) \\ \vdots & \vdots & \ddots & \vdots \\ H_{N-1,0}(\mathbf{z}) & H_{N-1,1}(\mathbf{z}) & \cdots & H_{N-1,|\mathbf{M}|-1}(\mathbf{z}) \end{bmatrix}. \quad (4.15)$$

Here $H_{i,j}(\mathbf{z})$ denotes the j th polyphase component of the filter $H_i(\mathbf{z})$.

Similarly, by using (4.13), the input-output relationship of the synthesis part of the filter bank can be expressed compactly in the polyphase domain as

$$\hat{\mathbf{x}}(\mathbf{z}) = \mathbf{G}(\mathbf{z})\mathbf{y}(\mathbf{z}). \quad (4.16)$$

Here, $\hat{\mathbf{x}}(\mathbf{z}) \stackrel{\text{def}}{=} (\hat{X}_0(\mathbf{z}), \dots, \hat{X}_{|\mathbf{M}|-1}(\mathbf{z}))^T$ is the vector formed by the polyphase components of the reconstructed signal $\hat{x}(\mathbf{z})$, and

$$\mathbf{G}(\mathbf{z}) = \begin{bmatrix} G_{0,0}(\mathbf{z}) & G_{1,0}(\mathbf{z}) & \cdots & G_{N-1,0}(\mathbf{z}) \\ G_{0,1}(\mathbf{z}) & G_{1,1}(\mathbf{z}) & \cdots & G_{N-1,1}(\mathbf{z}) \\ \vdots & \vdots & \ddots & \vdots \\ G_{0,|\mathbf{M}|-1}(\mathbf{z}) & G_{1,|\mathbf{M}|-1}(\mathbf{z}) & \cdots & G_{N-1,|\mathbf{M}|-1}(\mathbf{z}) \end{bmatrix}, \quad (4.17)$$

where $G_{i,j}(\mathbf{z})$ denotes the i th polyphase component of the j th synthesis filter $G_j(\mathbf{z})$.

The two identities in (4.14) and (4.16) point to an equivalent implementation of the original multidimensional filter bank in the polyphase domain, as shown in Figure 4.7(b). This polyphase-domain representation provides a powerful tool for the analysis and design of filter banks. For example, we can immediately obtain the following characterization of MD filter banks [47, 95, 100].

1. *Perfect reconstruction*: The filter bank has perfect reconstruction if $\hat{\mathbf{x}}(\mathbf{z}) = \mathbf{x}(\mathbf{z})$ for any input, or equivalently,

$$\mathbf{G}(\mathbf{z})\mathbf{H}(\mathbf{z}) = I_{|\mathbf{M}|}, \quad (4.18)$$

which means that $\mathbf{G}(\mathbf{z})$ is a left inverse of $\mathbf{H}(\mathbf{z})$.

2. *Critical sampling*: The filter bank is critically sampled if the sampling density is equal to the inverse of the number of channels, that is, $|\mathbf{M}| = N$. This means that both $\mathbf{H}(\mathbf{z})$ and $\mathbf{G}(\mathbf{z})$ are square matrices.

3. *Biorthogonality*: The filter bank is called biorthogonal if it has perfect reconstruction and is critically sampled. Since the left inverses of square matrices are also the right inverses, the perfect reconstruction condition (4.18) for biorthogonal filter bank becomes:

$$\mathbf{G}(\mathbf{z})\mathbf{H}(\mathbf{z}) = \mathbf{H}(\mathbf{z})\mathbf{G}(\mathbf{z}) = \mathbf{I}_{|\mathbf{M}|}. \quad (4.19)$$

4. *Orthogonality*: The filter bank is called orthogonal if it is biorthogonal and its analysis filters are simply time-flipped and complex-conjugated versions of the synthesis filters, that is,

$$h_j[\mathbf{n}] = g_j^*[-\mathbf{n}].$$

In the polyphase domain, this condition can be translated to $\mathbf{H}(\mathbf{z}) = \mathbf{G}_*^T(\mathbf{z}^{-1})$, where the notation $\mathbf{G}_*(\mathbf{z})$ denotes conjugation of coefficients without changing \mathbf{z} . For filters with real-valued coefficients, $\mathbf{G}_*(\mathbf{z}) = \mathbf{G}(\mathbf{z})$. It follows that a real-valued filter bank is orthogonal if and only if

$$\mathbf{H}(\mathbf{z}) = \mathbf{G}^T(\mathbf{z}^{-1}) \quad \text{and} \quad \mathbf{G}(\mathbf{z})\mathbf{G}^T(\mathbf{z}^{-1}) = \mathbf{G}^T(\mathbf{z}^{-1})\mathbf{G}(\mathbf{z}) = \mathbf{I}_{|\mathbf{M}|}.$$

4.4 Example: Two-Channel Filter Banks in 2D

To make our presentations on MD filter banks more concrete, we study here several examples of 2-D two-channel filter banks, whose general structure is shown in Figure 4.8. These filter banks will also serve as important building blocks for tree-structured directional filter banks that will be described in later sections.

We focus on critically-sampled two-channel filter banks. In this case, the sampling matrix \mathbf{M} of the filter bank must have $|\det(\mathbf{M})| = 2$. As we see in Example 3.8, there are a total of three such subsampling lattices, all of which are depicted in Figure 3.6.

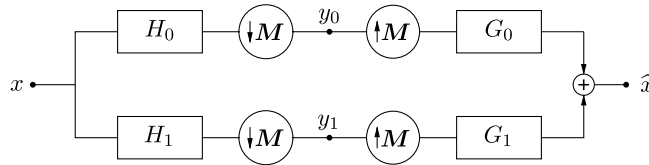


Fig. 4.8 The general structure of a two-channel filter bank.

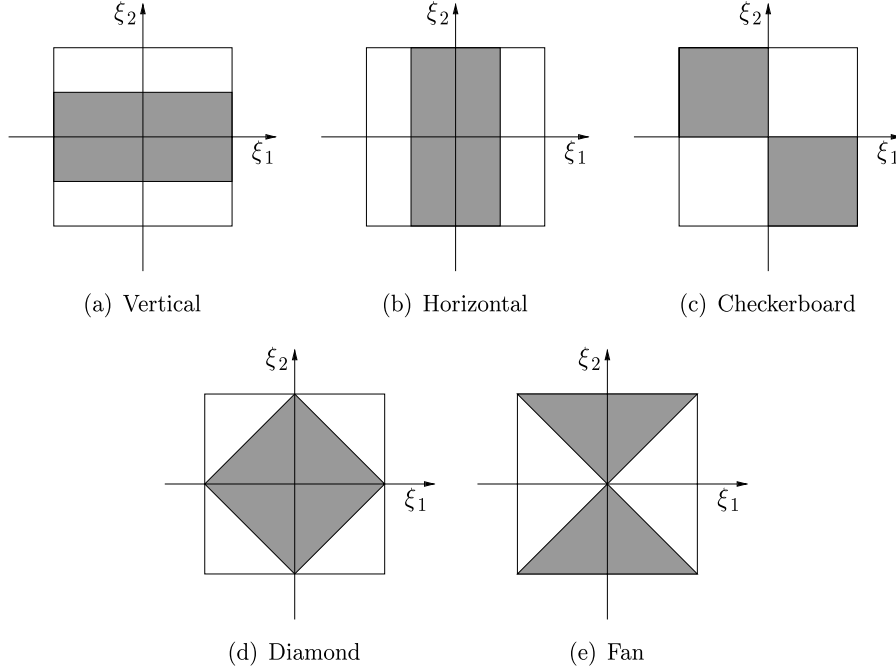


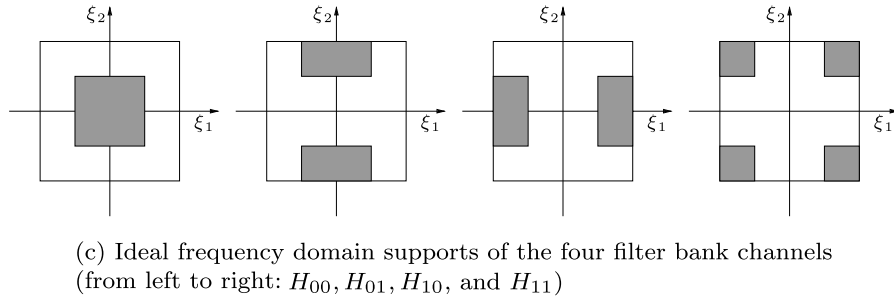
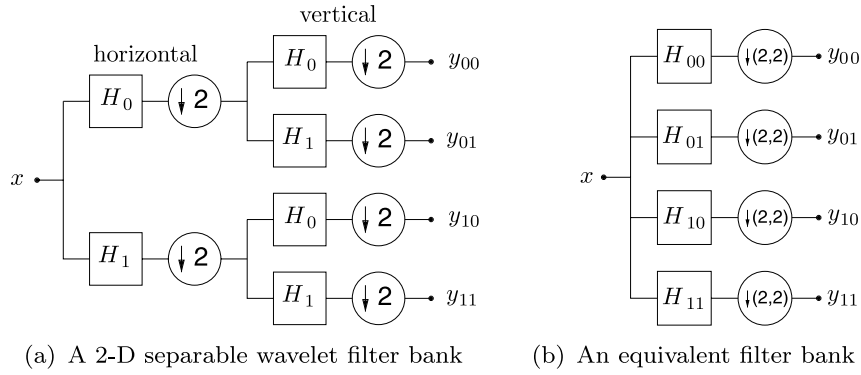
Fig. 4.9 Typical frequency support configurations for the channel filters in 2D two-channel filter banks. Each gray-colored region represents the ideal frequency support of a channel filter, and the complementary white region represent the ideal frequency support of the other channel filter.

The two channels in Figure 4.8 typically split the 2D frequency spectrum into two complementary regions. Figure 4.9 shows typical frequency partitions for commonly-used 2-channel filter banks. Except for the vertical and horizontal partitions [Figures 4.9(a) and 4.9(b)] which can be realized by separable filters, the remaining three frequency partitions have to be realized by nonseparable filters. Using the tools presented in Section 3 [in particular, the condition in (3.11)], we can identify the associated critical sampling lattices for each of these commonly-used 2D two-channel filter banks. Table 4.1 summarizes these pairings between frequency partitions and sampling lattices.

The vertical and horizontal two-channel filter banks are commonly used in the implementation of 2D *separable wavelet transforms*. For these cases, filtering and sampling are simplified into a series of

Table 4.1. Critical sampling lattices (from Figure 3.6) associated with the typical frequency decompositions shown in Figure 4.9.

Frequency Partition	Sampling Lattice
Vertical [Figure 4.9(a)]	Vertical [Figure 3.6(a)]
Horizontal [Figure 4.9(b)]	Horizontal [Figure 3.6(b)]
Diamond [Figure 4.9(d)]	Quincunx [Figure 3.6(c)]
Fan [Figure 4.9(e)]	Quincunx [Figure 3.6(c)]
Checker-board [Figure 4.9(c)]	Either vertical [Figure 3.6(a)] or horizontal [Figure 3.6(b)]

Fig. 4.10 Two-dimensional separable wavelet filter bank. H_0 and H_1 are the lowpass and highpass filters of a 1D two-channel filter bank, and $\{H_{i,j}\}$ are the equivalent 2D filters.

1D operations along each dimension, as depicted in Figure 4.10(a). Specifically, a 1D two-channel filter bank is applied horizontally to each row of the input image, and then applied vertically to each column of the previous output. Figure 4.11 shows the sequence input and output after each stage on an example image. For multiscale decomposition,

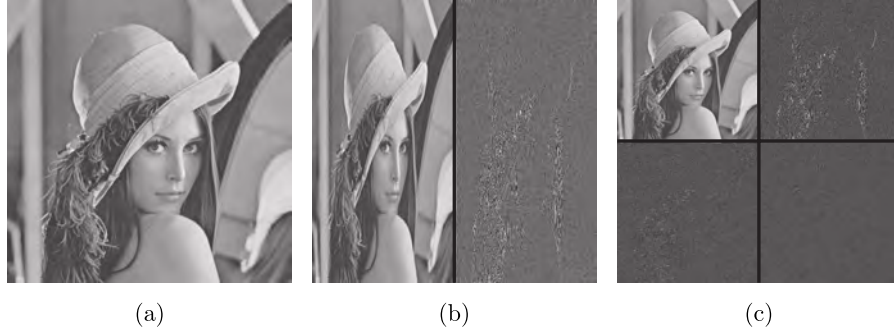


Fig. 4.11 Example of 2D separable wavelet filter bank decomposition. (a) Input image; (b) Output images after the horizontal step (two-channel filter bank decomposition applied to each row); (c) Output images after the vertical step (two-channel filter bank decomposition applied to each column). The final output consists of four subband images.

the entire filter bank in Figure 4.10(a) is iteratively applied on the y_{00} (that is, the low-low) channel.

The 2D separable wavelet filter bank in Figure 4.10(a) can be formally expressed as a two-level tree-structured filter bank as in Figure 4.5 (Left). Here, 1D horizontal filtering by H_i means 2D filtering by $H_i(z_1)$, and 1D horizontal sampling by 2 means 2D sampling by $\mathbf{M}_1 = \text{diag}(2, 1)$. Similarly, 1D vertical filtering by H_i means 2D filtering by $H_i(z_2)$, and 1D vertical sampling by 2 means 2D sampling by $\mathbf{M}_2 = \text{diag}(1, 2)$. Applying multirate identities, we can transform the tree-structured 2D separable wavelet filter bank in Figure 4.10(a) into an equivalent parallel form in Figure 4.10(b). The equivalent 2D filters are

$$H_{i,j}(z_1, z_2) = H_i(z_1)H_j(z_2), \quad i, j \in \{0, 1\}, \quad (4.20)$$

and the equivalent combined sampling matrix is $\mathbf{M} = \text{diag}(2, 2)$. If H_0 and H_1 are ideal half-band lowpass and highpass filters, respectively, then the equivalent 2D separable wavelet filters $H_{i,j}(z_1, z_2)$ in (4.20) partition the 2D spectrum into 4 subbands as depicted in Figure 4.10(c).

Quincunx filter banks (QFBs) associated with the quincunx sampling lattice form another class of widely used two-channel filter banks. The QFBs can split the frequency spectrum of the input signal into

lowpass and highpass channels by using a diamond-shaped filter pair as shown in Figure 4.9(d), or into horizontal and vertical channels by using a fan-shaped filter pair as shown in Figure 4.9(e). Note that these two types of QFBs (diamond and fan frequency partitions) are essentially the same: we can transform one into the other by simply shifting the frequency responses of the filters by 0.5 in either the ξ_1 or ξ_2 frequency variable. (This can be achieved by multiplying the filters by $(-1)^{n_1}$ or $(-1)^{n_2}$ in the space domain.) Alternatively, this transformation can also be done by modulating the input and output signals before and after the filter bank. Thus, the design problem for the fan-shaped QFB boils down to the design of the diamond-shaped QFB, and vice versa. In Section 6, we will use iterations of the fan-shaped QFBs to construct directional filter banks.

From (4.19), the perfect reconstruction condition for the general 2-channel critically-sampled filter bank in Figure 4.8 is

$$\begin{bmatrix} H_{0,0}(z) & H_{0,1}(z) \\ H_{1,0}(z) & H_{1,1}(z) \end{bmatrix} \begin{bmatrix} G_{0,0}(z) & G_{1,0}(z) \\ G_{0,1}(z) & G_{1,1}(z) \end{bmatrix} = \mathbf{I}_2.$$

In particular, this condition implies that

$$H_{0,0}(z)G_{0,0}(z) + H_{0,1}(z)G_{0,1}(z) = 1 \quad (4.21)$$

and

$$\begin{bmatrix} H_{0,0}(z) & H_{0,1}(z) \\ H_{1,0}(z) & H_{1,1}(z) \end{bmatrix} = \frac{1}{\det(\mathbf{G}(z))} \begin{bmatrix} G_{1,1}(z) & -G_{1,0}(z) \\ -G_{0,1}(z) & G_{0,0}(z) \end{bmatrix}. \quad (4.22)$$

When all filters are restricted to having finite impulse responses (FIR), the identity

$$\det[\mathbf{H}(z)] \det[\mathbf{G}(z)] = \det(\mathbf{I}) = 1$$

implies that $\det[\mathbf{G}(z)]$ must have the form $\alpha \mathbf{z}^{\mathbf{m}}$, for some $\alpha \in \mathbb{R} \setminus \{0\}$ and $\mathbf{m} \in \mathbb{Z}^2$. It then follows from (4.22) that the analysis filters $H_0(z)$ and $H_1(z)$ are uniquely determined (up to a scalar α and shift $\mathbf{z}^{\mathbf{m}}$) from the synthesis filters $G_1(z)$ and $G_0(z)$, respectively. Specifically,

$$\begin{bmatrix} H_{1,0}(z) & H_{1,1}(z) \end{bmatrix} = \alpha^{-1} \mathbf{z}^{-\mathbf{m}} \begin{bmatrix} -G_{0,1}(z) & G_{0,0}(z) \end{bmatrix}, \quad (4.23)$$

$$\begin{bmatrix} G_{1,0}(z) & G_{1,1}(z) \end{bmatrix} = \alpha \mathbf{z}^{\mathbf{m}} \begin{bmatrix} -H_{0,1}(z) & H_{0,0}(z) \end{bmatrix}. \quad (4.24)$$

Conversely, it can be verified by substitution that if 2×2 polyphase matrices $\mathbf{H}(\mathbf{z})$ and $\mathbf{G}(\mathbf{z})$ satisfy (4.21), (4.23), and (4.24) then they satisfy the biorthogonal condition (4.19). This leads to the following general design procedure for 2-channel FIR perfect reconstruction filter banks.

- (1) Design filters $H_0(\mathbf{z})$ and $G_0(\mathbf{z})$ in one channel such that they satisfy (4.21).
- (2) Form filters $H_1(\mathbf{z})$ and $G_1(\mathbf{z})$ in the other channel from $G_0(\mathbf{z})$ and $H_0(\mathbf{z})$, respectively, according to (4.23) and (4.24).

Sometimes it is convenient to convert the biorthogonal condition (4.21) from the polyphase domain to the filter domain. For example, for a filter bank with a quincunx sampling matrix, the condition (4.21) is equivalent to requiring [92]

$$P(z_1, z_2) + P(-z_1, -z_2) = 2 \quad (4.25)$$

$$\text{where } P(\mathbf{z}) = G_0(\mathbf{z})H_0(\mathbf{z}). \quad (4.26)$$

The constraint (4.25) is easy to enforce since it amounts to a set of linear equations involving the coefficients of $P(\mathbf{z})$. However, the factorization in (4.26) is hard to deal with in the MD case, since, unlike in 1-D, we no longer have the fundamental theorem of algebra. The lack of theory and tools to factor multivariate polynomials is the fundamental reason why multidimensional filter banks with perfect reconstruction are much more challenging to design than their 1-D counterparts.

5

Characterization and Design of Multidimensional Filter Banks

The design of 1D perfect reconstruction filter banks is a mature subject, with many tools and techniques available (see, for example, [95, 99]). However, the multidimensional case is much more challenging, mainly due to the difficulty in factoring multidimensional polynomials. In this section, we present several useful approaches to characterizing and designing multidimensional perfect reconstruction filter banks.

5.1 Characterizing MD Filter Banks Using Gröbner Bases

As shown in Section 4.3, the general multidimensional filter bank in Figure 4.7 can be represented by a pair of analysis and synthesis polyphase matrices $\mathbf{H}(\mathbf{z})$ and $\mathbf{G}(\mathbf{z})$ of size $N \times M$ and $M \times N$, respectively, where N is the number of channels and $M \stackrel{\text{def}}{=} |\mathbf{M}|$ is the absolute value of the determinant of the sampling matrix. The entries of the polyphase matrices $\mathbf{H}(\mathbf{z})$ and $\mathbf{G}(\mathbf{z})$ are the z -transform of the polyphase components of the analysis and synthesis filters and thus they are *multivariate Laurent polynomials*, which have the general form:

$$F(\mathbf{z}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} f[\mathbf{k}] \mathbf{z}^{\mathbf{k}} = \sum_{\mathbf{k} \in \mathbb{Z}^d} f[k_1, \dots, k_d] z_1^{k_1} \dots z_d^{k_d}. \quad (5.1)$$

Note that the powers in *Laurent polynomials* can be both positive and negative integers, whereas the powers in ordinary *polynomials* can only be non-negative integers.

