Design and Application of Quincunx Filter Banks

by

Yi Chen
B.Eng., Tsinghua University, China, 2002

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Design and Application of Quincunx Filter Banks

by

Yi Chen
B.Eng., Tsinghua University, China, 2002

Supervisory Committee

Dr. Michael D. Adams, (Department of Electrical and Computer Engineering)

Co-Supervisor

Dr. Wu-Sheng Lu, (Department of Electrical and Computer Engineering)

Co-Supervisor

Dr. Reinhard Illner, (Department of Mathematics and Statistics)

Outside Member

Supervisory Committee

Dr. Michael D. Adams, (Department of Electrical and Computer Engineering)

Co-Supervisor

Dr. Wu-Sheng Lu, (Department of Electrical and Computer Engineering)

Co-Supervisor

Dr. Reinhard Illner, (Department of Mathematics and Statistics)

Outside Member

ABSTRACT

Quincunx filter banks are two-dimensional, two-channel, nonseparable filter banks. They are widely used in many signal processing applications. In this thesis, we study the design and applications of quincunx filter banks in the processing of two-dimensional digital signals.

Symmetric extension algorithms for quincunx filter banks are proposed. In the one-dimensional case, symmetric extension is a commonly used technique to build nonexpansive transforms of finite-length sequences. We show how this technique can be extended to the nonseparable quincunx case. We consider three types of quadrantally-symmetric linear-phase quincunx filter banks, and for each of these types we show how nonexpansive transforms of two-dimensional sequences defined on arbitrary rectangular regions can be constructed.

New optimization-based techniques are proposed for the design of high-performance quincunx filter banks for the application of image coding. The new methods yield linear-phase perfect-reconstruction systems with high coding gain, good analysis/synthesis filter frequency responses, and certain prescribed vanishing moment properties. We present examples of filter banks designed with these techniques and demonstrate their efficiency for image coding relative to existing filter banks. The best filter banks in our design examples outperform other previously proposed quincunx filter banks in approximately 80% cases and sometimes even outperform the well-known 9/7 filter bank from the JPEG-2000 standard.

Contents

List of Tables

List of Figures

List of Acronyms

1D	One-dimensional
2D	Two-dimensional
CR	Compression ratio
HVS	Human visual system
ITI	Integer-to-integer
LTI	Linear time-invariant
MD	Multidimensional
MRA	Multiresolution approximation
MSE	Mean-squared error
PR	(Shift-free) perfect reconstruction
PSNR	Peak-signal-to-noise ratio
SOCP	Second-order cone programming
SVD	Singular value decomposition
UMD	Uniformly maximally decimated

Chapter 1

Introduction

1.1 Quincunx Filter Banks

One-dimensional (1D) and multidimensional (MD) filter banks have proven to be a highly effective tool for the processing of digital signals including speech, image, and video. Usually, the MD case is handled via tensor product, i.e., the MD signal is decomposed into 1D signals and processed by 1D filter banks along each dimension. Some of the more recent efforts concentrate on the nonseparable case, where nonseparable sampling and filtering are employed [1, 2, 3, 4, 5, 6, 7, 8]. The quincunx sampling scheme is the simplest two-dimensional (2D) nonseparable sampling scheme. It is used in many signal processing applications, such as the handling of images returned from remote sensors of satellites [5] and intraframe coding of HDTV [1, 9]. In contrast to the separable case, the quincunx sampling scheme leads to a two-channel filter bank and reduces the scale by a factor of $\sqrt{2}$.

Although the implementation of quincunx filter banks has higher computational complexity than the dyadic separable case, these filter banks offer several important advantages. Firstly, the quincunx filter bank is a good match to the human visual system (HVS) [10]. The HVS has a higher sensitivity to changes in the horizontal and vertical directions [11]. This is equivalent to saying that the HVS is more accurate in perceiving high frequencies in the horizontal and vertical directions than along diagonals. Figure 1.1 shows the frequency response of a typical quincunx lowpass filter, where the shaded and unshaded regions correspond to the passband and stopband, respectively. With the diamond-shaped passband, this filter conserves horizontal and vertical high frequencies, and cuts diagonal frequencies by half. In this way, the quincunx filter bank well matches the HVS. Another advantage of quincunx filter banks is that there are more degrees of

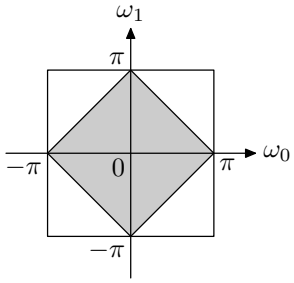


Figure 1.1: Frequency responses of a quincunx lowpass filter. The shaded and unshaded regions represent the passband and stopband, respectively.

freedom in the design of such filter banks. This may lead to filter banks with better performance for targeted applications.

1.2 Historical Perspective

Although 1D filter banks have been well studied, in the MD case, many problems remain unsolved. Filter banks are often defined to operate on signals of infinite extent. In practice, however, we frequently deal with signals of finite extent. This leads to the well-known boundary problem that can arise whenever a finite-extent signal is filtered. In the 1D case, several solutions have been proposed to solve this problem by extending the finite-extent signal into a signal with infinite extent. Zero padding and periodic extension [12, 13, 14] introduce sharp discontinuities in the extended signals, which cause distortion at edges of the reconstructed signals. Symmetric extension [14, 15, 16] is the most commonly used solution to the boundary problem in the 1D case. This extension scheme provides smooth extended signals and leads to desirable nonexpansive transforms. In the MD case, symmetric extension is often applied to the signals separably along each dimension.

For 1D filter banks, various design techniques have been successfully developed. In the nonseparable MD case, however, far fewer effective methods have been proposed. Variable transformation methods are commonly used for the design of MD filter banks. With such methods, a 1D prototype filter bank is designed first. Then it is mapped into an MD filter bank by a change of variables. For example, the McClellan transformation [17] has been used in several design approaches [18, 19, 20, 21]. In these designs, the frequency responses of the 1D filters are mapped into MD frequency responses. Other design techniques have also been proposed where a transformation is applied to the polyphase components of the filters instead of the original filter transfer functions [22, 5, 7, 23]. These transformation-based designs have the restriction that one cannot

explicitly control the shape of the MD frequency responses, while in some cases the transformed MD filter banks can only achieve approximate perfect reconstruction. Direct optimization of the filter coefficients has also been proposed [24, 2, 25], but because of the involvement of large numbers of variables and nonlinear, nonconvex constraints, such optimization typically leads to a very complicated system, which is often difficult to solve. Designs through the lifting framework [26, 27] have been proposed in [28, 6] for two-channel MD filter banks with an arbitrary number of vanishing moments. With these methods, however, only interpolating filter banks (i.e., filter banks with two lifting steps) are considered. Thus, good filter banks with more lifting steps cannot be designed with these approaches.

1.3 Overview and Contribution of This Thesis

This thesis is primarily concerned with the design and application of quincunx filter banks. A symmetric extension algorithm is presented to build nonexpansive transforms associated with quincunx filter banks. Then an optimization-based design algorithm with some variations is proposed for constructing quincunx filter banks with a number of desirable characteristics. Finally, the optimally designed filter banks are compared to some previously proposed ones in terms of their performance in image coding.

The remainder of this thesis is structured as follows. Chapter 2 introduces the background necessary to understand this work. We begin by discussing the notational conventions used herein. Then, we introduce multidimensional multirate systems and filter banks, and examine in detail the quincunx filter banks, which are of the most interest in this work. At last, we present some basic concepts related to subband image coding.

In the 1D case, when processing signals with finite lengths, symmetric extension is a very useful algorithm to handle the signal boundaries and build nonexpansive transforms for such signals. In Chapter 3, we show how this technique can be extended to the 2D quincunx case. To this end, we first define four ways to extend finite-extent 2D sequences to infinite-extent sequences with four-fold symmetry and periodicity. Then we discuss how these properties can be preserved under nonseparable sampling and filtering. Finally, we propose several symmetric extension algorithms for building nonexpansive transforms with quincunx filter banks, and illustrate the algorithms with several examples.

Chapter 4 presents new optimal design algorithms for quincunx filter banks. We begin with a lifting parametrization of quincunx filter banks such that all of the filters have symmetric or antisymmetric linear phase. Based on this parametrization, we further show how to build filter banks compatible with the symmetric extension algorithms discussed in Chapter 3. Then an optimization-based design algorithm is proposed for the design of quincunx filter banks with perfect reconstruction (PR), linear phase, high coding gain, good

frequency selectivity and prescribed numbers of vanishing moments. We show how this complex design problem can be formulated as a second-order cone programming (SOCP) problem. Several variations of the proposed algorithm are also investigated. Design examples are presented to demonstrate the effectiveness of our proposed design method. At the end of this chapter, we examine the performance of the optimal filter banks, as well as some existing filter banks, in an image coder, and comment on their coding performance. The experimental results show that our new filter banks outperform the previously proposed quincunx filter banks in most cases, and sometimes even outperform the 9/7 filter bank, which is considered to be one of the very best in the literature.

Chapter 5 summarizes the results presented in this thesis and suggests some related topics for future research.

Chapter 2

Preliminaries

2.1 Overview

In this chapter, we first explain some fundamental concepts related to this work. We begin with an introduction to the notation and terminology used herein. We then present some of the basic concepts on multirate systems and filter banks in the MD case. We conclude the chapter by a brief discussion on subband image coding.

2.2 Notation and Terminology

In this work, matrices and vectors are denoted by upper and lower case boldface letters, respectively. The symbols $\mathbb{C}$, $\mathbb{R}$, and $\mathbb{Z}$ denote the sets of complex numbers, real numbers, and integers, respectively. The symbol j denotes $\sqrt{-1}$. For $c \in \mathbb{C}$, c^* denotes the complex conjugate of c. In $\mathbb{R}$, (a,b), $[a,b]$, and $[a,b)$ denote the open interval $\{x : a < x < b\}$, the closed interval $\{x : a \leq x \leq b\}$, and the half-open half-closed interval $\{x : a \leq x < b\}$, respectively. The symbols $\mathbb{Z}^*$, $\mathbb{Z}^+$, $\mathbb{Z}^-$, $\mathbb{Z}_{odd}$, and $\mathbb{Z}_{even}$ denote the sets of nonnegative, positive, negative, odd, and even integers, respectively. For a set S and a scalar k, the notation kS denotes the set $\{ks\}_{s\in S}$. If $k \in \mathbb{Z}^+$, S^k denotes the k-fold Cartesian product of S, i.e., $S^k = \left\{\boldsymbol{s} = [s_0 \quad s_1 \quad \cdots \quad s_{k-1}]^T\right\}_{s_i \in S}$. As an example, $\mathbb{Z}^2$ denotes the set of ordered pairs of integers. Furthermore, for a $k \times k$ matrix $\boldsymbol{M}$, $\boldsymbol{M}S^k$ denotes the set $\{\boldsymbol{Ms}\}_{\boldsymbol{s}\in S^k}$. The difference of two sets A and B is denoted $A \setminus B$ and defined as $A \setminus B = \{x : x \in A, x \notin B\}$.

The symbols $\mathbf{0}$, $\mathbf{1}$ and $\boldsymbol{I}$ are used to denote a vector/matrix of all zeros, all ones, and an identity matrix, respectively, the dimensions of which should be clear from the context. In particular, $\boldsymbol{I}_k$ denotes an identity

matrix of size $k \times k$ for some $k \in \mathbb{Z}$. The symbols $\mathbf{0}_k$ and $\mathbf{1}_k$ are used to denote k-dimensional vectors of all zeros and ones, respectively, and $\mathbf{0}_{k_0 \times k_1}$ and $\mathbf{1}_{k_0 \times k_1}$ are used to denote $k_0 \times k_1$ matrices of all zeros and ones, respectively. For two vectors/matrices $\boldsymbol{u}$ and $\boldsymbol{v}$, $\boldsymbol{u} \circ \boldsymbol{v}$ denotes the **Schur product** (i.e., element-wise product) of $\boldsymbol{u}$ and $\boldsymbol{v}$. We write $\boldsymbol{u} \geq \boldsymbol{v}$ if every element in $\boldsymbol{u}$ is no less than its corresponding element in $\boldsymbol{v}$. The notations $\boldsymbol{u} > \boldsymbol{v}$, $\boldsymbol{u} \leq \boldsymbol{v}$ and $\boldsymbol{u} < \boldsymbol{v}$ are defined in a similar way. For two D-dimensional vectors $\boldsymbol{n} = [n_0 \quad n_1 \quad \cdots \quad n_{D-1}]^T$ and $\boldsymbol{z} = [z_0 \quad z_1 \quad \cdots \quad z_{D-1}]^T$, we define

$$|\boldsymbol{n}| = \sum_{i=0}^{D-1} n_i \quad \text{and} \quad \boldsymbol{z}^{\boldsymbol{n}} = \prod_{i=0}^{D-1} z_i^{n_i}.$$

Furthermore, for a $D \times D$ matrix $\boldsymbol{M} = [\boldsymbol{m}_0 \quad \boldsymbol{m}_1 \quad \cdots \quad \boldsymbol{m}_{D-1}]$ with $\boldsymbol{m}_k$ being the kth column of $\boldsymbol{M}$, we define

$$\boldsymbol{z}^{\boldsymbol{M}} = \left[\boldsymbol{z}^{\boldsymbol{m}_0} \quad \boldsymbol{z}^{\boldsymbol{m}_1} \quad \cdots \quad \boldsymbol{z}^{\boldsymbol{m}_{D-1}}\right]^T.$$

Note that $|\boldsymbol{n}|$ and $\boldsymbol{z}^{\boldsymbol{n}}$ are scalars, while $\boldsymbol{z}^{\boldsymbol{M}}$ is a vector. With these notations, it can be verified that $\left(\boldsymbol{z}^{\boldsymbol{M}}\right)^{\boldsymbol{n}} = \boldsymbol{z}^{\boldsymbol{M}\boldsymbol{n}}$ and $\left(\boldsymbol{z}^{\boldsymbol{M}}\right)^{\boldsymbol{L}} = \boldsymbol{z}^{\boldsymbol{M}\boldsymbol{L}}$. For matrix multiplication, we define the product notation as

$$\prod_{k=M}^{N} \boldsymbol{A}_k = \begin{cases} \boldsymbol{A}_N \boldsymbol{A}_{N-1} \cdots \boldsymbol{A}_{M+1} \boldsymbol{A}_M & \text{for } N \geq M \\ \boldsymbol{A}_N \boldsymbol{A}_{N+1} \cdots \boldsymbol{A}_{M-1} \boldsymbol{A}_M & \text{for } N < M. \end{cases}$$

For convenience, in the rest of this thesis a linear (or polynomial) function of the elements of a vector $\boldsymbol{x}$ is simply referred to as a linear (or polynomial) function of $\boldsymbol{x}$.

For $a \in \mathbb{R}$, $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a, and $\lceil a \rceil$ denotes the least integer no less than a. For an $M \times N$ matrix $\boldsymbol{A}$ with the (i,j)th element being $a_{i,j}$, $\lfloor \boldsymbol{A} \rfloor$ and $\lceil \boldsymbol{A} \rceil$ each denotes an $M \times N$ matrix where the (i,j)th element is $\lfloor a_{i,j} \rfloor$ and $\lceil a_{i,j} \rceil$, respectively. For $m, n \in \mathbb{Z}$, we define the **mod** function as $\operatorname{mod}(m,n) = m - n \lfloor m/n \rfloor$.

2.3 Multidimensional Multirate Systems

Multirate systems are very useful in processing digital signals. In this section, we explain the basic concepts of multirate signal processing and extend them to the MD case. We begin with an introduction to MD signals and filter banks, and then concentrate on the quincunx case. Next, we briefly comment on the relation between quincunx filter banks and dyadic wavelet systems. Lastly, we introduce the lifting scheme that can be used to efficiently design and implement filter banks.

2.3.1 Multidimensional Signals

We first introduce the notions of MD signals and filters. A D-dimensional signal x is a sequence of real numbers given by

$$x = \left\{ x[\boldsymbol{n}] \in \mathbb{R} \,\middle|\, \boldsymbol{n} \in \mathbb{Z}^D \right\}.$$

An element of x is denoted either as $x[\boldsymbol{n}]$ or $x[n_0, n_1, \ldots, n_{D-1}]$ (whichever is more convenient), where $\boldsymbol{n} = [n_0 \quad n_1 \quad \cdots \quad n_{D-1}]^T$ and $n_i \in \mathbb{Z}$. If only a finite number of $x[\boldsymbol{n}]$ are nonzero, the sequence x is said to have **finite support**. For a nonsingular integer matrix $\boldsymbol{P}$, if $x[\boldsymbol{n}] = x[\boldsymbol{n} + \boldsymbol{P}\boldsymbol{k}]$ for all $\boldsymbol{n}, \boldsymbol{k} \in \mathbb{Z}^D$, the sequence x is said to be $\boldsymbol{P}$**-periodic** and $\boldsymbol{P}$ is called a **periodicity matrix**. The Fourier transform $\hat{x}(\boldsymbol{\omega})$ of x and the inverse Fourier transform of $\hat{x}(\boldsymbol{\omega})$ are defined as

$$\hat{x}(\boldsymbol{\omega}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^D} x[\boldsymbol{n}] e^{-j\boldsymbol{\omega}^T \boldsymbol{n}} \quad \text{and} \quad x[\boldsymbol{n}] = \frac{1}{(2\pi)^D} \int_{[-\pi,\pi)^D} \hat{x}(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^T \boldsymbol{n}} \mathrm{d}\boldsymbol{\omega},$$

respectively. The z-transform of x is defined as

$$X(\boldsymbol{z}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^D} x[\boldsymbol{n}] \boldsymbol{z}^{-\boldsymbol{n}}.$$

For a D-dimensional FIR filter H, its impulse response h is a finitely supported sequence defined on $\mathbb{Z}^D$. The transfer function $H(\boldsymbol{z})$ and frequency response $\hat{h}(\boldsymbol{\omega})$ of H are given by

$$H(\boldsymbol{z}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^D} h[\boldsymbol{n}] \boldsymbol{z}^{-\boldsymbol{n}} \quad \text{and} \quad \hat{h}(\boldsymbol{\omega}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^D} h[\boldsymbol{n}] e^{-j\boldsymbol{\omega}^T \boldsymbol{n}},$$

respectively. Figure 2.1 shows a linear time-invariant (LTI) system characterized by the transfer function $H(\boldsymbol{z})$. The output sequence y is computed by the convolution of x and h as

$$y[\boldsymbol{n}] = \sum_{\boldsymbol{k} \in \mathbb{Z}^D} x[\boldsymbol{k}] h[\boldsymbol{n} - \boldsymbol{k}]. \tag{2.1}$$

The above input-output relation (2.1) is equivalent to $\hat{y}(\boldsymbol{\omega}) = \hat{x}(\boldsymbol{\omega})\hat{h}(\boldsymbol{\omega})$ and $Y(\boldsymbol{z}) = X(\boldsymbol{z})H(\boldsymbol{z})$ in the frequency domain and z-domain, respectively.

For a 2D filter H, for convenience, we express its impulse response h in the form of a matrix $\boldsymbol{A}_h$ and denote the relationship of h and $\boldsymbol{A}_h$ as

$$h[\boldsymbol{n}] \sim \boldsymbol{A}_h. \tag{2.2}$$

In $\boldsymbol{A}_h$, the element corresponding to $h[0,0]$ is framed. For example, a filter H with impulse response $h[-1,0] = 1$, $h[-1,1] = 2$, $h[-1,2] = 3$, $h[0,0] = 4$, $h[0,1] = 5$, and $h[0,2] = 6$ is denoted as

$$h[\boldsymbol{n}] \sim \begin{bmatrix} h[-1,2] & h[0,2] \\ h[-1,1] & h[0,1] \\ h[-1,0] & \boxed{\text{h[0,0]}} \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 2 & 5 \\ 1 & \boxed{4} \end{bmatrix}.$$

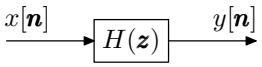


Figure 2.1: An MD digital filter.

A D-dimensional filter H with impulse response h is said to have **linear phase with group delay $\boldsymbol{c}$** if, for some $\boldsymbol{c} \in \frac{1}{2}\mathbb{Z}^D$ and $S \in \{-1, 1\}$,

$$h[\boldsymbol{n}] = Sh[2\boldsymbol{c} - \boldsymbol{n}] \quad \text{for all } \boldsymbol{n} \in \mathbb{Z}^D. \tag{2.3}$$

The filter H is said to be **symmetric** if $S = 1$, and **antisymmetric** if $S = -1$. For a linear-phase filter H, its transfer function $H(\boldsymbol{z})$ satisfies $H(\boldsymbol{z}) = S\boldsymbol{z}^{-2\boldsymbol{c}}H(\boldsymbol{z}^{-1})$, and its frequency response can be expressed as

$$\begin{aligned}
\hat{h}(\boldsymbol{\omega}) &= \sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]e^{-j\boldsymbol{\omega}^T\boldsymbol{n}} = S\sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[2\boldsymbol{c}-\boldsymbol{n}]e^{-j\boldsymbol{\omega}^T\boldsymbol{n}} = S\sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]e^{-j\boldsymbol{\omega}^T(2\boldsymbol{c}-\boldsymbol{n})} \\
&= \frac{1}{2}\sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]\left[e^{-j\boldsymbol{\omega}^T\boldsymbol{n}} + Se^{-j\boldsymbol{\omega}^T(2\boldsymbol{c}-\boldsymbol{n})}\right] \\
&= \frac{1}{2}e^{-j\boldsymbol{\omega}^T\boldsymbol{c}}\sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]\left[e^{-j\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{c})} + Se^{-j\boldsymbol{\omega}^T(\boldsymbol{c}-\boldsymbol{n})}\right] \\
&= \begin{cases} e^{-j\boldsymbol{\omega}^T\boldsymbol{c}}\sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]\cos\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{c})\right] & \text{for } S=1 \\ e^{-j\left(\boldsymbol{\omega}^T\boldsymbol{c}+\pi/2\right)}\sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]\sin\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{c})\right] & \text{for } S=-1. \end{cases}
\end{aligned} \tag{2.4}$$

For the case with $S = 1$, we define the **signed amplitude response** $\hat{h}_a(\boldsymbol{\omega})$ to be $\hat{h}(\boldsymbol{\omega})$ without the exponential factor $e^{-j\boldsymbol{\omega}^T\boldsymbol{c}}$, i.e.,

$$\hat{h}_a(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^D} h[\boldsymbol{n}]\cos\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{c})\right] \quad \text{for} \quad S = 1. \tag{2.5}$$

The quantity $\hat{h}_a(\boldsymbol{\omega})$ determines the shape of the frequency response, and $\left|\hat{h}_a(\boldsymbol{\omega})\right|$ is equivalent to the **amplitude response** of H.

The MD sequences that we have discussed above are all defined on the D-dimensional integer lattice $\mathbb{Z}^D$. In multirate systems, we often deal with sequences defined on a subset of $\mathbb{Z}^D$, called a **lattice**, associated with a generating matrix $\boldsymbol{M}$. Below we introduce some fundamentals on lattices.

Let $\boldsymbol{M} = [\boldsymbol{m}_0 \quad \boldsymbol{m}_1 \quad \cdots \quad \boldsymbol{m}_{D-1}]^T$ be a $D \times D$ nonsingular integer matrix with $\boldsymbol{m}_k \in \mathbb{Z}^D$ being the kth column of $\boldsymbol{M}$. Since $\boldsymbol{M}$ is nonsingular, the set $\{\boldsymbol{m}_k\}$ is linearly independent. The lattice $\mathrm{LAT}(\boldsymbol{M})$ is defined as the set of all possible vectors that can be represented as integer linear combinations of $\boldsymbol{m}_k$ [29], i.e.,

$$\mathrm{LAT}(\boldsymbol{M}) = \left\{ \boldsymbol{x} \in \mathbb{Z}^D \,\middle|\, \boldsymbol{x} = \sum_{k=0}^{D-1} n_k\boldsymbol{m}_k = \boldsymbol{M}\boldsymbol{n}, \forall \boldsymbol{n} = [n_0 \;\; n_1 \;\; \ldots \;\; n_{D-1}]^T \in \mathbb{Z}^D \right\}. \tag{2.6}$$

Using the notation we introduced in Section 2.2, $\mathrm{LAT}(\boldsymbol{M})$ can be written as $\boldsymbol{M}\mathbb{Z}^D$. The matrix $\boldsymbol{M}$ is called a generating matrix or sampling matrix of $\mathrm{LAT}(\boldsymbol{M})$, and its columns $\{\boldsymbol{m}_k\}$ are called the basis vectors. Note

that the generating matrix for a lattice is not unique. The **sampling density** of LAT($\boldsymbol{M}$) is defined as

$$d = \frac{1}{|\det \boldsymbol{M}|}, \tag{2.7}$$

which describes the number of lattice points in a unit volume.

Given a sampling matrix $\boldsymbol{M}$, the **fundamental parallelepiped**, denoted as FPD($\boldsymbol{M}$), is defined as

$$\mathrm{FPD}(\boldsymbol{M}) = \left\{ \boldsymbol{x} \in \mathbb{R}^D \,\middle|\, \boldsymbol{x} = \boldsymbol{M}\boldsymbol{\alpha}, \boldsymbol{\alpha} \in [0,1)^D \right\},$$

where $[0,1)^D$ denotes the D-fold Cartesian product of the half-open half-closed interval $[0,1)$. The finite set of integer vectors contained in FPD($\boldsymbol{M}$) is denoted as $\mathscr{N}(\boldsymbol{M})$ and $\mathscr{N}(\boldsymbol{M}) = \mathrm{FPD}(\boldsymbol{M}) \bigcap \mathbb{Z}^D$. Let $\boldsymbol{n}$ be an arbitrary vector in $\mathbb{Z}^D$, then $\boldsymbol{n}$ can be expressed as [20]

$$\boldsymbol{n} = \boldsymbol{k} + \boldsymbol{M}\boldsymbol{m}, \tag{2.8}$$

where $\boldsymbol{k}$ and $\boldsymbol{m}$ are unique vectors satisfying $\boldsymbol{k} \in \mathscr{N}(\boldsymbol{M})$ and $\boldsymbol{m} \in \mathbb{Z}^D$. For a given vector $\boldsymbol{n}$ and a matrix $\boldsymbol{M}$, we denote the unique vector $\boldsymbol{k}$ satisfying (2.8) as $\boldsymbol{k} = \mathrm{mod}(\boldsymbol{n}, \boldsymbol{M})$. A **coset** of LAT($\boldsymbol{M}$) in $\mathbb{Z}^D$ is the set of all vectors of the form (2.8), where $\boldsymbol{k}$ is fixed and called the **coset vector** of this coset. The number of distinct cosets of LAT($\boldsymbol{M}$) is $|\det \boldsymbol{M}|$.

Figure 2.2(a) shows a lattice with its fundamental parallelepiped and two basis vectors $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$. A generating matrix of this lattice is $\boldsymbol{M} = [\boldsymbol{m}_0 \;\; \boldsymbol{m}_1] = \left[\begin{smallmatrix} 1 & 1 \\ 2 & -1 \end{smallmatrix}\right]$, and the sampling density is $\frac{1}{3}$. There are also other matrices that generate this lattice, such as $\left[\begin{smallmatrix} 1 & -1 \\ 2 & 1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} 2 & 1 \\ 1 & -1 \end{smallmatrix}\right]$. Figure 2.2(b) shows the $|\det \boldsymbol{M}| = 3$ distinct cosets represented by symbols $\bullet$, $\circ$, and $\times$, which are associated with coset vectors $[0 \;\; 0]^T$, $[1 \;\; 1]^T$, and $[1 \;\; 0]^T$, respectively.

2.3.2 Multirate Fundamentals

In this part, we show the important multirate concepts for the MD case, including downsampling, upsampling and polyphase decomposition of signals and filters. The basic building blocks of a multirate system are the **downsampler** and **upsampler**, which perform the operations of downsampling and upsampling, respectively. Figure 2.3 shows a downsampler, where the input x is downsampled by a nonsingular integer matrix $\boldsymbol{M}$, and the output y is given by

$$y[\boldsymbol{n}] = (\downarrow \boldsymbol{M})x[\boldsymbol{n}] = x[\boldsymbol{M}\boldsymbol{n}], \tag{2.9}$$

that is, the output y contains all samples on LAT($\boldsymbol{M}$). Through the downsampler, the sampling density is reduced by a factor of $|\det \boldsymbol{M}|$. The Fourier transform $\hat{y}(\boldsymbol{\omega})$ of y can be written in terms of the Fourier

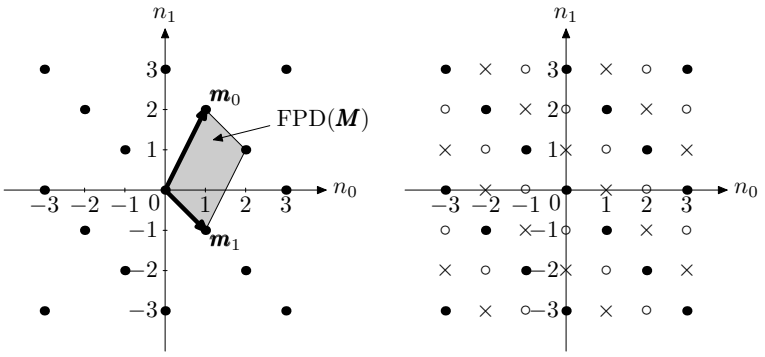


Figure 2.2: (a) A lattice with generating matrix $\left[\begin{smallmatrix}1 & 1\\ 2 & -1\end{smallmatrix}\right]$, and (b) its three distinct cosets.