To design perfect reconstruction filter banks, we need to solve the Laurent polynomial matrix equation

$$\mathbf{G}(\mathbf{z})\mathbf{H}(\mathbf{z}) = I_{|\mathbf{M}|}, \quad (5.2)$$

in the polyphase domain. In 1-D with univariate polynomials, the standard tool for solving this problem is the *Euclidean division algorithm*. In the MD case with multivariate polynomials, the Euclidean division algorithm is no longer applicable and we need to resort to the theory and algorithms of *Gröbner bases*, which were originally developed by Buchberger in the 1960s (see [4, 59] for reviews geared toward signal processing applications).

Intuitively, the Gröbner basis computation can be viewed equivalently as Gaussian elimination for solving the polynomial matrix equation (5.2). Formally, for a set of polynomial vectors $\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$, the *module* generated by this set is defined as

$$\begin{aligned} \text{Module}\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\} &\stackrel{\text{def}}{=} \{c_1(\mathbf{z})\mathbf{h}_1(\mathbf{z}) + \dots + c_N(\mathbf{z})\mathbf{h}_N(\mathbf{z}) : \\ &\text{for some polynomials } c_1(\mathbf{z}), \dots, c_N(\mathbf{z})\}. \end{aligned} \quad (5.3)$$

The $\text{Module}\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$ is analogous to the *span* of a set of vectors in linear algebra. The theory of Gröbner bases implies that $\text{Module}\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$ has a unique *reduced Gröbner basis* for a given order of power products in polynomials (such as the lexicographic order). Let $\{\mathbf{b}_1(\mathbf{z}), \dots, \mathbf{b}_K(\mathbf{z})\}$ be this Gröbner basis. Then it can be obtained from $\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$ by a finite sequence of reduction (division) steps. Tracing back this reduction process, we can express the basis vectors $\mathbf{b}_i(\mathbf{z})$ in terms of the original vectors $\mathbf{h}_j(\mathbf{z})$ through a $K \times N$ transformation matrix $[W_{ij}(\mathbf{z})]$ as

$$\mathbf{b}_i(\mathbf{z}) = \sum_{j=1}^N W_{ij}(\mathbf{z})\mathbf{h}_j(\mathbf{z}), \quad i = 1, \dots, K. \quad (5.4)$$

Algorithms for computing Gröbner bases are available in most computer algebra software systems, such as Maple, Mathematica, and Singular [25].

The first application of Gröbner bases to characterizing perfect reconstruction multidimensional filter banks is established by the following result.

Proposition 5.1. [53, 76] Suppose $\mathbf{H}(\mathbf{z})$ is a given $N \times M$ polynomial matrix. Let $\text{Module}\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$ be the module generated by the rows $\mathbf{h}_i(\mathbf{z})$ of $\mathbf{H}(\mathbf{z})$ as defined in (5.3). Then there exists an $M \times N$ polynomial matrix $\mathbf{G}(\mathbf{z})$ such that

$$\mathbf{G}(\mathbf{z})\mathbf{H}(\mathbf{z}) = \mathbf{I}_M, \quad (5.5)$$

if and only if the reduced Gröbner basis of $\text{Module}\{\mathbf{h}_1(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$ is $\{\mathbf{e}_i\}_{i=1, \dots, M}$, where \mathbf{e}_i is the i th row of the $M \times M$ identity matrix \mathbf{I}_M . In that case, the associated transformation matrix $[W_{ij}(\mathbf{z})]$ defined in (5.4) is a solution for $\mathbf{G}(\mathbf{z})$ of (5.5).

One issue with the above result is that it only deals with ordinary polynomial matrices. To make it applicable for multidimensional filter banks with general FIR filters, we need to extend this result from polynomial matrices to Laurent polynomial matrices. This problem was first tackled by Park et al. [74, 76], in which they propose a method to transform Laurent polynomials into ordinary polynomials through a series of elementary matrix multiplications. A simpler and more direct method for this problem was first developed in [104] for the special case $M = 1$ (which is also known as the Rabinowitsch trick [58]). That method was then generalized for arbitrary M in [53] as follows.

Proposition 5.2. [53] Suppose $\mathbf{H}(\mathbf{z})$ is a given $N \times M$ Laurent polynomial matrix of d variables. Consider the $(N + M) \times M$ polynomial matrix

$$\mathbf{H}'(\mathbf{z}, w) = \begin{bmatrix} \mathbf{z}^{\mathbf{m}} \mathbf{H}(\mathbf{z}) \\ (1 - z_1 \cdots z_d w) \mathbf{I}_M \end{bmatrix}, \quad (5.6)$$

where $\mathbf{m} \in \mathbb{Z}^d$ is such that $\mathbf{z}^{\mathbf{m}} \mathbf{H}(\mathbf{z})$ is a polynomial matrix and w is an additional variable. Then there exists a Laurent polynomial matrix $\mathbf{G}(\mathbf{z})$ such as $\mathbf{G}(\mathbf{z})\mathbf{H}(\mathbf{z}) = \mathbf{I}_M$ if and only if there exists a polynomial matrix $\mathbf{G}'(\mathbf{z})$ such as $\mathbf{G}'(\mathbf{z})\mathbf{H}'(\mathbf{z}) = \mathbf{I}_M$. In that case, a solution of the

Laurent polynomial left inverse problem is obtained from a solution of the associated polynomial left inverse problem by

$$G_{ij}(z) = z^m G'_{ij}(z, z_1^{-1} \dots z_d^{-1}), \quad i = 1, \dots, M; j = 1, \dots, N. \quad (5.7)$$

We will demonstrate this result through a numerical example.

Example 5.3. Suppose that

$$H(z) = \begin{bmatrix} 1 & 1 \\ z_1^{-1} z_2^2 + 3z_1^{-1} & z_1^{-1} z_2^2 + z_1^{-1} \end{bmatrix}$$

is the analysis polyphase matrix of a given 2D filter bank. We can apply Proposition 5.2 to design a synthesis polyphase matrix $G(z)$ so that the perfect reconstruction condition (4.18) is satisfied. Using the software Singular [25] we can solve the problem as follows.

```
> ring R = 0,(z(1),z(2),w),dp;  ▷ R is a ring with 3 variables;
dp specifies degree reverse lexicographical ordering
> matrix H'[4][2];
> H'=z(1),z(1),z(2)^2+3,z(2)^2+1,1-z(1)*z(2)*w,0,
    0,1-z(1)*z(2)*w;
> print(H');
z(1), z(1),
z(2)^2+3, z(2)^2+1,
-z(1)*z(2)*w+1, 0,
0, -z(1)*z(2)*w+1    ▷ H' is the 4 × 2 matrix as defined in (5.6)
with m = (1,0).
> matrix I[2][2] = unitmat(2);  ▷ I is the 2 × 2 identity
matrix.
> matrix G'[2][4];
> G'=transpose(lift(transpose(H'),I)); ▷ lift(A,B) uses the
aforementioned Gröbner bases algorithm to return a matrix X such
as AX = B. transpose is used to convert a right inverse into a left
inverse.
> print(G');
-1/2*z(2)^3*w-1/2*z(2)*w, 1/2*z(1)*z(2)*w,1,0,
1/2*z(2)^3*w+3/2*z(2)*w, -1/2*z(1)*z(2)*w,0,1
```


Given this output \mathbf{G}' , we obtain, according to Proposition 5.2, the following synthesis polyphase matrix

$$\mathbf{G}(\mathbf{z}) = \begin{bmatrix} -\frac{1}{2}z_2^2 - \frac{1}{2} & \frac{1}{2}z_1 \\ \frac{1}{2}z_2^2 + \frac{3}{2} & -\frac{1}{2}z_1 \end{bmatrix}$$

as a left inverse of $\mathbf{H}(\mathbf{z})$.

Once we have obtained a particular left-inverse, we can then parametrize the complete set of all left-inverses as follows:

Proposition 5.4. [75, 104] Suppose $\mathbf{H}(\mathbf{z})$ is an $N \times M$ Laurent polynomial matrix and $\tilde{\mathbf{G}}(\mathbf{z})$ is an $M \times N$ Laurent polynomial matrix such that $\tilde{\mathbf{G}}(\mathbf{z})\mathbf{H}(\mathbf{z}) = \mathbf{I}$. Then $\mathbf{G}(\mathbf{z})$ is an $M \times N$ Laurent polynomial matrix such that $\mathbf{G}(\mathbf{z})\mathbf{H}(\mathbf{z}) = \mathbf{I}$ if and only if $\mathbf{G}(\mathbf{z})$ can be written as

$$\mathbf{G}(\mathbf{z}) = \tilde{\mathbf{G}}(\mathbf{z}) + \mathbf{A}_1(\mathbf{z})(\mathbf{I} - \mathbf{H}(\mathbf{z})\tilde{\mathbf{G}}(\mathbf{z})), \quad (5.8)$$

or alternatively,

$$\mathbf{G}(\mathbf{z}) = \tilde{\mathbf{G}}(\mathbf{z}) + \mathbf{A}_2(\mathbf{z})\text{Syz}(\mathbf{h}_1(\mathbf{z}), \mathbf{h}_2(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})). \quad (5.9)$$

where $\mathbf{A}_1(\mathbf{z})$ and $\mathbf{A}_2(\mathbf{z})$ are arbitrary Laurent polynomial matrices of appropriate sizes, and $\text{Syz}(\mathbf{h}_1(\mathbf{z}), \mathbf{h}_2(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z}))$ is the *syzygy* [19] of the row vectors $\{\mathbf{h}_1(\mathbf{z}), \mathbf{h}_2(\mathbf{z}), \dots, \mathbf{h}_N(\mathbf{z})\}$ of $\mathbf{H}(\mathbf{z})$.

The above proposition contains two complete characterizations of all left inverses $\mathbf{G}(\mathbf{z})$ of a given matrix $\mathbf{H}(\mathbf{z})$. The first formulation, given in (5.8), is generally simpler to implement. The second method, given in (5.9), is more complicated but it requires fewer parameters: The size of the free parameter matrix $\mathbf{A}_1(\mathbf{z})$ in (5.8) is $M \times N$, while the smallest possible matrix size of $\mathbf{A}_2(\mathbf{z})$ in (5.9) is $M \times (N - M)$. Meanwhile, the syzygy used in (5.9) can be computed in most computer algebra softwares using Gröbner bases.

In several application scenarios, the desired multidimensional signals are acquired through a multichannel system that can be modeled by a filter bank (see, for example, [16, 36, 38, 42, 97]). The analysis

part of Figure 4.7(a) in Section 4 depicts such an acquisition system in which the desired signal x is captured via multiple filtered and down-sampled signals $\{y_0, y_1, \dots, y_{N-1}\}$, potentially contaminated by noise. We assume that the analysis filters $\{H_1(z), H_2(z), \dots, H_{N-1}(z)\}$ and the sampling matrix \mathbf{M} are known, for example through a calibration stage. In the operation stage, the goal is to efficiently reconstruct the desired signal x from the given measurements $\{y_0, y_1, \dots, y_{N-1}\}$.

By expressing the acquisition system *equivalently* in the polyphase domain [see the analysis part of Figure 4.7(b)], we can reconstruct the input signal x through a reconstruction polyphase system as shown in the synthesis part of Figure 4.7(b). We just need to design the synthesis polyphase matrix $\mathbf{G}(z)$ to satisfy the perfect reconstruction condition with $\mathbf{H}(z)$, that is,

$$\mathbf{G}(z)\mathbf{H}(z) = \mathbf{I}_M.$$

In the noiseless case, any particular inverse $\mathbf{G}(z)$ obtained through the application of Proposition 5.2 would be sufficient. In the presence of noise, we need to search for the optimal inverse (in terms of noise robustness) among all possible inverses as characterized in Proposition 5.4. For more details of this application of Gröbner bases in MD signal reconstruction from multichannel acquisition, see [52].

5.2 The Mapping-Based Design for MD Filter Banks

One challenge associated with the above Gröbner bases approach is to design filters with good frequency responses. In many filter bank design problems, we aim to achieve a desired frequency partition with our channel filters, but it is difficult to impose such frequency response constraints on the filters obtained in the form of (5.8) and (5.9). In the following sections, we will present two alternative design methods that take filter frequency responses into account.

A popular approach to the design of nonseparable multidimensional filter banks with good frequency responses is based on mapping (that is, transformation of variables) [49, 67, 84, 92]. Although the mapping approach imposes certain restrictions on the kind of filters we can get,

it brings many important advantages, including, as we will demonstrate later, efficient implementation via lifting/ladder structures.

Our discussions in this section will be restricted to a particular class of two-channel filter banks in 2-D, whose sampling matrices are

$$\mathbf{D}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that we lose no generality by focusing on this special case. In fact, as shown in Section 4.2 (see, in particular, Figure 4.4), any two-channel critically sampled filter bank in 2-D can be converted, through multi-rate identities (and possibly, dimension swapping), to an equivalent filter bank using \mathbf{D}_1 .

We show in Figure 5.1 several possible choices of the ideal frequency responses of the channel filters $H_0(\boldsymbol{\xi})$ and $G_0(\boldsymbol{\xi})$. Note that the other two filters $H_1(\boldsymbol{\xi})$ and $G_1(\boldsymbol{\xi})$ are supported on complementary regions. Using the tools presented in Section 3, we can easily verify that all the frequency regions in Figure 5.1 can indeed be critically sampled by the rectangular lattice spanned by \mathbf{D}_1 .

Assume now that the filter bank achieves perfect reconstruction with FIR filters. It then follows from the polyphase domain characterization in Section 4.4 that the filters $H_1(\mathbf{z})$ and $G_1(\mathbf{z})$ are completely specified by $H_0(\mathbf{z})$ and $G_0(\mathbf{z})$ as in (4.23) and (4.24), respectively. Consequently, our task boils down to designing $H_0(\mathbf{z})$ and $G_0(\mathbf{z})$, which have the desired frequency responses and satisfy the polyphase-domain

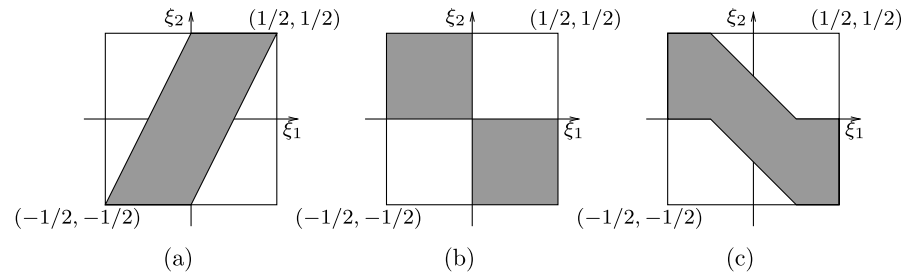


Fig. 5.1 Some examples of the frequency support configurations for 2D two-channel filter banks. All the frequency regions shown here can be critically sampled by the diagonal matrix \mathbf{D}_1 .

condition (4.21), or equivalently,¹ in the filter domain,

$$H_0(z_1, z_2)G_0(z_1, z_2) + H_0(-z_1, z_2)G_0(-z_1, z_2) = 2. \quad (5.10)$$

To achieve the above goal, we use the mapping-based technique originally proposed by Tay and Kingsbury [92], which can be regarded as a generalization of the McClellan transform [67]. The main ideas of this mapping approach can be described as follows. First, we design two 1D polynomials $h(x), g(x)$ that satisfy the following Bézout identity:

$$h(x)g(x) + h(-x)g(-x) = 2. \quad (5.11)$$

Meanwhile, we control the shape of the polynomials so that $h(x) \approx g(x)$, and that

$$h(1) \approx g(1) \approx \sqrt{2}, \quad \text{and} \quad h(-1) \approx g(-1) \approx 0. \quad (5.12)$$

The details of how to find these polynomials will be given in several design examples shown later.

Next, we find a mapping kernel $K(\mathbf{z})$ that satisfies the following property

$$K(-z_1, z_2) = -K(z_1, z_2). \quad (5.13)$$

Now the key step in the mapping-based design is to substitute the variable x in the polynomials with $K(\mathbf{z})$ and define the filters $H_0(\mathbf{z})$ and $G_0(\mathbf{z})$ as

$$H_0(\mathbf{z}) = h(K(\mathbf{z})) \quad \text{and} \quad G_0(\mathbf{z}) = g(K(\mathbf{z})). \quad (5.14)$$

Several nice properties come from this mapping of variables. First, the Bézout identity (5.11) of the 1D polynomials and the property (5.13) of the mapping kernel guarantee the perfect reconstruction condition in (5.10). Second, if the kernel $K(\mathbf{z})$ is a FIR and zero phase filter, then the mapping filters $H_0(\mathbf{z})$ and $G_0(\mathbf{z})$ in (5.14) are also FIR and have zero phase.

To control the frequency responses, we can design the mapping kernel such that, in the frequency domain, $K(e^{j2\pi\boldsymbol{\xi}}) \approx 1$ for $\boldsymbol{\xi} \in \mathcal{K}$ and

¹The equivalence between (4.21) and (5.10) follows from that fact that the left-hand sides of the two formulae are both calculating the first polyphase component of $H_0(\mathbf{z})G_0(\mathbf{z})$.

$K(e^{j2\pi\xi}) \approx -1$ for $\xi \in \mathcal{K}^C$, where $\mathcal{K} \subseteq \mathbb{R}^2$ is the desired passband support of the filter bank, and \mathcal{K}^C is the complement of \mathcal{K} . (See Figure 5.1 for examples of \mathcal{K} .) It then follows from the shape of the polynomials specified in (5.12) that in the Fourier domain the mapped filters in (5.14) can be written as $H_0(e^{j2\pi\xi}) \approx G_0(e^{j2\pi\xi}) \approx \sqrt{2}\mathbb{1}_{\mathcal{K}}(\xi)$, where $\mathbb{1}_{\mathcal{K}}(\xi)$ is the indicator function on \mathcal{K} .

In what follows, we will describe in detail the design of the mapping kernel $K(\mathbf{z})$ and 1D polynomials $h(x), g(x)$ so that they satisfy their respective conditions.

First, we rewrite the condition (5.13) for the mapping kernel as

$$K(z_1, z_2) + K(-z_1, z_2) = 0.$$

Let $k[\mathbf{n}]$ be the spatial domain representation of the kernel. We can easily verify that the above condition is equivalent to requiring

$$k[2n_1, n_2] = 0, \quad \text{for all } n_1, n_2 \in \mathbb{Z},$$

or, via the sampling matrix notation,

$$k[\mathbf{D}_1 \mathbf{n}] = 0 \quad \text{for all } \mathbf{n} \in \mathbb{Z}^2. \quad (5.15)$$

Let \mathcal{K} be the ideal frequency support we want to achieve. Note that \mathcal{K} can be critically sampled by \mathbf{D}_1 . It then follows from (3.11) that

$$\widehat{\mathbb{1}}_{\mathcal{K}}(\mathbf{D}_1 \mathbf{n}) = \frac{\delta[\mathbf{n}]}{2},$$

where $\widehat{\mathbb{1}}_{\mathcal{K}}(\mathbf{x}) = \int \mathbb{1}_{\mathcal{K}}(\xi) e^{-j2\pi\xi \cdot \mathbf{x}} d\xi$ is the Fourier transform of the indicator function $\mathbb{1}_{\mathcal{K}}(\xi)$.

If we choose the mapping kernel to be

$$k_0[\mathbf{n}] \stackrel{\text{def}}{=} 2\widehat{\mathbb{1}}_{\mathcal{K}}(\mathbf{n}) - \delta[\mathbf{n}], \quad (5.16)$$

then the condition in (5.15) is automatically satisfied. Meanwhile, the kernel $k_0[\mathbf{n}]$ defined above also has the desired frequency response: $K_0(e^{j2\pi\xi}) = 1$ for $\xi \in \mathcal{K}$ and $K_0(e^{j2\pi\xi}) = -1$ otherwise.

In order to obtain an FIR kernel, a simple approach is to truncate $k_0[\mathbf{n}]$ and get

$$k[\mathbf{n}] = k_0[\mathbf{n}]w[\mathbf{n}],$$

where $w[\mathbf{n}]$ is a zero-phase window (for example, the centrally-shifted Kaiser window) with finite spatial support. Obviously, the condition in (5.15) still holds for the truncated kernel $k[\mathbf{n}]$.

Example 5.5. Let \mathcal{K} be the parallelogram-shaped region shown in Figure 5.1(a). We can obtain the ideal kernel $k_0[\mathbf{n}]$ in (5.16) as

$$\begin{aligned} k_0[\mathbf{n}] &= 2 \left(\int_{\xi \in \mathcal{K}} e^{-j2\pi \xi \cdot \mathbf{n}} d\xi \right) - \delta[\mathbf{n}] \\ &= \text{sinc}\left(\frac{n_1}{2}\right) \text{sinc}\left(\frac{n_1 + 2n_2}{2}\right) - \delta[\mathbf{n}], \end{aligned} \quad (5.17)$$

where $\text{sinc}(x) \stackrel{\text{def}}{=} \sin(\pi x)/(\pi x)$ is the normalized sinc function. A closed-form formula for the integral in (5.17) is available when \mathcal{K} is an arbitrary polygonal region [63].

We can easily verify that the required condition (5.15) is indeed satisfied, that is, $k_0[2n_1, n_2] \equiv 0$. To obtain FIR filters, we truncate $k_0[\mathbf{n}]$ as

$$k[\mathbf{n}] = k_0[\mathbf{n}] w_\beta[n_1 - N] w_\beta[n_2 - N],$$

where $w_\beta[n]$ is a Kaiser window of length $2N + 1$ and shape parameter β . Figure 5.2 shows the frequency response of the designed kernel, when we choose $N = 10$ and $\beta = 4.0$. We can see that the obtained kernel indeed has good frequency domain localization on the desired parallelogram-shaped support region.

Although our discussions so far have focused on FIR filters, we note that the choices for suitable kernels $K(\mathbf{z})$ are much more flexible. In [93], for example, IIR kernels have been exploited to obtain much sharper frequency cut-off than FIR filters of similar complexity.

Next, we consider the design of the two 1-D polynomials $h(x)$ and $g(x)$. The Bézout condition given in (5.11) implies that the two polynomials $h(x)$ and $g(-x)$ are coprime, and hence we can always factor $h(x)$ and $g(-x)$ into lifting/ladder structures [3, 91] by using the Euclidean algorithm. For simplicity, we go one step farther by imposing that the

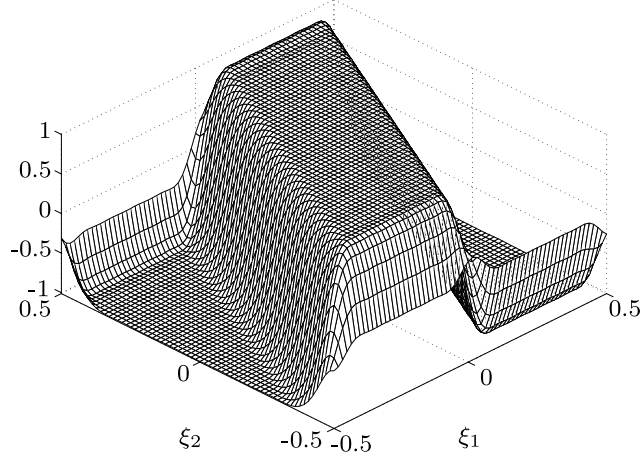


Fig. 5.2 The frequency response of the designed FIR mapping kernel with a parallelogram-shaped support.

prediction and update filters in the lifting/ladder structures be monomials, that is, we let

$$\begin{bmatrix} h(x) \\ g(-x) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c_n x & 1 \end{bmatrix} \begin{bmatrix} 1 & c_{n-1} x \\ 0 & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ c_2 x & 1 \end{bmatrix} \begin{bmatrix} 1 & c_1 x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k \\ 1/k \end{bmatrix}, \quad (5.18)$$

where k and c_1, \dots, c_n are free parameters. Writing

$$h(x)g(x) + h(-x)g(-x) = \begin{bmatrix} h(x) \\ g(-x) \end{bmatrix}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} h(-x) \\ g(x) \end{bmatrix},$$

we can easily verify that, for the polynomials specified in (5.18), the Bézout identity always holds, independent of the choices of the parameters k and c_1, \dots, c_n .

To satisfy the shape requirement (5.12) for 1D polynomials, we impose the constraints that $h(-1) = g(-1) = 0$ and $h(1) = \sqrt{2}$. Note that $g(1) = \sqrt{2}$ is then automatically guaranteed by the Bézout identity. Under these constraints, we can write k , c_1 and c_2 as functions of the other parameters. For example, when $n = 4$, we have

$$k = \frac{1 + c_3 + c_3 c_4}{\sqrt{2}}, \quad c_1 = \frac{(1 + c_3 c_4)^2 - c_3^2}{2}, \quad \text{and} \quad c_2 = \frac{-1 - c_4}{1 + c_3 + c_3 c_4}. \quad (5.19)$$

Table 5.1. The designed lifting/ladder coefficients.

k	c_1	c_2	c_3	c_4
0.998605	0.235258	-0.541031	0.539535	-0.235934

The remaining degrees of freedom from c_3 and c_4 can then be exploited to endow additional properties to the filter bank.

Example 5.6. When $h(x) \approx g(x)$, the analysis and synthesis filters in the mapping filter bank are approximately equal to each other. In other words, the designed filter bank is close to being orthogonal. To achieve this goal, we apply the relations given in (5.19) and solve the following unconstrained nonlinear optimization problem

$$\operatorname{argmin}_{c_3, c_4} \max_{|x| \leq 1} |h(x) - g(x)|,$$

with two free variables. Table 5.1 shows the optimized parameters c_1, \dots, c_4 and K (obtained using the MATLAB optimization toolbox). We also plot in Figure 5.3 the polynomials $h(x)$ and $g(-x)$ for $n = 4$ with their parameters taken from Table 5.1. We can see that $h(x)$ is indeed approximately a reversed version of $g(-x)$, that is, $h(x) \approx g(x)$.

An important advantage in using the factorization (5.18) is computational efficiency. Recall that in our mapping based design, the two lowpass filters $H_0(\mathbf{z})$ and $G_0(\mathbf{z})$ are obtained by replacing x in 1D polynomials with 2D mapping kernels $K(\mathbf{z})$, as in (5.14). For the high-pass filter $H_1(\mathbf{z})$, we can deduce from the polyphase characterization in (4.23) the following biorthogonal identity

$$H_1(\mathbf{z}) = z_1 G_0(-z_1, z_2). \quad (5.20)$$

Note that for simplicity, we have set the free parameters in (4.23) to $\alpha = 1$ and $\mathbf{m} = \mathbf{0}$. Replacing $G_0(\mathbf{z})$ by $g(K(\mathbf{z}))$ and using the property (5.13) of the kernel $K(\mathbf{z})$, we can rewrite (5.20) as

$$H_1(\mathbf{z}) = z_1 g(K(-z_1, z_2)) = z_1 g(-K(\mathbf{z})). \quad (5.21)$$

Similarly, from (4.24), we can write the other highpass filter $G_1(\mathbf{z})$ as

$$G_1(\mathbf{z}) = z_1^{-1} h(-K(\mathbf{z})).$$

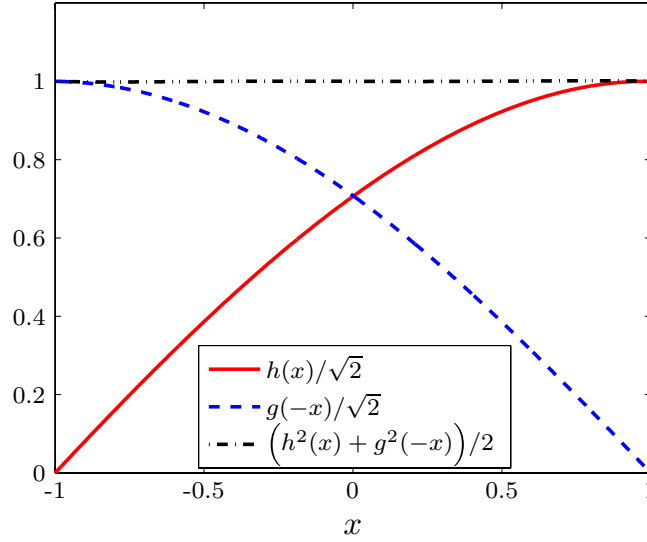
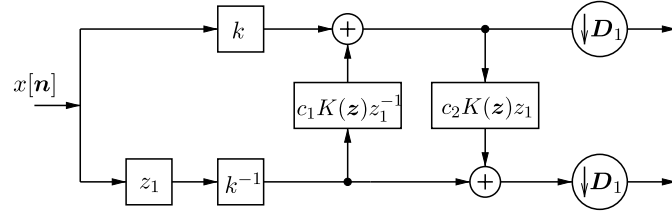


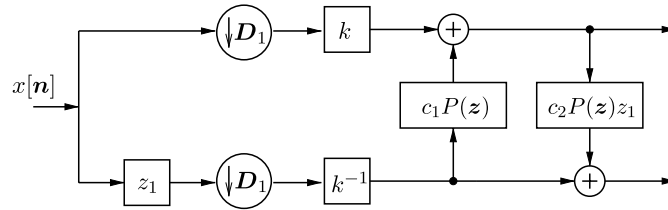
Fig. 5.3 Designed 1D polynomials $h(x)$ and $g(-x)$ which lead to approximately orthogonal filters.

Using the factorization form (5.18), we can thus implement our mapped filter bank via a lifting/ladder structure shown in Figure 5.4(a). Note that we only show the analysis part, while the synthesis part will be exactly symmetric. Compared with a direct implementation of the filter bank (in the form of Figure 4.8), the scheme in Figure 5.4(a) has advantages such as reduced computational complexities and robustness against the quantization of filter coefficients.

To quantify the improvement in computational efficiency, we assume that the lifting/ladder structure in (5.18) has n stages and that the 2D mapping kernel $K(\mathbf{z})$ is of size $m \times m$. It is then easy to verify that, after the mapping steps in (5.14) and (5.21), one of the channel filters — either $H_0(\mathbf{z})$ or $H_1(\mathbf{z})$, depending on the parity of n — will have a support size equal to $((m-1)n+1) \times ((m-1)n+1)$. Consequently, the direct implementation shown in Figure 4.8 requires $\mathcal{O}(n^2(m-1)^2)$ arithmetic operations per input sample. In contrast, with the lifting/ladder implementation shown in Figure 5.4(a), the number of arithmetic operations per input samples becomes $\mathcal{O}(nm^2)$, that is, an n -fold reduction.



(a) The lifting/ladder implementation



(b) An equivalent, but more efficient, implementation in the polyphase domain

Fig. 5.4 Efficient implementations of a two-channel filter bank designed by mapping. Only the analysis parts are shown. The synthesis parts are exactly symmetric.

It is possible to further reduce the computational complexity by working in the polyphase domain. The condition (5.15) implies that one of the two polyphase components of $K(\mathbf{z})$ is all zero. It follows that we can write

$$K(\mathbf{z}) = z_1 P(z_1^2, z_2), \quad (5.22)$$

where $P(\mathbf{z})$ is the remaining (nonzero) polyphase component of $K(\mathbf{z})$. Substituting (5.22) into the lifting/ladder structure in Figure 5.4(a) and applying the multirate identities, we can obtain an equivalent implementation of the filter bank in the polyphase domain, shown in Figure 5.4(b). Since the spatial domain support of the polyphase filter $P(\mathbf{z})$ is only half of that of $K(\mathbf{z})$, the polyphase implementation shown in Figure 5.4(b) is more efficient than the original scheme in Figure 5.4(a).

Furthermore, for certain frequency support shapes (notably, the diamond, fan, and checkerboard shapes shown in Figure 4.9), the polyphase component $P(\mathbf{z})$ can be designed to be a separable filter [80].

This desirable property allows us to implement the 2D filter bank by simple 1D operations in the polyphase domain, even though the original filter bank is not separable. The computational complexity in this case becomes $\mathcal{O}(nm)$ per input sample, a significant improvement over $\mathcal{O}(n^2(m-1)^2)$ as required by direct implementations.

5.3 Designing Filter Banks in the Frequency Domain

The design methods presented in the previous two sections aim at obtaining perfect reconstruction filter banks using FIR filters. We can greatly simplify the design problem if we do not confine ourselves to using FIR (or even IIR) filters and work in the frequency domain instead. As suggested by several independent works [37, 71, 103], multi-dimensional filters designed in the frequency domain can achieve quite satisfactory performance.

In the next section on constructing directional filter banks, we will use a 3-D nonsubsampled filter bank which decomposes the frequency spectrum of the input signal into three hourglass-shaped subbands (see Figure 6.9 for an illustration). In what follows, we use this filter bank as an example to highlight the frequency domain design method.

As shown in Figure 6.9, we use $H_i(\xi_1, \xi_2, \xi_3)$ and $G_i(\xi_1, \xi_2, \xi_3)$, ($i = 1, 2, 3$) to represent the three analysis and synthesis filters in the hourglass filter bank, respectively. As the first step of simplification, we assume the three analysis filters are rotated versions of each other, that is,

$$H_2(\xi_1, \xi_2, \xi_3) = H_1(\xi_2, \xi_3, \xi_1) \quad \text{and} \quad H_3(\xi_1, \xi_2, \xi_3) = H_1(\xi_3, \xi_1, \xi_2).$$

The same constraint also applies to the synthesis filters. Meanwhile, if the filter bank implements a tight frame expansion, we need the synthesis filters to be the time-reversed versions of the corresponding analysis filters, that is,

$$G_i(\boldsymbol{\xi}) = H_i(-\boldsymbol{\xi}) = H_i(\boldsymbol{\xi}),$$

for $i = 1, 2, 3$, where the second equality comes from the symmetry in the ideal frequency responses of $H_i(\boldsymbol{\xi})$. Combining the above two constraints, we get the condition for perfect reconstruction as

$$H_1^2(\xi_1, \xi_2, \xi_3) + H_1^2(\xi_2, \xi_3, \xi_1) + H_1^2(\xi_3, \xi_1, \xi_2) = 1. \quad (5.23)$$

In our design, we set

$$H_1(\xi_1, \xi_2, \xi_3) = \sqrt{\frac{K(\xi_1, \xi_2, \xi_3)^\lambda}{K(\xi_1, \xi_2, \xi_3)^\lambda + K(\xi_2, \xi_3, \xi_1)^\lambda + K(\xi_3, \xi_1, \xi_2)^\lambda}}, \quad (5.24)$$

where λ is a positive even integer and $K(\xi_1, \xi_2, \xi_3)$ is a positive and 2π periodic function of ξ_1, ξ_2 and ξ_3 . We can verify that the perfect reconstruction condition in (5.23) is satisfied by arbitrary choices of λ and $K(\xi_1, \xi_2, \xi_3)$. To control the filter frequency responses so that they approximate the desired hourglass shape, we let

$$K(\xi_1, \xi_2, \xi_3) = E(\xi_1, \xi_2)E(\xi_1, \xi_3),$$

where $E(\cdot, \cdot)$ is a bivariate 2π periodic function such that the values of $E(\xi_1, \xi_2)$ are approximately equal to one in the dark region in Figure 5.5(a) and zero in the white region, with smooth transition regions between the two. There can be many ways to design $E(\cdot, \cdot)$. For example, we can truncate the ideal “sinc”-like 2-D signal corresponding to the fan-shaped support in Figure 5.5(a) with a smooth Kaiser window and take $E(\xi_1, \xi_2)$ to be the Fourier transform of the truncated signal. The parameter λ in (5.24) can be used to adjust the sharpness of the frequency response.

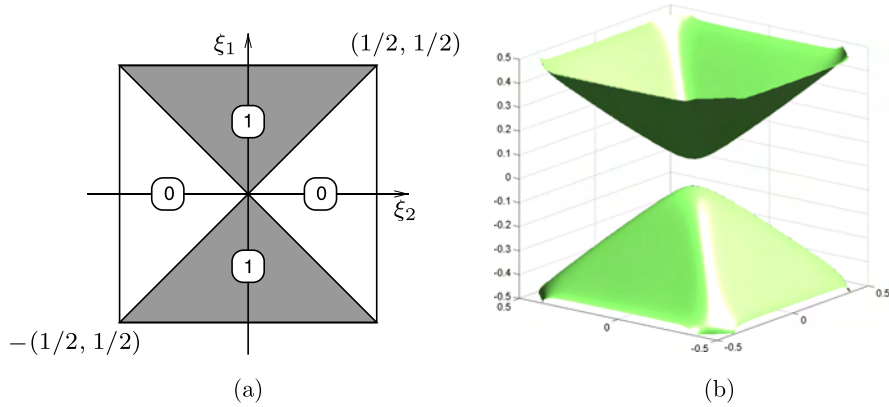


Fig. 5.5 (a) $E(\xi_1, \xi_2)$ approximately takes the value one in the dark region and the value zero in the white region. (b) The frequency response of one hourglass filter designed by the proposed frequency-domain method. The responses of the other two filters are rotationally-symmetric to this one.

Figure 5.5(b) shows an isosurface of the frequency response of one analysis filter. We can see that the frequency response approximates the ideal hourglass shape fairly well. Note that the responses of the other two hourglass filters are rotated versions of this one.

In general, the channel filters as defined in (5.24) are neither FIR nor IIR, and thus they have to be implemented in the frequency domain. This can be done efficiently by using the fast Fourier transform (FFT). The filter bank decomposition in this case amounts to two FFTs (forward and inverse) and a small number of pointwise multiplications in the Fourier domain. It follows that the number of arithmetic operations for implementing such filter banks is $\mathcal{O}(N \log(N))$, where N is the number of input samples.

Finally, we note that the frequency domain method is not limited to the design of nonsubsampled filter banks. We refer readers to [37] for an example of designing a critically-sampled filter bank in the frequency domain.

6

Iterated and Directional Filter Banks

Building upon the concepts and tools we have developed so far, we will present in this section the constructions of iterated and directional filter banks. They serve as the essential components of several multiscale geometric representations for MD signals.

6.1 Directional Filter Banks

In 1992, Bamberger and Smith [2] introduced a 2D directional filter bank (DFB) that can be maximally decimated while achieving perfect reconstruction. The DFB is efficiently implemented via an ℓ -level tree-structured decomposition that leads to 2^ℓ subbands with wedge-shaped frequency partition as shown in Figure 6.1.

The original construction of the DFB in [2] involves modulating the input signal and using diamond-shaped filters. Furthermore, to obtain the desired frequency partition, a complicated tree expanding rule has to be followed (see [77] for details). As a result, the frequency regions for the resulting subbands do not follow a simple ordering as shown in Figure 6.1 based on the channel indices.

A new and simpler construction for the DFB was proposed in [27], which avoids modulating the input image and has a straightforward

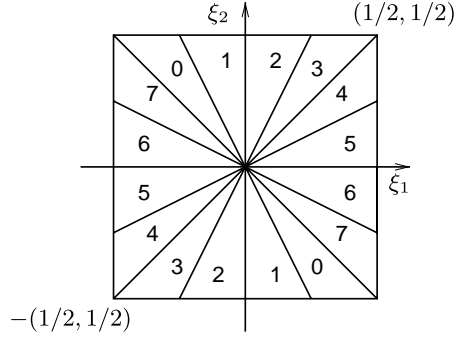


Fig. 6.1 The frequency partitioning of a 2-D directional filter bank with $2^3 = 8$ real wedge-shaped frequency bands.

rule for expanding the decomposition tree. The simplified DFB is intuitively constructed from two building blocks. The first building block is a two-channel quincunx filter bank [98] with fan filters that divides a 2D spectrum into two directions: horizontal and vertical [see Figure 6.3(a)]. The second building block of the DFB is a *shearing* operator, which amounts to just a reordering of image samples. Figure 3.5(b) shows an application of a shearing operator where a -45° direction edge becomes a vertical edge. By adding a pair of shearing operator and its inverse (“unshearing”) to before and after a quincunx filter bank as shown in Figure 6.4, we can obtain a different directional frequency partition while maintaining perfect reconstruction. Thus, the key in the DFB is to use an appropriate combination of shearing operators together with the two-direction partition of the quincunx filter bank at each node in a binary tree-structured filter bank, to obtain the desired 2D spectrum decomposition as shown in Figure 6.1.