Figure 2.3: An MD downsampler.

transform $\hat{x}(\boldsymbol{\omega})$ of x. The relation is given by

$$\hat{y}(\boldsymbol{\omega}) = \frac{1}{|\det \boldsymbol{M}|} \sum_{\boldsymbol{k} \in \mathcal{N}(\boldsymbol{M}^T)} \hat{x}\left(\boldsymbol{M}^{-T}(\boldsymbol{\omega} - 2\pi \boldsymbol{k})\right).$$

Let $X(\boldsymbol{z})$ and $Y(\boldsymbol{z})$ be the z-transforms of x and y, respectively. Then, downsampling in the z-domain can be expressed as

$$Y(\boldsymbol{z}) = \frac{1}{|\det \boldsymbol{M}|} \sum_{\boldsymbol{k} \in \mathcal{N}(\boldsymbol{M}^T)} X(\boldsymbol{e_l} \circ \boldsymbol{z}^{\boldsymbol{M}^{-1}}), \tag{2.10}$$

where $\boldsymbol{e_l} = \left[e^{l_0} \;\; e^{l_1} \;\; \cdots \;\; e^{l_{D-1}}\right]^T$ and $\boldsymbol{l} = [l_0 \;\; l_1 \;\; \cdots \;\; l_{D-1}]^T = \left(-j2\pi \boldsymbol{k}^T \boldsymbol{M}^{-1}\right)^T$. In the frequency domain, the spectrum of the downsampled signal is the average of $|\det \boldsymbol{M}|$ shifted and stretched versions of the spectrum of the original signal.

Figure 2.4 shows an upsampler, where $\boldsymbol{M}$ is a nonsingular integer matrix. The output y is given by

$$y[\boldsymbol{n}] = (\uparrow \boldsymbol{M}) x[\boldsymbol{n}] = \begin{cases} x[\boldsymbol{M}^{-1}\boldsymbol{n}] & \text{if } \boldsymbol{n} \in \mathrm{LAT}(\boldsymbol{M}) \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

The input-output relation in the Fourier domain and z-domain are similar to the 1D case, and are given by

$$\hat{y}(\boldsymbol{\omega}) = \hat{x}\left(\boldsymbol{M}^T \boldsymbol{\omega}\right) \quad \text{and} \quad Y(\boldsymbol{z}) = X\left(\boldsymbol{z}^{\boldsymbol{M}}\right),$$

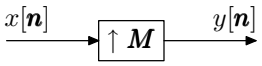


Figure 2.4: An MD upsampler.

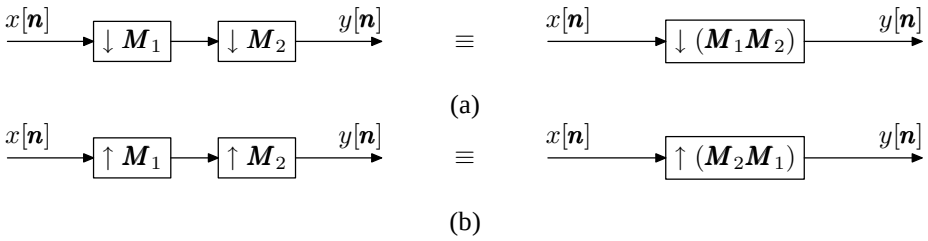


Figure 2.5: Cascade connections and their equivalent forms of the (a) downsamplers and (b) upsamplers.

respectively. The upsampled signal y is nonzero only at points on the lattice LAT($\boldsymbol{M}$). In the frequency domain, the upsampler performs a linear transformation of the frequency vector $\boldsymbol{\omega}$, and $|\det\boldsymbol{M}|$ copies of the original baseband spectrum are squeezed into the region $[-\pi,\pi)^2$.

Sometimes the downsamplers/upsamplers are applied in cascade. They can be combined as follows. Figure 2.5(a) shows a cascade of two downsamplers with the downsampling matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ and its equivalent form with a single downsampler $\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2$. Figure 2.5(b) shows a cascade of two upsamplers with $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ and its equivalent structure with a single upsampler $\boldsymbol{M} = \boldsymbol{M}_2\boldsymbol{M}_1$.

The downsampler and upsampler are often used in cascade with filters. The order of the downsampler/upsampler and the filter can be interchanged under certain circumstances. Figures 2.6(a) and (b) show the equivalent structures for the downsampling and upsampling operations, respectively. They are called the noble identities. With these identities, one can apply the convolution operation on the side of the downsampler or upsampler with lower sampling density, which is very useful to improve the computation efficiency.

Now we consider the polyphase decomposition of MD signals and filters. From Section 2.3.1, we know that an arbitrary MD integer vector $\boldsymbol{n}$ can be expressed uniquely in the form of (2.8). Therefore, given a sequence x and a sampling matrix $\boldsymbol{M}$, there are $M = |\det\boldsymbol{M}|$ unique subsequences

$$x_k[\boldsymbol{n}] = x[\boldsymbol{M}\boldsymbol{n} + \boldsymbol{m}_k], \tag{2.12}$$

for $k = 0,1,\ldots,M-1$, $\boldsymbol{m}_k \in \mathcal{N}(\boldsymbol{M})$ and $\{\boldsymbol{m}_k\}$ are distinct. The subsequence x_k is called the kth **type-1 polyphase component** of x. As $x_k[\boldsymbol{n}]$ is the $\boldsymbol{M}$-fold downsampled version of $x[\boldsymbol{n}+\boldsymbol{m}_k]$, the sequence x can be written as the sum of the upsampled and shifted versions of its polyphase components $\{x_k\}$ as

$$x[\boldsymbol{n}] = \sum_{k=0}^{M-1} ((\uparrow \boldsymbol{M})x_{\boldsymbol{k}})[\boldsymbol{n} - \boldsymbol{m}_k]. \tag{2.13}$$

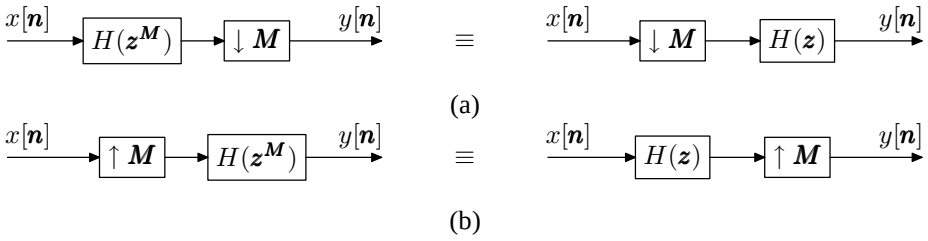


Figure 2.6: Noble identities of the (a) downsampler and (b) upsampler.

The above equation (2.13) is called the **type-1 polyphase representation** of x. In the Fourier domain and z-domain, (2.13) can be expressed as

$$\hat{x}(\boldsymbol{\omega}) = \sum_{k=0}^{M-1} e^{-j\boldsymbol{\omega}^T \boldsymbol{m}_k} \hat{x}_k(\boldsymbol{M}^T \omega) \quad \text{and} \quad X(\boldsymbol{z}) = \sum_{k=0}^{M-1} \boldsymbol{z}^{-\boldsymbol{m}_k} X_k\left(\boldsymbol{z}^{\boldsymbol{M}}\right),$$

respectively.

Similarly, we define the kth **type-2 polyphase component** of a sequence x as

$$x_k[\boldsymbol{n}] = x[\boldsymbol{M}\boldsymbol{n} - \boldsymbol{m}_k], \tag{2.14}$$

where $k \in \{0,1,\ldots,M-1\}$, $\boldsymbol{m}_k \in \mathscr{N}(\boldsymbol{M})$ and $\{\boldsymbol{m}_k\}$ are distinct. The time domain, Fourier domain, and z-domain expressions of the type-2 polyphase representation of a sequence x are respectively given by

$$x[\boldsymbol{n}] = \sum_{k=0}^{M-1} ((\uparrow \boldsymbol{M}) x_{\boldsymbol{k}})[\boldsymbol{n} + \boldsymbol{m}_k],$$

$$\hat{x}(\boldsymbol{\omega}) = \sum_{k=0}^{M-1} e^{j\boldsymbol{\omega}^T \boldsymbol{m}_k} \hat{x}_k(\boldsymbol{M}^T \omega), \quad \text{and} \quad X(\boldsymbol{z}) = \sum_{k=0}^{M-1} \boldsymbol{z}^{\boldsymbol{m}_k} X_k\left(\boldsymbol{z}^{\boldsymbol{M}}\right).$$

2.3.3 Uniformly Maximally Decimated Filter Banks

The uniformly maximally decimated (UMD) filter bank is of great importance in multirate systems. The block diagram of a UMD filter bank with $M = |\det \boldsymbol{M}|$ channels is shown in Figure 2.7. On the analysis side, the analysis filters $\{\mathrm{H}_k\}$ divide the input sequence x into subbands in the D-dimensional frequency domain. The output of each analysis filter is then downsampled by $\boldsymbol{M}$, yielding the subband sequences $\{y_k\}$. Since there are M analysis filters and each downsampler reduces the sampling density by a factor of M, the combined sampling rate of the subbands $\{y_k\}$ is the same as that of the input x. On the synthesis side, the subband sequences are upsampled by $\boldsymbol{M}$, and then pass through the synthesis filters $\{\mathrm{G}_k\}$. The outputs of the synthesis filters are added together to obtain the reconstructed sequence x_r. If $x_r[\boldsymbol{n}] = x[\boldsymbol{n}]$, the filter bank is said to have the **shift-free perfect reconstruction** (PR) property. The shift-free PR property is desirable in

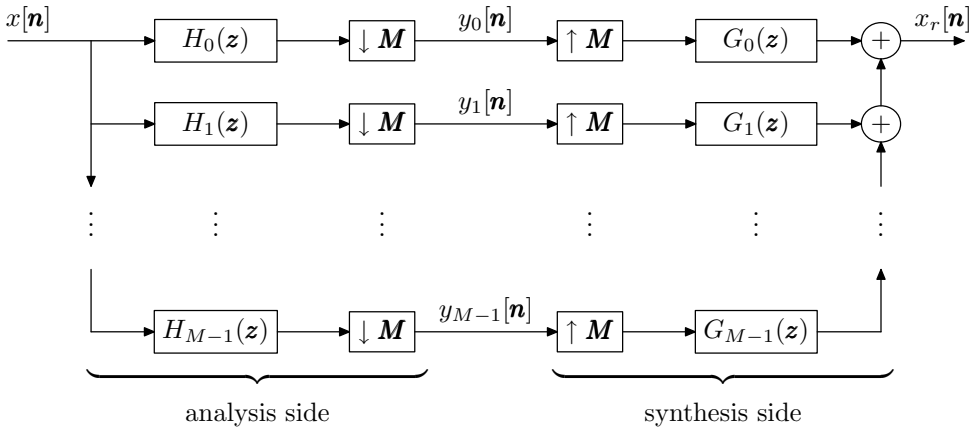


Figure 2.7: An M-channel UMD filter bank, where $M = |\det \boldsymbol{M}|$.

many signal processing applications. In this thesis, henceforth, the term PR shall denote shift-free perfect reconstruction unless explicitly noted otherwise.

With the polyphase representation introduced in Section 2.3.2, the UMD filter bank can also be implemented in the polyphase domain. Each analysis filter H_k can be represented in the form of

$$H_k(\boldsymbol{z}) = \sum_{i=0}^{M-1} \boldsymbol{z}^{\boldsymbol{m}_i} H_{k,i}\left(\boldsymbol{z}^{\boldsymbol{M}}\right), \tag{2.15}$$

where $H_{k,i}(\boldsymbol{z})$ is the ith type-2 polyphase component of H_k. The analysis filter transfer functions $\{H_k(\boldsymbol{z})\}$ can be written as

$$\begin{bmatrix} H_0(\boldsymbol{z}) \\ H_1(\boldsymbol{z}) \\ \vdots \\ H_{M-1}(\boldsymbol{z}) \end{bmatrix} = \underbrace{\begin{bmatrix} H_{0,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & H_{0,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & \cdots & H_{0,M-1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \\ H_{1,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & H_{1,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & \cdots & H_{1,M-1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & H_{M-1,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & \cdots & H_{M-1,M-1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \end{bmatrix}}_{\boldsymbol{H}_p\left(\boldsymbol{z}^{\boldsymbol{M}}\right)} \begin{bmatrix} \boldsymbol{z}^{\boldsymbol{m}_0} \\ \boldsymbol{z}^{\boldsymbol{m}_1} \\ \vdots \\ \boldsymbol{z}^{\boldsymbol{m}_{M-1}} \end{bmatrix}. \tag{2.16}$$

The matrix $\boldsymbol{H}_p(\boldsymbol{z})$ is called the **analysis polyphase matrix**.

Similarly, the synthesis filter transfer functions $\{G_k(\boldsymbol{z})\}$ can be written as

$$\begin{bmatrix} G_0(\boldsymbol{z}) \\ G_1(\boldsymbol{z}) \\ \vdots \\ G_{M-1}(\boldsymbol{z}) \end{bmatrix} = \underbrace{\begin{bmatrix} G_{0,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & G_{0,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & \cdots & G_{0,M-1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \\ G_{1,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & G_{1,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & \cdots & G_{1,M-1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \\ \vdots & \vdots & \ddots & \vdots \\ G_{M-1,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & G_{M-1,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) & \cdots & G_{M-1,M-1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \end{bmatrix}}_{\boldsymbol{G}_p^T\left(\boldsymbol{z}^{\boldsymbol{M}}\right)} \begin{bmatrix} \boldsymbol{z}^{-\boldsymbol{m}_0} \\ \boldsymbol{z}^{-\boldsymbol{m}_1} \\ \vdots \\ \boldsymbol{z}^{-\boldsymbol{m}_{M-1}} \end{bmatrix}, \tag{2.17}$$

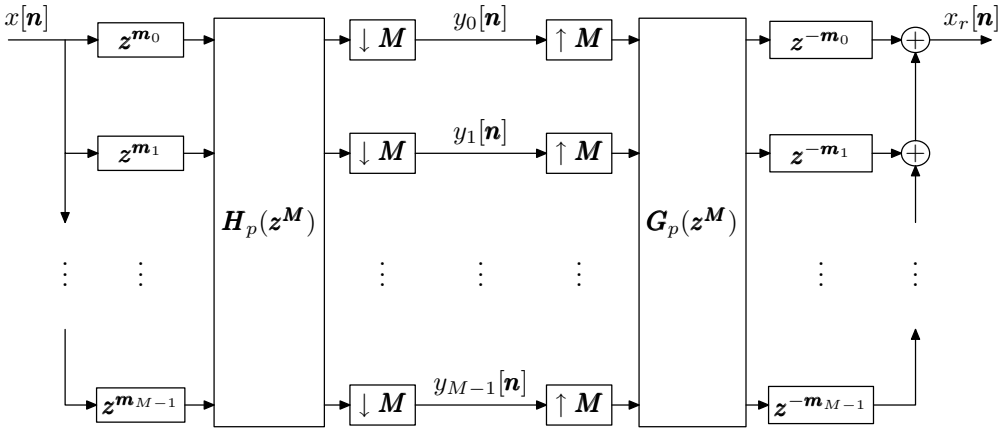


Figure 2.8: The polyphase representation of a UMD filter bank before simplification with the noble identities.

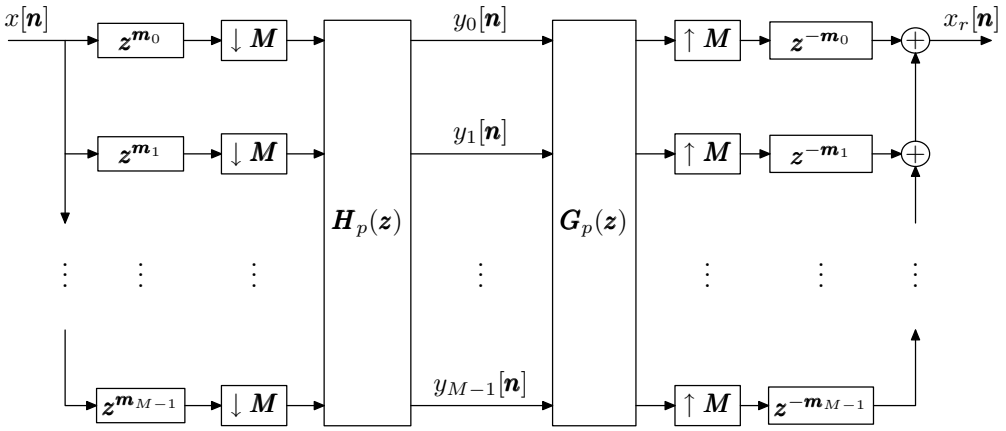


Figure 2.9: The polyphase representation of a UMD filter bank.

where $G_{k,i}(z)$ is the ith type-1 polyphase component of the synthesis filter G_k, i.e., $G_k(z) = \sum_{i=0}^{M-1} z^{-m_k} G_{k,i}\left(z^M\right)$, and $\boldsymbol{G}_p(z)$ is called the **synthesis polyphase matrix**. With (2.16) and (2.17), the filter bank can be implemented in its polyphase domain as shown in Figure 2.8. Using the noble identities, we can interchange the downsamplers/upsamplers and the polyphase matrices to obtain the simplified structure shown in Figure 2.9. This structure provides a convenient way to design and implement UMD filter banks. In order for the filter bank to have (shift-free) PR, the polyphase matrices must satisfy

$$\boldsymbol{H}_p(z)\boldsymbol{G}_p(z) = \boldsymbol{I}. \tag{2.18}$$

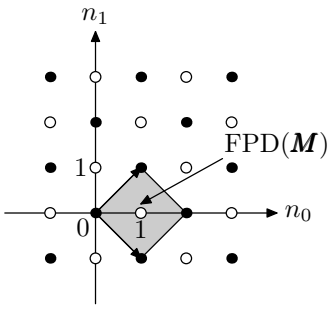


Figure 2.10: The quincunx lattice.

2.3.4 Quincunx Filter Banks

The two-dimensional (2D) **quincunx lattice** is the simplest nonseparable lattice. Figure 2.10 shows the quincunx lattice, where the symbols • and ◦ represent the two distinct cosets associated with coset vectors $\boldsymbol{k}_0 = [0\ \ 0]^T$ and $\boldsymbol{k}_1 = [1\ \ 0]^T$, respectively. There are many matrices that generate the quincunx lattice, such as $\left[\begin{smallmatrix} 2 & 1 \\ 0 & 1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$. Herein, we shall always choose the generating matrix to be $\boldsymbol{M} = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$. In this way, when two downsamplers are cascaded, the equivalent single downsampling matrix becomes a separable diagonal matrix $\boldsymbol{M}^2 = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$.

With the quincunx downsampling matrix $\boldsymbol{M} = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, the downsampling operation, as shown in Figure 2.3, with input sequence x and output sequence y is expressed in time domain, Fourier domain, and z-domain as

$$y[n_0, n_1] = x[n_0 + n_1, n_0 - n_1],$$

$$\hat{y}(\omega_0, \omega_1) = \frac{1}{2}\left[\hat{x}\left(\tfrac{\omega_0+\omega_1}{2}, \tfrac{\omega_0-\omega_1}{2}\right) + \hat{x}\left(\tfrac{\omega_0+\omega_1}{2} - \pi, \tfrac{\omega_0-\omega_1}{2} - \pi\right)\right], \quad \text{and}$$

$$Y(z_0, z_1) = \frac{1}{2}\left[X(z_0^{\frac{1}{2}} z_1^{\frac{1}{2}}, z_0^{\frac{1}{2}} z_1^{-\frac{1}{2}}) + X(-z_0^{\frac{1}{2}} z_1^{\frac{1}{2}}, -z_0^{\frac{1}{2}} z_1^{-\frac{1}{2}})\right],$$

respectively. The upsampling operation shown in Figure 2.4 is expressed in time domain, Fourier domain, and z-domain as

$$y[n_0, n_1] = \begin{cases} x\left[\frac{1}{2}(n_0 + n_1), \frac{1}{2}(n_0 - n_1)\right] & \text{if } [n_0\ \ n_1]^T \in \mathrm{LAT}(\boldsymbol{M}) \\ 0 & \text{otherwise,} \end{cases}$$

$$\hat{y}(\omega_0, \omega_1) = \hat{x}(\omega_0 + \omega_1, \omega_0 - \omega_1), \quad \text{and} \quad Y(z_0, z_1) = X\left(z_0 z_1, z_0 z_1^{-1}\right),$$

respectively.

Figure 2.11 shows a UMD filter bank based on quincunx sampling, where $\boldsymbol{M}$ denotes the quincunx generating matrix $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, $\{\mathrm{H}_k\}$ and $\{\mathrm{G}_k\}$ are the analysis and synthesis filters, respectively. The (shift-free) PR

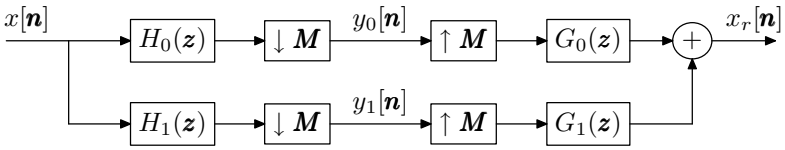
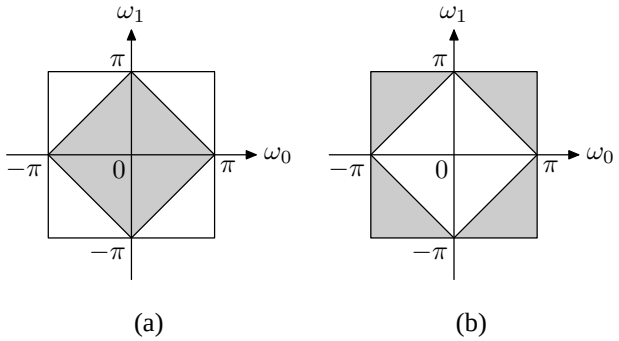


Figure 2.11: A two-channel quincunx UMD filter bank.



(a) (b)

Figure 2.12: Ideal frequency responses of quincunx filter banks for the (a) lowpass filters and (b) highpass filters.

condition for the quincunx UMD filter bank is

$$H_0(\boldsymbol{z})G_0(\boldsymbol{z}) + H_1(\boldsymbol{z})G_1(\boldsymbol{z}) = 2 \quad \text{and} \tag{2.19a}$$

$$H_0(-\boldsymbol{z})G_0(\boldsymbol{z}) + H_1(-\boldsymbol{z})G_1(\boldsymbol{z}) = 0, \tag{2.19b}$$

where $\{H_k(\boldsymbol{z})\}$ and $\{G_k(\boldsymbol{z})\}$ are the analysis and synthesis filter transfer functions, respectively.

The quincunx lowpass and highpass filters are often chosen to have diamond-shaped frequency responses as shown in Figures 2.12(a) and (b), respectively. In these figures, passband and stopband are represented by the shaded and unshaded areas, respectively. With the diamond-shaped frequency response, the lowpass filter can preserve high frequencies in the horizontal and vertical directions, which is a good match to the human visual system as the visual sensitivity is higher to changes in these two directions than in other directions.

In many image processing applications, a quincunx filter bank is typically applied in a recursive manner in the lowpass channel, resulting in an octave-band filter bank structure as shown in Figure 2.13. With the ideal frequency responses shown in Figure 2.12, this structure leads to a frequency decomposition shown in Figure 2.14. For an N-level octave-band filter bank generated from a quincunx filter bank with analysis filters $\{\mathrm{H}_k\}$, by using the noble identities and combining cascaded downsamplers, upsamplers and filters, we obtain the equivalent nonuniform filter bank shown in Figure 2.15. The equivalent filter bank has $N+1$ channels

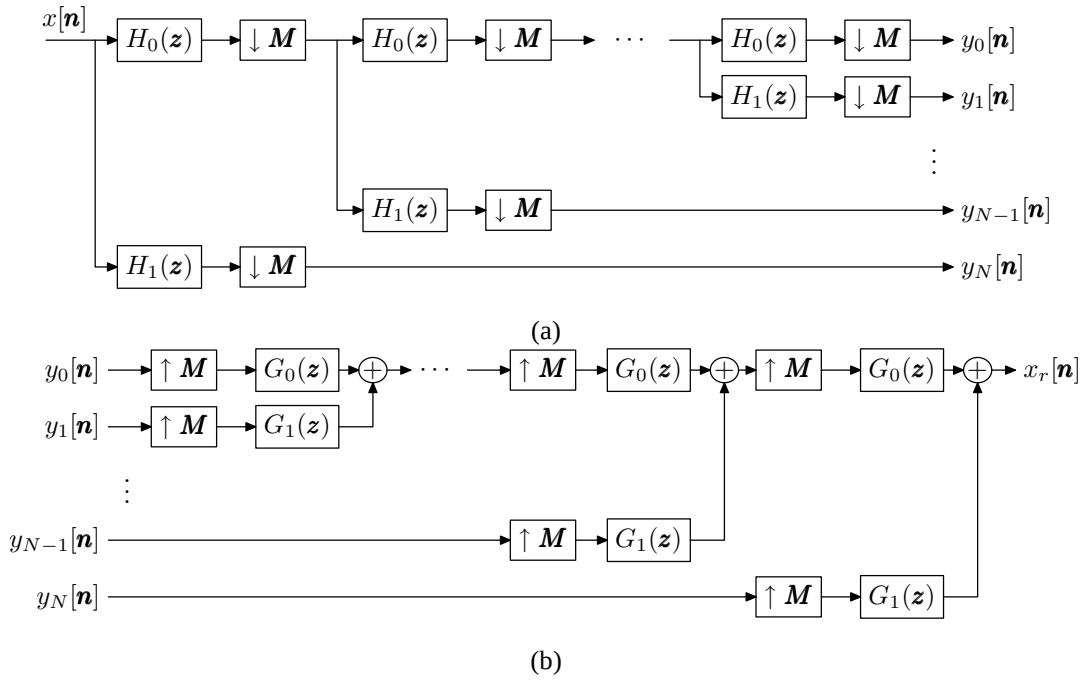


Figure 2.13: The structure of an *N*-level octave-band quincunx filter bank. (a) Analysis side and (b) synthesis side.

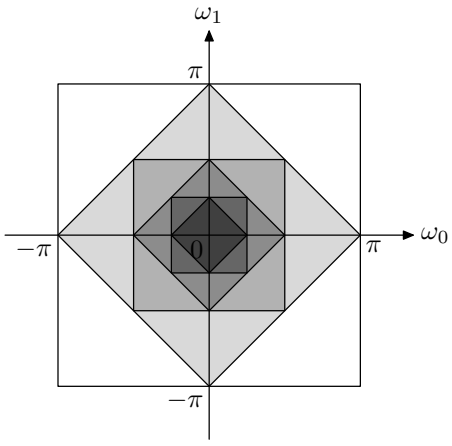


Figure 2.14: Frequency decomposition associated with octave-band quincunx scheme.

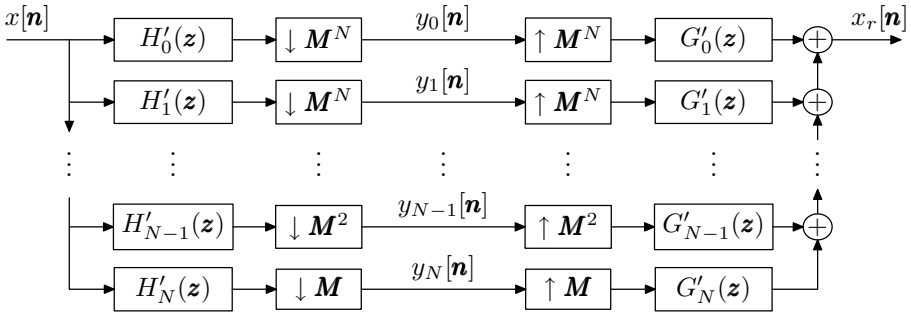


Figure 2.15: The equivalent nonuniform filter bank associated with the N-level octave-band filter bank.

with analysis filters $\{\mathrm{H}'_i\}$ and synthesis filters $\{\mathrm{G}'_i\}$. The impulse responses of these equivalent filters can be computed by iterative upsampling and convolution of the original analysis and synthesis filter impulse responses as

$$h'_i = \begin{cases} h_0 * (\uparrow M) h_0 * (\uparrow M^2) h_0 * \cdots * (\uparrow M^{N-1}) h_0 & \text{for } i = 0 \\ h_0 * (\uparrow M) h_0 * \cdots * (\uparrow M^{N-i-1}) h_0 * (\uparrow M^{N-i}) h_1 & \text{for } 1 \le i \le N-1 \\ h_1 & \text{for } i = N, \quad \text{and} \end{cases} \tag{2.20}$$

$$g'_i = \begin{cases} g_0 * (\uparrow M) g_0 * (\uparrow M^2) g_0 * \cdots * (\uparrow M^{N-1}) g_0 & \text{for } i = 0 \\ g_0 * (\uparrow M) g_0 * \cdots * (\uparrow M^{N-i-1}) g_0 * (\uparrow M^{N-i}) g_1 & \text{for } 1 \le i \le N-1 \\ g_1 & \text{for } i = N. \end{cases} \tag{2.21}$$

The transfer functions $\{H'_i(\boldsymbol{z})\}$ of $\{\mathrm{H}'_i\}$ are given by

$$H'_i(\boldsymbol{z}) = \begin{cases} \prod_{k=0}^{N-1} H_0\left(\boldsymbol{z}^{\boldsymbol{M}^k}\right) & i = 0 \\ H_1\left(\boldsymbol{z}^{\boldsymbol{M}^{N-i}}\right) \prod_{k=0}^{N-i-1} H_0\left(\boldsymbol{z}^{\boldsymbol{M}^k}\right) & 1 \le i \le N-1 \\ H_1(\boldsymbol{z}) & i = N. \end{cases} \tag{2.22}$$

The transfer functions $\{G'_i(\boldsymbol{z})\}$ of the equivalent synthesis filters $\{\mathrm{G}'_i\}$ can be derived in a similar way.

2.3.5 Relation Between Filter Banks and Wavelet Systems

Filter banks and wavelets are closely connected [30]. Filter banks can be viewed as discrete wavelet transforms [31], and continuous-time wavelet bases can be derived using iterated filter banks [32, 10]. Therefore, when an octave-band filter bank is applied to a signal, the shape of the basis functions of the associated

wavelet may appear as artifacts in the reconstructed signal if the transformed coefficients are quantized. In this section, we briefly explain how filter banks are related to wavelet systems in the quincunx case.