The first two decomposition levels of the DFB are given in Figure 6.2, which leads to a frequency partitioning with four directional subbands. We chose the sampling matrices in the first and second level to be \mathbf{Q}_0 and \mathbf{Q}_1 , respectively (recall (3.3) and (3.4) for their definitions), so that the overall sampling after two levels is $\mathbf{Q}_0\mathbf{Q}_1 = 2\mathbf{I}_2$, or downsampling by two in each dimension.

Using the multirate identities we can transform the tree-structured filter bank in Figure 6.2 into its equivalent parallel form as shown in

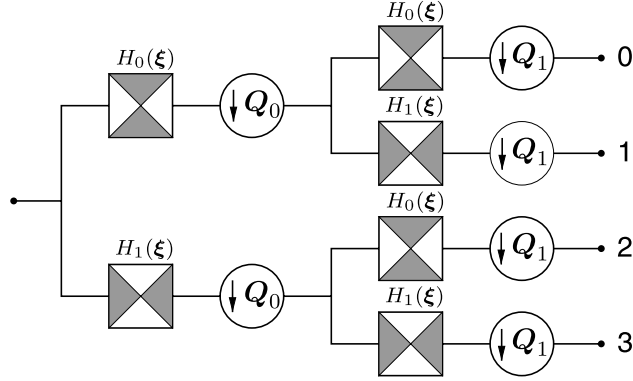


Fig. 6.2 The first two levels of the DFB. At each level, a QFB with fan filters is used. The dark regions represent the ideal frequency supports of the channel filters.

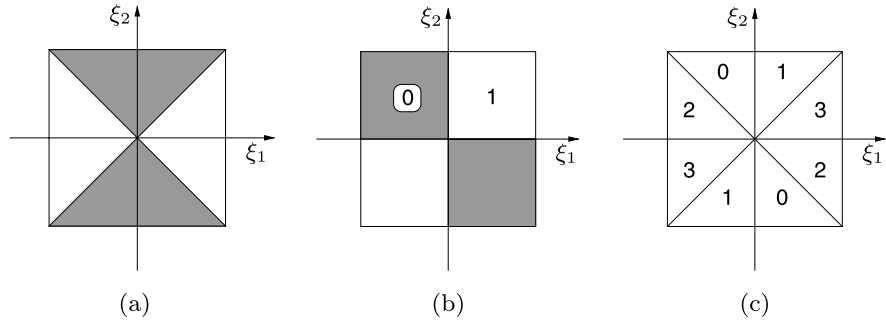


Fig. 6.3 The support configuration of the equivalent filters in the first two levels of the DFB. (a) First level: $H_i(\xi)$, $i \in \{0, 1\}$. (b) Second level: $H_j(Q_0^T \xi)$, $j \in \{0, 1\}$. (c) Combining two levels: $H_i(\xi)H_j(Q_0^T \xi)$, $i, j \in \{0, 1\}$. The equivalent filters correspond to four directional subbands labeled as 0, 1, 2, 3.



Fig. 6.4 Quincunx filter banks with resampling operations that are used in the third and higher levels of the DFB.

Figure 4.5. The support configurations of the equivalent filters when going through this transformation are depicted in Figure 6.3. We see that the combined equivalent filters of the four channels in the parallel form produce the desired directional subbands.

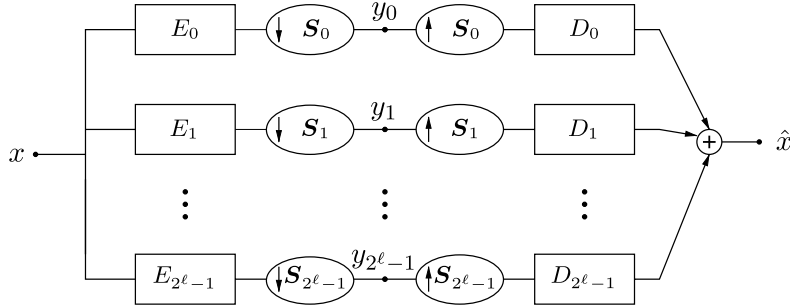


Fig. 6.5 The multichannel view of an ℓ -level tree-structured directional filter bank.

From the third level onwards, to achieve finer frequency partition, we can use quincunx filter banks together with resampling operations as shown in Figure 6.4. We omit further discussions on this step and refer readers to [27, Chapter 3] for details.

Generalizing our discussions on two-level DFBs, we can convert an ℓ -level tree-structured DFB to an equivalent parallel-structured filter bank with 2^ℓ channels, whose equivalent filters and overall sampling matrices are shown in Figure 6.5. Denote these equivalent (directional) synthesis filters as $D_k^{(\ell)}(\boldsymbol{\xi})$, $0 \leq k < 2^\ell$. The k th filter then corresponds to the k th subband as indexed in Figure 6.1. The corresponding overall sampling matrices can be shown [27] to have the following diagonal forms

$$\mathbf{S}_k^{(\ell)} = \begin{cases} \text{diag}(2^{\ell-1}, 2) & \text{for } 0 \leq k < 2^{\ell-1}, \\ \text{diag}(2, 2^{\ell-1}) & \text{for } 2^{\ell-1} \leq k < 2^\ell, \end{cases} \quad (6.1)$$

which means that all sampling operations are separable. The two sets of sampling matrices in (6.1) correspond to the *mostly horizontal* and *mostly vertical* set of directions, respectively.

From the equivalent parallel view of the DFB, we see that the family

$$\left\{ d_k^{(\ell)}[\mathbf{n} - \mathbf{S}_k^{(\ell)} \mathbf{m}] \right\}_{0 \leq k < 2^\ell, \mathbf{m} \in \mathbb{Z}^2}, \quad (6.2)$$

obtained by shifting the impulse responses of the equivalent synthesis filters $D_k^{(\ell)}$ over the sampling lattices by $\mathbf{S}_k^{(\ell)}$, provides a *basis* for discrete signals in $l^2(\mathbb{Z}^2)$. This basis exhibits both directional and localization properties. Figure 6.6 demonstrates this fact by showing the

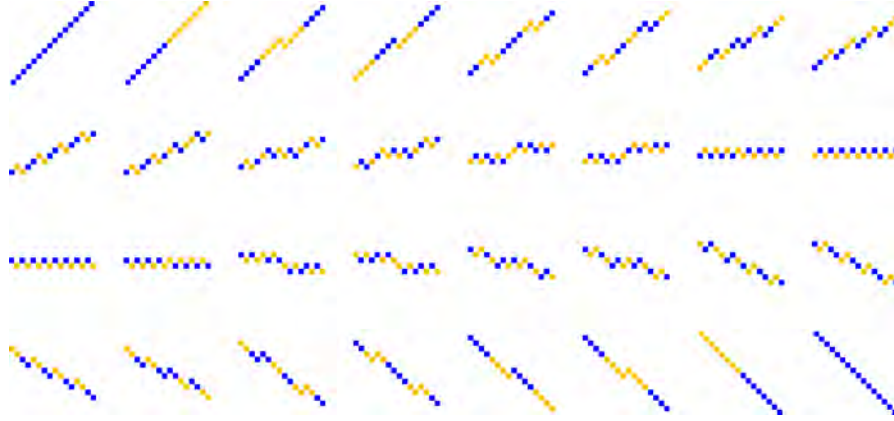


Fig. 6.6 Impulse responses of the first 32 channels of a 6-level DFB that uses Haar filters. All these channels correspond to the mostly horizontal directions. Blue and yellow squares represent $+1$ and -1 , respectively. Because the basis functions resemble “local lines,” we call them *Radonlets*.

impulse responses of equivalent filters from an example DFB. These basis functions have quasi-linear supports in space and span all directions. In other words, the basis (6.2) resembles a local Radon transform and is therefore called *Radonlets*. Furthermore, it can be shown [27] that if all building blocks — that is, the two-channel quincunx filter banks — use orthogonal filters, then the resulting DFB is orthogonal and (6.2) becomes an orthogonal basis.

6.2 Directional Filter Banks in Higher Dimensions

The concept of directional filter banks can be extended to higher dimensional cases [62]. In 3D, for example, the desirable passband support of a directional filter can be a symmetric pair of rectangular-based cones, as shown in Figure 6.7(a). A 3D directional filter bank can then cover the entire frequency space by a total of 48 (and in general $3 \times 4^\ell$ for $\ell \geq 0$) such cone-shaped filters, each radiating out from the origin at a different orientation.

The 2D version of the DFB presented in the previous section possesses several nice properties, including perfect reconstruction, efficient tree-structured implementation, simple expansion rule, and

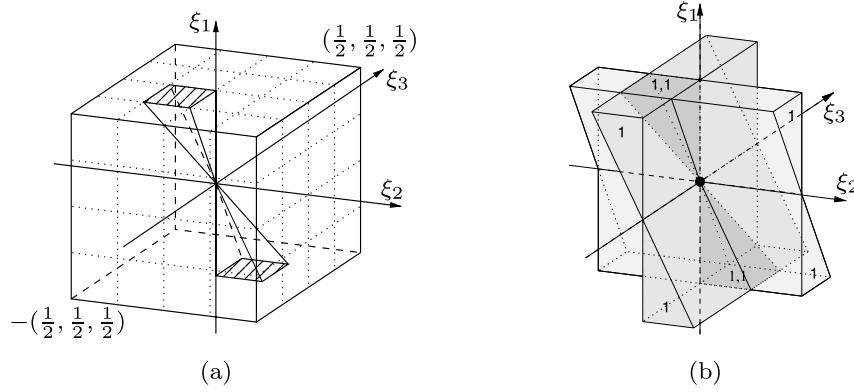


Fig. 6.7 (a) Frequency partitioning of the directional filter banks in 3D. (b) The cone-shaped subband support (dark gray region) is the intersection of two wedge-shaped supports (light gray regions).

critical sampling. In constructing the 3D DFB, all these desirable traits can be retained, except for one — critically sampling. As much as we would like to have a critically-sampled (that is, nonredundant) 3D DFB, this turns out to be theoretically impossible: In fact, applying the condition (3.11), we can show that, in 3D and beyond, cone-shaped spectrum supports [such as the one in Figure 6.7(a)] can never be critically sampled (see [63, Theorem 3] for a proof). In what follows, we describe the construction of a tree-structured 3D DFB with *low* redundancy.

We start from a simple geometric observation. As shown in Figure 6.7(b), the cone-shaped support (in dark gray color) can be obtained as the intersection of two wedge-shaped supports (in light gray color). This observation naturally leads to the idea of obtaining the frequency partitioning of the 3D DFB by a concatenation of two 2D DFB's on appropriate dimensions.

To be more specific, we show in Figure 6.8(a) a wedge-shaped decomposition of the 3D frequency spectrum, obtained by applying the 2D DFB on the (ξ_1, ξ_2) -plane of the 3D signal. Similarly, we show in Figure 6.8(b) the wedge-shaped frequency decomposition along the (ξ_1, ξ_3) -plane. Now, by taking the pairwise intersection of the wedge supports from Figures 6.8(a) and 6.8(b), we can get 16 “thinner” cone-shaped supports, as shown in Figure 6.8(c). In general, the entire region can be divided into $2^\ell \times 2^\ell = 4^\ell$ ($\ell \geq 0$) different square-based cones.

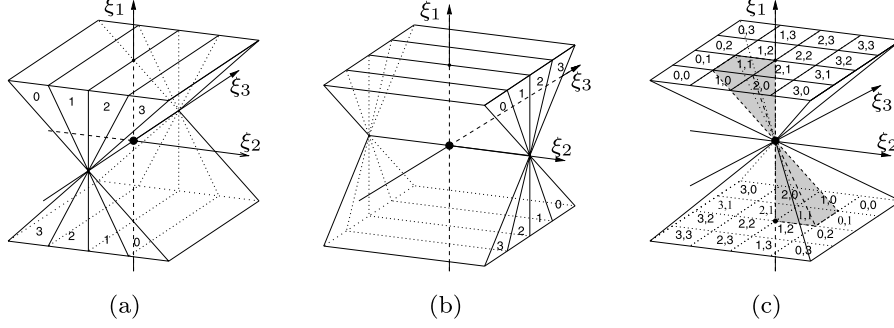


Fig. 6.8 (a) The ideal wedge-shaped frequency support of a 2D filter operating along the (n_1, n_2) -plane. (b) The wedge-shaped support of a 2D filter along the (n_1, n_3) -plane. (c) The ideal cone-shaped frequency decomposition.

The above idea seems to be working fine — but it has a hidden problem. To see this, we illustrate the flow of operations by applying two 2D DFB's sequentially along two signal planes as follows:

$$\rightarrow \boxed{E_i(\xi_1, \xi_2)} \rightarrow \underbrace{(\downarrow 2)}_{\text{along } n_1} \rightarrow \underbrace{(\downarrow 2^{\ell-1})}_{\text{along } n_2} \rightarrow \boxed{E_j(\xi_1, \xi_3)} \rightarrow \underbrace{(\downarrow 2)}_{\text{along } n_1} \rightarrow \underbrace{(\downarrow 2^{\ell-1})}_{\text{along } n_3}, \quad (6.3)$$

where $E_i(\xi_1, \xi_2)$ and $E_j(\xi_1, \xi_3)$ are two wedge-shaped analysis filters from the two DFBs, respectively. Since 2D DFBs are critically sampled, the filtering operations are always followed by downsampling. We know from (6.1) that the equivalent downsampling matrix for an ℓ -level DFB operating along the (n_1, n_2) -plane is a diagonal matrix, implemented separately as downsampling in the n_1 -dimension by 2 followed by downsampling in the n_2 -dimension by $2^{\ell-1}$. From multirate identities, the downsampling by 2 along the n_1 dimension inevitably scrambles the wedge-shaped frequency decomposition provided by $W_j(\xi_1, \xi_3)$. Thus it can be easily checked that the subsequent application of $W_j(\xi_1, \xi_3)$ will not provide the desired cone-shaped frequency decomposition as shown in Figure 6.8(c).

To overcome this problem, a simple fix is to use nonsubsampled 2D DFBs (that is, to remove all downsampling operations). However, this scheme leads to an excessively redundant system (2^ℓ -times redundant for ℓ levels of decomposition), which is undesirable in MD signal processing due to the high computational and memory costs.

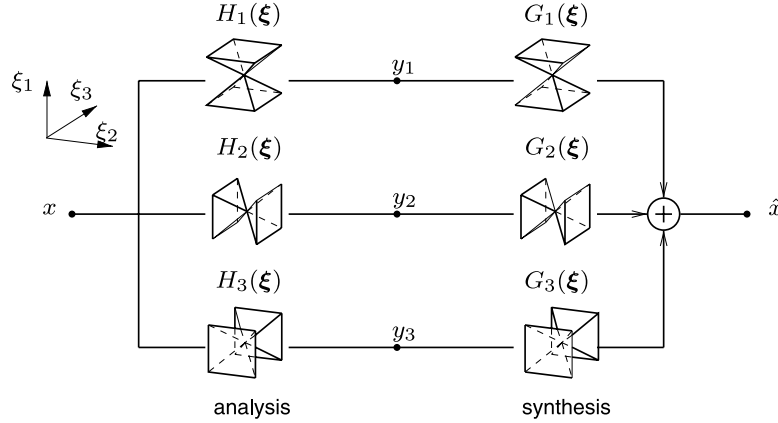


Fig. 6.9 The first level of decomposition: a three-channel nonsubsampled filter bank in 3D. The ideal frequency-domain supports of the component filters are hourglass-shaped regions, with their corresponding dominant directions aligned with the ξ_1 , ξ_2 , and ξ_3 axes, respectively.

In the following, we describe a different solution [62], which still makes use of the simple geometric observation in Figure 6.8, but does so without the excessive redundancy.

In the first level of decomposition, we employ a three-channel filter bank shown in Figure 6.9. This filter bank decomposes the 3D frequency spectrum of the input signal into three *hourglass-shaped* subbands, with their dominant directions aligned with the ξ_1 , ξ_2 , and ξ_3 axes, respectively. An important feature of the hourglass filter bank is that it is nonsubsampled, and hence brings a redundancy factor of 3 into the entire construction.

Figure 6.10 shows the block diagram of subsequent levels of decompositions after one of the three hourglass filters. (The situations for the other two channels can be inferred by symmetry.) After the 3D hourglass filter ($H_1(\xi)$), we sequentially decompose the signal by a concatenation of two 2D filter banks, with the first one, denoted as $\text{FB}_{12}^{(\ell)}$, operating along the (n_1, n_2) -plane and the second one, $\text{FB}_{13}^{(\ell)}$, along the (n_1, n_3) -plane.

The two filter banks $\text{FB}_{12}^{(\ell)}$ and $\text{FB}_{13}^{(\ell)}$ have exactly the same structure, and hence we will only focus on $\text{FB}_{12}^{(\ell)}$. As shown in Figure 6.11(b), the analysis part of $\text{FB}_{12}^{(\ell)}$ is constructed as an ℓ -level binary tree, with

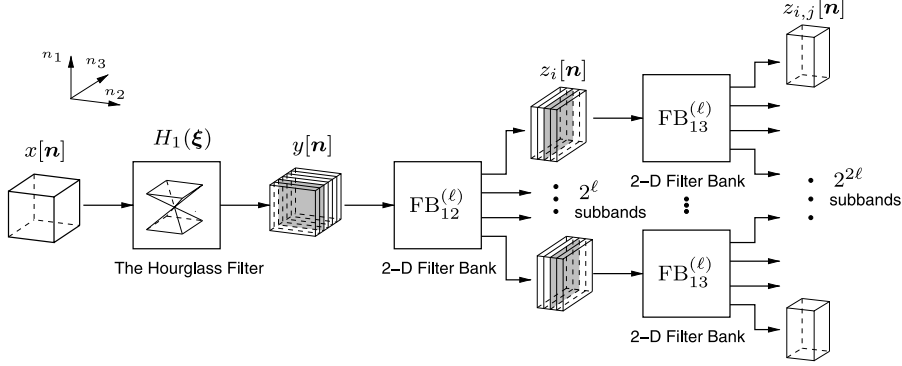


Fig. 6.10 One branch of the 3D DFB. The input signal $x[\mathbf{n}]$ first goes through a 3D filter $H_1(\xi)$, with an hourglass-shaped frequency domain support. The output $y[\mathbf{n}]$ is then fed into a 2D filter bank, denoted by $\text{FB}_{12}^{(\ell)}$, which operates on the (n_1, n_2) -planes (depicted by the gray colored slice under $y[\mathbf{n}]$). The ℓ -level tree-structured filter bank $\text{FB}_{12}^{(\ell)}$ produces 2^ℓ output subbands, denoted as $z_i[\mathbf{n}]$ for $0 \leq i < 2^\ell$. Each output is then fed into another 2D filter bank $\text{FB}_{13}^{(\ell)}$ operating on the (n_1, n_3) -planes (depicted by the gray colored slices under $z_i[\mathbf{n}]$). In the end, we get 4^ℓ outputs, represented by $z_{i,j}[\mathbf{n}]$ for $0 \leq i, j < 2^\ell$.

an appropriate resampling matrix $\mathbf{U}_k^{(\ell)}$ ($0 \leq k < 2^\ell$) attached to each of the 2^ℓ output channels of the tree. The building block of the tree is a two-channel 2D filter bank with a checkerboard-shaped frequency partition, as illustrated in Figure 6.11(b). Note that we need to attach two resampling operations, denoted as \mathbf{R}_0 and \mathbf{R}_1 , to channel 0 and channel 1, respectively. The corresponding sampling matrices are defined as

$$\mathbf{D}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \mathbf{R}_0 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \mathbf{R}_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}.$$

Using the multirate identities recursively, we can transform the tree-structured filter bank $\text{FB}_{12}^{(\ell)}$ into an equivalent parallel implementation with 2^ℓ channels. The i th channel contains a single equivalent filter $F_i(\xi_1, \xi_2)$ followed by an overall sampling matrix $\mathbf{M}_i^{(\ell)}$, which can be shown [62] to take the following form:

$$\mathbf{M}_i^{(\ell)} = \begin{pmatrix} 1 & 0 \\ 0 & 2^\ell \end{pmatrix}. \quad (6.4)$$

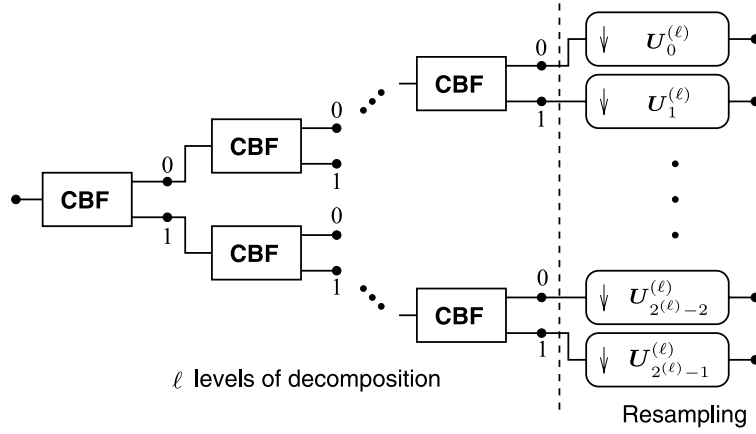
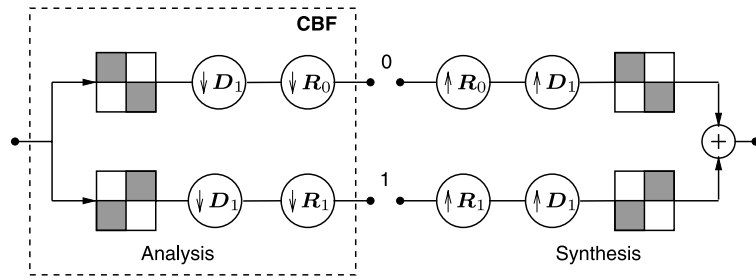

 (a) The construction of the filter bank $\text{FB}_{12}^{(\ell)}$.