We consider the dyadic wavelet systems, where functions are represented at different resolutions where successive resolution differs in scale by a factor of two. A wavelet system is a basis of $L^2\left(\mathbb{R}^2\right)$ derived from a **multiresolution approximation** (MRA) [32]. Consider an MRA associated with scaling function ϕ satisfying the refinement equation

$$\phi(\boldsymbol{x}) = \sqrt{2} \sum_{\boldsymbol{k}\in\mathbb{Z}^2} c[\boldsymbol{k}]\phi(\boldsymbol{M}\boldsymbol{x}-\boldsymbol{k}), \tag{2.23}$$

and wavelet function ψ satisfying the wavelet equation

$$\psi(\boldsymbol{x}) = \sqrt{2} \sum_{\boldsymbol{k}\in\mathbb{Z}^2} d[\boldsymbol{k}]\phi(\boldsymbol{M}\boldsymbol{x}-\boldsymbol{k}), \tag{2.24}$$

where $\boldsymbol{M}$ is the generating matrix of the quincunx lattice. The **dual MRA** is associated with scaling function $\tilde{\phi}$ and wavelet function $\tilde{\psi}$, where $\tilde{\phi}$ and $\tilde{\psi}$ are the dual Riesz bases of ϕ and ψ, and satisfy the scaling and wavelet equations

$$\tilde{\phi}(\boldsymbol{x}) = \sqrt{2} \sum_{\boldsymbol{k}\in\mathbb{Z}^2} \tilde{c}[\boldsymbol{k}]\tilde{\phi}(\boldsymbol{M}\boldsymbol{x}-\boldsymbol{k}) \quad \text{and} \quad \tilde{\psi}(\boldsymbol{x}) = \sqrt{2} \sum_{\boldsymbol{k}\in\mathbb{Z}^2} \tilde{d}[\boldsymbol{k}]\tilde{\phi}(\boldsymbol{M}\boldsymbol{x}-\boldsymbol{k}),$$

respectively.

A quincunx UMD filter bank as the one shown in Figure 2.11 is related to the above MRA as

$$h_0[\boldsymbol{n}] = \tilde{c}^*[-\boldsymbol{n}], \quad h_1[\boldsymbol{n}] = \tilde{d}^*[-\boldsymbol{n}], \quad g_0[\boldsymbol{n}] = c[\boldsymbol{n}], \quad \text{and} \quad g_1[\boldsymbol{n}] = d[\boldsymbol{n}]. \tag{2.25}$$

Therefore, the choice of filters determines the shape of the scaling and wavelet functions. Iteratively upsampling and convolving the lowpass analysis or synthesis filter produces a shape approximating the dual or primal scaling function, respectively. Similarly, the shape of the wavelet function can be approximated with a similar approach starting from the convolution of the lowpass and highpass filters followed by iterative upsampling and convolution with the lowpass filter. Referring to the N-level octave-band quincunx filter bank shown in Figure 2.13 and its equivalent form in Figure 2.15, the shape of the impulse responses $h_0'[\boldsymbol{n}]$ and $g_0'[\boldsymbol{n}]$ of the equivalent filters H_0' and G_0' approximate the shape of the dual and primal scaling functions, respectively, and the shape of $h_i'[\boldsymbol{n}]$ and $g_i'[\boldsymbol{n}]$ approximate that of the wavelet functions more and more accurately as i decreases from N to 1.

The number of *vanishing moments* is of interest herein. It corresponds to the highest order of polynomials that can be reproduced by the scaling function. From the filter bank point of view, it represents the highpass filter's ability to annihilate polynomials. If there are a certain number of vanishing moments, and the original

signal can be well approximated by polynomials, then the highpass and bandpass subbands contain few nonzero coefficients, which is favorable in many signal processing applications. The number of vanishing moments is equivalent to the order of zero at $[0\ \ 0]^T$ in the highpass filter frequency response, or the order of zero at $[\pi\ \ \pi]^T$ in the lowpass filter frequency response. Similar to the sum rule in the 1D case, we have the following lemma for the quincunx case.

Lemma 2.1 (Sum rule)**.** *Let c and d be sequences defined on $\mathbb{Z}^2$ with Fourier transforms $\hat{c}(\boldsymbol{\omega})$ and $\hat{d}(\boldsymbol{\omega})$, respectively. Then, $\hat{c}(\boldsymbol{\omega})$ has an Nth order zero at $\boldsymbol{\omega} = [\pi\ \ \pi]^T$ if and only if*

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} (-1)^{|\boldsymbol{n}|} \boldsymbol{n}^{\boldsymbol{m}} c[\boldsymbol{n}] = 0, \quad \textit{for } |\boldsymbol{m}| < N, \tag{2.26}$$

and $\hat{d}(\boldsymbol{\omega})$ has an Nth order zero at $\boldsymbol{\omega} = [0\ \ 0]^T$ if and only if

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} \boldsymbol{n}^{\boldsymbol{m}} d[\boldsymbol{n}] = 0, \quad \textit{for } |\boldsymbol{m}| < N. \tag{2.27}$$

Therefore, for a UMD quincunx filter bank to have N vanishing moments, the impulse response of the corresponding lowpass or highpass filter is required to satisfy the linear system (2.26) or (2.27), respectively. The presence of vanishing moments is desirable in many applications.

2.3.6 Lifting Realization of Quincunx Filter Banks

The *lifting scheme* [26, 27] is an efficient method used to design and implement filter banks. The lifting structure provides a number of advantages over the traditional filter bank realization. It features fast and in-place computation, satisfies the (shift-free) PR condition automatically, and can be used to construct reversible integer-to-integer (ITI) transforms [33]. Unlike the 1D case, only a subset of all PR quincunx filter banks can be implemented using the lifting scheme.

The lifting realization of a quincunx filter bank with 2λ lifting filters is shown in Figure 2.16. Without loss of generality, we assume that none of the 2λ lifting filter transfer functions $\{A_k(\boldsymbol{z})\}$ are identically zero, except possibly $A_1(\boldsymbol{z})$ and $A_{2\lambda}(\boldsymbol{z})$. With the lifting structure for the forward transform shown in Figure 2.16(a), the input sequence x is decomposed into its two polyphase components, and then each lifting step adds a filtered version of the sequence in one channel to the sequence in the other channel. The inverse transform has a similar structure which undoes each step of the forward transform as shown in Figure 2.16(b). In this way, the PR condition is satisfied structurally.

The analysis polyphase matrix can be derived from the lifting filters as

$$\boldsymbol{H}_p(\boldsymbol{z}) = \begin{bmatrix} H_{0,0}(\boldsymbol{z}) & H_{0,1}(\boldsymbol{z}) \\ H_{1,0}(\boldsymbol{z}) & H_{1,1}(\boldsymbol{z}) \end{bmatrix} = \prod_{k=1}^{\lambda} \left(\begin{bmatrix} 1 & A_{2k}(\boldsymbol{z}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{2k-1}(\boldsymbol{z}) & 1 \end{bmatrix} \right), \tag{2.28}$$

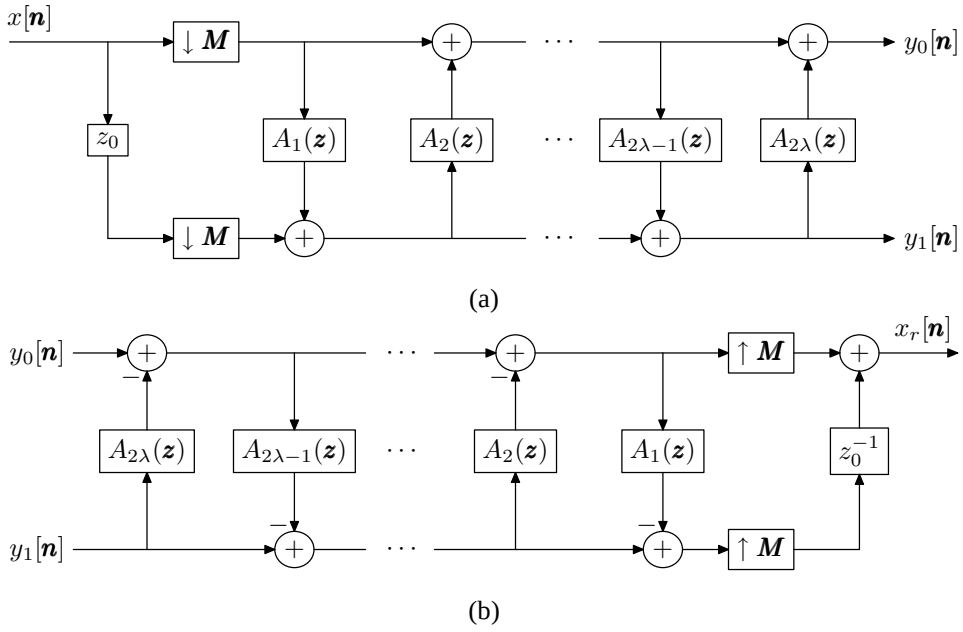


Figure 2.16: Lifting realization of a quincunx filter bank. (a) Analysis side and (b) synthesis side.

and the corresponding analysis filter transfer functions are calculated using (2.15) as

$$H_0(\boldsymbol{z}) = H_{0,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) + z_0 H_{0,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) \quad \text{and} \quad H_1(\boldsymbol{z}) = H_{1,0}\left(\boldsymbol{z}^{\boldsymbol{M}}\right) + z_0 H_{1,1}\left(\boldsymbol{z}^{\boldsymbol{M}}\right). \tag{2.29}$$

The synthesis filter transfer functions $G_0(\boldsymbol{z})$ and $G_1(\boldsymbol{z})$ can be trivially computed as $G_k(\boldsymbol{z}) = (-1)^{1-k} z_0^{-1} H_{1-k}(-\boldsymbol{z})$ for $k = 0, 1$.

The lifting structure can be used to construct reversible integer-to-integer transforms (i.e., PR filter banks which map integers to integers). For each lifting step on the analysis side, a rounding operator R_i is added to the output of the lifting filter A_i such that the sequences after each lifting step, including the subbands, contain only integers. On the synthesis side, the same rounding operator is added in the corresponding lifting step. With this method, the transform retains invertibility and maps integers to integers. The lifting realization of an integer-to-integer transform is shown in Figure 2.17.

2.4 Image Coding

In this thesis, we are sometimes interested in image coding applications of quincunx filter banks. Below, we briefly introduce the subband image compression system and some measures used to evaluate the coding performance.

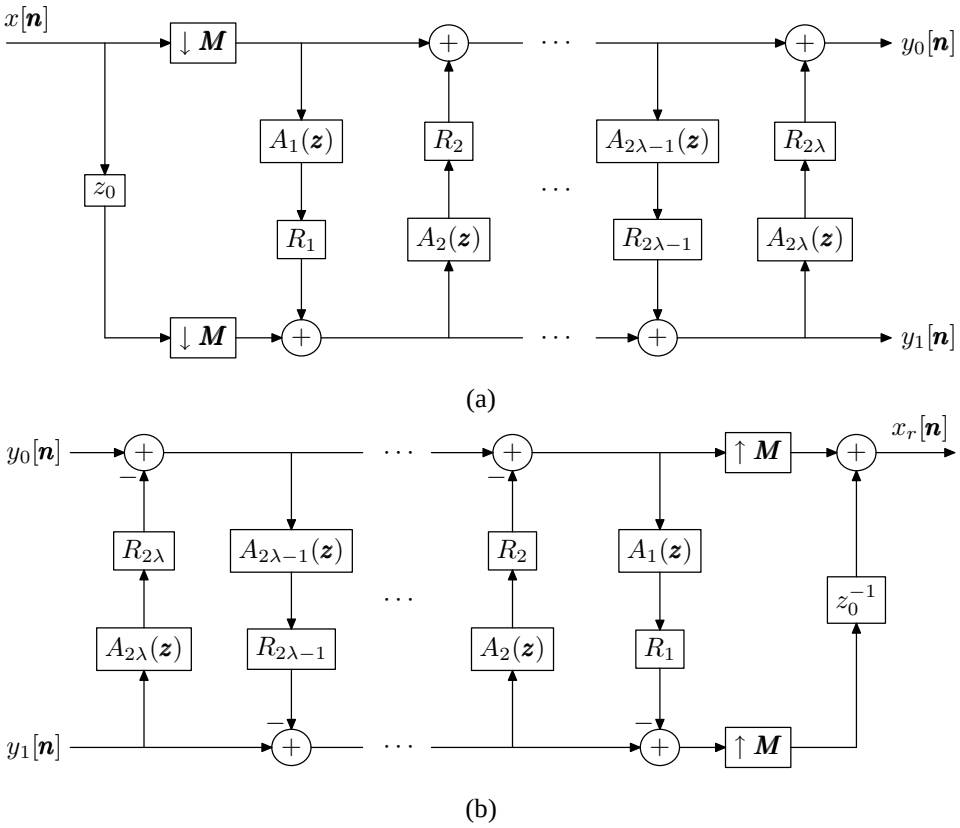


Figure 2.17: The lifting realization of a reversible integer-to-integer transform. (a) Analysis side and (b) synthesis side.

2.4.1 Subband Image Compression Systems

Figure 2.18 shows the structure of a subband image compression system. In this thesis, the subband transforms are computed by an N-level octave-band quincunx filter bank. On the encoder side as shown in Figure 2.18(a), the forward subband transform is applied to the original image to reduce the data redundancy by decomposing the image into a set of coefficients corresponding to subbands at multiple resolution levels and frequency segments. The filter coefficients are chosen such that there are considerably more small coefficients in the transformed data than in the original one, which leads to more efficient compression. Next, the transform coefficients are quantized and encoded to produce a bitstream of the coded image. On the decoder side shown in Figure 2.18(b), the bitstream is first decoded and dequantized. Then the inverse transform is applied to reconstruct the image. If the original image is exactly reconstructed from the coded data, the compression is said to be **lossless**. If the reconstructed image is only an approximation of the original one, the compression is said to be **lossy**. In the lossy case, the difference between the original and reconstructed images is referred to as distortion.

Next we introduce some measures used to evaluate the performance of the compression system. The **compression ratio** (CR) is usually used for lossless compression, which is defined as the ratio between the original and compressed image sizes in number of bits. In the lossy case, the **mean-squared error** (MSE) and **peak-signal-to-noise ratio** (PSNR) are commonly used to measure distortion. For the original image x and reconstructed image x_r of size $N_0 \times N_1$, MSE and PSNR are defined as

$$\mathrm{MSE} = \frac{1}{N_0 N_1} \sum_{n_0=0}^{N_0-1} \sum_{n_1=0}^{N_1-1} \big(x_r[n_0,n_1] - x[n_0,n_1]\big)^2 \quad \text{and} \tag{2.30}$$

$$\mathrm{PSNR} = 20 \log_{10} \left(\frac{2^P - 1}{\sqrt{\mathrm{MSE}}} \right), \tag{2.31}$$

respectively, where P is the number of bits used per sample in x. Higher PSNR often corresponds to better reconstructed images, but sometimes PSNR cannot exactly reflect the visual quality of reconstructed images. In this case, subjective image quality tests can also be performed by human observers.

2.4.2 Coding Gain

Coding gain [34, 35] is an analytical performance measure to evaluate the coding performance of filter banks. It is used to estimate the energy compaction ability of filter banks by computing the ratio between the reconstruction error variance obtained by quantizing a signal directly to that obtained by quantizing the corresponding subband coefficients using an optimal bit allocation strategy.

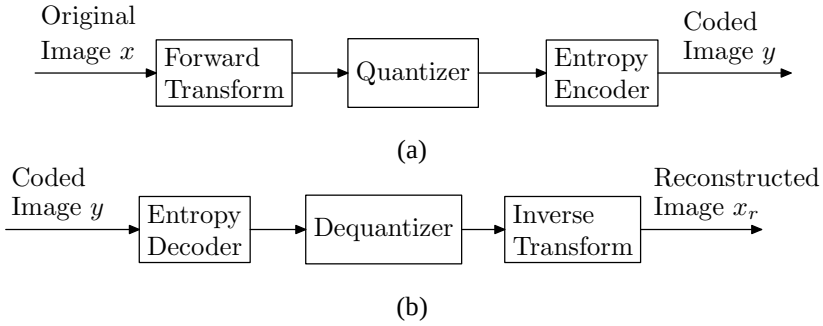


Figure 2.18: Block diagram of an image coder. (a) Encoder side and (b) decoder side.

For an N-level octave-band quincunx filter bank as the one shown in Figure 2.13, its equivalent nonuniform filter bank with $N+1$ channels is shown in Figure 2.15. The coding gain G_{SBC} of this N-level octave-band filter bank can be computed as [34]

$$G_{SBC} = \prod_{k=0}^{N} (A_k B_k / \alpha_k)^{-\alpha_k}, \tag{2.32}$$

where

$$A_k = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h'_k[\boldsymbol{m}] h'_k[\boldsymbol{n}] r[\boldsymbol{m}-\boldsymbol{n}],$$

$$B_k = \alpha_k \sum_{\boldsymbol{n}\in\mathbb{Z}^2} g'^2_k[\boldsymbol{n}],$$

$$\alpha_k = \begin{cases} 2^{-N} & \text{for } k = 0 \\ 2^{-(N+1-k)} & \text{for } k = 1, 2, \ldots, N, \end{cases}$$

$h'_k[\boldsymbol{n}]$ and $g'_k[\boldsymbol{n}]$ are the impulse responses of the equivalent analysis and synthesis filters H'_k and G'_k in Figure 2.15, and r is the normalized autocorrelation of the input. Depending on the source image model, r is given by

$$r[n_0, n_1] = \begin{cases} \rho^{|n_0|+|n_1|} & \text{for separable model} \\ \rho^{\sqrt{n_0^2+n_1^2}} & \text{for isotropic model,} \end{cases} \tag{2.33}$$

where ρ is the correlation coefficient (typically, $0.90 \leq \rho \leq 0.95$).

Filter banks with high coding gain can efficiently compact energy, which generally leads to good performance in subband coding systems. Therefore, high coding gain is a desirable property in filter bank design.

Chapter 3

Symmetric Extension for Quincunx Filter Banks

3.1 Overview

Symmetric extension is a commonly used technique for constructing nonexpansive transforms for 1D sequences of finite length. In this chapter, we show how to extend this technique to the case of 2D nonseparable quincunx filter banks. In particular, we show how one can construct nonexpansive transforms for input sequences defined on arbitrary rectangular regions. Some of the material in this chapter has also been presented in [36, 37].

3.2 Introduction

Filter banks have proven to be a highly effective tool for in many signal processing applications. They are often defined so as to operate on sequences of infinite extent. In practice, however, we almost invariably deal with sequences of finite extent. Therefore, we usually require some means for adapting filter banks to such sequences. This leads to the well known boundary filtering problem that can arise whenever a finite-extent sequence is filtered. Furthermore, in many signal processing applications such as image coding, the objective is to reduce the redundancy of the original sequence and represent it with as few bits as possible. Therefore, it is desirable to employ a transform that is nonexpansive (i.e., maps a sequence of N samples to a new sequence of no more than N samples). Consequently, we seek a solution to the boundary problem that

yields nonexpansive transforms.

In the case of 1D filter banks, various methods have been proposed to solve the boundary problem. The simplest way is zero padding, where the region beyond the boundaries of the finite-extent sequence are padded with zeros. In this way, the number of samples increases due to the effect of linear convolution, resulting in an expansive transform. Although truncation can be used to obtain nonexpansive transform, it causes distortion in the reconstructed signal near the boundaries. Periodic extension is another solution to the boundary problem. This method concatenates the original finite-extent sequence periodically, usually generating sharp transitions at the splice points between periods. Unfortunately, this method has the disadvantage that the discontinuities in the extended sequence introduce undesirable high frequencies, which is detrimental in many applications.

In the 1D case, symmetric extension [14, 16] is a commonly used technique for constructing nonexpansive transforms of finite-extent sequences. This scheme uses a structure similar to the one shown in Figure 3.1, where the filter bank should be viewed as an 1D filter bank with $\boldsymbol{M} = 2$. The input sequence is first mirrored across its boundary, and then this symmetric pattern is repeated periodically. Therefore, the continuity is maintained at the splice points between periods, as illustrated by the example shown in Figure 3.2. The key point to build a nonexpansive transform in this approach is that the subband sequences should also have certain symmetry and periodicity properties, such that only a small finite number of samples are independent in each subband. This requires the analysis filters to have linear phase with group delays satisfying certain conditions.

In this chapter, we explain how the symmetric extension technique can be extended to the case of quincunx filter banks. In particular, we show how one can construct nonexpansive transforms for input sequences defined on arbitrary rectangular regions. We use a structure for the forward transform like that shown in Figure 3.1(a). The input 2D sequence $\tilde{x}$ is first extended to an infinite-extent periodic symmetric sequence x. The periodicity and symmetry properties may propagate across the nonseparable downsampler by carefully constraining the choice of the analysis filters H_0 and H_1. In this way, the independent samples of the subbands y_0 and y_1 are each located in a finite region, and then we can extract these samples from y_0 and y_1. The structure for the inverse transform is shown in Figure 3.1(b).

The remaining part of this chapter is organized as follows. Section 3.3 defines several types of MD symmetries. Section 3.4 introduces a scheme that maps a 2D finite-extent sequence into an infinite-extent sequence. Section 3.5 discusses how symmetry and periodicity can be preserved under the operations of a quincunx filter bank. These results are then used in Section 3.6 to produce our new symmetric extension algorithms. Finally, Section 3.7 summarizes the proposed symmetric extension algorithms.

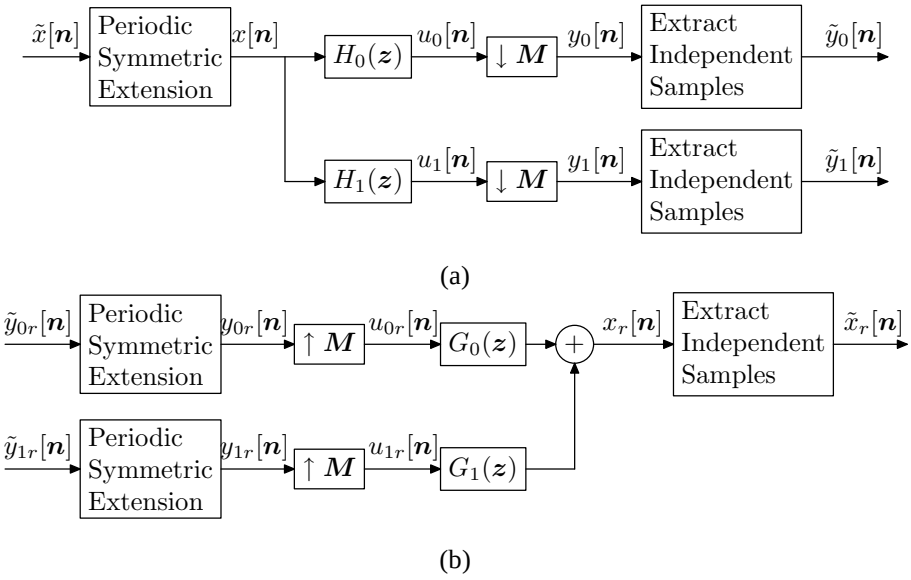


Figure 3.1: Filter bank with symmetric extension. (a) Analysis side and (b) synthesis side.

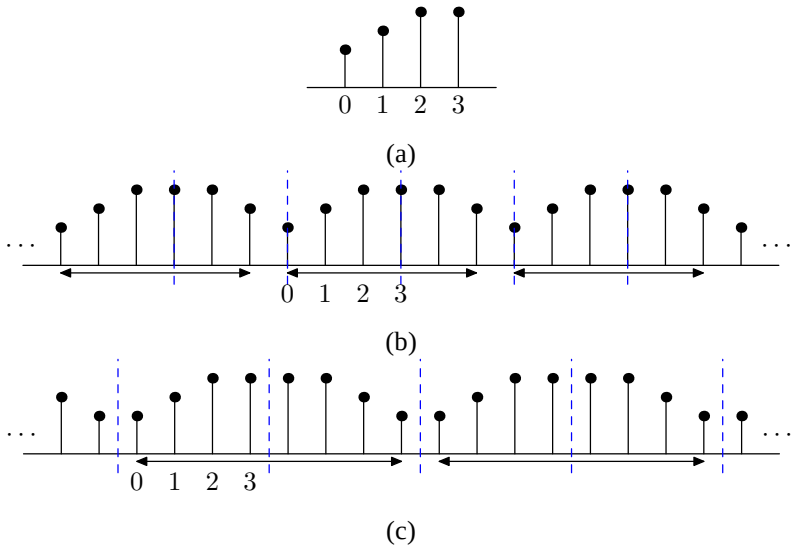


Figure 3.2: 1D symmetric extension. (a) Original sequence, (b) whole-sample symmetrically extended sequence, and (c) half-sample symmetrically extended sequence.

3.3 Types of Symmetries

The notion of symmetry is of fundamental importance herein. In the 1D case, only a very limited number of symmetry types is possible. In the MD case, however, there are considerably more possibilities. Below, we define several types of MD symmetry relevant to this work. Recall that we have defined linear phase in (2.3) for MD filters. For MD sequences, it is also called centrosymmetry as shown below.

Definition 3.1 (Centrosymmetry)**.** A sequence x defined on $\mathbb{Z}^D$ is said to be **centrosymmetric** about $\boldsymbol{c}$ (i.e., has **linear phase** with **group delay** $\boldsymbol{c}$) if, for some $\boldsymbol{c} \in \frac{1}{2}\mathbb{Z}^D$ and $S \in \{-1,1\}$,

$$x[\boldsymbol{n}] = Sx[2\boldsymbol{c}-\boldsymbol{n}] \quad \text{for all } \boldsymbol{n} \in \mathbb{Z}^D. \tag{3.1}$$

The sequence x is referred to as symmetric if $S=1$, and antisymmetric if $S=-1$.

Centrosymmetry is a kind of two-fold symmetry, where about half of the samples are independent. In the 1D case, a centrosymmetric sequence x is said to have whole-sample symmetry/antisymmetry if its symmetry center $c \in \mathbb{Z}$, and half-sample symmetry/antisymmetry if $c \in \frac{1}{2}\mathbb{Z}_{odd}$.

In the MD case, there exist some types of higher-order symmetry. We first introduce the hyper-octantal centrosymmetry.

Definition 3.2 (Hyper-octantal centrosymmetry)**.** A sequence x defined on $\mathbb{Z}^D$ is said to be **hyper-octantally centrosymmetric** [38] about $\boldsymbol{c}$ if, for some $\boldsymbol{c} \in \frac{1}{2}\mathbb{Z}^D$ and $A \in \{1,2,\ldots,2^D-1\}$,

$$x[\boldsymbol{n}] = s[A]x\big[\boldsymbol{c} \circ (\boldsymbol{1}-\boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A]\big] \quad \text{for all } \boldsymbol{n} \in \mathbb{Z}^D, \tag{3.2}$$

where $s[A] \in \{-1,1\}$, $\boldsymbol{v}[A] = \big[\,(-1)^{a_0}\ (-1)^{a_1}\ \cdots\ (-1)^{a_D}\,\big]^T$, $a_i \in \{0,1\}$, and $A = \sum_{i=0}^{D-1} a_i 2^i$.

In order for satisfy the centrosymmetry condition (3.1), the function $s[\cdot]$ must be chosen to satisfy

$$s\left[2^D-1-A\right] = Ss[A], \tag{3.3}$$

for all $A = 0,1,\ldots,2^D-1$ and $S \in \{-1,1\}$. Note that by definition $s[0]=1$.

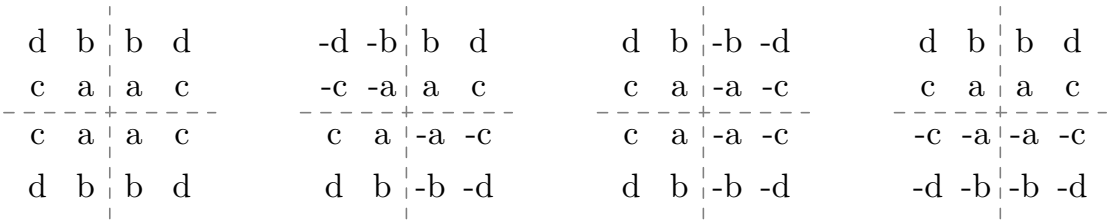
In the 2D case, the hyper-octantal centrosymmetry is called **quadrantal centrosymmetry**, and (3.2) can be equivalently expressed as

$$x[n_0,n_1] = STx[2c_0-n_0,2c_1-n_1] = Sx[2c_0-n_0,n_1] = Tx[n_0,2c_1-n_1], \tag{3.4}$$

where $S,T \in \{-1,1\}$. In terms of S and T, four types of quadrantal centrosymmetry are possible [38] as listed in Table 3.1. Examples of the four types of 2D quadrantally centrosymmetric sequences are shown in Figure 3.3. Clearly, quadrantal centrosymmetry is a type of four-fold symmetry, where only (approximately) $\frac{1}{4}$ of the samples are independent (e.g., those with indices $\boldsymbol{n} \geq \boldsymbol{c}$).

Table 3.1: Four types of quadrantal centrosymmetry

Type	even-even	odd-odd	even-odd	odd-even
S	1	-1	1	-1
T	1	-1	-1	1



(a) (b) (c) (d)

Figure 3.3: Four types of quadrantal centrosymmetry: (a) even-even, (b) odd-odd, (c) odd-even, and (d) even-odd.

For a filter with the last three types of quadrantal centrosymmetry in Table 3.1, its frequency response is zero along one or both of the ω_0- and ω_1-axes. Therefore, such filters cannot be used as lowpass filters in horizontal and/or vertical directions. This statement can be shown as follows. Let H be a quadrantally centrosymmetric filter with impulse response h. Its frequency response can be expressed as

$$\hat{h}(\boldsymbol{\omega}) = \sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} h[n_0,n_1]e^{-j(\omega_0 n_0+\omega_1 n_1)}, \tag{3.5a}$$

$$\hat{h}(\boldsymbol{\omega}) = \sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} STh[2c_0-n_0,2c_1-n_1]e^{-j(\omega_0 n_0+\omega_1 n_1)}, \tag{3.5b}$$

$$\hat{h}(\boldsymbol{\omega}) = \sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} Sh[2c_0-n_0,n_1]e^{-j(\omega_0 n_0+\omega_1 n_1)}, \tag{3.5c}$$

and

$$\hat{h}(\boldsymbol{\omega}) = \sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} Th[n_0,2c_1-n_1]e^{-j(\omega_0 n_0+\omega_1 n_1)}. \tag{3.5d}$$

Equation (3.5) can be rewritten as

$$\hat{h}(\boldsymbol{\omega}) = e^{-j(\omega_0 c_0+\omega_1 c_1)} \sum_{n_0\in\mathbb{Z}} \sum_{n_1\in\mathbb{Z}} h[n_0,n_1] e^{j[\omega_0(c_0-n_0)+\omega_1(c_1-n_1)]}, \tag{3.6a}$$

$$\hat{h}(\boldsymbol{\omega}) = e^{-j(\omega_0 c_0+\omega_1 c_1)} \sum_{n_0\in\mathbb{Z}} \sum_{n_1\in\mathbb{Z}} STh[n_0,n_1] e^{-j[\omega_0(c_0-n_0)+\omega_1(c_1-n_1)]}, \tag{3.6b}$$

$$\hat{h}(\boldsymbol{\omega}) = e^{-j(\omega_0 c_0+\omega_1 c_1)} \sum_{n_0\in\mathbb{Z}} \sum_{n_1\in\mathbb{Z}} Sh[n_0,n_1] e^{-j[\omega_0(c_0-n_0)-\omega_1(c_1-n_1)]}, \tag{3.6c}$$

and

$$\hat{h}(\boldsymbol{\omega}) = e^{-j(\omega_0 c_0+\omega_1 c_1)} \sum_{n_0\in\mathbb{Z}} \sum_{n_1\in\mathbb{Z}} Th[n_0,n_1] e^{j[\omega_0(c_0-n_0)-\omega_1(c_1-n_1)]}. \tag{3.6d}$$

Averaging the expressions for $\hat{h}(\boldsymbol{\omega})$ in (3.6), we have

$$\begin{aligned}\hat{h}(\boldsymbol{\omega}) &= \frac{1}{4} e^{-j\boldsymbol{\omega}^T\boldsymbol{c}} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}] \left[e^{j\omega_0(c_0-n_0)} + Se^{-j\omega_0(c_0-n_0)}\right] \left[e^{j\omega_1(c_1-n_1)} + Te^{-j\omega_1(c_1-n_1)}\right] \\ &= \begin{cases} e^{-j\boldsymbol{\omega}^T\boldsymbol{c}} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}] \cos[\omega_0(c_0-n_0)] \cos[\omega_1(c_1-n_1)] & \text{for even-even} \\ e^{-j(\boldsymbol{\omega}^T\boldsymbol{c}-\pi)} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}] \sin[\omega_0(c_0-n_0)) \sin[\omega_1(c_1-n_1)] & \text{for odd-odd} \\ e^{-j(\boldsymbol{\omega}^T\boldsymbol{c}-\pi/2)} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}] \cos[\omega_0(c_0-n_0)] \sin[\omega_1(c_1-n_1)] & \text{for even-odd} \\ e^{-j(\boldsymbol{\omega}^T\boldsymbol{c}-\pi/2)} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}] \sin[\omega_0(c_0-n_0)] \cos[\omega_1(c_1-n_1)] & \text{for odd-even.} \end{cases}\end{aligned}$$

Therefore, an odd-odd quadrantally centrosymmetric filter H has its frequency response $\hat{h}(\boldsymbol{\omega}) = 0$ along both the ω_0- and ω_1-axis. Similarly, in the even-odd case, $\hat{h}(\boldsymbol{\omega}) = 0$ along the ω_0-axis (i.e., $\omega_1 = 0$), and in the odd-even case, $\hat{h}(\boldsymbol{\omega}) = 0$ along the ω_1-axis (i.e., $\omega_0 = 0$).

For a quadrantally centrosymmetric sequence x with symmetry center $\boldsymbol{c}$, the modulated sequence x', defined as $x'[\boldsymbol{n}] = (-1)^{|\boldsymbol{n}|} x[\boldsymbol{n}]$, also has quadrantal centrosymmetry with symmetry center $\boldsymbol{c}$, where the symmetry type depends on the location of $\boldsymbol{c}$ and the symmetry of x. This relationship is shown in Table 3.2. Note that the z-transform of x' is $X'(\boldsymbol{z}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} (-1)^{|\boldsymbol{n}|} x[\boldsymbol{n}] \boldsymbol{z}^{\boldsymbol{n}} = X(-\boldsymbol{z})$.

Next we introduce another type of four-fold symmetry, named rotated quadrantal centrosymmetry, for 2D sequences. It is similar to the quadrantal centrosymmetry, but with a rotation of 45 degrees.

Definition 3.3 (Rotated quadrantal centrosymmetry)**.** A sequence x defined on $\mathbb{Z}^2$ is said to be **rotated quadrantally centrosymmetric** about $\boldsymbol{c}$ if, for some $S,T \in \{-1,1\}$ and $\boldsymbol{c} = [c_0 \;\; c_1]^T \in \frac{1}{2}\mathbb{Z}^2$ satisfying

Table 3.2: Symmetry type for x' where $x'[\boldsymbol{n}] = (-1)^{|\boldsymbol{n}|}x[\boldsymbol{n}]$ and $X'(\boldsymbol{z}) = X(-\boldsymbol{z})$

symmetry center	Symmetry type of x			
$\boldsymbol{c} = [c_0 \;\; c_1]^T$	even-even	even-odd	odd-even	odd-odd
$\boldsymbol{c} \in \mathbb{Z}^2$	even-even	even-odd	odd-even	odd-odd
$\boldsymbol{c} \in \frac{1}{2}\mathbb{Z}^2_{odd}$	odd-odd	odd-even	even-odd	even-even
$c_0 \in \frac{1}{2}\mathbb{Z}_{odd}$, $c_1 \in \mathbb{Z}$	odd-even	odd-odd	even-even	even-odd
$c_0 \in \mathbb{Z}$, $c_1 \in \frac{1}{2}\mathbb{Z}_{odd}$	even-odd	even-even	odd-odd	odd-even

(a) (b)

Figure 3.4: Rotated quadrantal centrosymmetry: (a) $\boldsymbol{c} \in \mathbb{Z}^2$, and (b) $\boldsymbol{c} \in \frac{1}{2}\mathbb{Z}^2_{odd}$.

$c_0 + c_1 \in \mathbb{Z}$,

$$\begin{aligned} x[n_0, n_1] &= STx[2c_0 - n_0, 2c_1 - n_1] \\ &= Sx[c_0 + c_1 - n_1, c_0 + c_1 - n_0] \\ &= Tx[c_0 - c_1 + n_1, c_1 - c_0 + n_0] \text{ for all } n_0, n_1 \in \mathbb{Z}. \end{aligned} \tag{3.7}$$

Rotated quadrantal centrosymmetry is also a type of four-fold symmetry, where only approximately $\frac{1}{4}$ of the samples are independent (e.g., those with indices $\boldsymbol{n}$ satisfying $\boldsymbol{Mn} \geq \boldsymbol{Mc}$). In terms of the location of symmetry center, there are only two possibilities for this kind of symmetry (i.e., $\boldsymbol{c} \in \mathbb{Z}^2$ and $\boldsymbol{c} \in \frac{1}{2}\mathbb{Z}^2_{odd}$). Examples of rotated quadrantally centrosymmetric sequences are shown in Figure 3.4.

3.4 Mapping Scheme

We now introduce a scheme for mapping a finite-extent 2D sequence defined on a rectangular region to an infinite-extent sequence that is quadrantally centrosymmetric and periodic. This process is called symmetric extension.

In the 1D case, there are two ways to extend a finite-length sequence such that the extended sequence has whole-sample or half-sample symmetry as shown in Figures 3.2(b) and (c). (In this work, we do not consider antisymmetric extension as this extension scheme yields extended signals with large discontinuities, which is undesirable in most applications.) The symmetric extension of a 2D sequence can be viewed as 1D extension operations applied independently along each dimension of the sequence. Therefore, there are four types of symmetric extension for a 2D sequence as defined below.

Definition 3.4 (Symmetric extension of sequence)**.** Let $\tilde{x}$ be a 2D sequence defined on the rectangular region $\{0,1,\ldots,L_0-1\}\times\{0,1,\ldots,L_1-1\}$. Then, the symmetric extension x of $\tilde{x}$ is defined as

$$x[n_0,n_1]=\begin{cases}\tilde{x}[f_w[n_0,L_0],f_w[n_1,L_1]] & \text{type 1}\\ \tilde{x}[f_h[n_0,L_0],f_w[n_1,L_1]] & \text{type 2}\\ \tilde{x}[f_w[n_0,L_0],f_h[n_1,L_1]] & \text{type 3}\\ \tilde{x}[f_h[n_0,L_0],f_h[n_1,L_1]] & \text{type 4,}\end{cases} \tag{3.8}$$

where the functions f_w and f_h are used to compute the corresponding indices for whole-sample symmetry and half-sample symmetry respectively, and are given by

$$f_w[n,L]=\min\{\text{mod}(n,2L-2),2L-2-\text{mod}(n,2L-2)\},\quad \text{and}$$
$$f_h[n,L]=\min\{\text{mod}(n,2L),2L-1-\text{mod}(n,2L)\}.$$

The 1D horizontal slices of the 2D extended sequence are whole- or half-sample symmetric and $(2L_0-2)$- or $2L_0$-periodic in the horizontal direction depending on whether f_w or f_h is applied to n_0 in (3.8). Similarly, the 1D vertical slices of the 2D extended sequence are also symmetric and periodic in the vertical direction. This leads to the symmetry and periodicity properties of a 2D symmetrically extended sequence as summarized by the below lemma.

Lemma 3.1 (Properties of symmetrically extended sequences)**.** *Let $\tilde{x}$ be a sequence defined on the rectangular region $\{0,1,\ldots,L_0-1\}\times\{0,1,\ldots,L_1-1\}$, and x be the symmetric extension of $\tilde{x}$ as defined by (3.8). Let $\boldsymbol{M}$ denote the quincunx generating matrix $\left[\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right]$. Then, x is even-even quadrantally centrosymmetric about*

$\boldsymbol{c}_x$ and $\boldsymbol{P}$-periodic with $\boldsymbol{M}^{-1}\boldsymbol{P}$ being an integer matrix. For the four types of symmetric extension, $\boldsymbol{c}_x$ and $\boldsymbol{P}$ are as given in Table 3.3.

Table 3.3: Properties of the extended sequences

Type	1	2	3	4
$\boldsymbol{c}_x$	$\begin{bmatrix}0 & 0\end{bmatrix}^T$	$\begin{bmatrix}-\frac{1}{2} & 0\end{bmatrix}^T$	$\begin{bmatrix}0 & -\frac{1}{2}\end{bmatrix}^T$	$\begin{bmatrix}-\frac{1}{2} & -\frac{1}{2}\end{bmatrix}^T$
$\boldsymbol{P}$	$\begin{bmatrix}2L_0-2 & 0\\ 0 & 2L_1-2\end{bmatrix}$	$\begin{bmatrix}2L_0 & 0\\ 0 & 2L_1-2\end{bmatrix}$	$\begin{bmatrix}2L_0-2 & 0\\ 0 & 2L_1\end{bmatrix}$	$\begin{bmatrix}2L_0 & 0\\ 0 & 2L_1\end{bmatrix}$

Proof. We only prove the properties of the type-2 symmetric extension. Proofs for the other types can be derived similarly. First, we show that x is $\boldsymbol{P}$-periodic with $\boldsymbol{P} = \boldsymbol{M}\begin{bmatrix}L_0 & L_1-1\\ L_0 & -L_1+1\end{bmatrix}$. Since $\mathrm{mod}(u+kv,v) = \mathrm{mod}(u,v)$ for $k \in \mathbb{Z}$, we have $f_h[n_0+2L_0k_0, L_0] = f_h[n_0, L_0]$ and $f_w[n_1+(2L_1-2)k_1, L_1] = f_w[n_1, L_1]$, for k_0, $k_1 \in \mathbb{Z}$. This implies that $x[\boldsymbol{n}+\boldsymbol{P}\boldsymbol{k}] = x[\boldsymbol{n}]$ for $\boldsymbol{k} = [k_0 \quad k_1]^T \in \mathbb{Z}^2$ with $\boldsymbol{P} = \begin{bmatrix}2L_0 & 0\\ 0 & 2L_1-2\end{bmatrix}$. Therefore, x is $\boldsymbol{P}$-periodic, and $\boldsymbol{M}^{-1}\boldsymbol{P} = \begin{bmatrix}L_0 & L_1-1\\ L_0 & -L_1+1\end{bmatrix}$ is an integer matrix.

Now, we show that x is quadrantally centrosymmetric about $\begin{bmatrix}-\frac{1}{2} & 0\end{bmatrix}^T$. For $u, v \in \mathbb{Z}$, if $v \nmid u$, $\mathrm{mod}(-u,v) = v - \mathrm{mod}(u,v)$; otherwise, $\mathrm{mod}(-u,v) = 0$. It follows that $f_h[-n_0, L_0] = f_h[n_0-1, L_0]$ and $f_w[-n_1, L_1] = f_w[n_1, L_1]$. Therefore, we have

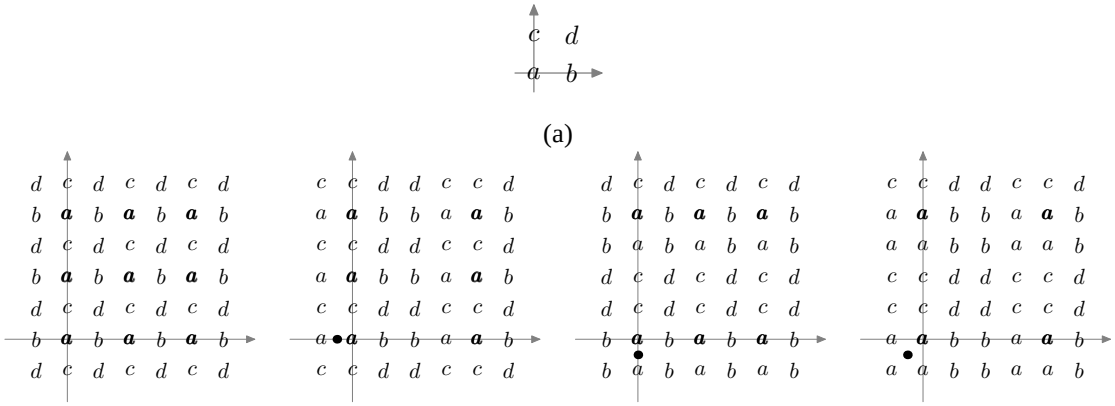
$$x[-1-n_0, -n_1] = \tilde{x}[f_h[-1-n_0, L_0], f_w[-n_1, L_1]] = \tilde{x}[f_h[n_0, L_0], f_w[n_1, L_1]] = x[n_0, n_1].$$

Similarly, we have that $x[-1-n_0, n_1] = x[n_0, n_1]$ and $x[n_0, -n_1] = x[n_0, n_1]$. Thus, from (3.4), x is quadrantally centrosymmetric about $\boldsymbol{c}_x = \begin{bmatrix}-\frac{1}{2} & 0\end{bmatrix}^T$. Since $S = T = 1$ in (3.4), x has the even-even symmetry. (Due to $\boldsymbol{P}$-periodicity, x is also quadrantally centrosymmetric about $\boldsymbol{P}\boldsymbol{k} + \boldsymbol{c}_x$ for $\boldsymbol{k} \in \frac{1}{2}\mathbb{Z}^2$.) □

Example 3.1 (Symmetric extension of a 2D sequence)**.** Consider an input sequence with four samples a, b, c, and d defined on a 2×2 rectangular region as shown in Figure 3.5(a). The four types of symmetrically extended sequences are shown in Figures 3.5(b), (c), (d) and (e). It can be seen from Figure 3.5(b) that the type-1 symmetrically extended sequence has $\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$-periodicity and even-even quadrantal centrosymmetry with symmetry center $\begin{bmatrix}0 & 0\end{bmatrix}^T$. For the other three types, the extended sequences also have quadrantal centrosymmetry and periodicity corresponding to Table 3.3. □

3.5 Preservation of Symmetry and Periodicity

Recall that with the symmetric extension scheme, the structure of the analysis filter bank is shown in Figure 3.1(a). The symmetrically extended sequence x is fed into the analysis side of the filter bank, which



(a) (b) (c) (d) (e)

Figure 3.5: 2D symmetric extension. (a) Input sequence $\tilde{x}$, and the (b) type-1, (c) type-2, (d) type-3, and (e) type-4 symmetrically extended sequences.

consists of filters followed by downsamplers. In order for the subband transform to be nonexpansive, the preservation of four-fold symmetry and periodicity under convolution and downsampling is very important. In what follows, we discuss the effects of these operations on symmetry and periodicity. We begin by considering the effect of convolution on symmetry.

Lemma 3.2 (Preservation of symmetry under convolution)**.** *Let x and h be sequences defined on $\mathbb{Z}^2$, and define $y = x * h$. Then, the following statements hold:*

1. *If x and h are centrosymmetric about $\boldsymbol{c}_x$ and $\boldsymbol{c}_h$, respectively, then y is centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{c}_x + \boldsymbol{c}_h$.*
2. *If x and h are quadrantally centrosymmetric about $\boldsymbol{c}_x$ and $\boldsymbol{c}_h$, respectively, then y is quadrantally centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{c}_x + \boldsymbol{c}_h$.*
3. *If x and h are rotated quadrantally centrosymmetric about $\boldsymbol{c}_x$ and $\boldsymbol{c}_h$, respectively, then y is rotated quadrantally centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{c}_x + \boldsymbol{c}_h$.*

Proof. Since $y = x * h$, we have

$$y[\boldsymbol{n}] = \sum_{\boldsymbol{k} \in \mathbb{Z}^2} x[\boldsymbol{k}] h[\boldsymbol{n} - \boldsymbol{k}]. \tag{3.9}$$

Proof of 1. For the centrosymmetric case, x and h satisfy

$$x[\boldsymbol{n}] = S_x x[2\boldsymbol{c}_x - \boldsymbol{n}] \quad \text{and} \quad h[\boldsymbol{n}] = S_h h[2\boldsymbol{c}_h - \boldsymbol{n}]. \tag{3.10}$$

Substituting (3.10) into (3.9), we have

$$y[\boldsymbol{n}] = S_x S_h \sum_{k\in\mathbb{Z}^2} x[2\boldsymbol{c}_x - \boldsymbol{k}]h[2\boldsymbol{c}_h - \boldsymbol{n} + \boldsymbol{k}]. \tag{3.11}$$

Let $\boldsymbol{k}' = 2\boldsymbol{c}_x - \boldsymbol{k}$, then (3.11) becomes

$$\begin{aligned} y[\boldsymbol{n}] &= S_x S_h \sum_{\boldsymbol{k}'\in\mathbb{Z}^2} x[k']h[2\boldsymbol{c}_h - \boldsymbol{n} + \left(2\boldsymbol{c}_x - \boldsymbol{k}'\right)] \\ &= S_x S_h \sum_{\boldsymbol{k}\in\mathbb{Z}^2} x[\boldsymbol{k}]h[(2\boldsymbol{c}_x + 2\boldsymbol{c}_h - \boldsymbol{n}) - \boldsymbol{k}] \\ &= S_x S_h y[2\left(\boldsymbol{c}_x + \boldsymbol{c}_h\right) - \boldsymbol{n}]. \end{aligned}$$

Thus, y is centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{c}_x + \boldsymbol{c}_h$.

Proof of 2. We prove the quadrantally centrosymmetric case with the Schur product form in (3.2). We have

$$x[\boldsymbol{n}] = s_x[A]x[\boldsymbol{c}_x \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A]] \quad \text{and} \tag{3.12}$$

$$h[\boldsymbol{n}] = s_h[A]h[\boldsymbol{c}_h \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A]], \tag{3.13}$$

for all $\boldsymbol{n} \in \mathbb{Z}^2$. Substituting (3.12) and (3.13) into (3.9), we obtain

$$y[\boldsymbol{n}] = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} s_x[A]x[\boldsymbol{c}_x \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{k} \circ \boldsymbol{v}[A]]s_h[A]h[\boldsymbol{c}_h \circ (1 - \boldsymbol{v}[A]) + (\boldsymbol{n} - \boldsymbol{k}) \circ \boldsymbol{v}[A]]. \tag{3.14}$$

Let $\boldsymbol{k}' = \boldsymbol{c}_x \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{k} \circ \boldsymbol{v}[A]$. Solving for $\boldsymbol{k}$ in terms of $\boldsymbol{k}'$ yields $\boldsymbol{k} = \boldsymbol{c}_x \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{k}' \circ \boldsymbol{v}[A]$. Applying the change of variable to (3.14), for $A = 1, 2, 3$, we obtain

$$\begin{aligned} y[\boldsymbol{n}] &= \sum_{\boldsymbol{k}'\in\mathbb{Z}^2} s_x[A]x[\boldsymbol{k}']s_h[A]h[\boldsymbol{c}_h \circ (1 - \boldsymbol{v}[A]) + (\boldsymbol{n} - \{\boldsymbol{c}_x \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{k}' \circ \boldsymbol{v}[A]\}) \circ \boldsymbol{v}[A]] \\ &= \sum_{\boldsymbol{k}'\in\mathbb{Z}^2} s_x[A]x[\boldsymbol{k}']s_h[A]h[\boldsymbol{c}_h \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A] - \boldsymbol{k}' - \boldsymbol{c}_x \circ (1 - \boldsymbol{v}[A]) \circ \boldsymbol{v}[A]] \\ &= s_x[A]s_h[A] \sum_{\boldsymbol{k}\in\mathbb{Z}^2} x[\boldsymbol{k}]h[(\boldsymbol{c}_h + \boldsymbol{c}_x) \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A] - \boldsymbol{k}] \\ &= s_y[A] \sum_{\boldsymbol{k}\in\mathbb{Z}^2} x[\boldsymbol{k}]h[(\boldsymbol{c}_x + \boldsymbol{c}_h) \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A] - \boldsymbol{k}] \\ &= s_y[A]y[(\boldsymbol{c}_x + \boldsymbol{c}_h) \circ (1 - \boldsymbol{v}[A]) + \boldsymbol{n} \circ \boldsymbol{v}[A]], \end{aligned}$$

where $s_y[A] = s_x[A]s_h[A]$. As $s_x[\cdot]$ and $s_h[\cdot]$ each satisfies (3.3), $s_y[\cdot]$ also satisfies (3.3). Thus, y is quadrantally centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{c}_x + \boldsymbol{c}_h$. The relation between the symmetry type of y and that of x and h is shown in Table 3.4.

Table 3.4: Symmetry type of y where $y = x * h$

Symmetry type of h	Symmetry type of x			
	even-even	even-odd	odd-even	odd-odd
even-even	even-even	even-odd	odd-even	odd-odd
even-odd	even-odd	even-even	odd-odd	odd-even
odd-even	odd-even	odd-odd	even-even	even-odd
odd-odd	odd-odd	odd-even	even-odd	even-even

Proof of 3. For the rotated quadrantal centrosymmetry case, let $\boldsymbol{c}_x = [c_{x,0} \;\; c_{x,1}]^T$ and $\boldsymbol{c}_h = \begin{bmatrix} c_{h,0} & c_{h,1} \end{bmatrix}^T$. We have

$$\begin{aligned} x[n_0,n_1] &= S_x T_x x[2c_{x,0}-n_0, 2c_{x,1}-n_1] \\ &= S_x x[c_{x,0}+c_{x,1}-n_1, c_{x,0}+c_{x,1}-n_0] \qquad (3.15) \\ &= T_x x[c_{x,0}-c_{x,1}+n_1, c_{x,1}-c_{x,0}+n_0], \quad \textit{and} \\ h[n_0,n_1] &= S_h T_h h[2c_{h,0}-n_0, 2c_{h,1}-n_1] \\ &= S_h h[c_{h,0}+c_{h,1}-n_1, c_{h,0}+c_{h,1}-n_0] \qquad (3.16) \\ &= T_h h[c_{h,0}-c_{h,1}+n_1, c_{h,1}-c_{h,0}+n_0]. \end{aligned}$$

From the proof of statement 1, we know that $y = x * h$ has centrosymmetry, i.e., $y[\boldsymbol{n}] = S_x T_x S_h T_x y[2(\boldsymbol{c}_x + \boldsymbol{c}_h) - \boldsymbol{n}]$. Next we prove that $y[n_0,n_1] = S_x S_h y[c_{x,0}+c_{x,1}+c_{h,0}+c_{h,1}-n_1, c_{x,0}+c_{x,1}+c_{h,0}+c_{h,1}-n_0]$. Substituting (3.15) and (3.16) into (3.9), we obtain

$$\begin{aligned} y[n_0,n_1] &= S_x S_h \sum_{k_0\in\mathbb{Z}} \sum_{k_1\in\mathbb{Z}} x[c_{x,0}+c_{x,1}-k_1, c_{x,0}+c_{x,1}-k_0] h[c_{h,0}+c_{h,1}-n_1+k_1, c_{h,0}+c_{h,1}-n_0+k_1] \\ &= S_x S_h \sum_{k_0'\in\mathbb{Z}} \sum_{k_1'\in\mathbb{Z}} x[k_1', k_0'] h[c_{h,0}+c_{h,1}-n_1+\left(c_{x,0}+c_{x,1}-k_1'\right), c_{h,0}+c_{h,1}-n_0+\left(c_{x,0}+c_{x,1}-k_0'\right)] \\ &= S_x S_h \sum_{k_0\in\mathbb{Z}} \sum_{k_1\in\mathbb{Z}} x[k_0,k_1] h[\left(c_{h,0}+c_{h,1}+c_{x,0}+c_{x,1}-n_1\right)-k_0, \left(c_{h,0}+c_{h,1}+c_{x,0}+c_{x,1}-n_0\right)-k_1] \\ &= S_x S_h y[c_{h,0}+c_{h,1}+c_{x,0}+c_{x,1}-n_1, c_{h,0}+c_{h,1}+c_{x,0}+c_{x,1}-n_0]. \end{aligned}$$

Similarly, we obtain that $y[n_0,n_1] = T_x T_h y[c_{x,0}+c_{h,0}-c_{x,1}-c_{h,1}+n_1, c_{x,1}+c_{h,1}-c_{x,0}-c_{h,0}+n_0]$. Thus, y is rotated quadrantally centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{c}_x + \boldsymbol{c}_h$. □

Note that statement 2 of this lemma holds for the case of higher-dimensional hyper-octantal centrosymmetry. The proof can be derived easily using an approach similar to that of the quadrantal centrosymmetry case. Next, we consider the effect of convolution on periodicity in the lemma below.

Lemma 3.3 (Preservation of periodicity under convolution)**.** *Let x and h be sequences defined on $\mathbb{Z}^2$, with x being $\boldsymbol{P}$-periodic. Then, $y = x * h$ is $\boldsymbol{P}$-periodic.*

Proof. Since x is $\boldsymbol{P}$-periodic, $x[\boldsymbol{n}] = x[\boldsymbol{n} + \boldsymbol{P}\boldsymbol{m}]$ for $\boldsymbol{m} \in \mathbb{Z}^2$. We have

$$\begin{aligned} y[\boldsymbol{n}] &= \sum_{\boldsymbol{k}\in\mathbb{Z}^2} x[\boldsymbol{k}]h[\boldsymbol{n}-\boldsymbol{k}] = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} x[\boldsymbol{k}+\boldsymbol{P}\boldsymbol{m}]h[\boldsymbol{n}-\boldsymbol{k}] \\ &= \sum_{\boldsymbol{k}'\in\mathbb{Z}^2} x[\boldsymbol{k}']h[\boldsymbol{n}-(\boldsymbol{k}'-\boldsymbol{P}\boldsymbol{m})] = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} x[\boldsymbol{k}]h[(\boldsymbol{n}+\boldsymbol{P}\boldsymbol{m})-\boldsymbol{k}] \\ &= y[\boldsymbol{n}+\boldsymbol{P}\boldsymbol{m}], \end{aligned}$$

for $\boldsymbol{m} \in \mathbb{Z}^2$. Thus, $y = x * h$ is $\boldsymbol{P}$-periodic. □

Next, we examine the effects of downsampling on periodicity and symmetry. First, we consider the case of periodicity in the lemma below.

Lemma 3.4 (Downsampling of periodic sequence)**.** *Let $\boldsymbol{M}$ be an arbitrary sampling (i.e., nonsingular integer) matrix. Let x be $\boldsymbol{P}$-periodic such that $\boldsymbol{M}^{-1}\boldsymbol{P}$ is an integer matrix. Then, $(\downarrow \boldsymbol{M})x$ is $(\boldsymbol{M}^{-1}\boldsymbol{P})$-periodic.*

Proof. Since x is $\boldsymbol{P}$-periodic, $x[\boldsymbol{n}] = x[\boldsymbol{n} + \boldsymbol{P}\boldsymbol{k}]$ for $\boldsymbol{k} \in \mathbb{Z}^2$. The downsampled sequence y is given by

$$y[\boldsymbol{n}] = (\downarrow \boldsymbol{M})x[\boldsymbol{n}] = x[\boldsymbol{M}\boldsymbol{n}] = x[\boldsymbol{M}\boldsymbol{n} + \boldsymbol{P}\boldsymbol{k}] = x[\boldsymbol{M}(\boldsymbol{n} + (\boldsymbol{M}^{-1}\boldsymbol{P})\boldsymbol{k})]. \tag{3.17}$$

Since $\boldsymbol{M}^{-1}\boldsymbol{P}$ is an integer matrix, we have

$$x[\boldsymbol{M}(\boldsymbol{n} + (\boldsymbol{M}^{-1}\boldsymbol{P})\boldsymbol{k})] = y[\boldsymbol{n} + (\boldsymbol{M}^{-1}\boldsymbol{P})\boldsymbol{k}]. \tag{3.18}$$

Substituting (3.18) into (3.17), we have that $y[\boldsymbol{n}] = y[\boldsymbol{n} + (\boldsymbol{M}^{-1}\boldsymbol{P})\boldsymbol{k}]$ for $\boldsymbol{k} \in \mathbb{Z}^2$. Therefore, y is $\boldsymbol{M}^{-1}\boldsymbol{P}$-periodic. □

Thus, from above, if a $\boldsymbol{P}$-periodic sequence has both of its periodicity vectors (i.e., columns of $\boldsymbol{P}$) on $\mathrm{LAT}(\boldsymbol{M})$, the sequence downsampled by $\boldsymbol{M}$ is periodic with the number of samples in one period being reduced by a factor of $|\det \boldsymbol{M}|$ relative to the original sequence. (As an aside, we notice that Lemmas 3.3 and 3.4 are not restricted to the quincunx case. They also hold for the general MD case.) Next, we consider the effects of downsampling on symmetry. The key results are given by the lemma below.