 (b) The building block of $\text{FB}_{12}^{(\ell)}$: a resampled checkerboard filter bank.

Fig. 6.11 (a) The filter bank $\text{FB}_{12}^{(\ell)}$ is an ℓ -level tree-structured expansion of the checkerboard filter bank (denoted by CBF) given in (b), with resampling matrices $\mathbf{U}_k^{(\ell)}$ attached at the end of each channel. (b) A two-channel 2D checkerboard filter bank with resampling. The dark regions represent the ideal passband.

Consequently, we can simplify the flow of operations in each channel of Figure 6.10 as follows:

$$\begin{aligned}
 &\rightarrow \boxed{H_1(\xi_1, \xi_2, \xi_3)} \rightarrow \boxed{F_i(\xi_1, \xi_2)} \rightarrow \underbrace{(\downarrow 2^{\ell_1})}_{\text{along } n_2} \rightarrow \boxed{F_j(\xi_1, \xi_3)} \rightarrow \underbrace{(\downarrow 2^{\ell_1})}_{\text{along } n_3}, \\
 &\hspace{25em} (6.5)
 \end{aligned}$$

Compared with (6.3), a key difference here is that the downsampling operation after the first 2D filter $F_i(\xi_1, \xi_2)$ is performed on

the n_2 -dimension only, and is thus “transparent” to the second filter $F_j(\xi_1, \xi_3)$, which operates on the (n_1, n_3) dimensions. We can then freely interchange the order of the downsampling and filtering, and obtain the equivalent filter as

$$H_1(\xi_1, \xi_2, \xi_3)F_i(\xi_1, \xi_2)F_j(\xi_1, \xi_3). \quad (6.6)$$

We can show [62] that the two filters $F_i(\xi_1, \xi_2)$ and $F_j(\xi_1, \xi_3)$, in combination with the hourglass filter $H_1(\xi_1, \xi_2, \xi_3)$, can decompose the 3D spectrum into wedge-shaped regions as in Figures 6.8(a) and 6.8(b), respectively. Consequently, their combination in (6.6) leads to the desired 3D cone-shaped filters as in Figure 6.8(c).

As a final note: although our discussions focus on the 3D case, the above construction can be easily extended to higher dimensions, where the ideal frequency supports of directional filters are hypercube-based cones. In general, these directional filter banks are d -times redundant where $d = 3, 4, \dots$ is the dimensionality. More details on these higher dimensional generalizations can be found in [62].

7

Multiscale Geometric Representations

Recall that our original motivation as outlined in Section 1 is to improve separable wavelets and filter banks and to obtain efficient representations for typical multidimensional signals with intrinsic geometrical structures. In this section, we will study examples of such multiscale geometric representations that are based on the multidimensional filter banks described earlier.

7.1 The Contourlet and Surfacelet Transforms

Comparing the wavelet (“pointillist”) scheme with the new geometric scheme shown in Figure 1.2, we see that the improvement of the new scheme in representation efficiency can be attributed to the grouping of nearby wavelet coefficients, since they are locally correlated due to the smoothness of the image contours. Intuitively, this observation suggests that we can obtain a sparse expansion for natural images by first applying a wavelet-like transform, followed by a local directional transform to gather the nearby basis functions at the same scale into linear structures. In essence, we first use a multiscale transform for *edge* detection, and then a local directional transform for *contour segment* detection.

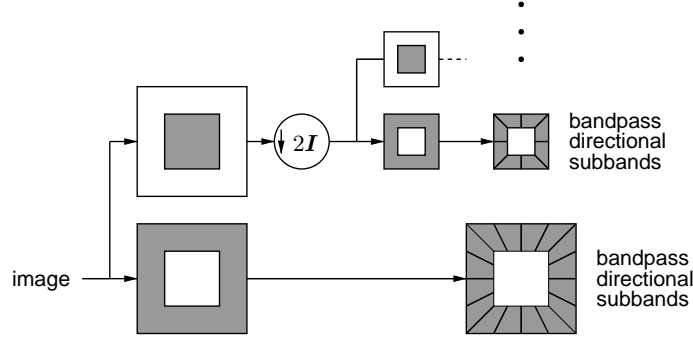


Fig. 7.1 The contourlet filter bank: first, a multiscale decomposition into octave bands by the Laplacian pyramid is computed, and then a directional filter bank is applied to each bandpass channel.

Based on this insight, we proposed a *double filter bank* structure (see Figure 7.1) [28] to obtain sparse expansions in 2D for typical images having smooth contours. In this double filter bank, the Laplacian pyramid (LP) [5] is first used to capture the *point discontinuities*; it is then followed by a directional filter bank (DFB) [2] to link point discontinuities into linear structures. The overall result is an image expansion using basic elements like contour segments (thus named *contourlets*) [29, 31]. In the spatial domain, contourlets have elongated supports at various scales, directions, and aspect ratios. This property allows contourlets to efficiently approximate a smooth contour at multiple resolutions in much the same way as the geometric scheme shown in Figure 1.2.

In the frequency domain, the contourlet transform provides a multiscale and directional decomposition. As shown in Figure 7.1, bandpass images from the LP are fed into a DFB so that directional information can be captured. The scheme can be iterated on the coarse image. The combined result is a double iterated filter bank structure, named *contourlet filter bank*, which decomposes images into directional subbands at multiple scales.

Let $a_0[\mathbf{n}]$ be the input image to the contourlet transform. The output after the multiscale stage is J bandpass images $b_j[\mathbf{n}]$, $j = 1, 2, \dots, J$ (from fine-to-coarse) and a lowpass image $a_J[\mathbf{n}]$. That means, the j th level of the LP decomposes the image $a_{j-1}[\mathbf{n}]$ into a coarser image

$a_j[\mathbf{n}]$ and a detail image $b_j[\mathbf{n}]$. Next, in the multidirectional stage, each bandpass image $b_j[\mathbf{n}]$ is further decomposed by an ℓ_j -level DFB into 2^{ℓ_j} bandpass directional images $c_{j,k}^{(\ell_j)}[\mathbf{n}]$, $k = 0, 1, \dots, 2^{\ell_j} - 1$. The main properties of the discrete contourlet transform are summarized in the following theorem.

Theorem 7.1. [31] In a contourlet filter bank, the following hold:

- (1) If both the LP and the DFB use perfect reconstruction filters, then the discrete contourlet transform achieves perfect reconstruction. In that case the analysis and synthesis operators form a frame and its dual frame pair, respectively.
 - (2) If both the LP and the DFB use orthogonal filters, then the discrete contourlet transform provides a tight frame with frame bounds equal to 1.
 - (3) The redundancy ratio of the contourlet transform, which is defined as the ratio of the number of transform coefficients over the number of input signal samples, is less than 4/3.
 - (4) Suppose an ℓ_j -level DFB is applied at the pyramidal level j of the LP, then the synthesis and analysis vectors of the discrete contourlet transform (that is, the equivalent filters of the contourlet filter bank, or, contourlets) have an essential support size of *width* $\approx C2^j$ and *length* $\approx C2^{j+\ell_j-2}$.
 - (5) When using FIR filters, the computational complexity of the discrete contourlet transform is $O(N)$ for N -pixel images.
-

Figure 7.2 illustrates the subband decomposition by the contourlet transform and the associated representation vectors. Recall from Lemma 4.1 that the contourlet transform, as a filter bank, can be viewed as a signal representation with analysis and synthesis vectors. As defined in (4.1) and (4.3), these vectors are filter impulse responses and their shifted versions on sampling lattices. The elongated supports of contourlets as shown in Figure 7.2(b) allow them to efficiently represent images with smooth contours.

Since the multiscale and directional decomposition stages are decoupled in the discrete contourlet transform, we can have a different

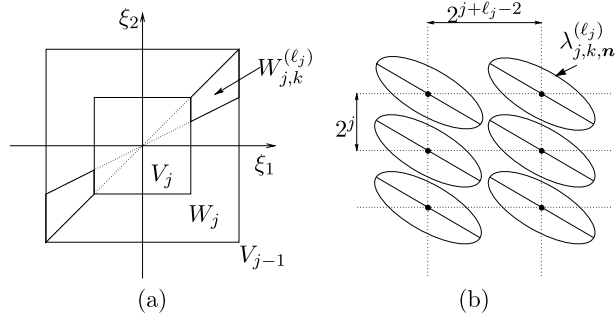


Fig. 7.2 Contourlet subband decomposition and representation vectors. (a) Multiscale and multidirectional subbands decomposed by the contourlet transform. (b) Sampling grid and approximate support of contourlet representation vectors for a “mostly horizontal” subband $W_{j,k}^{(\ell_j)}$ [shaded area in subfigure (a)]. For “mostly vertical” subbands, the grid is transposed.

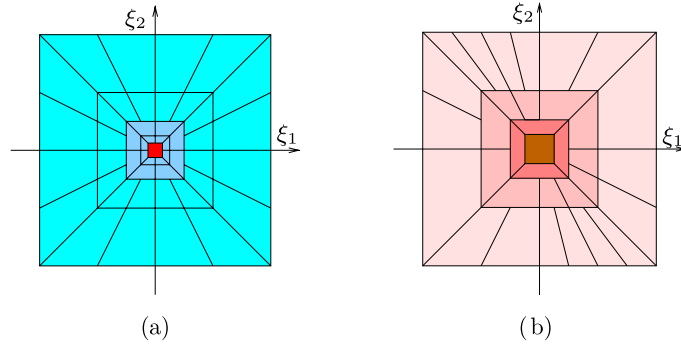


Fig. 7.3 Examples of possible frequency decompositions by the contourlet transform (a) and contourlet packets (b).

number of directions at different scales, thus offering a flexible multiscale and directional expansion. Moreover, the full binary tree decomposition of the DFB in the contourlet transform can be generalized to arbitrary tree structures, similar to the *wavelet packets* generalization of the wavelet transform [18]. The result is a family of directional multiresolution expansions, which we call *contourlet packets*. Figure 7.3 shows examples of possible frequency decompositions by the contourlet transform and contourlet packets. In particular, contourlet packets allow finer angular resolution decomposition at any scale or direction, at the cost of reduced spatial resolution. In addition, from Theorem 7.1 (part 4) we see that by altering the depth of the DFB decomposition

tree at different scales (and even at different orientations in a contourlet packets transform), we obtain a rich set of contourlets with a variety of support sizes and aspect ratios. This flexibility allows the contourlet transform and the contourlet packets to fit smooth contours of various curvatures well.

We would like to point out that the decoupling of multiscale and directional decomposition stages offers a simple and flexible transform, but at the cost of a small redundancy (up to 33%, which comes from the Laplacian pyramid). In [60], we developed a *critically sampled* contourlet transform, which we call *CRISP-contourlets*, using a combined iterated nonseparable filter bank for both multiscale and directional decompositions.

Extending the contourlet transform in 2D to arbitrary higher dimensions, we can similarly combine a multiscale decomposition with the new directional filter bank (NDFB) in higher dimensions as described in Section 6.2. The resulting transform provides a frames for signal expansions in multidimensional (MD) cases. The basis elements are multiscale surface patches, and are thus named *surfacelets* [62]. Figure 7.4 depicts the surfacelet transform in MD.

An important improvement of the surfacelet transform is that, instead of using the Laplacian pyramid as in contourlets, we employ

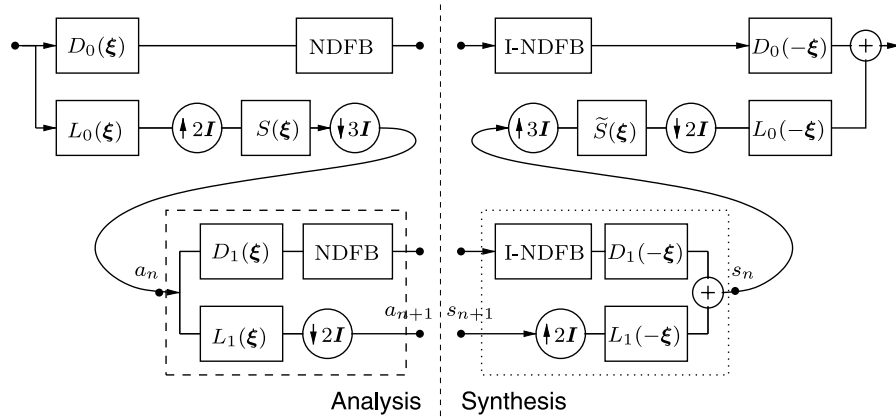


Fig. 7.4 The block diagram of the surfacelet transform. The forward directional filter bank NDFB and its inverse I-NDFB are attached to the highpass subbands of the multiscale pyramid at each scale.

a new multiscale pyramid structure which is conceptually similar to the one used in the steerable pyramid [86]. In Figure 7.4, we use $L_i(\xi)$ ($i = 0, 1$) to represent the lowpass filters and $D_i(\xi)$ ($i = 0, 1$) to represent the highpass filters in the multiscale decomposition. $S(\xi)$ and $\tilde{S}(\xi)$ are two anti-aliasing filters used to cancel the aliasing caused by the upsampling operations. The NDFB is attached to the highpass branch at the finest scale and bandpass branches at coarser scales. To have more levels of decomposition, we can recursively insert at point a_{n+1} a copy of the diagram contents enclosed by the dashed rectangle in the analysis part, and at point s_{n+1} a copy of the diagram contents enclosed by the dotted rectangle in the synthesis part.

In the new multiscale pyramid depicted in Figure 7.4, the lowpass filter $L_0(\xi)$ in the first level is downsampled by a non-integer factor of 1.5 (upsampling by 2 followed by downsampling by 3) along each dimension. Although this makes the new pyramid slightly more redundant than the Laplacian pyramid (for example, 1.34 vs. 1.14 in redundancy ratio in 3-D), we find this to be a crucial step in reducing the frequency-domain aliasing of the NDFB. We refer readers to [61] for a detailed explanation of this point.

Figure 7.5 shows a constructed 3D surfacelet (a representation vector of the surfacelet transform) in both frequency and spatial domains.

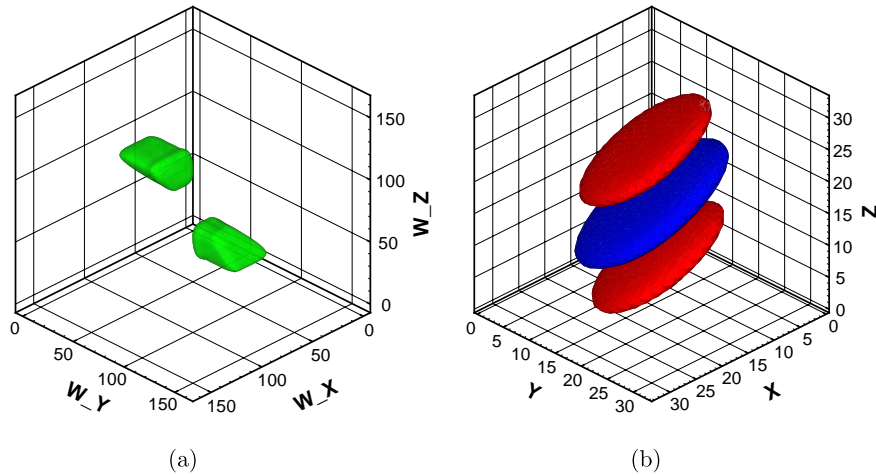


Fig. 7.5 A 3D surfacelet: (a) In the frequency domain; (b) In the spatial domain. We plot the isosurfaces of these functions to illustrate their supports.

We see that surfacelets are well-localized in both domains. In particular, surfacelets in the spatial domain are localized surface patches, smooth along the tangent planes and oscillatory along the normal directions (which are equal to the directions of the corresponding frequency supports). This feature allows surfacelets to efficiently represent multidimensional signals with discontinuities along smooth surfaces.

7.2 Multiresolution Directional Analysis

As for the wavelet filter bank, the contourlet and surfacelet filter banks have associated continuous-domain expansions in $L_2(\mathbb{R}^d)$ using contourlet and surfacelet functions, respectively. In this section, the connection between these discrete transforms and their associated continuous-domain expansions will be made precise via a new multiresolution analysis framework that is similar to the link between wavelets and filter banks [65]. The new elements in this framework are multidirection and its combination with multiscale. In what follows, we focus on the multiresolution directional analysis framework for the contourlet transform. The extension to the surfacelet transform is similar.

7.2.1 Multiscale

We start with the analysis for the multiscale part of the contourlet filter bank as shown in Figure 7.1. This multiscale filter bank is similar to the one for wavelets.

Suppose that the iterated lowpass filters in this filter bank are orthogonal and undergo two-fold downsampling in each dimension (that is, downsampling with $2\mathbf{I}$). Under certain regularity conditions, the lowpass synthesis filter G in the iterated multiscale filter bank uniquely defines a scaling function $\phi(\mathbf{t}) \in L_2(\mathbb{R}^2)$ that satisfies the following two-scale equation [65, 99]

$$\phi(\mathbf{t}) = 2 \sum_{\mathbf{n} \in \mathbb{Z}^2} g[\mathbf{n}] \phi(2\mathbf{t} - \mathbf{n}). \quad (7.1)$$

Let

$$\phi_{j,\mathbf{n}} = 2^{-j} \phi\left(\frac{\mathbf{t} - 2^j \mathbf{n}}{2^j}\right), \quad j \in \mathbb{Z}, \mathbf{n} \in \mathbb{Z}^2. \quad (7.2)$$

Then the family $\{\phi_{j,\mathbf{n}}\}_{\mathbf{n} \in \mathbb{Z}^2}$ forms an orthonormal basis for an approximation subspace V_j at scale 2^j . Furthermore, $\{V_j\}_{j \in \mathbb{Z}}$ provides a sequence of multiresolution nested subspaces, that is, $\{\mathbf{0}\} \subset \cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \cdots \subset L_2(\mathbb{R}^2)$, where V_j is associated with a uniform grid of intervals $2^j \times 2^j$ that characterizes signal approximation at scale 2^j .