Lemma 3.5 (Downsampling of symmetric sequence)**.** *Let x be a sequence defined on $\mathbb{Z}^2$. Define $y = (\downarrow \boldsymbol{M})x$ with $\boldsymbol{M}$ being the quincunx generating matrix. Then, the following statements are true:*

1. *If x is centrosymmetric about $\boldsymbol{c}_x$ with $2\boldsymbol{c}_x \in \mathrm{LAT}(\boldsymbol{M})$, then y is centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{M}^{-1}\boldsymbol{c}_x$.*

2. *If x is quadrantally centrosymmetric about $\boldsymbol{c}_x \in \mathbb{Z}^2$, then, y is rotated quadrantally centrosymmetric about $\boldsymbol{M}^{-1}\boldsymbol{c}_x$.*

3. *If x is rotated quadrantally centrosymmetric about $\boldsymbol{c}_x$, then, y is quadrantally centrosymmetric about $\boldsymbol{M}^{-1}\boldsymbol{c}_x$.*

Proof. The downsampled sequence y is given by

$$y[n_0,n_1] = (\downarrow \boldsymbol{M})x[\boldsymbol{n}] = x[\boldsymbol{M}\boldsymbol{n}] = x[n_0+n_1, n_0-n_1]. \tag{3.19}$$

Proof of 1. As x is centrosymmetric about $\boldsymbol{c}_x = [c_0 \quad c_1]^T$, we have

$$x[n_0+n_1,n_0-n_1] = Sx[2c_0-n_0-n_1, 2c_1-n_0+n_1]. \tag{3.20}$$

Since $2\boldsymbol{c}_x \in \mathrm{LAT}(\boldsymbol{M})$, $x[2c_0-n_0-n_1,2c_1-n_0+n_1] = y[c_0+c_1-n_0, c_0-c_1-n_1]$. Thus, we obtain $y[\boldsymbol{n}] = Sy[2\boldsymbol{M}^{-1}\boldsymbol{c}_x - \boldsymbol{n}]$. That is, y is centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{M}^{-1}\boldsymbol{c}_x$.

Proof of 2. As x is quadrantally centrosymmetric about $\boldsymbol{c}_x = [c_0 \quad c_1]^T$, we have

$$\begin{aligned} x[n_0+n_1,n_0-n_1] &= STx[2c_0-n_0-n_1, 2c_1-n_0+n_1] \\ &= Sx[2c_0-n_0-n_1, n_0-n_1] \\ &= Tx[n_0+n_1, 2c_1-n_0+n_1] \end{aligned} \tag{3.21}$$

for $S,T \in \{-1,1\}$. Since $c_0, c_1 \in \mathbb{Z}$, we have

$$\begin{aligned} x[2c_0-n_0-n_1, 2c_1-n_0+n_1] &= y[c_0+c_1-n_0, c_0-c_1-n_1] \\ x[2c_0-n_0-n_1, n_0-n_1] &= y[c_0-n_1, c_0-n_0] \\ x[n_0+n_1, 2c_1-n_0+n_1] &= y[c_1+n_1, -c_1+n_0]. \end{aligned}$$

Substituting (3.19) and the above three equations into (3.21), we obtain

$$y[n_0,n_1] = STy[c_0+c_1-n_0, c_0-c_1-n_1] = Sy[c_0-n_1, c_0-n_0] = Ty[c_1+n_1, -c_1+n_0].$$

Therefore, the downsampled sequence y has rotated quadrantal centrosymmetry about $\boldsymbol{M}^{-1}\boldsymbol{c}_x = \left[\frac{c_0+c_1}{2} \quad \frac{c_0-c_1}{2}\right]^T$.

Proof of 3. As x is rotated quadrantally centrosymmetric about $\boldsymbol{c}_x = [c_0 \quad c_1]^T$, we have

$$\begin{aligned} x[n_0+n_1,n_0-n_1] &= STx[2c_0-n_0-n_1, 2c_1-n_0+n_1] \\ &= Sx[c_0+c_1-n_0+n_1, c_0+c_1-n_0-n_1] \\ &= Tx[c_0-c_1+n_0-n_1, c_1-c_0+n_0+n_1] \end{aligned} \tag{3.22}$$

for $S,T \in \{-1,1\}$. Since $\boldsymbol{c}$ is the symmetry center of a rotated quadrantally centrosymmetric sequence, $c_0 \pm c_1 \in \mathbb{Z}$. Then, we have

$$x[2c_0 - n_0 - n_1, 2c_1 - n_0 + n_1] = y[c_0 + c_1 - n_0, c_0 - c_1 - n_1]$$
$$x[c_0 + c_1 - n_0 + n_1, c_0 + c_1 - n_0 - n_1] = y[c_0 + c_1 - n_0, n_1]$$
$$x[c_0 - c_1 + n_0 - n_1, c_1 - c_0 + n_0 + n_1] = y[n_0, c_0 - c_1 + n_0].$$

Substituting (3.19) and the above three equations into (3.22), we obtain

$$y[n_0,n_1] = STy[c_0 + c_1 - n_0, c_0 - c_1 - n_1] = Sy[c_0 + c_1 - n_0, n_1] = Ty[n_0, c_0 - c_1 + n_0].$$

Therefore, the downsampled sequence y has rotated quadrantal centrosymmetry about $\boldsymbol{M}^{-1}\boldsymbol{c}_x = \left[\frac{c_0+c_1}{2} \quad \frac{c_0-c_1}{2}\right]^T$. □

Now, we consider the effects of the upsampler on the periodicity and symmetry properties. These results are useful in the inverse transform shown in Figure 3.1(b). They are also used in Chapter 4 for the design of linear-phase quincunx filter banks.

Lemma 3.6 (Upsampling of periodic sequence)**.** *Let $\boldsymbol{M}$ be an arbitrary nonsingular integer matrix. Let x be $\boldsymbol{P}$-periodic. Then, $(\uparrow \boldsymbol{M})x$ is $(\boldsymbol{MP})$-periodic.*

Proof. Since x is $\boldsymbol{P}$-periodic, $x[\boldsymbol{n}] = x[\boldsymbol{n} + \boldsymbol{P}\boldsymbol{k}]$ for $\boldsymbol{k} \in \mathbb{Z}^D$. The upsampled sequence $y = (\uparrow \boldsymbol{M})x$ is given by

$$y[\boldsymbol{n}] = \begin{cases} x[\boldsymbol{M}^{-1}\boldsymbol{n}] & \text{if } n \in \mathrm{LAT}(\boldsymbol{M}) \\ 0 & \text{otherwise} \end{cases}$$
$$= \begin{cases} x[\boldsymbol{M}^{-1}\boldsymbol{n} + \boldsymbol{P}\boldsymbol{k}] & \text{if } n \in \mathrm{LAT}(\boldsymbol{M}) \\ 0 & \text{otherwise} \end{cases}$$
$$= \begin{cases} x[\boldsymbol{M}^{-1}(\boldsymbol{n} + \boldsymbol{MP}\boldsymbol{k})] & \text{if } n + \boldsymbol{MP}\boldsymbol{k} \in \mathrm{LAT}(\boldsymbol{M}) \\ 0 & \text{otherwise} \end{cases}$$
$$= y[\boldsymbol{n} + (\boldsymbol{MP})\boldsymbol{k}],$$

for $\boldsymbol{k} \in \mathbb{Z}^D$. Therefore, y is $(\boldsymbol{MP})$-periodic. □

At last, we consider preservation of symmetry under upsampling. The key results are given in the lemma below.

Lemma 3.7 (Upsampling of symmetric sequence)**.** *Let x be a sequence defined on $\mathbb{Z}^2$. Define $y = (\uparrow \boldsymbol{M})x$ with $\boldsymbol{M}$ being the quincunx generating matrix. Then, the following statements hold:*

1. *If x is centrosymmetric about $\boldsymbol{c}_x$, then y is centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{M}\boldsymbol{c}_x$.*
2. *If x is quadrantally centrosymmetric about $\boldsymbol{c}_x$, then y is rotated quadrantally centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{M}\boldsymbol{c}_x$.*
3. *If x is rotated quadrantally centrosymmetric about $\boldsymbol{c}_x$, then y is quadrantally centrosymmetric about $\boldsymbol{c}_y = \boldsymbol{M}\boldsymbol{c}_x$.*

Proof. The proof is similar to that of Lemmas 3.5 and 3.6 and is omitted here. □

The above results show that the symmetry and periodicity properties may be preserved under convolution and the downsampling and upsampling operations of the quincunx filter banks. These results will be used in later derivations of our new symmetric extension algorithms.

3.6 Symmetric Extension Algorithm

Using our previous results, we now derive a scheme based on symmetric extension that allows for the construction of nonexpansive transforms based on a quincunx filter bank. For nonexpansive transforms, the number of independent samples in each of the subbands y_0 and y_1 is approximately half of that in the extended input sequence x, where the sequences are as defined in Figure 3.1(a). This suggests that the subband sequences also have four-fold symmetry and periodicity. Therefore, from Lemma 3.5, the analysis filters H_0 and H_1 should have quadrantal centrosymmetry and their group delays should be chosen such that the symmetry centers of u_0 and u_1 are both on the integer lattice.

We recall the PR condition for a quincunx filter bank is given by

$$H_0(\boldsymbol{z})G_0(\boldsymbol{z}) + H_1(\boldsymbol{z})G_1(\boldsymbol{z}) = 2 \quad \text{and} \tag{3.23a}$$

$$H_0(-\boldsymbol{z})G_0(\boldsymbol{z}) + H_1(-\boldsymbol{z})G_1(\boldsymbol{z}) = 0, \tag{3.23b}$$

where $H_0(\boldsymbol{z})$, $H_1(\boldsymbol{z})$, $G_0(\boldsymbol{z})$, and $G_1(\boldsymbol{z})$ are the lowpass analysis, highpass analysis, lowpass synthesis, and highpass synthesis filter transfer functions, respectively. If we let $G_0(\boldsymbol{z}) = -\boldsymbol{z}^{\boldsymbol{l}}H_1(-\boldsymbol{z})$ and $G_1(\boldsymbol{z}) = \boldsymbol{z}^{\boldsymbol{l}}H_0(-\boldsymbol{z})$ with some $\boldsymbol{l} = [l_0 \;\; l_1]^T \in \mathbb{Z}^2$, then (3.23b) is satisfied. If we further define $P(\boldsymbol{z}) = H_0(\boldsymbol{z})H_1(-\boldsymbol{z})$, then (3.23a) becomes

$$P(\boldsymbol{z}) - P(-\boldsymbol{z}) = -2\boldsymbol{z}^{-\boldsymbol{l}}. \tag{3.24}$$

If H_0 and H_1 have quadrantal centrosymmetry with group delays $\boldsymbol{d}_0 = [d_{0,0} \;\; d_{0,1}]^T$ and $\boldsymbol{d}_1 = [d_{1,0} \;\; d_{1,1}]^T$, respectively, then $P(\boldsymbol{z})$ and $P(-\boldsymbol{z})$ are also quadrantally centrosymmetric with group delay $\boldsymbol{d}_p = \boldsymbol{d}_0 + \boldsymbol{d}_1$. We can see that when P has more than two nonzero coefficients, in order to satisfy (3.24), P must have even-even quadrantal centrosymmetry with symmetry center $\boldsymbol{d}_p \in \mathbb{Z}^2 \setminus \mathrm{LAT}(\boldsymbol{M})$, and the coefficients of P are zero at all non-lattice points except $\boldsymbol{d}_p$.

Now we examine the possible choices of the quadrantally centrosymmetric analysis filters. As previously shown in Section 3.3, a filter with odd-even, even-odd, or odd-odd quadrantal centrosymmetry cannot be a good lowpass filter. Therefore, the lowpass analysis filter H_0 must have the even-even type of symmetry. In order for P to have the even-even type of symmetry with group delay $\boldsymbol{d}_p \in \mathbb{Z}^2$, by using Tables 3.2 and 3.4, there are altogether four possibilities of the choice of the highpass analysis filter H_1 according to the location of the group delay $\boldsymbol{d}_0 = [d_{0,0} \quad d_{0,1}]^T$ of the lowpass analysis filter H_0:

1. when $\boldsymbol{d}_0 \in \mathbb{Z}^2$, H_1 must have even-even type symmetry with $\boldsymbol{d}_1 \in \mathbb{Z}^2$;
2. when $d_{0,0} \in \frac{1}{2}\mathbb{Z}_{odd}$ and $d_{0,1} \in \mathbb{Z}$, H_1 must have odd-even type symmetry with $d_{1,0} \in \frac{1}{2}\mathbb{Z}_{odd}$ and $d_{1,1} \in \mathbb{Z}$;
3. when $d_{0,0} \in \mathbb{Z}$ and $d_{0,1} \in \frac{1}{2}\mathbb{Z}_{odd}$, H_1 must have even-odd type symmetry with $d_{1,0} \in \mathbb{Z}$ and $d_{1,1} \in \frac{1}{2}\mathbb{Z}_{odd}$;
4. when $\boldsymbol{d}_0 \in \frac{1}{2}\mathbb{Z}^2_{odd}$, H_1 must have odd-odd type symmetry with $\boldsymbol{d}_1 \in \frac{1}{2}\mathbb{Z}^2_{odd}$.

In the remainder of this chapter, we discuss, for each of the above four cases, how one can construct nonexpansive transforms with the mapping scheme introduced in Section 3.4.

3.6.1 Type-1 Symmetric Extension Algorithm

We first consider the case where both analysis filters have even-even quadrantal centrosymmetry with group delays $\boldsymbol{d}_0, \boldsymbol{d}_1 \in \mathbb{Z}^2$. As mentioned earlier, the group delay of $P(\boldsymbol{z}) = H_0(\boldsymbol{z})H_1(-\boldsymbol{z})$ should satisfy that $\boldsymbol{d}_p = \boldsymbol{d}_0 + \boldsymbol{d}_1 \in \mathbb{Z}^2 \setminus \mathrm{LAT}(\boldsymbol{M})$. Therefore, it follows that $\boldsymbol{d}_0$ and $\boldsymbol{d}_1$ are in different cosets of the quincunx lattice. This suggests that such a PR filter bank is compatible with the type-1 symmetric extension scheme in (3.8) and leads to the result below.

Theorem 3.8 (Type-1 symmetric extension algorithm)**.** *Consider the filter bank shown in Figure 3.1, where the input sequence $\tilde{x}$ is defined on the rectangular region $\{0,1,\ldots,L_0-1\} \times \{0,1,\ldots,L_1-1\}$ and x is the type-1 symmetric extension of $\tilde{x}$ as given by (3.8). If H_0 and H_1 are quadrantally centrosymmetric with group delays $\boldsymbol{d}_0 = [d_{0,0} \quad d_{0,1}]^T \in \mathbb{Z}^2$ and $\boldsymbol{d}_1 = [d_{1,0} \quad d_{1,1}]^T \in \mathbb{Z}^2$, respectively, and $\boldsymbol{d}_0$ and $\boldsymbol{d}_1$ are in different cosets of the quincunx lattice, then the subband output y_0 can be completely characterized by N_0 samples*

with indices $\boldsymbol{n} = [n_0 \;\; n_1]^T$ *given by*

$$\begin{gathered}\left\lceil \frac{d_{0,0}+d_{0,1}}{2} \right\rceil \le n_0 \le \left\lfloor \frac{d_{0,0}+d_{0,1}+L_0+L_1}{2} \right\rfloor - 1, \quad \textit{and} \\ \max\{d_{0,0}-n_0, n_0-d_{0,1}-L_1+1\} \le n_1 \le \min\{d_{0,0}+L_0-1-n_0, n_0-d_{0,1}\};\end{gathered} \tag{3.25}$$

y_1 *can be completely characterized by* N_1 *samples with indices* $\boldsymbol{n} = [n_0 \;\; n_1]^T$ *given by*

$$\begin{gathered}\left\lceil \frac{d_{1,0}+d_{1,1}}{2} \right\rceil \le n_0 \le \left\lfloor \frac{d_{1,0}+d_{1,1}+L_0+L_1}{2} \right\rfloor - 1, \quad \textit{and} \\ \max\{d_{1,0}-n_0, n_0-d_{1,1}-L_1+1\} \le n_1 \le \min\{d_{1,0}+L_0-1-n_0, n_0-d_{1,1}\};\end{gathered} \tag{3.26}$$

and $N_0 + N_1 = L_0L_1$*, i.e., the transform is nonexpansive.*

Proof. In what follows, we refer to the definitions of sequences in Figure 3.1(a). From Lemma 3.1, we know that x is $\boldsymbol{P}$-periodic with $\boldsymbol{P} = \begin{bmatrix} 2L_0-2 & 0 \\ 0 & 2L_1-2 \end{bmatrix}$, and is quadrantally centrosymmetric about $\boldsymbol{0}$. Consider the first channel, where H_0 is quadrantally centrosymmetric with group delay $\boldsymbol{d}_0 \in \mathbb{Z}^2$. Then, the analysis filter output u_0 is $\boldsymbol{P}$-periodic from Lemma 3.3, and quadrantally centrosymmetric about $\boldsymbol{0} + \boldsymbol{d}_0 = \boldsymbol{d}_0$ from Lemma 3.2. Since $\boldsymbol{M}^{-1}\boldsymbol{P} = \begin{bmatrix} L_0-1 & L_1-1 \\ L_0-1 & -L_1+1 \end{bmatrix}$ is an integer matrix and $\boldsymbol{d}_0 \in \mathbb{Z}^2$, y_0 is $\boldsymbol{M}^{-1}\boldsymbol{P}$-periodic from Lemma 3.4, and rotated quadrantally centrosymmetric about $\boldsymbol{M}^{-1}\boldsymbol{d}_0$ from Lemma 3.5.

Therefore, y_0 can be completely characterized by samples with indices $\boldsymbol{n} = [n_0 \;\; n_1]^T$ given by

$$\boldsymbol{M}\boldsymbol{n} \in \{d_{0,0}, d_{0,0}+1, \ldots, d_{0,0}+L_0-1\} \times \{d_{0,1}, d_{0,1}+1, \ldots, d_{0,1}+L_1-1\}. \tag{3.27}$$

Solving (3.27), we obtain the conditions for n_0 and n_1 as shown in (3.25). The number N_0 of characteristic samples of y_0 is given by

$$N_0 = \begin{cases} \frac{1}{2}L_0L_1 & \text{for } L_0L_1 \text{ even} \\ \frac{1}{2}(L_0L_1+1) & \text{for } L_0L_1 \text{ odd}, \boldsymbol{d}_0 \in \mathrm{LAT}(\boldsymbol{M}) \\ \frac{1}{2}(L_0L_1-1) & \text{for } L_0L_1 \text{ odd}, \boldsymbol{d}_0 \notin \mathrm{LAT}(\boldsymbol{M}) \end{cases} \tag{3.28}$$

which can be equivalently written as

$$N_0 = \left\lfloor \tfrac{1}{2}(L_0L_1 + d_{0,0} + d_{0,1} + 1) \right\rfloor - \left\lceil \tfrac{1}{2}(d_{0,0}+d_{0,1}) \right\rceil.$$

Similarly, y_1 is characterized by samples with indices $\boldsymbol{n} = [n_0 \;\; n_1]^T$ given by (3.26). The number N_1 of characteristic samples of y_1 is given by

$$N_1 = \begin{cases} \frac{1}{2}L_0L_1 & \text{for } L_0L_1 \text{ even} \\ \frac{1}{2}(L_0L_1+1) & \text{for } L_0L_1 \text{ odd}, \boldsymbol{d}_1 \in \mathrm{LAT}(\boldsymbol{M}) \\ \frac{1}{2}(L_0L_1-1) & \text{for } L_0L_1 \text{ odd}, \boldsymbol{d}_1 \notin \mathrm{LAT}(\boldsymbol{M}). \end{cases} \tag{3.29}$$

Since only one of $\boldsymbol{d}_0$ and $\boldsymbol{d}_1$ is on the quincunx lattice, from (3.28) and (3.29), we have $N_0 + N_1 = L_0 L_1$. □

Below, we show an example of the type-1 symmetric extension algorithm. We first give a quincunx filter bank satisfying the conditions in Theorem 3.8, then apply this filter bank to a finite-extent input sequence.

Example 3.2 (Type-1 symmetric extension algorithm)**.** The filter bank is constructed using the method proposed in [6] with two primal and two dual vanishing moments. This filter bank was also proposed in [34]. Using the matrix notation defined in (2.2) in Section 2.3.1, the impulse responses of the analysis filters H_0 and H_1 are expressed by

$$h_0[\boldsymbol{n}] \sim \tfrac{1}{32}\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & 4 & -2 & 0 \\ -1 & 4 & \boxed{28} & 4 & -1 \\ 0 & -2 & 4 & -2 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad h_1[\boldsymbol{n}] \sim \tfrac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & \boxed{\text{-1}} \\ 0 & -1 & 0 \end{bmatrix}, \tag{3.30}$$

and the group delays are $[0\ \ 0]^T$ and $[-1\ \ 0]^T$, respectively. The synthesis filter transfer functions are computed as $G_0(\boldsymbol{z}) = -z_0^{-1} H_1(-\boldsymbol{z})$ and $G_1(\boldsymbol{z}) = z_0^{-1} H_0(-\boldsymbol{z})$. The frequency responses of the filters are shown in Figure 3.6, and the primal/dual scaling and wavelet functions are depicted in Figure 3.7.

We apply this filter bank to the input sequence $\tilde{x}$ in Example 3.1 as shown in Figure 3.5(a), which contains four independent samples a, b, c, and d defined on $\{0,1\} \times \{0,1\}$. We use the symmetric extension structure shown in Figure 3.1(a). The type-1 extended sequence x is shown in Figure 3.5(b). The sequences u_0, u_1 and y_0, y_1 are shown in Figure 3.8. For each sequence, the boldface samples represent points on the lattice generated by the periodicity matrix of this sequence, the dot represents the symmetry center, and the samples inside the dashed lines are the independent samples of this sequence. We can see from Figures 3.8(c) and (d) that the subband sequence $\tilde{y}_0$ has two independent samples $\frac{1}{4}(3a - d + b + c)$ and $\frac{1}{4}(3d - a + b + c)$ located at $(0,0)$ and $(1,0)$, respectively, and $\tilde{y}_1$ has two independent samples $c - \frac{1}{2}(a+d)$ and $b - \frac{1}{2}(a+d)$ located at $(1,-1)$ and $(1,0)$, respectively. The above results are consistent with the results in Theorem 3.8. □

3.6.2 Type-2 Symmetric Extension Algorithm

Recall that at the beginning of Section 3.6, we have introduced four types of quadrantally centrosymmetric PR quincunx filter banks. In this section, we consider the second type of PR filter bank, where H_1 has the odd-even quadrantal centrosymmetry, and the group delays of H_0 and H_1 satisfy $d_{0,0}, d_{1,0} \in \frac{1}{2}\mathbb{Z}_{odd}$, $d_{0,1}, d_{1,1} \in \mathbb{Z}$, and $\boldsymbol{d}_0 + \boldsymbol{d}_1 \in \mathbb{Z}^2 \setminus \mathrm{LAT}(\boldsymbol{M})$. In order for this filter bank to have the nonexpansive property, the sequences

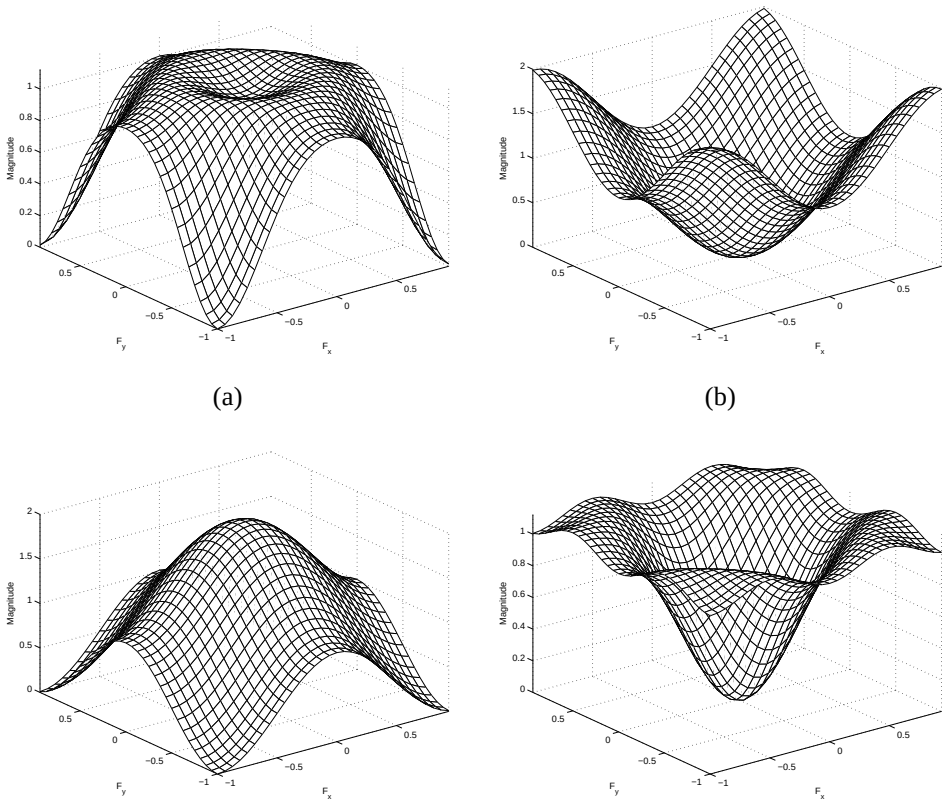


Figure 3.6: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for a type-1 filter bank.

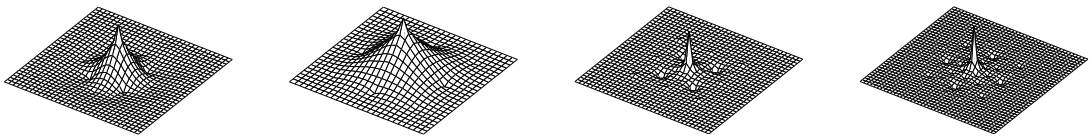


Figure 3.7: Scaling and wavelet functions for a type-1 filter bank. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

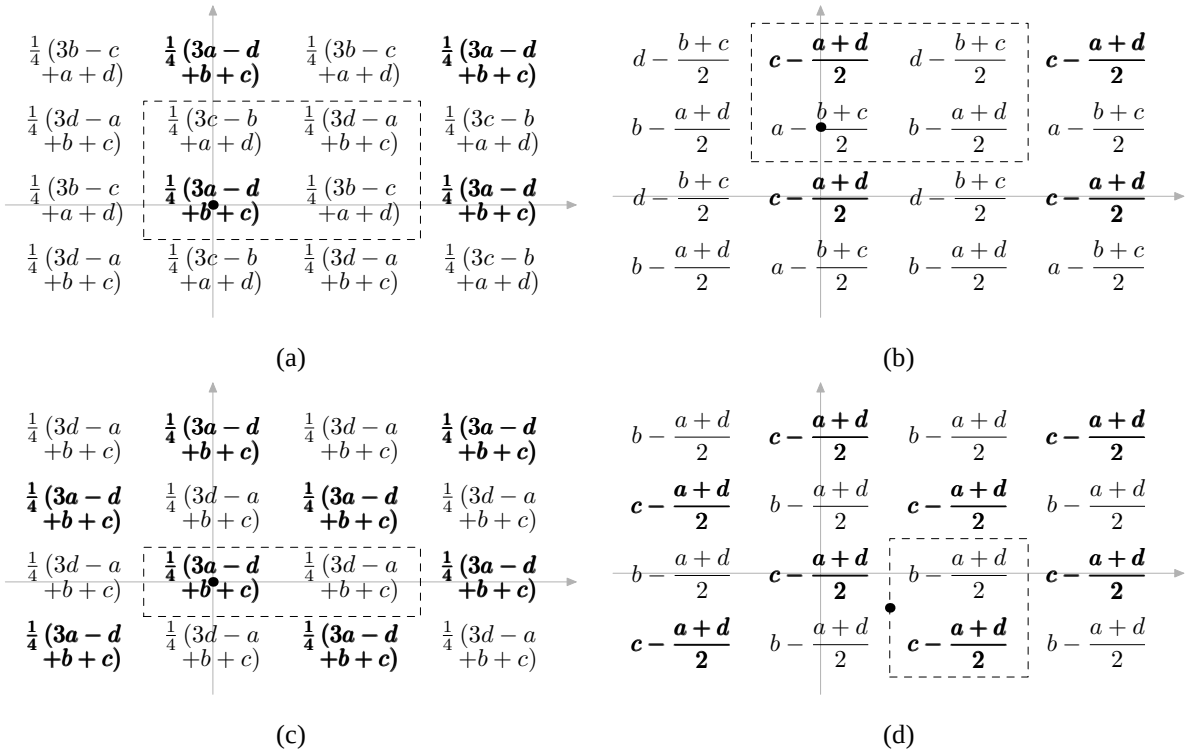


Figure 3.8: The intermediate sequences (a) u_0, (b) u_1 and output sequences (c) y_0, (d) y_1 in the type-1 filter bank. The sequences are as defined in Figure 3.1(a).

y_0 and y_1, as defined in Figure 3.1(a), should have four-fold symmetry and periodicity. This requires the sequences u_0 and u_1 to be quadrantally centrosymmetric with symmetry centers on the integer lattice $\mathbb{Z}^2$. These observations suggest that this type of PR filter bank may be compatible with the type-2 symmetric extension defined in (3.8), where the extended sequence x is half-sample symmetric in the horizontal direction and whole-sample symmetric in the vertical direction. This, in fact, is the case, as demonstrated by the theorem below.

Theorem 3.9 (Type-2 symmetric extension algorithm)**.** *Consider the analysis filter bank shown in Figure 3.1(a), where the input sequence $\tilde{x}$ is defined on the rectangular region $\{0,1,\ldots,L_0-1\}\times\{0,1,\ldots,L_1-1\}$ and x is the type-2 symmetric extension of $\tilde{x}$ as given by (3.8). Let $\boldsymbol{M}$ denote the quincunx generating matrix $\left[\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right]$. Suppose that the analysis filters H_0 and H_1 satisfy the following conditions: 1) H_0 has even-even quadrantal centrosymmetry with group delay $\boldsymbol{d}_0=[d_{0,0}\quad d_{0,1}]^T$, $d_{0,0}\in\frac{1}{2}\mathbb{Z}_{odd}$ and $d_{0,1}\in\mathbb{Z}$; 2) H_1 has odd-even quadrantal centrosymmetry with group delay $\boldsymbol{d}_1=[d_{1,0}\quad d_{1,1}]^T$, $d_{1,0}\in\frac{1}{2}\mathbb{Z}_{odd}$ and $d_{1,1}\in\mathbb{Z}$; 3) $\boldsymbol{d}_0-\boldsymbol{d}_1\in\mathrm{LAT}(\boldsymbol{M})$. In this case, the subband output y_0 can be completely characterized by N_0 samples with indices $\boldsymbol{n}=[n_0\quad n_1]^T$ given by*

$$\left\lceil\frac{d_{0,0}+d_{0,1}-\frac{1}{2}}{2}\right\rceil\le n_0\le\left\lfloor\frac{d_{0,0}+d_{0,1}+L_0+L_1-\frac{3}{2}}{2}\right\rfloor\quad\textit{and}$$
$$\max\{d_{0,0}-n_0-\tfrac{1}{2},n_0-d_{0,1}-L_1+1\}\le n_1\le\min\{d_{0,0}+L_0-n_0-\tfrac{1}{2},n_0-d_{0,1}\}; \tag{3.31}$$

y_1 can be completely characterized by N_1 samples with indices $\boldsymbol{n}=[n_0\quad n_1]^T$ given by

$$\left\lceil\frac{d_{1,0}+d_{1,1}+\frac{1}{2}}{2}\right\rceil\le n_0\le\left\lfloor\frac{d_{1,0}+d_{1,1}+L_0+L_1-\frac{5}{2}}{2}\right\rfloor\quad\textit{and}$$
$$\max\{d_{1,0}-n_0+\tfrac{1}{2},n_0-d_{1,1}-L_1+1\}\le n_1\le\min\{d_{1,0}+L_0-n_0-\tfrac{3}{2},n_0-d_{1,1}\}; \tag{3.32}$$

and $N_0+N_1=L_0L_1$, i.e., the transform is nonexpansive.

Proof. In what follows, we refer to the definitions of sequences in Figure 3.1(a). From Lemma 3.1, we know that x is $\boldsymbol{P}$-periodic with $\boldsymbol{P}=\left[\begin{smallmatrix}2L_0 & 0\\ 0 & 2L_1-2\end{smallmatrix}\right]$ and quadrantally centrosymmetric about $\left[-\frac{1}{2}\ 0\right]^T$. Consider the first channel, where H_0 is quadrantally centrosymmetric with group delay $\boldsymbol{d}_0=[d_{0,0}\quad d_{0,1}]^T$ satisfying $d_{0,0}\in\frac{1}{2}\mathbb{Z}_{odd}$ and $d_{0,1}\in\mathbb{Z}$. Then, the analysis filter output u_0 is $\boldsymbol{P}$-periodic from Lemma 3.3 and quadrantally centrosymmetric about $\boldsymbol{c}_{u_0}=\left[d_{0,0}-\frac{1}{2}\quad d_{0,1}\right]^T$ from Lemma 3.2. Since $\boldsymbol{M}^{-1}\boldsymbol{P}=\left[\begin{smallmatrix}L_0 & L_1-1\\ L_0 & -L_1+1\end{smallmatrix}\right]$ is an integer matrix and $\boldsymbol{c}_{u_0}\in\mathbb{Z}^2$, y_0 is $\boldsymbol{M}^{-1}\boldsymbol{P}$-periodic from Lemma 3.4 and rotated quadrantally centrosymmetric about $\boldsymbol{M}^{-1}\boldsymbol{c}_{u_0}$ from Lemma 3.5. Therefore, y_0 can be completely characterized by samples with indices $\boldsymbol{n}=[n_0\quad n_1]^T$ given by

$$\boldsymbol{M}\boldsymbol{n}\in\{d_{0,0}-\tfrac{1}{2},d_{0,0}+\tfrac{1}{2},\ldots,d_{0,0}+L_0-\tfrac{1}{2}\}\times\{d_{0,1},d_{0,1}+1,\ldots,d_{0,1}+L_1-1\}. \tag{3.33}$$

Solving (3.33), we obtain the conditions for n_0 and n_1 as shown in (3.31). The number N_0 of characteristic samples of y_0 is given by

$$N_0 = \begin{cases} \frac{1}{2}(L_0+1)L_1 & \text{for } (L_0+1)L_1 \text{ even} \\ \frac{1}{2}\left[(L_0+1)L_1+1\right] & \text{for } (L_0+1)L_1 \text{ odd}, \boldsymbol{c}_{u_0} \in \mathrm{LAT}(\boldsymbol{M}) \\ \frac{1}{2}\left[(L_0+1)L_1-1\right] & \text{for } (L_0+1)L_1 \text{ odd}, \boldsymbol{c}_{u_0} \notin \mathrm{LAT}(\boldsymbol{M}). \end{cases} \tag{3.34}$$

The above equation can be equivalently written as

$$N_0 = \left\lfloor \tfrac{1}{2}((L_0+1)L_1+d_{0,0}+d_{0,1}+\tfrac{1}{2}) \right\rfloor - \left\lceil \tfrac{1}{2}(d_{0,0}+d_{0,1}-\tfrac{1}{2}) \right\rceil.$$

Now we consider the second channel. Similar to the case of the first channel, we know that u_1 is $\boldsymbol{P}$-periodic and quadrantally centrosymmetric about $\boldsymbol{c}_{u_1} = \begin{bmatrix} d_{1,0}-\frac{1}{2} & d_{1,1} \end{bmatrix}^T$. Since H_1 has odd-even symmetry, u_1 also has odd-even symmetry. Note that each 1D horizontal slice of u_1 has whole-sample antisymmetry. Then, y_1 can be completely characterized by samples with indices $\boldsymbol{n} = \begin{bmatrix} n_0 & n_1 \end{bmatrix}^T$ given by

$$\boldsymbol{M}\boldsymbol{n} \in \{d_{1,0}+\tfrac{1}{2}, d_{1,0}+\tfrac{3}{2}, \ldots, d_{1,0}+L_0-\tfrac{3}{2}\} \times \{d_{1,1}, d_{1,1}+1, \ldots, d_{1,1}+L_1-1\}. \tag{3.35}$$

Solving (3.35), we obtain the conditions for n_0 and n_1 as shown in (3.32). The number of characteristic samples of y_1 is given by

$$N_1 = \begin{cases} \frac{1}{2}(L_0-1)L_1 & \text{for } (L_0-1)L_1 \text{ even} \\ \frac{1}{2}\left[(L_0-1)L_1-1\right] & \text{for } (L_0-1)L_1 \text{ odd}, \boldsymbol{c}_{u_1} \in \mathrm{LAT}(\boldsymbol{M}) \\ \frac{1}{2}\left[(L_0-1)L_1+1\right] & \text{for } (L_0-1)L_1 \text{ odd}, \boldsymbol{c}_{u_1} \notin \mathrm{LAT}(\boldsymbol{M}). \end{cases} \tag{3.36}$$

The above equation can be simplified as

$$N_1 = \left\lfloor \tfrac{1}{2}((L_0-1)L_1+d_{1,0}+d_{1,1}+\tfrac{3}{2}) \right\rfloor - \left\lceil \tfrac{1}{2}(d_{1,0}+d_{1,1}+\tfrac{1}{2}) \right\rceil.$$

Since $\boldsymbol{d}_0 - \boldsymbol{d}_1 \in \mathrm{LAT}(\boldsymbol{M})$, $\boldsymbol{c}_{u_0}$ and $\boldsymbol{c}_{u_1}$ are in the same coset of the quincunx lattice. Therefore, from (3.34) and (3.36), we have $N_0 + N_1 = L_0 L_1$. □

The above theorem is illustrated by the following example. A quincunx filter bank satisfying the three conditions in Theorem 3.9 is given, then it is applied to the input sequence in Example 3.1.

Example 3.3 (Type-2 symmetric extension algorithm)**.** An example of this type of PR filter bank is the Haar-like filter bank given by the transfer functions

$$H_0(z_0,z_1) = \tfrac{1}{2}(1+z_0), H_1(z_0,z_1) = 1-z_0, G_0(z_0,z_1) = 1+z_0^{-1}, \text{ and } G_1(z_0,z_1) = \tfrac{1}{2}\left(-1+z_0^{-1}\right).$$

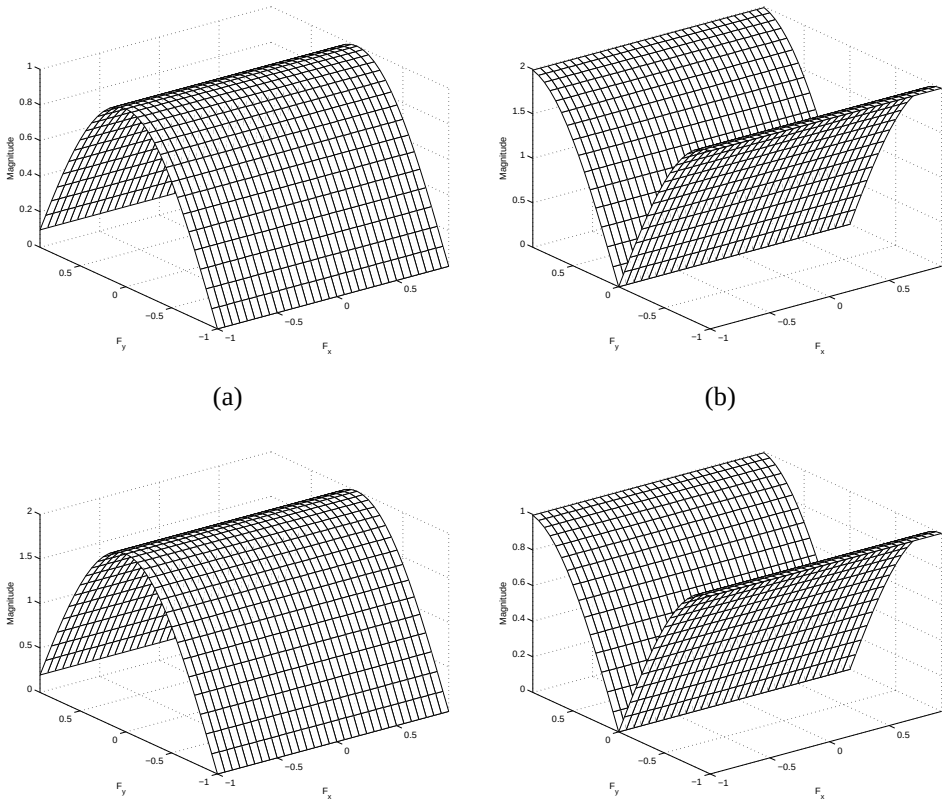


Figure 3.9: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for the Haar-like filter bank.

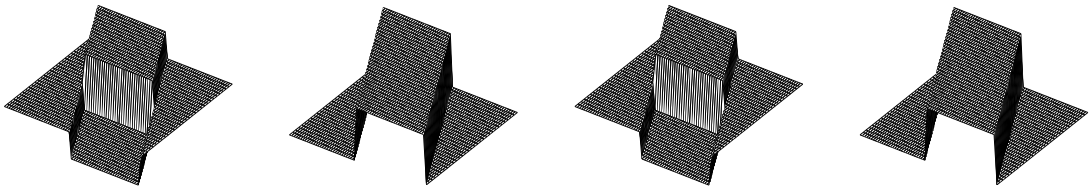


Figure 3.10: Scaling and wavelet functions for the Haar-like filter bank. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

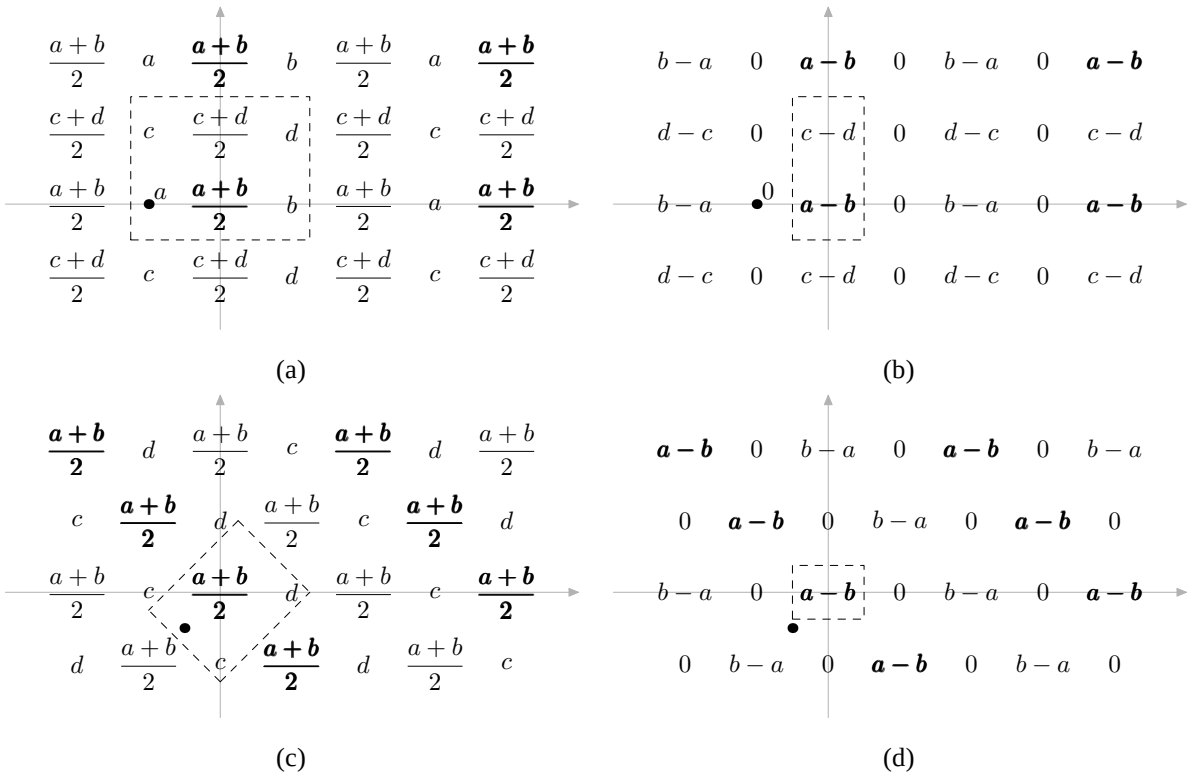


Figure 3.11: The intermediate sequences (a) u_0, (b) u_1 and output sequences (c) y_0, (d) y_1 in the Haar-like filter bank. The sequences are as defined in Figure 3.1(a).

The group delays of the analysis filters are both $[-\frac{1}{2}\ 0]^T$. The frequency responses are shown in Figure 3.9, and the scaling and wavelet functions associated with this filter bank are illustrated in Figure 3.10.

We apply this filter bank to Example 3.1 given in Section 3.4 with the type-2 symmetric extension, where the original input sequence $\tilde{x}$ with four samples is shown in Figure 3.5(a), and the extended sequence x is shown in Figure 3.5(c). The outputs u_0, u_1, y_0, and y_1, as defined in Figure 3.1(a), are shown in Figure 3.11, where the boldface samples represent points on the periodicity lattice, the dot represents the symmetry center, and the independent samples of each sequence are located inside the area surrounded by the dashed lines. From Figures 3.11(c) and (d), we see that $\tilde{y}_0$ has three independent samples $\frac{1}{2}(a+b)$, c and d at $(0,0)$, $(0,-1)$ and $(1,0)$, respectively, and $\tilde{y}_1$ has only one independent sample $a-b$ at $(0,0)$. The total number of independent samples in the subbands is four. Thus, the transform is nonexpansive. □

3.6.3 Type-3 Symmetric Extension Algorithm

Recall that there are four types of PR quincunx filter banks with quadrantally centrosymmetric filters. Next we consider the third type, where H_1 is even-odd quadrantally centrosymmetric. In the below theorem, we show this type of filter bank is compatible with the type-3 symmetric extension defined in (3.8), where the extended sequence x has whole-sample symmetry in the horizontal direction and half-sample symmetry in the vertical direction.

Theorem 3.10 (Type-3 symmetric extension algorithm)**.** *Consider the analysis filter bank shown in Figure 3.1(a), where $\tilde{x}$ is defined on the rectangular region $\{0,1,\ldots,L_0-1\}\times\{0,1,\ldots,L_1-1\}$ and x is the type-3 symmetric extension of $\tilde{x}$ as given by (3.8). Let $\boldsymbol{M}$ denote the quincunx generating matrix $\left[\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right]$. Suppose that the analysis filters H_0 and H_1 satisfy the following conditions: 1) H_0 has even-even quadrantal centrosymmetry with group delay $\boldsymbol{d}_0=[d_{0,0}\quad d_{0,1}]^T$, $d_{0,0}\in\mathbb{Z}$ and $d_{0,1}\in\frac{1}{2}\mathbb{Z}_{odd}$; 2) H_1 has even-odd quadrantal centrosymmetry with group delay $\boldsymbol{d}_1=[d_{1,0}\quad d_{1,1}]^T$, $d_{1,0}\in\mathbb{Z}$ and $d_{1,1}\in\frac{1}{2}\mathbb{Z}_{odd}$; 3) $\boldsymbol{d}_0-\boldsymbol{d}_1\in\mathrm{LAT}(\boldsymbol{M})$. In this case, the subband output y_0 can be completely characterized by N_0 samples with indices $\boldsymbol{n}=[n_0\quad n_1]^T$ given by*

$$\begin{gathered}\left\lceil\frac{d_{0,0}+d_{0,1}-\frac{1}{2}}{2}\right\rceil\le n_0\le\left\lfloor\frac{d_{0,0}+d_{0,1}+L_0+L_1-\frac{3}{2}}{2}\right\rfloor\ \textit{and}\\ \max\{d_{0,0}-n_0,n_0-d_{0,1}-L_1+\tfrac{1}{2}\}\le n_1\le\min\{d_{0,0}+L_0-n_0-1,n_0-d_{0,1}+\tfrac{1}{2}\};\end{gathered}\tag{3.37}$$

y_1 can be completely characterized by N_1 samples given by

$$\begin{gathered}\left\lceil\frac{d_{1,0}+d_{1,1}+\frac{1}{2}}{2}\right\rceil\le n_0\le\left\lfloor\frac{d_{1,0}+d_{1,1}+L_0+L_1-\frac{5}{2}}{2}\right\rfloor\ \textit{and}\\ \max\{d_{1,0}-n_0,n_0-d_{1,1}-L_1+\tfrac{3}{2}\}\le n_1\le\min\{d_{1,0}+L_0-n_0-1,n_0-d_{1,1}-\tfrac{1}{2}\};\end{gathered}\tag{3.38}$$

and $N_0+N_1=L_0L_1$, i.e., the transform is nonexpansive.

This theorem can be proved in a way similar to that of Theorem 3.9 with interchanged horizontal and vertical indices. An example of this type of PR filter bank is given by the analysis and synthesis filter transfer functions

$$H_0(z_0,z_1)=\tfrac{1}{2}(1+z_1),H_1(z_0,z_1)=1-z_1,G_0(z_0,z_1)=1+z_1^{-1},\text{ and }G_1(z_0,z_1)=\tfrac{1}{2}\left(z_1^{-1}-1\right).$$

3.6.4 Type-4 PR Quincunx Filter Banks

We have discussed three out of the four types of quadrantally centrosymmetric PR quincunx filter banks in the preceding sections. Now we show that for the last type of PR quincunx filter bank, where H_1 is odd-odd

quadrantally centrosymmetric with group delay $\boldsymbol{d}_1 \in \frac{1}{2}\mathbb{Z}^2_{odd}$, no nonexpansive transform can be constructed for any of the symmetric extension schemes defined in (3.8). In this case, the total number of independent samples in the subband sequences y_0 and y_1 combined is always greater than that in the original finite-extent sequence $\tilde{x}$. We prove this statement in a constructive way. In what follows, we refer to the definitions of sequences in Figure 3.1(a).

For this type of PR quincunx filter bank, the analysis filters H_0 and H_1 have even-even and odd-odd quadrantal centrosymmetry, respectively, and their group delays satisfy that $\boldsymbol{d}_0, \boldsymbol{d}_1 \in \frac{1}{2}\mathbb{Z}^2_{odd}$. In order for the filter bank to be nonexpansive, the subband sequences y_0 and y_1 are required to have four-fold symmetry and periodicity. It follows that the symmetry center of the extended sequence x satisfies $\boldsymbol{c}_x \in \frac{1}{2}\mathbb{Z}^2_{odd}$. From Lemma 3.1, we know the original input $\tilde{x}$ should be extended using the type-4 symmetric extension defined in (3.8). Then, x is quadrantally centrosymmetric about $\boldsymbol{c}_x = \left[-\frac{1}{2} \; -\frac{1}{2}\right]^T$, and $\begin{bmatrix} 2L_0 & 0 \\ 0 & 2L_1 \end{bmatrix}$-periodic. In this case, the subband sequence y_0 has rotated quadrantal centrosymmetry with symmetry center $\boldsymbol{c}_{y_0} \in \mathbb{Z}^2$ and periodicity with $\boldsymbol{M}^{-1}\boldsymbol{P} = \begin{bmatrix} L_0 & L_1 \\ L_0 & -L_1 \end{bmatrix}$. Therefore, y_0 can be completely characterized by samples with indices $\boldsymbol{n} = [n_0 \quad n_1]^T$ given by

$$\boldsymbol{M}\boldsymbol{n} \in \{d_{0,0} - \tfrac{1}{2}, d_{0,0} + \tfrac{1}{2}, \ldots, d_{0,0} + L_0 - \tfrac{1}{2}\} \times \{d_{0,1} - \tfrac{1}{2}, d_{0,1} + \tfrac{1}{2}, \ldots, d_{0,1} + L_1 - \tfrac{1}{2}\},$$

and the number of independent samples in y_0 is

$$N_0 = \begin{cases} \frac{1}{2}(L_0+1)(L_1+1) & \text{for } (L_0+1)(L_1+1) \text{ even} \\ \frac{1}{2}\left[(L_0+1)(L_1+1)+1\right] & \text{for } (L_0+1)(L_1+1) \text{ odd, } (\boldsymbol{d}_0+\boldsymbol{c}_x) \in \mathrm{LAT}(\boldsymbol{M}) \\ \frac{1}{2}\left[(L_0+1)(L_1+1)-1\right] & \text{for } (L_0+1)(L_1+1) \text{ odd, } (\boldsymbol{d}_0+\boldsymbol{c}_x) \notin \mathrm{LAT}(\boldsymbol{M}). \end{cases}$$

Similarly, for the second channel, since u_1 has odd-odd type quadrantal centrosymmetry, y_1 can be completely characterized by N_1 samples with indices $\boldsymbol{n} = [n_0 \quad n_1]^T$ given by

$$\boldsymbol{M}\boldsymbol{n} \in \{d_{1,0} + \tfrac{1}{2}, d_{1,0} + \tfrac{3}{2}, \ldots, d_{1,0} + L_0 - \tfrac{3}{2}\} \times \{d_{1,1} + \tfrac{1}{2}, d_{1,1} + \tfrac{3}{2}, \ldots, d_{1,1} + L_1 - \tfrac{3}{2}\}, \text{ and}$$

$$N_1 = \begin{cases} \frac{1}{2}(L_0-1)(L_1-1) & \text{for } (L_0-1)(L_1-1) \text{ even} \\ \frac{1}{2}\left[(L_0-1)(L_1-1)+1\right] & \text{for } (L_0-1)(L_1-1) \text{ odd, } (\boldsymbol{d}_1+\boldsymbol{c}_x) \in \mathrm{LAT}(\boldsymbol{M}) \\ \frac{1}{2}\left[(L_0-1)(L_1-1)-1\right] & \text{for } (L_0-1)(L_1-1) \text{ odd, } (\boldsymbol{d}_1+\boldsymbol{c}_x) \notin \mathrm{LAT}(\boldsymbol{M}). \end{cases}$$

We can see that when $(L_0+1)(L_1+1) \in \mathbb{Z}_{even}$, the total number of independent samples in y_0 and y_1 is always L_0L_1+1. Therefore, we cannot construct nonexpansive transforms using this type of PR filter bank.

3.6.5 Octave-Band Decomposition

A weakness of the symmetric extension algorithms introduced in the preceding sections is that they cannot be used in an octave-band decomposition where the filter bank is applied recursively in the lowpass channel. With the proposed algorithms in Theorems 3.8, 3.9, and 3.10, the extent of the lowpass output sequence $\tilde{y}_0$ assumes the shape of a 45°-rotated rectangle. At the second level of an octave-band decomposition, $\tilde{y}_0$ becomes the input sequence to the symmetric extension structure in Figure 3.1(a), while this structure only operates on sequences defined in rectangular regions. Therefore, these algorithms cannot be applied to an octave-band filter bank. For the type-1 symmetric extension algorithm, however, if we constrain the analysis filters more tightly, the filter bank can still be used to construct nonexpansive transforms with an octave-band decomposition, as demonstrated by the below theorem.

Theorem 3.11 (Type-1 symmetric extension algorithm for octave-band decompositions)**.** *Consider the two-level octave-band filter bank shown in Figure 3.12, where $\tilde{x}$ is defined on the rectangular region $\{0,1,\ldots,L_0-1\}\times\{0,1,\ldots,L_1-1\}$ and x is the type-1 symmetric extension of $\tilde{x}$ as given by (3.8). If H_0 and H_1 have both quadrantal and rotated-quadrantal centrosymmetry, and their group delays $\boldsymbol{d}_0$ and $\boldsymbol{d}_1$ satisfy that $\boldsymbol{d}_0 = [d_{0,0} \quad d_{0,1}]^T \in \mathrm{LAT}(\boldsymbol{M})$ and $\boldsymbol{d}_1 = [d_{1,0} \quad d_{1,1}]^T \in \mathbb{Z}^2 \setminus \mathrm{LAT}(\boldsymbol{M})$, respectively, then the subband output y_{00} can be completely characterized by N_{00} samples with indices $\boldsymbol{n} = [n_0 \quad n_1]^T$ given by*

$$d_{0,0} + \left\lceil \frac{d_{0,1}}{2} \right\rceil \le n_0 \le d_{0,0} + \left\lfloor \frac{d_{0,1}}{2} \right\rfloor + \left\lfloor \frac{L_0-1}{2} \right\rfloor \text{ and } \left\lceil \frac{d_{0,0}}{2} \right\rceil \le n_1 \le \left\lfloor \frac{d_{0,0}}{2} \right\rfloor + \left\lfloor \frac{L_1-1}{2} \right\rfloor; \tag{3.39}$$

y_{01} can be completely characterized by N_{01} samples with indices $\boldsymbol{n} = [n_0 \quad n_1]^T$ given by

$$\begin{aligned} \left\lceil \frac{d_{0,0}+d_{1,0}+d_{1,1}}{2} \right\rceil \le n_0 \le \left\lfloor \frac{d_{0,0}+d_{1,0}+d_{1,1}}{2} \right\rfloor + \left\lfloor \frac{L_0-1}{2} \right\rfloor, \text{ and} \\ \left\lceil \frac{d_{0,1}+d_{1,0}-d_{1,1}}{2} \right\rceil \le n_1 \le \left\lfloor \frac{d_{0,1}+d_{1,0}-d_{1,1}}{2} \right\rfloor + \left\lfloor \frac{L_1-1}{2} \right\rfloor; \end{aligned} \tag{3.40}$$

y_1 can be completely characterized by N_1 samples with indices $\boldsymbol{n} = [n_0 \quad n_1]^T$ given by (3.26); and $N_{00} + N_{01} + N_1 = L_0 L_1$, i.e., the transform is nonexpansive.

Proof. From Theorem 3.8, we know that y_0 has rotated quadrantal centrosymmetry about $\boldsymbol{c}_{y_0} = \boldsymbol{M}^{-1}\boldsymbol{d}_0 \in \mathbb{Z}^2$ and $\boldsymbol{P}_{y_0}$-periodicity with $\boldsymbol{P}_{y_0} = \boldsymbol{M}^{-1}\boldsymbol{P} = \begin{bmatrix} L_0-1 & L_1-1 \\ L_0-1 & -L_1+1 \end{bmatrix} \in \mathrm{LAT}(\boldsymbol{M})$, and its independent samples are located in a finite region given by (3.25).