The difference signals in the Laplacian pyramid (LP) for the contourlet filter bank contain the details necessary to increase the resolution between two consecutive approximation subspaces. Therefore, the difference signals live in a subspace W_j that is the orthogonal complement of V_j in V_{j-1} , or

$$V_{j-1} = V_j \oplus W_j. \quad (7.3)$$

In [30] we show that the LP can be considered as an oversampled filter bank where each of the four polyphase components of the difference signal $b[\mathbf{n}]$ in the LP filter bank, together with the coarse signal $a[\mathbf{n}]$, comes from a separate filter bank channel with the same sampling matrix $2\mathbf{I}$. Let $F_i(\mathbf{z}), 0 \leq i \leq 3$ be the synthesis filters for these polyphase components. As for wavelets, we associate with each of these filters a continuous function $\psi^{(i)}(\mathbf{t})$ where

$$\psi^{(i)}(\mathbf{t}) = 2 \sum_{\mathbf{n} \in \mathbb{Z}^2} f_i[\mathbf{n}] \phi(2\mathbf{t} - \mathbf{n}). \quad (7.4)$$

Proposition 7.2 ([30]). Using $\psi^{(i)}(\mathbf{t})$ in (7.4), we can define a family of functions

$$\psi_{j,\mathbf{n}}^{(i)}(\mathbf{t}) = 2^{-j} \psi^{(i)}\left(\frac{\mathbf{t} - 2^j \mathbf{n}}{2^j}\right), \quad j \in \mathbb{Z}, \mathbf{n} \in \mathbb{Z}^2. \quad (7.5)$$

Then, at scale 2^j , $\{\psi_{j,\mathbf{n}}^{(i)}\}_{0 \leq i \leq 3, \mathbf{n} \in \mathbb{Z}^2}$ is a tight frame for W_j . For all scales, $\{\psi_{j,\mathbf{n}}^{(i)}\}_{j \in \mathbb{Z}, 0 \leq i \leq 3, \mathbf{n} \in \mathbb{Z}^2}$ is a tight frame for $L_2(\mathbb{R}^2)$. In both cases, the frame bounds are equal to 1.

We define

$$\mu_{j,2\mathbf{n}+\mathbf{k}_i}(\mathbf{t}) = \psi_{j,\mathbf{n}}^{(i)}(\mathbf{t}), \quad 0 \leq i \leq 3, \quad (7.6)$$

where $\{\mathbf{k}_i\}$ are the coset representatives for downsampling by $2\mathbf{I}$. With this notation, the family $\{\mu_{j,\mathbf{n}}\}_{\mathbf{n} \in \mathbb{Z}^2}$ associated with a uniform grid of intervals 2^{j-1} in each dimension of \mathbb{R}^2 provides a tight frame for W_j .

7.2.2 Multidirection

In the iterated contourlet filter bank, the discrete basis (6.2) of the DFB can be regarded as a change of basis for the continuous-domain subspaces obtained from the multiscale analysis in the last section. Suppose that the DFBs in the contourlet filter bank use orthogonal filters. Applying the directional decomposition by the family (6.2) onto the detail subspace W_j as done by the contourlet transform, we obtain the following result.

Proposition 7.3 ([30]). Let

$$\lambda_{j,k,\mathbf{n}}^{(\ell)}(\mathbf{t}) = \sum_{\mathbf{m} \in \mathbb{Z}^2} d_k^{(\ell)}[\mathbf{m} - \mathbf{S}_k^{(\ell)} \mathbf{n}] \mu_{j,\mathbf{m}}(\mathbf{t}). \quad (7.7)$$

The family $\{\lambda_{j,k,\mathbf{n}}^{(\ell)}\}_{\mathbf{n} \in \mathbb{Z}^2}$ is a tight frame of a detail directional subspace $W_{j,k}^{(\ell)}$ with frame bounds equal to 1, for each $k = 0, \dots, 2^\ell - 1$. Furthermore, the subspaces $W_{j,k}^{(\ell)}$ are mutually orthogonal across scales and directions.

Figure 7.2(a) illustrates the detail directional subspaces $W_{j,k}^{(\ell)}$ in the frequency domain. The indices j , k , and \mathbf{n} specify the scale, direction, and location, respectively. Note that the number of DFB decomposition levels ℓ can be different at different scales j , in which case it will be denoted by ℓ_j .

The following result establishes that each $W_{j,k}^{(\ell)}$ is a shift-invariant subspace generated by a single function and its translations.

Proposition 7.4. Let

$$\lambda_{j,k}^{(\ell)}(\mathbf{t}) = \sum_{\mathbf{m} \in \mathbb{Z}^2} d_k^{(\ell)}[\mathbf{m}] \mu_{j,\mathbf{m}}(\mathbf{t}). \quad (7.8)$$

Then for $\ell \geq 2$,

$$\lambda_{j,k,\mathbf{n}}^{(\ell)}(\mathbf{t}) = \lambda_{j,k}^{(\ell)}(\mathbf{t} - 2^{j-1} \mathbf{S}_k^{(\ell)} \mathbf{n}). \quad (7.9)$$

It follows that the translated family of $\lambda_{j,k}^{(\ell)}$:

$$\left\{ \lambda_{j,k,\mathbf{n}}^{(\ell)}(\mathbf{t}) = \lambda_{j,k}^{(\ell)}(\mathbf{t} - 2^{j-1} \mathbf{S}_k^{(\ell)} \mathbf{n}) \right\}_{\mathbf{n} \in \mathbb{Z}^2} \quad (7.10)$$

is a frame of $W_{j,k}^{(\ell)}$. As a result, the subspace $W_{j,k}^{(\ell)}$ is defined on a rectangular grid with intervals $2^{j+\ell-2} \times 2^j$ or $2^j \times 2^{j+\ell-2}$, depending on whether it is mostly horizontal or vertical [see Figure 7.2(b)].

7.2.3 Multiscale and Multidirection

Integrating the multidirectional analysis over scales, we obtain the following result for the *contourlet frames* of $L_2(\mathbb{R}^2)$.

Theorem 7.5. For a sequence of finite positive integers $\{\ell_j\}_{j \leq j_0}$, the family

$$\{\phi_{j_0,\mathbf{n}}(\mathbf{t}), \lambda_{j,k,\mathbf{n}}^{(\ell_j)}(\mathbf{t})\}_{j \leq j_0, 0 \leq k \leq 2^{\ell_j}-1, \mathbf{n} \in \mathbb{Z}^2} \quad (7.11)$$

is a tight frame of $L_2(\mathbb{R}^2)$. For a sequence of finite positive integers $\{\ell_j\}_{j \in \mathbb{Z}}$, the family

$$\{\lambda_{j,k,\mathbf{n}}^{(\ell_j)}(\mathbf{t})\}_{j \in \mathbb{Z}, 0 \leq k \leq 2^{\ell_j}-1, \mathbf{n} \in \mathbb{Z}^2} \quad (7.12)$$

is a tight frame of $L_2(\mathbb{R}^2)$. In both cases, the frame bounds are equal to 1.

Finally, similar to the link between wavelets and filter banks [65], the following theorem establishes the precise connection between the continuous-domain expansions by contourlet functions defined in (7.11) and (7.12) and the discrete contourlet transform constructed in Section 7.1.

Theorem 7.6. Suppose $a_0[\mathbf{n}] = \langle f, \phi_{L,\mathbf{n}} \rangle$ are the inner products of a function $f(\mathbf{t}) \in L_2(\mathbb{R}^2)$ with the scaling functions at scale L . Furthermore, suppose that the image $a_0[\mathbf{n}]$ is decomposed by the discrete contourlet transform into coefficients $\{a_J[\mathbf{n}], c_{j,k}^{(\ell_j)}[\mathbf{n}]\}$, $j = 1, 2, \dots, J$ and $0 \leq k < 2^{\ell_j} - 1$. Then

$$a_J[\mathbf{n}] = \langle f, \phi_{L+J,\mathbf{n}} \rangle, \quad \text{and} \quad c_{j,k}^{(\ell_j)}[\mathbf{n}] = \langle f, \lambda_{L+j,k,\mathbf{n}}^{(\ell_j)} \rangle. \quad (7.13)$$

In what follows, we summarize several important features of the contourlet transform:

- (1) The contourlet expansions are defined on rectangular grids, and thus offer a seamless translation (as demonstrated in Theorem 7.6) to the discrete world, where image pixels are sampled on a rectangular grid. To achieve this “digital-friendly” feature, the contourlet kernel functions $\lambda_{j,k}^{(\ell_j)}$ have to be different for different directions k and cannot be obtained by simply rotating a single function. This is a key difference between the contourlet and the curvelet systems [7, 9].
- (2) Since the contourlet functions are defined via iterated filter banks like wavelets, the contourlet transform allows for fast filter bank implementations and has convenient tree structures.
- (3) It is easy to see that with FIR filters, the iterated contourlet filter bank leads to compactly supported contourlet frames. More precisely, the contourlet function $\lambda_{j,k,n}^{(\ell_j)}$ has support of size *width* $\approx C2^j$ and *length* $\approx C2^{j+\ell_j-2}$. In other words, at each scale and direction, the set $\{\lambda_{j,k,n}^{(\ell_j)}\}_{n \in \mathbb{Z}^2}$ “tiles” the plane \mathbb{R}^2 [see Figure 7.2(b)].
- (4) The contourlet construction provides a space-domain multiresolution scheme that offers flexible refinements for the spatial resolution and the angular resolution as illustrated in Figure 7.3.

7.3 Other Multiscale Geometric Representations

In addition to contourlets and surfacelets, there have been many other multiscale geometric representations developed in the literature. Broadly speaking, signal representations can be classified into two groups: *fixed* transforms and *adaptive* transforms. Contourlets and surfacelets belong to the first group, which also includes 2D Gabor wavelets [24], the cortex transform [102], the steerable pyramid [86], 2D directional wavelets [1], brushlets [68], complex wavelets [48], curvelets [7, 9], and shearlets [40, 50], all of which aim for providing a multiscale and

directional image representation. In what follows, we provide a brief review of several of these transforms, and refer readers to a recent article [45] for a more comprehensive survey.

The 2D *Gabor wavelet transforms* are based on a family of 2D Gabor functions, specified as

$$g(x_1, x_2) = e^{-((x_1/\sigma_1)^2 + (x_2/\sigma_2)^2)} e^{-j2\pi(\omega_1 x_1 + \omega_2 x_2)}.$$

The Gabor functions achieve optimal joint concentration in the spatial and frequency domains and resemble the measured 2D receptive fields in the mammalian visual cortex [23]. The parameters (ω_1, ω_2) specify the orientation, while the parameters (σ_1, σ_2) specify the spatial extent of the Gabor functions. However, constructing complete discrete 2D Gabor representations with structured and fast forward and inverse transforms has been challenging. For example, the discrete 2D Gabor transforms constructed in [24] have to be approximated by neural networks.

The *cortex transform* [102] approximates the 2D Gabor shape in the visual cortex by a simple frequency-domain partition with a set of bandlimited multiscale and directional filters. Cortex transform coefficients are computed in the frequency domain by multiplying the Fourier transform of the input image by each filter and then computing the inverse Fourier transform. The use of bandlimited filters allows aliasing-free subsampling for multiscale decomposition in a pyramid structure to reduce the redundancy ratio and computational complexity. For a cortex transform with Ω directions (typically $\Omega = 4$), the redundancy ratio is about $(4/3)\Omega + 1$ and the computational complexity is approximately equal to computing $(\Omega + 1)$ FFT's.

The *steerable pyramid* [86] aims to provide multiscale and directional responses of the input image with a fixed set of directions so that the responses to an arbitrary direction can be obtained by linear combinations of those fixed directional responses. Because of this *steerability* requirement, the design of directional filters for steerable pyramids is quite restricted. For instance, some steerable filters are based on directional derivative operators. In terms of filter bank implementations, the steerable pyramid makes use of *self-inverting* filters in the analysis and

synthesis parts (recall (4.7) for a precise definition). It follows that the associated signal representation is a *tight frame* and thus the adjoint of the forward transform is an inverse transform [21, 65]. The steerable pyramid has the same redundancy ratio as the cortex transform, about $(4/3)\Omega + 1$ for Ω directions.

The *complex wavelet transform* [48] uses the Hilbert-pair approach to achieving directionally selective filters from separable filter banks with analytic basis functions in 1D. More specifically, the complex wavelet transform uses a dual-tree of two discrete wavelet transforms in parallel such that the wavelets generated by these trees form a Hilbert pair. As a result, the complex wavelet transform provides finer frequency localization through analytic basis functions with supports in only one-half of the frequency axis. With separable extension to 2D, the complex wavelet basis functions have finer directional selectivity than the standard 2D wavelets. For example, a dual-tree separable 2-band filter bank leads to a 2D complex wavelet transform with 6 directions, roughly at 15, 45, 75, 105, 135, and 165 degree angles. Furthermore, the Hilbert-pair condition on the generated wavelets is equivalent to requiring that the lowpass filters in the dual-tree filter banks are related by a *half-sample shift* [82]. As a result, the dual-tree filter banks effectively provide a 2:1 oversampling at each scale as compared to the standard wavelet transform, and thus largely avoid the aliasing or shift-variant problems typically associated with subsampling. Due to the use of Hilbert-pair or dual-tree filter banks, the redundancy of the complex wavelet transform is 4 in 2D, and in general, 2^d in d -dimensions.

The *curvelet transform* was developed initially in the *continuous domain* [7] via multiscale filtering and then applying a block ridgelet transform [8] on each bandpass image. Later, the authors proposed the *second generation curvelet transform* [9] that was defined directly via frequency partitioning without using the ridgelet transform. The curvelet transform pioneers an important condition, namely, the *parabolic scaling* property for the support of the curvelets as $width \propto length^2$, which is important for achieving the optimal approximation rate for 2-D piecewise smooth functions with C^2 (twice continuously

differentiable) contours. The parabolic scaling condition implies that the number of directions doubles at every other finer octave scale. Both curvelet constructions require a rotation operation and correspond to a 2-D frequency partition based on the polar coordinate. This makes the curvelet construction simple in the continuous domain but causes the implementation for discrete images — typically sampled on rectangular grids — to be challenging. In particular, approaching critical sampling seems difficult in such discretized constructions. The latest digital implementations of the curvelet transform in [6], which are based on unequally-spaced fast Fourier transforms or wrappings of Fourier samples, have the redundancy ratio of about 2.8 when wavelets are chosen at the finest scale, and 7.2 otherwise.

The *shearlet transform* aims for a unified treatment of the continuous and digital worlds. In the continuous domain, shearlets are built similarly to curvelets, but rotation is replaced with shears [40, 50]. There are two discrete implementations of the shearlet transform for images. The first method, applicable for bandlimited shearlets, is based on the shear-like pseudo-polar grids and the pseudo-polar Fourier transform [51]. This frequency domain method has a fairly high redundancy ratio, about $4R$ where R is the oversampling factor for the employed pseudo-polar Fourier transform. In practice, R has to be large, such as $R = 8$, so that the discrete shearlet transform is almost a tight frame with a simple reconstruction algorithm. The second method, applicable for compactly supported shearlets, is based on resampling the input image into digital shear grids and applying the separable wavelet transform [56]. This space domain method has a reasonable redundancy ratio of about 4, but the transform is not a tight frame, and reconstruction has to resort to iterative methods such as conjugate gradient. The reported computational complexities of the shearlet transform by these two methods are, respectively, about 280 and 40 times that of a 2D FFT on the input image [51, 56].

As mentioned before, another group of approaches in developing efficient representations for geometrical regularity is based on *adaptive* transforms. Well-known examples include bandelets [78], edge-adapted multiscale transform [17], wedgelets [32, 101], wavelet

footprints [35], best tree-based representations [43, 85], directionlets [96], motion-adaptive transform for videos [81], adaptive directional lifting [13, 26], and grouplets [66]. We omit further discussions on these adaptive signal representations and refer readers to the references cited above for more details.

8

Example Applications

In this section, we demonstrate several applications of the multi-dimensional filter banks and multiscale geometric representations for image and signal processing. Software toolboxes for implementing the contourlet [20, 31, 61] and surfacelet [62] transforms are available for download from the authors' homepages.

8.1 Signal Decomposition and Nonlinear Approximation

Figure 8.1 shows an example of the contourlet transform applied to an image. We notice that only those contourlets that match the locations and directions of image contours produce significant coefficients.

Next, we compare the nonlinear approximation (NLA) performances of the wavelet and contourlet transforms. In these experiments, for a given value M , we select the M -most significant coefficients (in absolute values) from each transform domain, and then compare the reconstructed images from these sets of M coefficients. Since the two transforms share the same detail subspaces, it is possible to restrict the comparison in these subspaces. We expect that most of the refinement happens around the image edges.

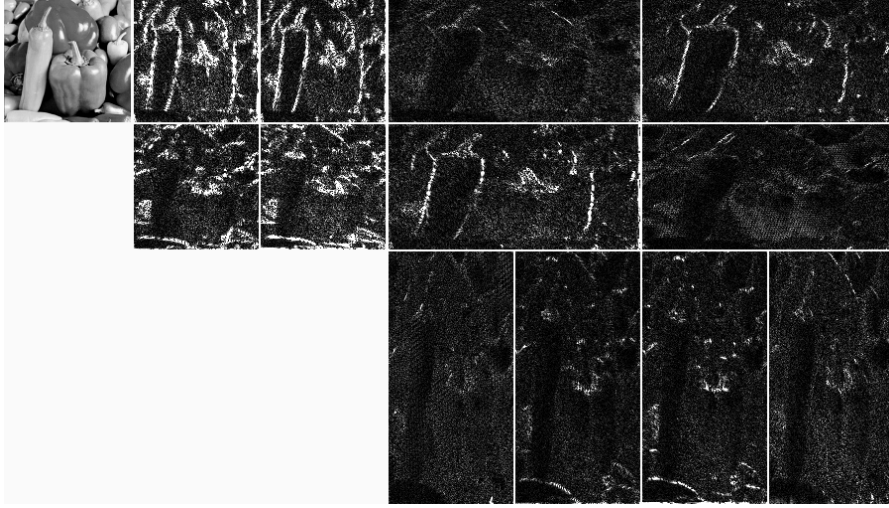


Fig. 8.1 The contourlet transform of the *Peppers* image in Figure 1.1(a). The input image is decomposed into two pyramidal levels, which are then further decomposed into four and eight directional subbands, respectively. Small coefficients (in absolute values) are shown in black while large coefficients are shown in white.

Figure 8.2 shows sequences of nonlinear approximated images at the finest detailed subspace W_j using the wavelet and the contourlet transforms, respectively, for the input image *Peppers* as shown in Figure 1.1(a). The wavelet scheme is seen to slowly capture contours by isolated “dots”. By contrast, the contourlet scheme quickly refines its reconstructions by using well-adapted “sketches,” in much the same way as the new painting style shown in Figure 1.2.

Figure 8.3 shows a detailed comparison of two nonlinear approximations of the *Barbara* image by the wavelet and contourlet transforms using the same number of coefficients. Contourlets are shown to be superior as compared to wavelets in capturing fine contours (for example, the directional textures on cloths). In addition, there is a significant gain of 1.46 dB in peak signal-to-noise ratio (PSNR) for contourlets.

More rigorously, consider 2-D piecewise smooth functions that are C^2 (that is, twice continuously differentiable) except for discontinuities along C^2 curves. They can serve as idealized models for natural images with smooth contours. For this class of functions, the decay rate of the best M -term approximation error (in squared L_2 -norm) using Fourier

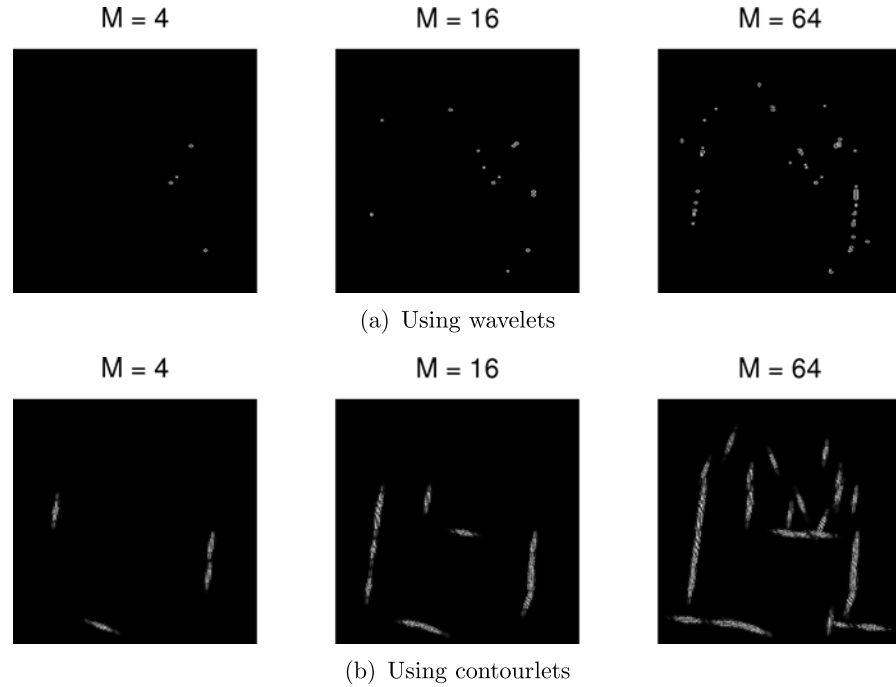


Fig. 8.2 Sequence of images showing the nonlinear approximations of the *Peppers* image using M most significant coefficients at the finest detailed subspace W_j , which is shared by both the wavelet and contourlet transforms.