At the second decomposition level, since H_0 has rotated quadrantal centrosymmetry, from Lemma 3.2, u_{00} is also rotated quadrantally centrosymmetric with symmetry center $\boldsymbol{c}_{u_{00}} = \boldsymbol{c}_{y_0} + \boldsymbol{d}_0 \in \mathbb{Z}^2$ and periodic with periodicity matrix $\boldsymbol{P}_{y_0} \in \mathrm{LAT}(\boldsymbol{M})$. From Lemmas 3.4 and 3.5, y_{00} has quadrantal centrosymmetry with

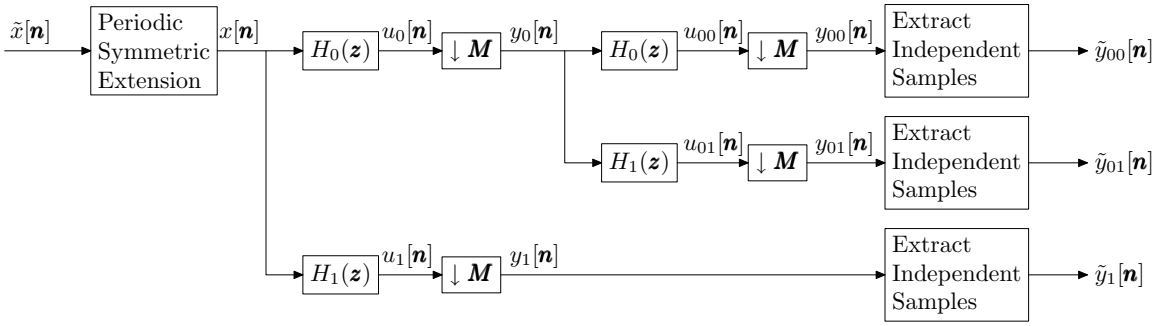


Figure 3.12: Symmetric extension for two-level octave-band filter banks.

symmetry center $\boldsymbol{c}_{y_{00}}$ and $\boldsymbol{P}_{y_{00}}$-periodicity, where

$$\begin{aligned} \boldsymbol{c}_{y_{00}} = \boldsymbol{M}^{-1}\boldsymbol{c}_{u_{00}} = \boldsymbol{M}^{-2}\boldsymbol{d}_0 + \boldsymbol{d}_0 = \begin{bmatrix} d_{0,0} + \frac{1}{2}d_{0,1} & \frac{1}{2}d_{0,0} \end{bmatrix}^T \text{ and} \\ \boldsymbol{P}_{y_{00}} = \boldsymbol{M}^{-2}\boldsymbol{P} = \begin{bmatrix} L_0-1 & 0 \\ 0 & L_1-1 \end{bmatrix}. \end{aligned} \tag{3.41}$$

Therefore, y_{00} can be completely characterized by samples with indices $\boldsymbol{n}$ given by

$$\lceil \boldsymbol{c}_{y_{00}} \rceil \le \boldsymbol{n} \le \lfloor \boldsymbol{c}_{y_{00}} \rfloor + \left\lfloor \tfrac{1}{2} \begin{bmatrix} L_0-1 & L_1-1 \end{bmatrix}^T \right\rfloor. \tag{3.42}$$

Substituting (3.41) into (3.42), we obtain the indices of the independent samples in y_{00} as in (3.39). The number of the independent samples is calculated as

$$N_{00} = \begin{cases} \left\lfloor \frac{L_0+1}{2} \right\rfloor \left\lfloor \frac{L_1+1}{2} \right\rfloor & \text{for } \boldsymbol{d}_0 \in \mathbb{Z}^2_{even} \\ \left\lceil \frac{L_0-1}{2} \right\rceil \left\lceil \frac{L_1-1}{2} \right\rceil & \text{for } \boldsymbol{d}_0 \in \mathbb{Z}^2_{odd}. \end{cases} \tag{3.43}$$

Next we consider the channel with subband sequence y_{01}. Similarly, y_{01} has quadrantal centrosymmetry with symmetry center $\boldsymbol{c}_{y_{01}}$ and $\boldsymbol{P}_{y_{01}}$-periodicity, where

$$\boldsymbol{c}_{y_{01}} = \begin{bmatrix} \frac{1}{2}(d_{0,0} + d_{1,0} + d_{1,1}) & \frac{1}{2}(d_{0,1} + d_{1,0} - d_{1,1}) \end{bmatrix}^T \text{ and } \boldsymbol{P}_{y_{01}} = \boldsymbol{M}^{-2}\boldsymbol{P} = \begin{bmatrix} L_0-1 & 0 \\ 0 & L_1-1 \end{bmatrix}.$$

Since $\boldsymbol{d}_1 \in \mathbb{Z}^2 \setminus \mathrm{LAT}(\boldsymbol{M})$, we have $d_{1,0} + d_{1,1} \in \mathbb{Z}_{odd}$. Therefore, y_{01} can be completely characterized by samples with indices $\boldsymbol{n}$ given by

$$\lceil \boldsymbol{c}_{y_{01}} \rceil \le \boldsymbol{n} \le \lfloor \boldsymbol{c}_{y_{01}} \rfloor + \left\lfloor \tfrac{1}{2} \begin{bmatrix} L_0-1 & L_1-1 \end{bmatrix}^T \right\rfloor.$$

The preceding inequality can be rewritten as (3.40). Since $\boldsymbol{d}_1 \in \mathbb{Z}^2 \setminus \mathrm{LAT}(\boldsymbol{M})$, $d_{1,0} + d_{1,1} \in \mathbb{Z}_{odd}$, the number

of independent samples in y_{01} can be calculated as

$$N_{01} = \begin{cases} \left\lceil \frac{L_0-1}{2} \right\rceil \left\lceil \frac{L_1-1}{2} \right\rceil & \text{for } \boldsymbol{d}_0 \in \mathbb{Z}^2_{even} \\ \left\lfloor \frac{L_0+1}{2} \right\rfloor \left\lfloor \frac{L_1+1}{2} \right\rfloor & \text{for } \boldsymbol{d}_0 \in \mathbb{Z}^2_{odd}. \end{cases} \tag{3.44}$$

Combining (3.43) and (3.44), we obtain

$$\begin{aligned} N_{00} + N_{01} &= \left\lceil \frac{L_0-1}{2} \right\rceil \left\lceil \frac{L_1-1}{2} \right\rceil + \left\lfloor \frac{L_0+1}{2} \right\rfloor \left\lfloor \frac{L_1+1}{2} \right\rfloor \\ &= \begin{cases} \frac{1}{2} L_0 L_1 & \text{for } L_0 L_1 \in \mathbb{Z}_{even} \\ \frac{1}{2}\left(L_0 L_1 + 1\right) & \text{for } L_0 L_1 \in \mathbb{Z}_{odd}. \end{cases} \end{aligned} \tag{3.45}$$

From Theorem 3.8, y_1 can be characterized by N_1 samples with indices given by (3.26), and

$$N_1 = \begin{cases} \frac{1}{2} L_0 L_1 & \text{for } L_0 L_1 \in \mathbb{Z}_{even} \\ \frac{1}{2}(L_0 L_1 - 1) & \text{for } L_0 L_1 \in \mathbb{Z}_{odd}. \end{cases} \tag{3.46}$$

Combining (3.45) and (3.46), we have $N_{00} + N_{01} + N_1 = L_0 L_1$. □

The independent samples in the lowpass subband sequence are located in a rectangular region. Therefore, if we cascade this two-level structure, with the original structure in Figure 3.1(a) cascaded as the last step, in the form of an octave-band quincunx filter bank with any number of decomposition levels, a nonexpansive transform can still be obtained.

Consider the filter bank from Example 3.2. The analysis filters are given by the impulse responses

$$h_0[\boldsymbol{n}] \sim \frac{1}{32} \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & 4 & -2 & 0 \\ -1 & 4 & \boxed{28} & 4 & -1 \\ 0 & -2 & 4 & -2 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad h_1[\boldsymbol{n}] \sim \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & \boxed{\text{-1}} \\ 0 & -1 & 0 \end{bmatrix},$$

and group delays $\boldsymbol{d}_0 = [0 \;\; 0]^T$ and $\boldsymbol{d}_1 = [-1 \;\; 0]^T$. The analysis filters satisfy the conditions in Theorem 3.11. Therefore, this filter bank can be used in conjunction with type-1 symmetric extension to construct nonexpansive transform for an octave-band decomposition with an arbitrary number of levels.

3.7 Summary

In this chapter, we have shown four ways to extend a 2D finite-extent input sequence of a quincunx filter bank to an infinite-extent periodic symmetric sequence, and discussed how the periodicity and symmetry

properties of the extended sequence can be preserved by the operations of the quincunx filter banks. Then, we have proposed new algorithms to construct nonexpansive transforms for three types of quadrantally centrosymmetric PR quincunx filter banks. At last, we have shown how the type-1 algorithm can be extended for use with multi-level octave-band decompositions. These symmetric extension schemes are potentially useful in applications that process finite-extent sequences using quincunx filter banks.

Chapter 4

Optimal Design of Quincunx Filter Banks

4.1 Overview

In this chapter, new optimization-based techniques are proposed for the design of high-performance quincunx filter banks for the application of image coding. These new methods are used to build linear-phase PR systems with high coding gain, good analysis/synthesis filter frequency responses, and certain prescribed vanishing moment properties. Examples of filter banks designed using these techniques are presented and shown to be highly effective for image coding. The material in this chapter has also been partly presented in [39].

4.2 Introduction

Filter banks have proven to be a highly effective tool for image coding applications [40]. Compared to the case with 1D filter banks, the nonseparable 2D filter banks are much more difficult to design and far fewer effective methods have been proposed. In image coding applications, one typically desires filter banks to have PR, linear phase, high coding gain, good frequency selectivity, and satisfactory vanishing moment properties. The PR property is desirable as it facilitates the construction of a lossless compression system. That is, if the system possesses this property, then in absence of quantization, the original image can be precisely reproduced from the subband coefficients. The linear phase property is crucial to avoiding phase distortion. High coding gain often leads to efficient energy compaction of images. The presence of vanishing moments

helps reduce the number of nonzero coefficients in the highpass subbands, and good frequency selectivity can diminish aliasing in the subband signals. Designing nonseparable 2D filter banks with all of the preceding properties is an extremely challenging task. In this chapter, several new optimization-based techniques are proposed for constructing quincunx filter banks with all of the aforementioned desirable characteristics.

The remainder of this chapter is organized as follows. Section 4.3 presents a parametrization of linear-phase PR quincunx filter banks based on the lifting scheme. We choose the lifting-based parametrization because the PR and linear-phase properties can be structurally imposed on the filter bank, which simplifies the later optimization algorithms. Then, an optimal design algorithm for quincunx filter banks with two lifting steps is proposed in Section 4.4. In this section, we show how coding gain, vanishing moments, and frequency selectivity are related to the lifting-filter coefficients, and discuss how the design of quincunx filter banks with all of the desired characteristics can be formulated as a constrained optimization problem. In Section 4.5, a scheme is proposed for the design of filter banks with more than two lifting steps. In this case, the relationships between the desirable properties and the lifting filter coefficients become more complicated than in the two-lifting-step case. We explain how the design can be formulated as a similar optimization problem. In Section 4.6, a suboptimal design algorithm is proposed for filter banks with at least three lifting steps using the above two algorithms. Several design examples are presented in Section 4.7 and their effectiveness for image coding is demonstrated in Section 4.8. Finally, the results obtained in this chapter are summarized in Section 4.9.

4.3 Lifting Parametrization of Linear-Phase PR Quincunx Filter Banks

For filter banks in image processing applications, the PR and linear phase properties are often highly desirable. In this section, we introduce a parametrization of a subset of linear-phase PR quincunx filter banks based on the lifting scheme. The use of the lifting-based parametrization is helpful in several respects. First, as discussed in Section 2.3.6, the PR condition is automatically satisfied by such a parametrization. Furthermore, the linear-phase condition can be imposed with relative ease. This eliminates the need for additional cumbersome constraints for PR and linear phase during optimization. Lastly, reversible integer-to-integer mappings can be readily constructed from the lifting realization [33]. Recall that with the lifting structure, the synthesis filters are completely determined by the analysis filters. Therefore, in what follows, we only consider the analysis side of the filter bank.

The canonical form of the analysis side of a quincunx filter bank with analysis filters H_0 and H_1 is shown in Figure 4.1. The lifting realization of the analysis side with 2λ lifting filters $\{\mathrm{A}_k\}$ is shown in Figure 4.2.

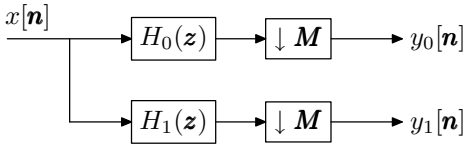


Figure 4.1: Analysis side of a quincunx filter bank.

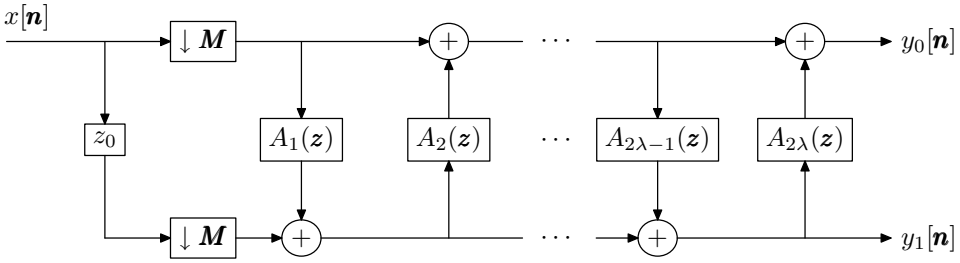


Figure 4.2: Analysis side of the lifting realization of a quincunx filter bank.

Without loss of generality, we assume none of the transfer functions $\{A_k(\boldsymbol{z})\}$ are identically zero, except possibly $A_1(\boldsymbol{z})$ and $A_{2\lambda}(\boldsymbol{z})$. Using this lifting structure, the linear-phase condition can be satisfied with a prudent choice of lifting filters $\{\mathrm{A}_k\}$. In particular, in what follows, we will introduce three lifting-based parameterizations of PR quincunx filter banks that result in linear-phase analysis/synthesis filters. The filter banks associated with these parameterizations are henceforth referred to as type-1, type-2, and type-3 filter banks.

4.3.1 Type-1 Filter Banks

The first lifting parameterization results in symmetric analysis filters H_0 and H_1 with group delays on the integer lattice $\mathbb{Z}^2$. The detail regarding the choice of the lifting filters $\{\mathrm{A}_k\}$ in Figure 4.2 is given by the theorem below.

Theorem 4.1 (Construction of type-1 filter banks)**.** *Consider a quincunx filter bank constructed from the lifting scheme with 2λ lifting filters as shown in Figure 4.2, where none of the transfer functions $\{A_k(\boldsymbol{z})\}$ are identically zero, except possibly $A_1(\boldsymbol{z})$ and $A_{2\lambda}(\boldsymbol{z})$. If the lifting filters $\{A_k\}$ are symmetric, and the filter A_k has group delay $\boldsymbol{c}_k$ satisfying*

$$\boldsymbol{c}_k = (-1)^k \left[\, \tfrac{1}{2} \;\; \tfrac{1}{2} \,\right]^T,$$

then the analysis filters H_0 and H_1 are symmetric with group delays $[0 \;\; 0]^T$ and $[-1 \;\; 0]^T$, respectively.

Proof. Let the analysis filters associated with the first k lifting steps be denoted as $\left\{\mathrm{H}_i^{(k)}\right\}$ for $i = 0,1$. We prove this theorem by induction on λ. Consider $\lambda = 0$. In this case, $H_0^{(0)}(\boldsymbol{z}) = 1$ and $H_1^{(0)}(\boldsymbol{z}) = z_0$. Thus, the analysis filters $\mathrm{H}_0^{(0)}$ and $\mathrm{H}_1^{(0)}$ have symmetric linear phase with group delays $[0\ 0]^T$ and $[-1\ 0]^T$, respectively. Therefore, the theorem holds for $\lambda = 0$.

Now, assume that the theorem holds for $\lambda = k$ (i.e., $\mathrm{H}_0^{(2k)}$ and $\mathrm{H}_1^{(2k)}$ have symmetric linear phase with group delays $[0\ 0]^T$ and $[-1\ 0]^T$, respectively). For $\lambda = k+1$, from (2.28), we have

$$\begin{bmatrix} H_{0,0}^{(2k+2)}(\boldsymbol{z}) & H_{0,1}^{(2k+2)}(\boldsymbol{z}) \\ H_{1,0}^{(2k+2)}(\boldsymbol{z}) & H_{1,1}^{(2k+2)}(\boldsymbol{z}) \end{bmatrix} = \begin{bmatrix} 1 & A_{2k+2}(\boldsymbol{z}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A_{2k+1}(\boldsymbol{z}) & 1 \end{bmatrix} \begin{bmatrix} H_{0,0}^{(2k)}(\boldsymbol{z}) & H_{0,1}^{(2k)}(\boldsymbol{z}) \\ H_{1,0}^{(2k)}(\boldsymbol{z}) & H_{1,1}^{(2k)}(\boldsymbol{z}) \end{bmatrix}.$$

Using the above equation, the analysis filter transfer functions $\left\{H_i^{(2k+2)}(\boldsymbol{z})\right\}$ can be computed as

$$H_1^{(2k+2)}(\boldsymbol{z}) = H_1^{(2k+1)}(\boldsymbol{z}) = H_1^{(2k)}(\boldsymbol{z}) + A_{2k+1}(\boldsymbol{z}^{\boldsymbol{M}})H_0^{(2k)}(\boldsymbol{z}) \quad \text{and} \tag{4.1}$$

$$H_0^{(2k+2)}(\boldsymbol{z}) = H_0^{(2k+1)}(\boldsymbol{z}) + A_{2k+2}(\boldsymbol{z}^{\boldsymbol{M}})H_1^{(2k+1)}(\boldsymbol{z}) = H_0^{(2k)}(\boldsymbol{z}) + A_{2k+2}(\boldsymbol{z}^{\boldsymbol{M}})H_1^{(2k+1)}(\boldsymbol{z}). \tag{4.2}$$

In (4.1), since the lifting filter A_{2k+1} has symmetric linear phase with group delay $\boldsymbol{c}_{2k+1} = \left[-\frac{1}{2}\ -\frac{1}{2}\right]^T$, from Lemma 3.7, $A_{2k+1}(\boldsymbol{z}^{\boldsymbol{M}})$ has symmetric linear phase with group delay $\boldsymbol{M}\boldsymbol{c}_{2k+1} = [-1\quad 0]^T$. By assumption, $H_0^{(2k)}(\boldsymbol{z})$ is symmetric with group delay $[0\quad 0]^T$. Then, from Lemma 3.2, $A_{2k+1}(\boldsymbol{z}^{\boldsymbol{M}})H_0^{(2k)}(\boldsymbol{z})$ has symmetric linear phase with group delay $[0\quad 0]^T + [-1\quad 0]^T = [-1\quad 0]^T$. Therefore, $\mathrm{H}_1^{(2k+2)}$ has linear phase with group delay $[-1\quad 0]^T$. Similarly, in (4.2), $A_{2k+2}(\boldsymbol{z}^{\boldsymbol{M}})$ has symmetric linear phase with group delay $\boldsymbol{M}\boldsymbol{c}_{2k+2} = [1\quad 0]^T$. Thus, it follows that $\mathrm{H}_0^{(2k+2)}$ has linear phase with group delay $[0\quad 0]^T$. Therefore, the theorem holds for $\lambda = k+1$. This completes the proof. □

We henceforth refer to the filter banks constructed using Theorem 4.1 as **type-1 filter banks**. Below we show an example of a type-1 filter bank. This filter bank is proposed in [6, 34] and previously discussed in Example 3.2 in Section 3.6.1.

Example 4.1 (Type-1 filter bank)**.** Consider the filter bank from Example 3.2 with the analysis filter impulse responses

$$h_0[\boldsymbol{n}] \sim \frac{1}{32}\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & 4 & -2 & 0 \\ -1 & 4 & \boxed{28} & 4 & -1 \\ 0 & -2 & 4 & -2 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad h_1[\boldsymbol{n}] \sim \frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & \boxed{\text{-1}} \\ 0 & -1 & 0 \end{bmatrix}.$$

This filter bank can be realized using two lifting steps. The impulse responses of the lifting filters A_1 and A_2 are given by

$$a_1[\boldsymbol{n}] \sim -\frac{1}{4}\begin{bmatrix} 1 & \boxed{1} \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad a_2[\boldsymbol{n}] \sim \frac{1}{8}\begin{bmatrix} 1 & 1 \\ \boxed{1} & 1 \end{bmatrix},$$

and the group delays of A_1 and A_2 are $\boldsymbol{c}_1 = \left[-\frac{1}{2}\ -\frac{1}{2}\right]^T$ and $\boldsymbol{c}_2 = \left[\frac{1}{2}\ \frac{1}{2}\right]^T$, respectively. □

In order to allow the frequency responses of the lifting and analysis filters to be expressed in a compact form, we introduce some notation in this regard. Since the lifting filter A_k has linear phase with group delay $\boldsymbol{c}_k = (-1)^k\left[\frac{1}{2}\ \frac{1}{2}\right]^T$, the support region of A_k is a rectangle of size $2l_{k,0} \times 2l_{k,1}$ for some $l_{k,0}, l_{k,1} \in \mathbb{Z}^+$, and the number of independent coefficients in A_k is $2l_{k,0}l_{k,1}$. Let $\boldsymbol{a}_k$ be a vector containing the independent coefficients of A_k. Then, there are $2l_{k,0}l_{k,1}$ elements in $\boldsymbol{a}_k$ indexed from 0 to $2l_{k,0}l_{k,1}-1$.

For an odd-indexed lifting filter A_{2k-1}, its support region can be expressed as $\{-l_{2k-1,0}, -l_{2k-1,0}+1, \ldots, l_{2k-1,0}-1\} \times \{-l_{2k-1,1}, -l_{2k-1,1}+1, \ldots, l_{2k-1,1}-1\}$. The coefficient vector $\boldsymbol{a}_{2k-1}$ of A_{2k-1} is then defined as

$$\boldsymbol{a}_{2k-1} = \Big[a_{2k-1}[0,-l_{2k-1,1}] \quad a_{2k-1}[0,-l_{2k-1,1}+1] \quad \cdots \quad a_{2k-1}[l_{2k-1,0}-1, l_{2k-1,1}-1]\Big]^T. \tag{4.3}$$

That is, the nth element of $\boldsymbol{a}_{2k-1}$ is defined as $a_{2k-1}[n_0, n_1]$ with n_0 and n_1 computed by

$$\begin{aligned} n_0 &= \lfloor n/(2l_{2k-1,1})\rfloor \in \{0,1,\ldots,l_{2k-1,0}-1\} \quad \text{and} \\ n_1 &= \mathrm{mod}(n, 2l_{2k-1,1}) - l_{2k-1,1} \in \{-l_{12k-1,1}, -l_{2k-1,1}+1, \ldots, l_{2k-1,1}-1\}. \end{aligned} \tag{4.4}$$

Since A_{2k-1} has symmetric linear phase, the frequency response of A_{2k-1} can be written from (2.4) as

$$\begin{aligned} \hat{a}_{2k-1}(\boldsymbol{\omega}) &= e^{-j\boldsymbol{\omega}^T\boldsymbol{c}_{2k-1}} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_{2k-1}[\boldsymbol{n}] \cos\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{c}_{2k-1})\right] \\ &= 2e^{j\frac{1}{2}(\omega_0+\omega_1)} \sum_{n_0=0}^{l_{2k-1,0}-1} \sum_{n_1=-l_{2k-1,1}}^{l_{2k-1,1}-1} a_{2k-1}[n_0,n_1] \cos\left[\omega_0(n_0+\tfrac{1}{2}) + \omega_1(n_1+\tfrac{1}{2})\right]. \end{aligned}$$

In the upsampled domain, $\hat{a}_{2k-1}(\boldsymbol{M}^T\boldsymbol{\omega})$ can be expressed as

$$\hat{a}_{2k-1}(\boldsymbol{M}^T\boldsymbol{\omega}) = 2e^{j\omega_0} \sum_{n_0=0}^{l_{2k-1,0}-1} \sum_{n_1=-l_{2k-1,1}}^{l_{2k-1,1}-1} a_{2k-1}[n_0,n_1] \cos\left[\omega_0(n_0+n_1+1) + \omega_1(n_0-n_1)\right].$$

With the notation $\boldsymbol{a}_{2k-1}$, $\hat{a}_{2k-1}(\boldsymbol{M}^T\boldsymbol{\omega})$ can be compactly written as

$$\hat{a}_{2k-1}(\boldsymbol{M}^T\boldsymbol{\omega}) = e^{j\omega_0}\boldsymbol{a}_{2k-1}^T\boldsymbol{v}_{2k-1}, \tag{4.5}$$

where $\boldsymbol{v}_{2k-1}$ is a vector of $2l_{2k-1,0}l_{2k-1,1}$ elements indexed from 0 to $2l_{2k-1,0}l_{2k-1,1}-1$, and the nth element of $\boldsymbol{v}_{2k-1}$ is

$$\boldsymbol{v}_{2k-1}[n] = 2\cos\left[\omega_0(n_0+n_1+1)+\omega_1(n_0-n_1)\right] \tag{4.6}$$

with n_0 and n_1 given by (4.4).

Similarly, for an even-indexed lifting filter A_{2k}, its support region is $\{-l_{2k,0}+1,-l_{2k,0}+2,\ldots,l_{2k,0}\}\times\{-l_{2k,1}+1,-l_{2k,1}+2,\ldots,l_{2k,1}\}$. The coefficient vector $\boldsymbol{a}_{2k}$ of A_{2k} is then defined as

$$\boldsymbol{a}_{2k} = \begin{bmatrix} a_{2k}[1,-l_{2k,1}+1] & a_{2k}[2,-l_{2k,1}+2] & \cdots & a_{2k}[l_{2k,0},l_{2k,1}] \end{bmatrix}^T. \tag{4.7}$$

That is, the nth element of $\boldsymbol{a}_{2k}$ is $a_{2k}[n_0,n_1]$ with n_0 and n_1 given by

$$\begin{aligned} n_0 &= \left\lfloor n/(2l_{2k,1})\right\rfloor + 1 \in \{1,2,\ldots,l_{2k,0}\} \quad \text{and} \\ n_1 &= \mathrm{mod}(n,2l_{2k,1}) - l_{2k,1} + 1 \in \{-l_{2k,1}+1,-l_{2k,1}+2,\ldots,l_{2k,1}\}, \end{aligned} \tag{4.8}$$

respectively. The frequency response $\hat{a}_{2k}(\boldsymbol{\omega})$ of A_{2k} is computed by

$$\hat{a}_{2k}(\boldsymbol{\omega}) = 2e^{-j\frac{1}{2}(\omega_0+\omega_1)}\sum_{n_0=1}^{l_{2k,0}}\sum_{n_1=1-l_{2k,1}}^{l_{2k,1}} a_{2k}[n_0,n_1]\cos\left[\omega_0(n_0-\tfrac{1}{2})+\omega_1(n_1-\tfrac{1}{2})\right],$$

and in the upsampled domain $\hat{a}_{2k}(\boldsymbol{M}^T\boldsymbol{\omega})$ can be expressed as

$$\hat{a}_{2k}(\boldsymbol{M}^T\boldsymbol{\omega}) = e^{-j\omega_0}\boldsymbol{a}_{2k}^T\boldsymbol{v}_{2k}. \tag{4.9}$$

In (4.9), $\boldsymbol{v}_{2k}$ is a vector of $2l_{2k,0}l_{2k,1}$ elements indexed from 0 to $2l_{2k,0}l_{2k,1}-1$, and the nth element of $\boldsymbol{v}_{2k}$ is defined as

$$\boldsymbol{v}_{2k}[n] = 2\cos\left[\omega_0(n_0+n_1-1)+\omega_1(n_0-n_1)\right] \tag{4.10}$$

with n_0 and n_1 given by (4.8).

We rewrite (2.28) and (2.16) in the Fourier domain as

$$\boldsymbol{H}_p(\boldsymbol{\omega}) = \begin{bmatrix} \hat{h}_{0,0}(\boldsymbol{\omega}) & \hat{h}_{0,1}(\boldsymbol{\omega}) \\ \hat{h}_{1,0}(\boldsymbol{\omega}) & \hat{h}_{1,1}(\boldsymbol{\omega}) \end{bmatrix} = \prod_{k=1}^{\lambda}\left(\begin{bmatrix} 1 & \hat{a}_{2k}(\boldsymbol{\omega}) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ \hat{a}_{2k-1}(\boldsymbol{\omega}) & 1 \end{bmatrix}\right) \text{ and} \tag{4.11}$$

$$\begin{bmatrix} \hat{h}_0(\boldsymbol{\omega}) \\ \hat{h}_1(\boldsymbol{\omega}) \end{bmatrix} = \begin{bmatrix} \hat{h}_{0,0}\left(\boldsymbol{M}^T\boldsymbol{\omega}\right) & \hat{h}_{0,1}\left(\boldsymbol{M}^T\boldsymbol{\omega}\right) \\ \hat{h}_{1,0}\left(\boldsymbol{M}^T\boldsymbol{\omega}\right) & \hat{h}_{1,1}\left(\boldsymbol{M}^T\boldsymbol{\omega}\right) \end{bmatrix}\begin{bmatrix} 1 \\ e^{j\omega_0} \end{bmatrix}, \tag{4.12}$$

respectively. Substituting (4.11), (4.5) and (4.9) into (4.12), we obtain the frequency responses of the analysis filters as

$$\begin{bmatrix} \hat{h}_0(\boldsymbol{\omega}) \\ \hat{h}_1(\boldsymbol{\omega}) \end{bmatrix} = \prod_{k=1}^{\lambda}\left(\begin{bmatrix} 1 & e^{-j\omega_0}\boldsymbol{a}_{2k}^T\boldsymbol{v}_{2k} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ e^{j\omega_0}\boldsymbol{a}_{2k-1}^T\boldsymbol{v}_{2k-1} & 1 \end{bmatrix}\right)\begin{bmatrix} 1 \\ e^{j\omega_0} \end{bmatrix}. \tag{4.13}$$

We further define a vector $\boldsymbol{x}$ containing all of the independent coefficients $\{\boldsymbol{a}_k\}$ of the lifting filters $\{\mathrm{A}_k\}$ as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1^T & \boldsymbol{a}_2^T & \cdots & \boldsymbol{a}_{2\lambda}^T \end{bmatrix}^T. \tag{4.14}$$

Thus, $\boldsymbol{x}$ has $l_x = 2\sum_{i=1}^{2\lambda} l_{i,0} l_{i,1}$ elements. Each vector $\boldsymbol{a}_k$ can be expressed in terms of $\boldsymbol{x}$ as

$$\boldsymbol{a}_k = \underbrace{\begin{bmatrix} \boldsymbol{0}_{2l_{k,0}l_{k,1}\times\alpha_0} & \boldsymbol{I}_{2l_{k,0}l_{k,1}} & \boldsymbol{0}_{2l_{k,0}l_{k,1}\times\beta_0} \end{bmatrix}}_{\boldsymbol{E}_k} \boldsymbol{x} = \boldsymbol{E}_k \boldsymbol{x}, \tag{4.15}$$

where $\alpha_0 = 2\sum_{i=1}^{k-1} l_{i,0} l_{i,1}$ and $\beta_0 = 2\sum_{i=k+1}^{2\lambda} l_{i,0} l_{i,1}$. Substituting (4.15) into (4.13), we have

$$\begin{bmatrix} \hat{h}_0(\boldsymbol{\omega}) \\ \hat{h}_1(\boldsymbol{\omega}) \end{bmatrix} = \prod_{k=1}^{\lambda} \left(\begin{bmatrix} 1 & e^{-j\omega_0} \boldsymbol{x}^T \boldsymbol{E}_{2k}^T \boldsymbol{v}_{2k} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ e^{j\omega_0} \boldsymbol{x}^T \boldsymbol{E}_{2k-1}^T \boldsymbol{v}_{2k-1} & 1 \end{bmatrix} \right) \begin{bmatrix} 1 \\ e^{j\omega_0} \end{bmatrix}. \tag{4.16}$$

By expanding the preceding equation, each of the analysis filter frequency responses can be viewed as a polynomial in $\boldsymbol{x}$, the order of which depends on the number of lifting steps.