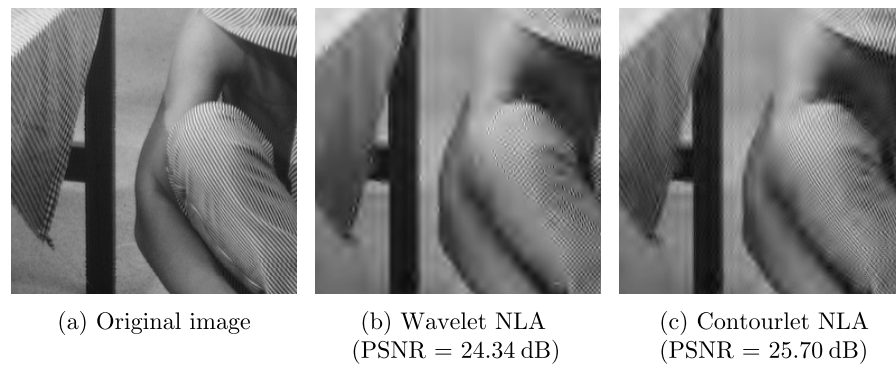


Fig. 8.3 Nonlinear approximations (NLA) by the wavelet and contourlet transforms. In each case, the original image *Barbara* of size 512×512 is reconstructed from the 4096 most significant coefficients. Only part of images are shown for detailed comparisons.

basis can be shown to be $O(M^{-1/2})$ [34, 65]; using wavelets, the rate becomes $O(M^{-1})$; and using curvelets [9], contourlets [31], and shearlets [40], the rate can be further improved to $O((\log M)^3 M^{-2})$. Furthermore, this last decay rate is essentially the best rate one can achieve for the class of C^2/C^2 piecewise smooth functions [9]. These theoretical results suggest that, for typical images with smooth contours, we may expect a significant improvement in representation-efficiency by using curvelet-like methods over wavelets. And this is comparable to the improvement brought by wavelets over the Fourier basis in representing 1D piecewise smooth signals.

8.2 Image and Video Denoising

Next, we consider an application of geometric signal representations in removing additive white Gaussian noise from visual data. Let \mathbf{x} represent a clean image, and suppose that we can only observe a noisy version of \mathbf{x} , that is,

$$\mathbf{x}_n = \mathbf{x} + \mathbf{n},$$

where \mathbf{n} is a realization of independent Gaussian noise of zero mean and variance σ^2 . Our goal is to separate the noise \mathbf{n} from the observation \mathbf{x}_n and get a denoised image $\hat{\mathbf{x}}$.

To that end, we apply a simple thresholding estimator in the transform domain. Denote by \mathbf{C} and \mathbf{C}^{-1} the forward and inverse transforms, respectively, of some geometric signal representation (for example, the contourlet transform presented in Section 7.1). We can then represent both the clean and noisy images in the transform domain as

$$\mathbf{x} = \mathbf{C}^{-1}\mathbf{u} \quad \text{and} \quad \mathbf{x}_n = \mathbf{C}^{-1}\mathbf{u}_n,$$

where $\mathbf{u} = \mathbf{C}\mathbf{x}$ and $\mathbf{u}_n = \mathbf{C}\mathbf{x}_n$ are the corresponding expansion coefficients. To denoise the image, we obtain

$$\hat{\mathbf{x}} = \mathbf{C}^{-1}\text{Th}_\tau(\mathbf{u}_n),$$

where $\text{Th}_\tau(\cdot)$ is a component-wise hard-thresholding function

$$\text{Th}_\tau(u) = \begin{cases} u, & \text{if } |u| \geq \tau, \\ 0, & \text{otherwise} \end{cases} \quad (8.1)$$

acting on each element of \mathbf{u}_n . The thresholding scheme is a very simple yet often effective way to removing noise. Intuitively, this method works because most natural images are compressible through geometric representations, meaning, the energy of the transformed images \mathbf{u} is usually concentrated on a small number of significant coefficients. In contrast, due to the lack of geometric structures, the energy of the noise component \mathbf{n} is more uniformly distributed in the transform domain. As a result, most of the small coefficients that are set to zero in (8.1) are due to noise, whereas the significant coefficients that come from the original image are more likely to survive the thresholding.

In our experiment, we consider three geometric representations: the contourlet transform described in Section 7.1 with a new multiscale pyramid (Figure 7.4; see also [61] for details); the nonsubsampling contourlet transform [20]; and a complex-valued curvelet transform [6, 9]. For both versions of the contourlet transform, we use 5 levels of decomposition and set the number of directional subbands at each scale, from fine to coarse, to be 16, 16, 8, 8, and 4, respectively. For benchmark, we also show the denoising performance of a non-geometric signal representation — the 2-D separable wavelet transform, with biorthogonal 9/7 filters. Finally, the threshold τ in (8.1) is chosen to be 4σ for the finest scale of decomposition and 3σ for all other scales.

Figure 8.4 shows a “zoom-in” comparison of the original *Peppers* image, its noisy version (with $\sigma = 20$), and the denoised results using different transforms. We can see that the separable wavelet transform exhibits many point-like denoising artifacts (recall our early discussions in Section 1 on the “pointillist” style of wavelets in representing images). In contrast, the three geometric signal representations produce much better results, especially along smooth object boundaries. All of them significantly outperform the wavelet transform, both in terms of peak signal-to-noise ratios (PSNRs) and in visual quality.

The three geometric representations used in our test — contourlets, curvelets, and nonsubsampling contourlets — have different redundancy ratios: 2.33, 14.5, and 53.0, respectively. In general, extra redundancy is beneficial in denoising tasks. The nonsubsampling contourlet transform, being the most redundant among the three, also achieves the highest PSNR. Of course, the price for higher redundancies is increased

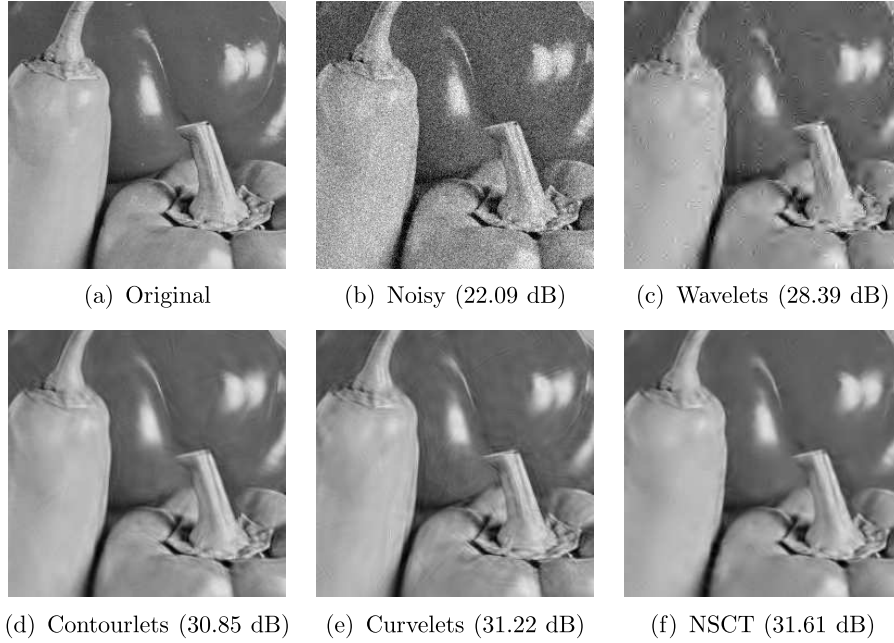


Fig. 8.4 Comparison of denoised *Peppers* images. Top row, from left to right: the original image, the noisy version, and the denoised image by using the wavelet transform; Bottom row, from left to right: the denoised images obtained by using the contourlet transform, the curvelet transform, and the nonsampled contourlet transform (NSCT). Shown in parentheses are the associated PSNR values calculated on the whole images (instead of the cropped regions).

computational complexity and memory footprint. For example, for the test images of size 512×512 and on a computer with a 2.2 GHz CPU, the running time of the denoising algorithm based on the three transforms is 0.76, 4.76, and 670 seconds, respectively.

We can also apply the same thresholding-based denoising algorithm to videos, which can be seen as a special type of 3-D signals with two spatial dimensions and one temporal dimension. Denoising video signals using velocity selective 3-D transforms was first studied by Selesnick et al. [83]. In our experiment, we use the 3-D separable wavelet transform and the 3-D surfacelet transform presented in Section 7.1, the latter of which can efficiently capture and represent signal singularities lying on smooth surfaces. Such singularities are often observed in video

signals, in which moving objects carve out smooth surfaces in the 3-D spatial/temporal space.

We decompose the input SIF-sized video sequences into 4 scales. For the surfacelet transform, the number of directional subbands at each scale, from fine to coarse, is set to 192, 192, 48, and 12, respectively.

Figure 8.5 shows one frame from the denoised *Foreman* sequence. We can see that the surfacelet transform, being a geometric representation, performs much better than separable wavelets in preserving image details. This difference in denoising quality is even more conspicuous when viewing the video sequences.



Fig. 8.5 Video denoising: Denoised frames from the *Foreman* sequence. Shown in parentheses are the PSNR values calculated on each frame.

8.3 Edge-Preserving Image Interpolation

Image interpolation is perhaps one of the most fundamental image processing tasks. To reduce artifacts and improve final image quality, various interpolation algorithms have been proposed in the literature, aiming at obtaining images with regularity (that is, smoothness) along object boundaries. In this section, we describe an iterative image interpolation algorithm [69] based on the contourlet transform. We will show that enforcing sparsity of the contourlet coefficients provides an efficient mechanism that helps improve the regularity along edges in the resulting images.

The main idea of our algorithm [69] is to alternately enforce two constraints — *observation* and *sparsity*. Figure 8.6 illustrates the observation constraint, in which we assume that the given low-resolution image, denoted by \mathbf{x}_L , is the lowpass subband of an n -level wavelet transform of the *unknown* high-resolution image \mathbf{x} , while all the coefficients in the highpass subbands have been discarded. As a simple way to get an estimate $\hat{\mathbf{x}}_0$ of the high-resolution image, we can take the inverse wavelet transform by keeping \mathbf{x}_L as the lowpass band and inserting zeros at all highpass subbands. It has been observed by several authors [39, 54] that this simple scheme often outperforms other more sophisticated interpolation methods, which do not take into account the anti-aliasing operation in the image generation process.

Let \mathbf{W} and \mathbf{W}^{-1} represent the forward and inverse wavelet transforms, respectively; denote by \mathbf{P} the diagonal matrix of ones and zeros that keeps the highpass wavelet coefficients and zeros out the low frequency subband coefficients. If we use orthonormal wavelet transforms, then the projection of any image $\hat{\mathbf{x}}^{(n)}$ onto the observation constraint set can be calculated by

$$\hat{\mathbf{x}}^{(n+1)} = \mathbf{W}^{-1} \mathbf{P} \mathbf{W} \hat{\mathbf{x}}^{(n)} + \hat{\mathbf{x}}_0, \quad (8.2)$$

where $\hat{\mathbf{x}}_0$ is the linear estimation of the high-resolution image obtained as in Figure 8.6.

The contourlet transform provides a sparse image representation (that is, there are many more insignificant coefficients than significant ones), and these significant coefficients compactly represent the edge

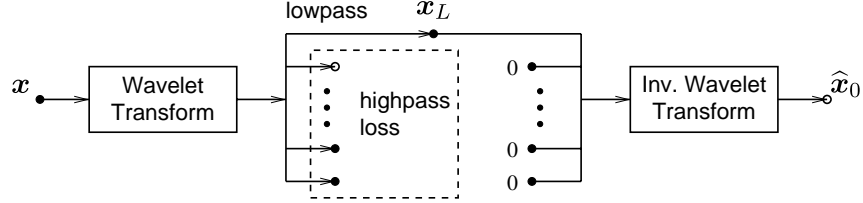


Fig. 8.6 The observation model. We assume that the given low-resolution image x_L is the lowpass subband of a wavelet transform applied on the unknown high-resolution image x . A simple linear estimation of the high-resolution image, denoted by \hat{x}_0 , can be obtained by an inverse wavelet transform, which uses x_L as the lowpass subband and zero coefficients for all highpass subbands.

regions of the image. In our algorithm, we choose to use the hard-thresholding scheme described in the previous section to enforce the sparseness constraint. For any input image $\hat{x}^{(n)}$, the sparseness constraint by hard-thresholding can be implemented as

$$\hat{x}^{(n+1)} = C^{-1} \text{Th}_\tau \left(C \hat{x}^{(n)} \right), \quad (8.3)$$

where C and C^{-1} represent the forward and inverse contourlet transforms, respectively; $\text{Th}_\tau(\cdot)$ is the component-wise thresholding function defined in (8.1).

As an initial estimate, our algorithm starts from $\hat{x}^{(0)}$, the image obtained by the simple wavelet interpolation shown in Figure 8.6. It then alternately enforces the sparseness constraint (8.3) and the observation constraint (8.2) until the generated images converge or a predetermined maximum iteration number has been reached. We note that, when combined together, the two steps (8.2) and (8.3) in our algorithm amount to a thresholded Landweber iteration scheme, whose convergence has been studied in [22].

In what follows, we show some experimental results of the iterative image interpolation algorithm described above. For comparison, we also show the results obtained by four other interpolation methods, including bilinear interpolation, simple wavelet-based linear interpolation implemented by zero-padding highpass subbands, the data-dependent triangulation (DDT) method [90], and the new edge-directed interpolation (NEDI) [55]. In all experiments, we use the *symlet* of length 16 for the wavelet transform.

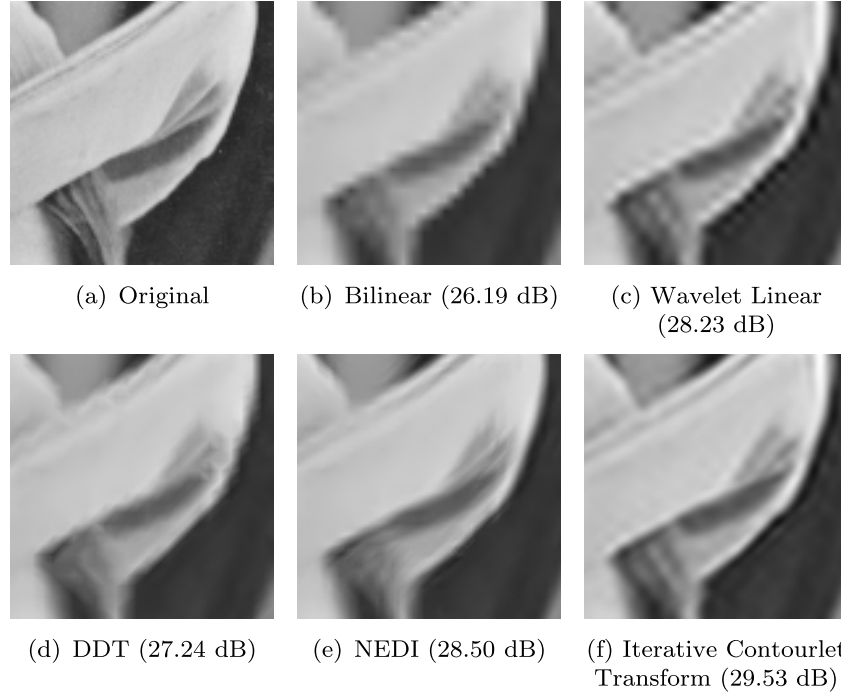


Fig. 8.7 The zoom-in comparison of the *Lena* image. Shown in parentheses are the corresponding PSNR values for the region shown.

We show the zoom-in comparisons of different algorithms on the test images *Lena* in Figure 8.7, together with the PSNR values of the reconstructed images. The original image of size 512×512 is first downsampled by a factor of four along each dimension. We then interpolate the resulting low-resolution version back to its original size. We can see from the figures that our algorithm outperforms the simple wavelet-based linear interpolation scheme. It is also clearly superior to both bilinear and DDT interpolation, as it is able to better reconstruct the high-frequency areas of the image without distorting the smooth regions. Along edge regions, the NEDI algorithm performs very similarly to our algorithm. However, our algorithm seems to be able to reconstruct the smooth and texture regions without excessive smearing. This advantage also leads to a higher PSNR value in the reconstruction result. Figure 8.8 shows the reconstructed images after each iteration of the

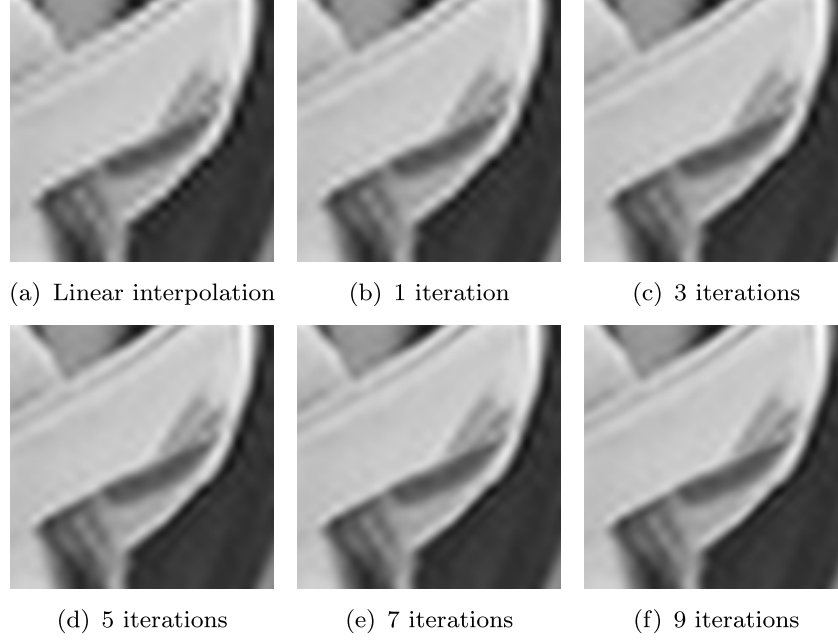


Fig. 8.8 Reconstructed *Lena* images at each iteration of the algorithm. The generated images begin to converge after about 5 iterations.

algorithm. We can clearly see the improvement of reconstruction quality along smooth image boundaries as we have more iterations. This indicates the effectiveness of the sparsity constraint in the contourlet domain.

8.4 Compressed Sensing

Sparse signals representations are at the core of the nascent field of compressed sensing (CS) [10, 11, 33]. In this section, we present an application of the contourlet transform in the CS reconstruction of images from highly incomplete Fourier measurements.

More specifically, given a set of randomly subsampled Fourier coefficients of an image \mathbf{x} , we would like to reconstruct \mathbf{x} with high visual fidelity. Denoting by \mathbf{y} the measurement vector, we can write

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \quad (8.4)$$

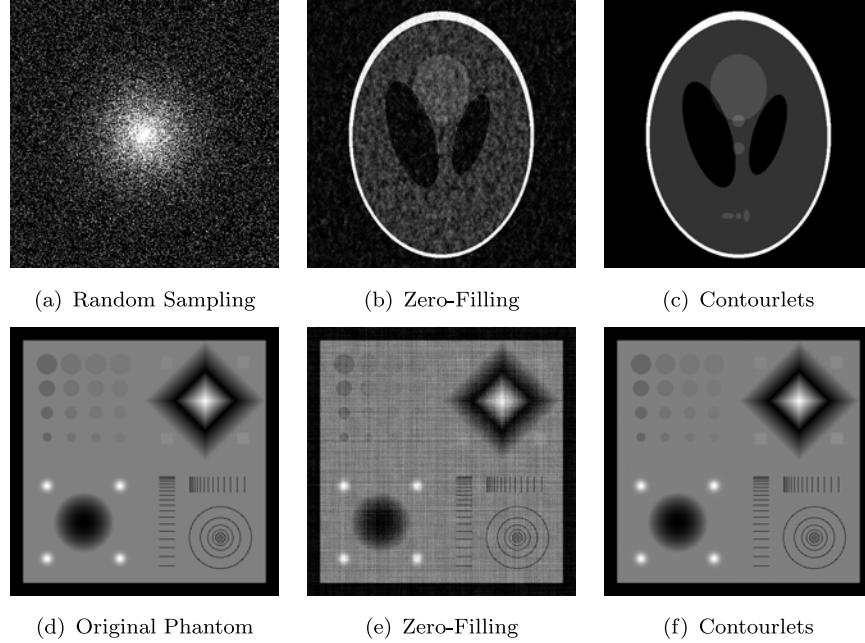


Fig. 8.9 CS reconstruction of images from highly incomplete Fourier measurements. (a) A random sampling pattern which keeps 15% of the Fourier coefficients. (b) Reconstruction of the Shepp-Logan phantom by zero-filling the missing Fourier measurements. (c) Reconstruction using the contourlet transform. (d) A more challenging MRI phantom. (e) Reconstruction by zero-filling. (f) Reconstruction using the contourlet transform.

where \mathbf{A} is a measurement matrix made of randomly picked rows of the discrete Fourier transform matrix, and \mathbf{n} is some measurement noise. For example, we show in Figure 8.9(a) a random sampling pattern in the Fourier domain, keeping only 15% of the Fourier coefficients. In magnetic resonance imaging (MRI), measurements of diagnostic images are directly taken in the Fourier domain, in a time-sequential fashion. Taking fewer Fourier samples can thus lead to significantly reduced online measurement time.

We note that (8.4) is an underdetermined linear system. To faithfully reconstruct \mathbf{x} from \mathbf{y} , an effective approach is to exploit the sparsity of \mathbf{x} in some transform domain. In our experiment, we use the new version of the contourlet transform described in [61]. For most natural images with geometric regularities, their contourlet coefficients are

sparse. To explore this sparsity, we regularize the inversion of the above linear system by solving an optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{l_2}^2 + \lambda \|\mathbf{C}\mathbf{x}\|_{l_1}, \quad (8.5)$$

where \mathbf{C} is the forward (analysis) contourlet operator, and λ is a regularization constant. In our experiments, we solve the above minimization problem by using an iterated thresholding scheme [22].

Figures 8.9(b) and 8.9(c) show the reconstruction of the Shepp-Logan phantom from 15% of its Fourier coefficients.¹ We see that the contourlet-based scheme leads to an almost perfect reconstruction, substantially outperforming the simple linear reconstruction obtained by zero-filling the missing Fourier data. Figure 8.9(d) shows a more challenging MRI phantom [87], containing features that are difficult to reproduce with partial Fourier measurements. Again, we observe that the minimization scheme in (8.5) using contourlets leads to a high quality reconstruction [see Figure 8.9(f)], which recovers image details that are completely obfuscated by the simple linear reconstruction shown in Figure 8.9(e).

¹We thank Jianwei Ma [64] for sharing his Matlab code for the iterated thresholding scheme.

Conclusions

Multidimensional filter banks provide a powerful computational tool in the acquisition and processing of multidimensional data sets. The involvement of nonseparable operations (filtering and sampling) makes the theory and constructions of multidimensional filter banks substantially richer than those of their 1-D counterpart, leading to both technical challenges and enhanced design freedoms. Starting from basic concepts such as multidimensional filtering and nonseparable sampling, we presented a systematic overview of the common notation, key tools, and main results in the characterization and design of multidimensional filter banks. In doing so, we hope that this survey can serve as a starting point and helpful guide for those readers who would like to explore this exciting topic further.

Many signals, including natural images, medical diagnostic images, videos, music, and certain physical fields are sparse when represented in an appropriately chosen signal representation. Consequently, sparse representations are one of the most fundamental tools in signal processing with numerous applications, including denoising, compression, feature extraction, and compressed sensing. Building upon the results of multidimensional perfect reconstruction filter banks, we presented

the constructions of the contourlet and surfacelet transforms, which can adapt to the inherent geometric structures in multidimensional discrete data in an efficient and robust way. As demonstrated by several numerical examples, these sparse representations can facilitate the development of new or improved algorithms in a wide range of areas in science and engineering, including for example, image and video processing, medical image acquisition, geophysical imaging, and astrophysics.

Acknowledgments

We would like to express our foremost appreciation and gratitude to Martin Vetterli, who did the pioneering work on multidimensional filter banks and guided the construction of the contourlet transform. Our understanding of multidimensional filter banks and multiscale geometric representations has been deeply influenced by our fortunate collaboration with him. We also thank the reviewers and editor Robert Gray for excellent comments and suggestions that help to improve the contents and presentation of this survey. The work of MND was supported in part by the US National Science Foundation under grants CCF-0237633, CCF-0635234, and CCF-0916953.

References

- [1] J. P. Antoine, P. Carrette, R. Murenzi, and B. Plette, “Image analysis with two-dimensional continuous wavelet transform,” *Signal Processing*, vol. 31, pp. 241–272, 1993.
- [2] R. H. Bamberger and M. J. T. Smith, “A filter bank for the directional decomposition of images: Theory and design,” *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 882–893, April 1992.
- [3] F. Bruekers and A. W. M. van den Enden, “New networks for perfect inversion and perfect reconstruction,” *IEEE Journal on Selected Areas in Communication*, vol. 10, no. 1, pp. 129–137, January 1992.
- [4] B. Buchberger, “Gröbner bases: An algorithmic method in polynomial ideal theory,” in *Multidimensional Systems Theory: Progress, Directions and Open Problems*, (N. K. Bose, ed.), pp. 184–232, Dordrecht, The Netherlands: Reidel, 1985.
- [5] P. J. Burt and E. H. Adelson, “The Laplacian pyramid as a compact image code,” *IEEE Transactions on Communication*, vol. 31, no. 4, pp. 532–540, April 1983.
- [6] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying, “Fast discrete curvelet transforms,” *Multiscale Modeling and Simulation*, vol. 5, pp. 861–899, 2006.
- [7] E. J. Candès and D. L. Donoho, “Curvelets — a surprisingly effective non-adaptive representation for objects with edges,” in *Curve and Surface Fitting*, (A. Cohen, C. Rabut, and L. L. Schumaker, eds.), Saint-Malo: Vanderbilt University Press, 1999.
- [8] E. J. Candès and D. L. Donoho, “Ridgelets: A key to higher-dimensional intermittency?,” *Philosophical Transactions on Royal Society of London A*, pp. 2495–2509, 1999.

- [9] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Communications on Pure and Applied Mathematics*, pp. 219–266, February 2004.
- [10] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, February 2006.
- [11] E. J. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, pp. 5406–5425, December 2006.
- [12] J. W. Cassels, *An Introduction to the Geometry of Numbers*. Berlin: Springer-Verlag, 1971.
- [13] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform for image compression," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1289–1302, May 2007.
- [14] T. Chen and P. P. Vaidyanathan, "Multidimensional multirate filters and filter banks derived from one-dimensional filters," *IEEE Transactions on Signal Processing*, vol. 41, no. 5, pp. 1749–1765, May 1993.
- [15] T. Chen and P. P. Vaidyanathan, "Recent developments in multidimensional multirate systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 2, pp. 116–137, April 1993.
- [16] K. F. Cheung, "A multidimensional extension of Papoulis' generalized sampling expansion with application in minimum density sampling," in *Advanced Topics in Shannon Sampling and Interpolation Theory*, (R. J. Marks, II, ed.), Springer-Verlag, 1993.
- [17] A. Cohen and B. Matei, "Compact representation of images by edge adapted multiscale transforms," in *Proceedings of IEEE International Conference on Image Processing, Special Session on Image Processing and Non-Linear Approximation*, Thessaloniki, Greece, October 2001.
- [18] R. R. Coifman, Y. Meyer, and M. V. Wickerhauser, "Wavelet Analysis and Signal Processing," in *Wavelets and their Applications*, (M. B. R. et al, ed.), pp. 153–178, Boston: Jones and Barlett, 1992.
- [19] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*. Springer-Verlag, 3rd Edition, 1992.
- [20] A. L. Cunha, J. Zhou, and M. N. Do, "The nonsubsampling contourlet transform: Theory, design and applications," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3089–3101, October 2006.
- [21] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [22] I. Daubechies, M. D. Frieze, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, pp. 3601–3608, 2004.
- [23] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 1980.
- [24] J. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Transactions on Signal Processing*, vol. 36, no. 7, pp. 1169–1179, July 1988.

- [25] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, "Singular — A computer algebra system for polynomial computations," <http://www.singular.uni-kl.de>, 2011.
- [26] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 416–427, February 2007.
- [27] M. N. Do, "Directional multiresolution image representations," PhD thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, <http://www.ifp.uiuc.edu/minhdo/publications>, December 2001.
- [28] M. N. Do and M. Vetterli, "Pyramidal directional filter banks and curvelets," in *Proceedings of IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.
- [29] M. N. Do and M. Vetterli, "Contourlets," in *Beyond Wavelets*, (G. V. Welland, ed.), New York: Academic Press, 2003.
- [30] M. N. Do and M. Vetterli, "Framing pyramids," *IEEE Transactions on Signal Processing*, pp. 2329–2342, September 2003.
- [31] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, pp. 2091–2106, December 2005.
- [32] D. L. Donoho, "Wedgelets: Nearly-minimax estimation of edges," *Annals Statistics*, vol. 27, pp. 859–897, 1999.
- [33] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, April 2006.
- [34] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2435–2476, October 1998.
- [35] P. L. Dragotti and M. Vetterli, "Wavelet footprints: Theory, algorithms and applications," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1306–1323, May 2003.
- [36] Y. C. Eldar and A. V. Oppenheim, "Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples," *IEEE Transactions on Signal Processing*, vol. 48, pp. 2864–2875, October 2000.
- [37] M. Feilner, D. V. D. Ville, and M. Unser, "An orthogonal family of quincunx wavelets with continuously adjustable order," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 499–510, April 2005.
- [38] A. Feuer and G. C. Goodwin, "Reconstruction of multi-dimensional bandlimited signals from nonuniform and generalized samples," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4273–4282, November 2005.
- [39] O. G. Guleryuz, "Predicting wavelet coefficients over edges using estimates based on nonlinear approximants," in *Proceedings of IEEE Data Compression Conference*, April 2004.
- [40] K. Guo and D. Labate, "Optimally sparse multidimensional representation using shearlets," *SIAM Journal on Mathematical Analysis*, vol. 39, pp. 298–318, 2007.
- [41] G. H. Hardy and E. M. Weight, *An Introduction to the Theory of Numbers*. Oxford University Press, 1979.

- [42] C. Herley and P. W. Wong, "Minimum rate sampling and reconstruction of signals with arbitrary frequency support," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1555–1564, July 1999.
- [43] Y. Huang, I. Pollak, M. N. Do, and C. A. Bouman, "Fast search for best representations on multitree dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1779–1793, July 2006.
- [44] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, no. 160, pp. 106–154, 1962.
- [45] L. Jacques, L. Duval, C. Chaux, and G. Peyré, "A panorama on multi-scale geometric representations, intertwining spatial, directional and frequency selectivity," *Signal Processing*, vol. 91, no. 12, pp. 2699–2730, 2011.
- [46] A. Jerri, "The Shannon sampling theorem — its various extensions and applications: A tutorial review," *Proceedings of IEEE*, vol. 65, no. 11, pp. 1565–1596, November 1977.
- [47] G. Karlsson and M. Vetterli, "Theory of two-dimensional multirate filter banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 6, pp. 925–937, June 1990.
- [48] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Applied and Computational Harmonic Analysis*, vol. 10, pp. 234–253, 2001.
- [49] J. Kovačević and M. Vetterli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for \mathcal{R}^n ," *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, vol. 38, no. 2, pp. 533–555, March 1992.
- [50] G. Kutyniok and D. Labate, "Construction of regular and irregular shearlets," *Journal of Wavelet Theory and Applications*, vol. 1, pp. 1–10, 2007.
- [51] G. Kutyniok, M. Shahram, and X. Zhuang, "Shearlab: A rational design of a digital parabolic scaling algorithm," submitted.
- [52] K. L. Law and M. N. Do, "Multidimensional filter bank signal reconstruction from multichannel acquisition," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 317–326, February 2011.
- [53] K. L. Law, R. Fossum, and M. N. Do, "Generic invertibility of multidimensional FIR filter banks and MIMO systems," *IEEE Transactions on Signal Processing*, vol. 11, pp. 4282–4291, November 2009.
- [54] X. Li, "Image resolution enhancement via data-driven parametric models in the wavelet space," *Journal of Image Video Processing*, vol. 2007, no. 1, p. 12, 2007.
- [55] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, October 2001.
- [56] W. Lim, "The discrete shearlet transform: A new directional transform and compactly supported shearlet frames," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1166–1180, 2010.
- [57] Y.-P. Lin and P. P. Vaidyanathan, "Theory and design of two-dimensional filter banks: A review," *Multidimensional Systems and Signal Processing*, vol. 7, pp. 263–330, 1996.

- [58] Z. Lin, L. Xu, and N. K. Bose, "A tutorial on Gröbner bases with applications in signals and systems," *IEEE Transactions on Circuits and Systems*, vol. 55, no. 1, pp. 445–461, February 2008.
- [59] Z. Lin, L. Xu, and Q. Wu, "Applications of Gröbner bases to signal and image processing: A survey," *Linear Algebra and its Applications*, vol. 391, no. 1, pp. 169–202, 2004.
- [60] Y. Lu and M. N. Do, "CRISP-contourlet: A critically sampled directional multiresolution image representation," in *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing*, San Diego, August 2003.
- [61] Y. Lu and M. N. Do, "A new contourlet transform with sharp frequency localization," in *Proceedings of IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.
- [62] Y. M. Lu and M. N. Do, "Multidimensional directional filter banks and surfacelets," *IEEE Transactions on Image Processing*, vol. 16, pp. 918–931, April 2007.
- [63] Y. M. Lu, M. N. Do, and R. S. Laugesen, "A computable Fourier condition generating alias-free sampling lattices," *IEEE Transactions on Signal Processing*, no. 5, pp. 1768–1782, May 2009.
- [64] J. Ma and G. Plonka, "The curvelet transform: A review of recent applications," *IEEE Signal Processing Magazine*, pp. 118–133, March 2010.
- [65] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 3rd Edition, 2008.
- [66] S. Mallat, "Geometrical grouplets," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 161–180, 2009.
- [67] J. McClellan, "The design of two-dimensional filters by transformations," in *Annals of Princeton Conference on Information Science and System*, pp. 247–251, Princeton, NJ, 1973.
- [68] F. G. Meyer and R. R. Coifman, "Brushlets: A tool for directional image analysis and image compression," *Applied and Computational Harmonic Analysis*, vol. 5, pp. 147–187, 1997.
- [69] N. Mueller, Y. M. Lu, and M. N. Do, "Image interpolation using multiscale geometric representations," in *Proceedings of SPIE Symposium on Electronic Imaging*, San Jose, January 2007.
- [70] M. Newman, *Integral Matrices*. Academic Press, 1972.
- [71] F. Nicolier, O. Laligant, and F. Truchetet, "Discrete wavelet transform implementation in Fourier domain for multidimensional signal," *Journal of Electronic Imaging*, vol. 11, no. 3, pp. 338–346, July 2002.
- [72] B. A. Olshausen, "Principles of image representation in visual cortex," in *The Visual Neurosciences*, (L. Chalupa and J. Werner, eds.), MIT Press, 2003.
- [73] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, pp. 607–609, 1996.
- [74] H. Park, "A computational theory of Laurent polynomial rings and multi-dimensional FIR systems," PhD thesis, University of California at Berkeley, 1995.

- [75] H. Park, "Optimal design of synthesis filters in multidimensional perfect reconstruction FIR filter banks using Gröbner bases," *IEEE Transactions on Circuits and Systems*, vol. 49, no. 5, pp. 843–851, May 2002.
- [76] H. Park, T. Kalker, and M. Vetterli, "Gröbner bases and multidimensional FIR multirate systems," *Multidimensional Systems and Signal Processing*, vol. 8, pp. 11–30, 1997.
- [77] S. Park, M. J. T. Smith, and R. M. Mersereau, "A new directional filterbank for image analysis and classification," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1417–1420, 1999.
- [78] E. L. Pennec and S. Mallat, "Sparse geometric image representation with bandelets," *IEEE Transactions on Image Processing*, vol. 14, pp. 423–438, April 2005.
- [79] D. P. Peterson and D. Middleton, "Sampling and reconstruction of wavenumber-limited functions in N -dimensional Euclidean spaces," *Information Control*, vol. 5, pp. 279–323, 1962.
- [80] S.-M. Phoong, C. W. Kim, P. P. Vaidyanathan, and R. Ansari, "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Transactions on Signal Processing*, vol. 43, no. 3, pp. 649–665, March 1995.
- [81] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–1542, December 2003.
- [82] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, 2005.
- [83] I. W. Selesnick and K. Y. Li, "Video denoising using 2D and 3D dual-tree complex wavelet transforms," in *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing X*, pp. 607–618, San Diego, USA, August 2003.
- [84] I. Shah and A. Kalker, "Theory and design of multidimensional QMF sub-band filters from 1-D filters and polynomials using transforms," *Proceedings of the IEE*, vol. 140, no. 1, pp. 67–71, February 1993.
- [85] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree structured compression algorithms for piecewise smooth images," *IEEE Transactions on Image Processing*, vol. 14, pp. 343–359, March 2005.
- [86] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, vol. 38, no. 2, pp. 587–607, March 1992.
- [87] D. Smith, "Compressed sensing MRI phantom," [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/29364-compressed-sensing-mri-phantom-v1-1>, 2010.
- [88] H. J. S. Smith, "On systems of linear indeterminate equations and congruences," *Philosophical Transactions of the Royal Society of London*, pp. 293–326, 1861.

- [89] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Boston: Wellesley Cambridge Press, 1996.
- [90] D. Su and P. Willis, "Image interpolation by pixel level data-dependent triangulation," *Computer Graphics Forum*, vol. 23, no. 2, pp. 189–201, June 2004.
- [91] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp. 186–200, 1996.
- [92] D. B. H. Tay and N. G. Kingsbury, "Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 466–480, October 1993.
- [93] D. B. H. Tay and N. G. Kingsbury, "Design of 2-D perfect reconstruction filter banks using transformations of variables: IIR case," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 274–279, March 1996.
- [94] M. Unser, "Sampling — 50 years after Shannon," *Proceedings of IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.
- [95] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [96] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. Dragotti, "Directionlets: Anisotropic Multidirectional representation with separable filtering," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1916–1933, 2006.
- [97] R. Venkataramani and Y. Bresler, "Filter design for MIMO sampling and reconstruction," *IEEE Transactions on Signal Processing*, vol. 51, no. 12, pp. 3164–3176, December 2003.
- [98] M. Vetterli, "Multidimensional subband coding: Some theory and algorithms," *Signal Processing*, vol. 6, no. 2, pp. 97–112, February 1984.
- [99] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Prentice-Hall, 1995.
- [100] E. Viscito and J. P. Allebach, "The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 1, pp. 29–42, January 1991.
- [101] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Rate-distortion optimized image compression using wedgelets," in *Proceedings of IEEE International Conference on Image Processing*, Rochester, New York, October 2002.
- [102] A. B. Watson, "The cortex transform: Rapid computation of simulated neural images," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 311–327, 1987.
- [103] L. Ying, L. Demanet, and E. Candès, "3D discrete curvelet transform," in *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing XI*, pp. 344–354, San Diego, USA, 2005.
- [104] J. Zhou and M. N. Do, "Multidimensional multichannel FIR deconvolution using Gröbner bases," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2998–3007, October 2006.