Now we consider a special case of Theorem 4.1. If, in addition to the conditions in Theorem 4.1, the lifting filters $\{\mathrm{A}_k\}$ have rotated quadrantal centrosymmetry with $S = T = 1$, where S and T are defined in (3.7), then the analysis filters H_0 and H_1 have even-even quadrantal centrosymmetry with group delays $[0 \;\; 0]^T$ and $[-1 \;\; 0]^T$, respectively. The above statement can be easily proved by induction with the help of statement 3 in Lemma 3.7. In this case, the analysis filters H_0 and H_1 satisfy the three conditions in Theorem 3.8. Therefore, this filter bank can be used to build nonexpansive transforms with the type-1 symmetric extension of sequences defined in Theorem 3.8.

4.3.2 Type-2 and Type-3 Filter Banks

We consider now two related lifting-based parameterizations of quincunx filter banks, where the lowpass and highpass analysis filters are symmetric and antisymmetric, respectively. The first parameterization is described by the theorem below.

Theorem 4.2 (Construction of type-2 filter banks)**.** *Consider a quincunx filter bank constructed from the lifting scheme with 2λ lifting filters as illustrated in Figure 4.2, where none of the transfer functions $\{A_k(\boldsymbol{z})\}$ are identically zero, except possibly $A_{2\lambda}(\boldsymbol{z})$. Suppose that the lifting filters $\{A_k\}$ satisfy the following conditions: 1) $A_1(\boldsymbol{z}) = -1$; 2) $A_2(\boldsymbol{z}) = \frac{1}{2}$; and 3) A_k is antisymmetric with group delay $\boldsymbol{c}_k = [0 \;\; 0]^T$ for $k \geq 3$. Then, H_0 is symmetric with group delay $[-\frac{1}{2} \;\; 0]^T$ and H_1 is antisymmetric with group delay $[-\frac{1}{2} \;\; 0]^T$.*

Proof. Let $\mathrm{H}_i^{(k)}$ for $i = 0, 1$ denote the ith analysis filter associated with the first k lifting steps. We prove this theorem by induction on λ. Assume $\lambda = 1$. Then, $H_0^{(2)}(\boldsymbol{z}) = \frac{1}{2}(1 + z_0)$ and $H_1^{(2)}(\boldsymbol{z}) = -1 + z_0$. Therefore, H_0

is symmetric with group delay $\left[-\frac{1}{2}\ 0\right]^T$, and H_1 is antisymmetric with group delay $\left[-\frac{1}{2}\ 0\right]^T$. Therefore, the theorem holds for $\lambda = 1$.

Now, we assume that the theorem holds for $\lambda = k$ (i.e., $\mathrm{H}_0^{(2k)}$ and $\mathrm{H}_1^{(2k)}$ have symmetric and antisymmetric linear phase with group delays $\left[-\frac{1}{2}\ 0\right]^T$, respectively). Then, for $\lambda = k+1$, we have

$$H_1^{(2k+2)}(\boldsymbol{z}) = H_1^{(2k+1)}(\boldsymbol{z}) = H_1^{(2k)}(\boldsymbol{z}) + A_{2k+1}(\boldsymbol{z}^{\boldsymbol{M}})H_0^{(2k)}(\boldsymbol{z}) \quad \text{and} \tag{4.17}$$

$$H_0^{(2k+2)}(\boldsymbol{z}) = H_0^{(2k)}(\boldsymbol{z}) + A_{2k+2}(\boldsymbol{z}^{\boldsymbol{M}})H_1^{(2k+1)}(\boldsymbol{z}). \tag{4.18}$$

In (4.17), since A_{2k+1} has antisymmetric linear phase with group delay $\boldsymbol{c}_{2k+1} = [0\ \ 0]^T$, $A_{2k+1}(\boldsymbol{z}^{\boldsymbol{M}})$ has antisymmetric linear phase with group delay $\boldsymbol{M}\boldsymbol{c}_{2k+1} = [0\ \ 0]^T$. Then, from Lemma 3.2, $A_{2k+1}(\boldsymbol{z}^{\boldsymbol{M}})H_0^{(2k)}(\boldsymbol{z})$ has antisymmetric linear phase with group delay $\left[-\frac{1}{2}\ \ 0\right]^T$. Thus, it follows that $\mathrm{H}_1^{(2k+2)}$ is antisymmetric with group delay $\left[-\frac{1}{2}\ \ 0\right]^T$. Similarly, from (4.18), $\mathrm{H}_0^{(2k+2)}$ is symmetric with group delay $\left[-\frac{1}{2}\ \ 0\right]^T$. Therefore, this theorem holds for $\lambda = k+1$. This completes the proof. □

Henceforth, we refer to quincunx filter banks constructed using Theorem 4.2 as **type-2 filter banks**. The filter bank from Example 3.3 in Section 3.6.2, where $H_0(z_0, z_1) = \frac{1}{2}(1 + z_0)$ and $H_1(z_0, z_1) = 1 - z_0$, can be parameterized using the above theorem with $\lambda = 1$.

For type-2 filter banks, if for $k \geq 3$ the lifting filters $\{\mathrm{A}_k\}$ are not only antisymmetric but also rotated quadrantally centrosymmetric with $S = -1$ and $T = 1$, then the resulting analysis filters satisfy the three conditions in Theorem 3.9. Thus, the filter bank can be used to construct nonexpansive transforms for finite-extent sequences extended with the type-2 symmetric extension scheme defined in Theorem 3.9.

A variation of Theorem 4.2 is given below. In this case, the analysis filters still have symmetric and antisymmetric linear phase, but their group delays are different from those obtained by Theorem 4.2.

Theorem 4.3 (Construction of type-3 filter banks)**.** *Consider a quincunx filter bank constructed from the lifting scheme with 2λ lifting filters as shown in Figure 4.2, where none of the transfer functions $\{A_k(\boldsymbol{z})\}$ are identically zero, except possibly $A_{2\lambda}(\boldsymbol{z})$. Suppose that the lifting filters $\{A_k\}$ satisfy the following conditions: 1) $A_1(\boldsymbol{z}) = -z_1$; 2) $A_2(\boldsymbol{z}) = \frac{1}{2}z_1^{-1}$; and 3) A_k is antisymmetric with group delay $\boldsymbol{c}_k = (-1)^k[0\ \ 1]^T$ for $k \geq 3$. Then, the analysis filter H_0 is symmetric with group delay $\left[0\ \ -\frac{1}{2}\right]^T$ and H_1 is antisymmetric with group delay $\left[-1\ \ \frac{1}{2}\right]^T$.*

The above theorem can be proved with an approach similar to that of Theorem 4.2. Theorem 4.3 can also be extended to build filter banks that lead to nonexpansive transforms. Suppose that in addition to the conditions in Theorem 4.3, the lifting filters $\{\mathrm{A}_k\}$ further have rotated quadrantal centrosymmetry with $S = 1$ and $T = -1$ for $k \geq 3$. Then, the resulting analysis filters satisfy the conditions in Theorem 3.10. That is,

the filter bank can be used to construct nonexpansive transforms for finite-extent sequences extended with the type-3 symmetric extension scheme defined in Theorem 3.10.

Although type-2 and type-3 filter banks have PR and linear phase, it is likely that these filter banks will not lead to good results in image coding applications. This is due to the antisymmetry of the highpass analysis filters, as we explain in what follows.

Consider a type-2 filter bank, where H_1 has antisymmetry with group delay $\boldsymbol{d}_1 = [d_{1,0} \quad d_{1,1}]^T = [-\frac{1}{2} \; 0]^T$. Then, from (2.4), we have

$$\hat{h}_1(\boldsymbol{\omega}) = e^{-j\left(\boldsymbol{\omega}^T \boldsymbol{d}_1 + \pi/2\right)} \sum_{\boldsymbol{n} \in \mathbb{Z}^D} h_1[\boldsymbol{n}] \sin\left[\boldsymbol{\omega}^T (\boldsymbol{n} - \boldsymbol{d}_1)\right].$$

Let $\omega_0 = 0$, we obtain

$$\hat{h}_1(0, \omega_1) = e^{-j\pi/2} \sum_{n_0 \in \mathbb{Z}} \sum_{n_1 \in \mathbb{Z}} h_1[\boldsymbol{n}] \sin(\omega_1 n_1).$$

Therefore, $\hat{h}_1(0,0) = 0$ and $\hat{h}_1(0, \pm\pi) = 0$. The first equality implies that the filter bank has at least one dual vanishing moment. The second equality, however, prevents the filter H_1 from having good diamond-shaped stopband. A similar result also holds for the synthesis filters. We have that there is at least one primal vanishing moment, but the frequency response of the highpass synthesis filter is zero at $[0 \;\; \pm\pi]^T$. For type-3 filter banks, similar results can be derived. Due to the restriction on the frequency responses, these filter banks would likely not perform well in image coding applications. An example illustrating the above statement will be given in Section 4.7. In the following part of this chapter, we focus on the type-1 filter banks where both analysis filters are symmetric.

4.4 Design of Type-1 Filter Banks with Two Lifting Steps

Consider a quincunx filter bank with two lifting steps as shown in Figure 4.3. For image coding application, we seek a filter bank with PR, linear phase, high coding gain, certain vanishing moment properties, and good frequency selectivity. We use the lifting-based parametrization introduced in Theorem 4.1 to enforce the PR and symmetric linear-phase properties. In what follows, we first investigate how the other desirable characteristics (i.e., high coding gain, vanishing moments, and good frequency selectivity) are related to the lifting filter coefficients. Then, we show how this design problem can be formulated as a constrained optimization problem.

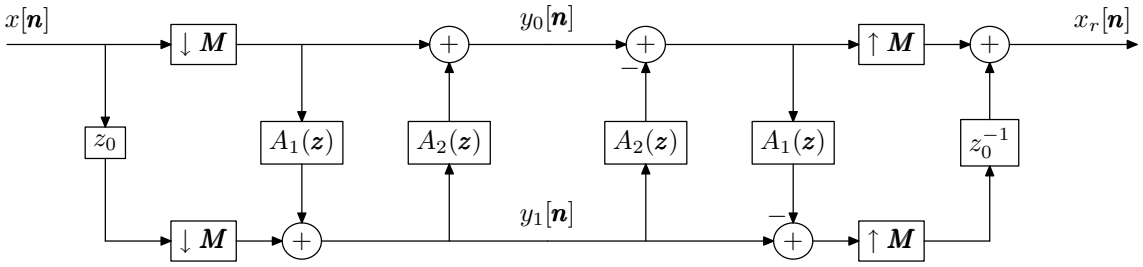


Figure 4.3: A quincunx filter bank with two lifting steps.

4.4.1 Coding Gain

Recall that for an N-level octave-band quincunx filter bank, the coding gain is computed by

$$G_{SBC} = \prod_{k=0}^{N} (A_k B_k / \alpha_k)^{-\alpha_k}, \tag{4.19}$$

where

$$A_k = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h'_k[\boldsymbol{m}] h'_k[\boldsymbol{n}] r[\boldsymbol{m}-\boldsymbol{n}],$$

$$B_k = \alpha_k \sum_{\boldsymbol{n}\in\mathbb{Z}^2} g'^2_k[\boldsymbol{n}],$$

$$\alpha_k = \begin{cases} 2^{-N} & \text{for } k = 0 \\ 2^{-(N+1-k)} & \text{for } k = 1, 2, \ldots, N, \end{cases}$$

$h'_k[\boldsymbol{n}]$ and $g'_k[\boldsymbol{n}]$ are the impulse responses of the equivalent analysis and synthesis filters H'_k and G'_k, respectively, and r is the normalized autocorrelation of the input. The N-level octave-band filter bank and its equivalent nonuniform filter banks are depicted in Figures 2.13 and 2.15, respectively.

For a type-1 filter bank, the analysis filter impulse responses are functions of the lifting filter coefficient vector $\boldsymbol{x}$. Therefore, the N-level coding gain can be expressed as a function of $\boldsymbol{x}$. Note that the coding gain function is potentially highly nonlinear.

4.4.2 Vanishing Moments

In Section 2.3.5, we explained that for a UMD filter bank, the number of vanishing moments is equivalent to the order of zero at $[0 \;\; 0]^T$ or $[\pi \;\; \pi]^T$ in the highpass or lowpass filter frequency response, respectively. In order to have N vanishing moments, the impulse response of the highpass or lowpass filter must satisfy a set of $\frac{N(N+1)}{2}$ linear equations of the form of (2.27) or (2.26), respectively. If a filter H has symmetric linear-phase with group delay on the lattice $\mathbb{Z}^2$, then the number of equations in (2.26) or (2.27) can be reduced.

This can be shown as follows. For a symmetric linear-phase filter H with group delay $\boldsymbol{d} \in \mathbb{Z}^2$, its frequency response $\hat{h}(\boldsymbol{\omega})$ can be computed by (2.4) with $S = 1$. The $\boldsymbol{m}$th-order partial derivative of its signed amplitude response $\hat{h}_a(\boldsymbol{\omega})$ defined in (2.5) is then given by

$$\frac{\partial^{m_0+m_1}\hat{h}_a(\boldsymbol{\omega})}{\partial\omega_0^{m_0}\partial\omega_1^{m_1}} = \begin{cases} \sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}]\,(\boldsymbol{n}-\boldsymbol{d})^{\boldsymbol{m}}\cos\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{d})\right] & \text{for } |\boldsymbol{m}| \in \mathbb{Z}_{even} \\ -\sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}]\,(\boldsymbol{n}-\boldsymbol{d})^{\boldsymbol{m}}\sin\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{d})\right] & \text{otherwise,} \end{cases}$$

where $\boldsymbol{m} = [m_0 \;\; m_1]^T$. From the above equation, we see that when $|\boldsymbol{m}| \in \mathbb{Z}_{odd}$, the $\boldsymbol{m}$th-order partial derivative of $\hat{h}_a(\boldsymbol{\omega})$ is automatically zero at $[0 \;\; 0]^T$ and $[\pi \;\; \pi]^T$. Therefore, in order to have an Nth order zero at $\boldsymbol{\omega} = [0 \;\; 0]^T$, the filter coefficients need only satisfy

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} h[\boldsymbol{n}]\,(\boldsymbol{n}-\boldsymbol{d})^{\boldsymbol{m}} = 0 \quad \text{for all } |\boldsymbol{m}| \in \mathbb{Z}_{even} \text{ such that } |\boldsymbol{m}| < N. \tag{4.20}$$

Similarly, in order to have an Nth order zero at $\boldsymbol{\omega} = [\pi \;\; \pi]^T$, the filter coefficients need only satisfy

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} (-1)^{|\boldsymbol{n}-\boldsymbol{d}|} h[\boldsymbol{n}]\,(\boldsymbol{n}-\boldsymbol{d})^{\boldsymbol{m}} = 0 \quad \text{for all } |\boldsymbol{m}| \in \mathbb{Z}_{even} \text{ such that } |\boldsymbol{m}| < N. \tag{4.21}$$

Since we only need to consider the case with $|\boldsymbol{m}| \in \mathbb{Z}_{even}$ in (4.20) and (4.21), the number of linear equations is reduced to $\lceil N/2 \rceil^2$. Therefore, in order for a filter bank to have $\tilde{N}$ dual and N primal vanishing moments, the analysis filter coefficients are required to satisfy equations like those shown in (4.20) and (4.21). Since we use the lifting-based parametrization, the relationships need to be expressed in terms of the lifting filter coefficients.

For a quincunx filter bank constructed with two lifting filters A_1 and A_2, as depicted in Figure 4.3, the constraints on dual and primal vanishing moments form a linear system of equations in the lifting filter coefficients. In what follows, we introduce the reader to some key results from [6] and then apply these results to type-1 filter banks.

In order for the filter bank shown in Figure 4.3 to have $\tilde{N}$ dual and N primal vanishing moments, the lifting filter impulse responses $a_1[\boldsymbol{n}]$ and $a_2[\boldsymbol{n}]$ of A_1 and A_2 should satisfy

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_1[\boldsymbol{n}](-\boldsymbol{n})^{\boldsymbol{m}} = -\boldsymbol{\tau}_1^{\boldsymbol{m}}, \text{ for } \boldsymbol{m} \in (\mathbb{Z}^*)^2 \text{ and } |\boldsymbol{m}| < \tilde{N} \quad \text{and} \tag{4.22}$$

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_2[\boldsymbol{n}](-\boldsymbol{n})^{\boldsymbol{m}} = \frac{1}{2}\boldsymbol{\tau}_2^{\boldsymbol{m}}, \text{ for } \boldsymbol{m} \in (\mathbb{Z}^*)^2 \text{ and } |\boldsymbol{m}| < N, \tag{4.23}$$

respectively, where $\boldsymbol{\tau}_1 = [\frac{1}{2} \;\; \frac{1}{2}]^T$ and $\boldsymbol{\tau}_2 = -\boldsymbol{\tau}_1 = [-\frac{1}{2} \;\; -\frac{1}{2}]^T$ [6]. The total number of equations in (4.22) and (4.23) combined is $\binom{\tilde{N}+1}{2} + \binom{N+1}{2} = \frac{(\tilde{N}+1)\tilde{N}+(N+1)N}{2}$. To illustrate the use of the above result, we consider the simple example below.

Example 4.2 (Vanishing moments condition)**.** Consider a quincunx filter bank with two lifting steps, where each of the lifting filters A_1 and A_2 has a support size of 2×2. The support regions of A_1 and A_2 are chosen to be centered around $-\boldsymbol{\tau}_1 = -\left[\frac{1}{2}\ \frac{1}{2}\right]^T$ and $-\boldsymbol{\tau}_2 = \left[\frac{1}{2}\ \frac{1}{2}\right]^T$, respectively. In order for this filter bank to have two dual and two primal vanishing moments, the impulse response a_1 of A_1 is required to satisfy

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_1[\boldsymbol{n}](-\boldsymbol{n})^{[0\ 0]^T} = -\left(\left[\tfrac{1}{2}\ \tfrac{1}{2}\right]^T\right)^{[0\ 0]^T},$$

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_1[\boldsymbol{n}](-\boldsymbol{n})^{[0\ 1]^T} = -\left(\left[\tfrac{1}{2}\ \tfrac{1}{2}\right]^T\right)^{[0\ 1]^T}, \text{ and}$$

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_1[\boldsymbol{n}](-\boldsymbol{n})^{[1\ 0]^T} = -\left(\left[\tfrac{1}{2}\ \tfrac{1}{2}\right]^T\right)^{[1\ 0]^T}.$$

In turn, this implies that

$$a_1[-1,-1] + a_1[-1,0] + a_1[0,-1] + a_1[0,0] = -1,$$

$$a_1[-1,-1] + a_1[0,-1] = -\frac{1}{2}, \text{ and}$$

$$a_1[-1,-1] + a_1[-1,0] = -\frac{1}{2}.$$

Similarly, the impulse response $a_2[\boldsymbol{n}]$ of A_2 is required to satisfy

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_2[\boldsymbol{n}](-\boldsymbol{n})^{[0\ 0]^T} = \frac{1}{2}\left(-\left[\tfrac{1}{2}\ \tfrac{1}{2}\right]^T\right)^{[0\ 0]^T},$$

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_2[\boldsymbol{n}](-\boldsymbol{n})^{[0\ 1]^T} = \frac{1}{2}\left(-\left[\tfrac{1}{2}\ \tfrac{1}{2}\right]^T\right)^{[0\ 1]^T}, \text{ and}$$

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_2[\boldsymbol{n}](-\boldsymbol{n})^{[1\ 0]^T} = \frac{1}{2}\left(-\left[\tfrac{1}{2}\ \tfrac{1}{2}\right]^T\right)^{[1\ 0]^T}.$$

This, in turn, implies that

$$a_2[0,0] + a_2[0,1] + a_2[1,0] + a_2[1,1] = \frac{1}{2},$$

$$-a_2[0,1] - a_2[1,1] = -\frac{1}{4}, \text{ and}$$

$$-a_2[1,0] - a_2[1,1] = -\frac{1}{4}.$$

Thus, the total number of equations that must be satisfied is $\frac{2(2+1)+2(2+1)}{2} = 6$. □

The above results on vanishing moments can be applied to the type-1 filter banks, where the analysis filters have symmetric linear phase with group delays $\boldsymbol{d}_0, \boldsymbol{d}_1 \in \mathbb{Z}^2$. The support region of A_1 is $\{-l_{1,0}, -l_{1,0}+1, \ldots, l_{1,0}-1\} \times \{-l_{1,1}, -l_{1,1}+1, \ldots, l_{1,1}-1\}$ for some $l_{1,0}, l_{1,1} \in \mathbb{Z}$. Then, (4.22) can be rewritten as

$$\sum_{n_0=0}^{l_{1,0}-1} \sum_{n_1=-l_{1,1}}^{l_{1,1}-1} a_1[n_0,n_1]\left[(n_0+1)^{m_0}(n_1+1)^{m_1} + (-n_0)^{m_0}(-n_1)^{m_1}\right] = -2^{-(m_0+m_1)}, \tag{4.24}$$

for $m_0, m_1 \in \mathbb{Z}^*$ and $m_0 + m_1 < \tilde{N}$. As previously discussed, we only need to consider the case with $m_0 + m_1 \in \mathbb{Z}_{even}$. Therefore, the number of equations in (4.24) is reduced to $\lceil \tilde{N}/2 \rceil^2$. If we use $\boldsymbol{a}_1$ to denote the independent coefficients of A_1 as defined in (4.3), the set of linear equations in (4.24) can be expressed in a compact form as

$$\boldsymbol{A}_1 \boldsymbol{a}_1 = \boldsymbol{b}_1, \tag{4.25}$$

where $\boldsymbol{A}_1$ is an $M_0 \times M_1$ matrix with $M_0 = \lceil \tilde{N}/2 \rceil^2$ and $M_1 = 2l_{1,0}l_{1,1}$, and $\boldsymbol{b}_1$ is a vector with $\lceil \tilde{N}/2 \rceil^2$ elements. Each element of $\boldsymbol{A}_1$ assumes the form $(n_0+1)^{m_0}(n_1+1)^{m_1} + (-n_0)^{m_0}(-n_1)^{m_1}$, and each element of $\boldsymbol{b}_1$ assumes the form $-2^{-(m_0+m_1)}$.

Similarly, because of the linear-phase property of the second lifting filter A_2, (4.23) becomes

$$\sum_{n_0=1}^{l_{2,0}} \sum_{n_1=-l_{2,1}+1}^{l_{2,1}} a_2[n_0, n_1] \left[(n_0-1)^{m_0}(n_1-1)^{m_1} + (-n_0)^{m_0}(-n_1)^{m_1} \right] = -(-2)^{-(m_0+m_1+1)}, \tag{4.26}$$

for $m_0, m_1 \in \mathbb{Z}^*$, $m_0 + m_1 \in \mathbb{Z}_{even}$ and $m_0 + m_1 < N$. With $\boldsymbol{a}_2$ denoting the $2l_{2,0}l_{2,1}$ independent coefficients of A_2 as defined in (4.7), (4.26) can be rewritten as

$$\boldsymbol{A}_2 \boldsymbol{a}_2 = \boldsymbol{b}_2, \tag{4.27}$$

where $\boldsymbol{A}_2$ is an $M_0 \times M_1$ matrix with $M_0 = \lceil N/2 \rceil^2$ and $M_1 = 2l_{2,0}l_{2,1}$, and $\boldsymbol{b}_2$ is a vector with $\lceil N/2 \rceil^2$ elements. Elements of $\boldsymbol{A}_2$ and $\boldsymbol{b}_2$ assume the forms of $(n_0-1)^{m_0}(n_1-1)^{m_1} + (-n_0)^{m_0}(-n_1)^{m_1}$ and $-(-2)^{-(m_0+m_1+1)}$, respectively.

Combining (4.25) and (4.27), we have a linear system of equations involving the lifting filter coefficient vector $\boldsymbol{x}$ as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{4.28}$$

where $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix}$, $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}$, and $\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}$. The number of equations in (4.28) is $\lceil \tilde{N}/2 \rceil^2 + \lceil N/2 \rceil^2$.

Example 4.3 (Vanishing moment condition for type-1 filter banks)**.** Consider the filter bank in Example 4.2, which has two 2×2 lifting filters A_1 and A_2. In the linear-phase case, the conditions for two dual and two primal vanishing moments becomes

$$2a_1[0,-1] + 2a_1[0,0] = -1 \quad \text{and} \quad 2a_2[1,0] + 2a_2[1,1] = \frac{1}{2},$$

respectively. Using the vector form $\boldsymbol{a}_1 = \begin{bmatrix} a_1[0,-1] & a_1[0,0] \end{bmatrix}^T$, $\boldsymbol{a}_2 = \begin{bmatrix} a_2[1,0] & a_2[1,1] \end{bmatrix}^T$, and $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}$, the condition for two dual and two primal vanishing moments becomes a linear system of equations involving $\boldsymbol{x}$ as

$$\begin{bmatrix} 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} -1 \\ 1/2 \end{bmatrix}.$$

□

It is worth noting that for a type-1 filter bank with two lifting steps, the analysis filter frequency responses have some special properties if this filter bank has at least one dual vanishing moment. In particular, we have the theorem below.

Theorem 4.4 (Frequency responses of type-1 filter banks with two lifting steps)**.** *Consider a type-1 filter bank with two lifting steps. Let $\hat{h}_0(\boldsymbol{\omega})$ and $\hat{h}_1(\boldsymbol{\omega})$ be the frequency responses of the lowpass and highpass analysis filters H_0 and H_1, respectively. If this filter bank has at least one dual vanishing moments, then*

$$\hat{h}_0(0,0) = 1 \quad \textit{and} \tag{4.29a}$$

$$\hat{h}_1(\pi,\pi) = -2 \tag{4.29b}$$

(i.e., the DC gain of the lowpass analysis filter H_0 is one and the Nyquist gain of the highpass analysis filter H_1 is two).

Proof. From (2.28), we have

$$\hat{h}_0(\boldsymbol{\omega}) = 1 + \hat{a}_1(\boldsymbol{M}^T\boldsymbol{\omega})\hat{a}_2(\boldsymbol{M}^T\boldsymbol{\omega}) + e^{j\omega_0}\hat{a}_2(\boldsymbol{M}^T\boldsymbol{\omega}) \quad \text{and} \tag{4.30}$$

$$\hat{h}_1(\boldsymbol{\omega}) = \hat{a}_1(\boldsymbol{M}^T\boldsymbol{\omega}) + e^{j\omega_0}. \tag{4.31}$$

Since the filter bank has at least one dual vanishing moment, we obtain $\hat{h}_1(0,0) = 0$. From (4.31), $\hat{a}_1(\boldsymbol{M}^T\boldsymbol{\omega}) = -e^{j\omega_0}$ for $\boldsymbol{\omega} = [0\ \ 0]^T$. It follows from (4.30) that for $\boldsymbol{\omega} = [0\ \ 0]^T$

$$\begin{aligned}\hat{h}_0(\boldsymbol{\omega}) &= 1 + \hat{a}_1(\boldsymbol{M}^T\boldsymbol{\omega})\hat{a}_2(\boldsymbol{M}^T\boldsymbol{\omega}) + e^{j\omega_0}\hat{a}_2(\boldsymbol{M}^T\boldsymbol{\omega}) \\ &= 1 - e^{j\omega_0}\hat{a}_2(\boldsymbol{M}^T\boldsymbol{\omega}) + e^{j\omega_0}\hat{a}_2(\boldsymbol{M}^T\boldsymbol{\omega}) \\ &= 1.\end{aligned}$$

Therefore, the DC gain of H_0 is one.

The lifting filter A_1 has symmetric linear phase with group delay $\boldsymbol{c}_1 = \left[-\frac{1}{2}\ -\frac{1}{2}\right]^T$. Then, from (2.4), its frequency response $\hat{a}_1(\boldsymbol{\omega})$ can be written as

$$\hat{a}_1(\boldsymbol{\omega}) = e^{-j\boldsymbol{\omega}^T\boldsymbol{c}_1}\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_1[\boldsymbol{n}]\cos\left[\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{c}_1)\right]. \tag{4.32}$$

Substituting (4.32) into (4.31), we obtain

$$\begin{aligned}\hat{h}_1(\boldsymbol{\omega}) &= e^{-j\boldsymbol{\omega}^T\boldsymbol{M}\boldsymbol{c}_1}\sum_{\boldsymbol{n}\in\mathbb{Z}^2} a_1[\boldsymbol{n}]\cos\left[\boldsymbol{\omega}^T(\boldsymbol{M}\boldsymbol{n}-\boldsymbol{M}\boldsymbol{c}_1)\right] + e^{j\omega_0} \\ &= e^{j\omega_0}\left\{\sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} a_1[n_0,n_1]\cos\left[\omega_0(n_0+n_1+1)+\omega_1(n_0-n_1)\right]+1\right\}.\end{aligned}$$

Since $\hat{h}_1(0,0) = 0$, we have that $\sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} a_1[n_0,n_1] = -1$. Therefore, we conclude

$$\begin{aligned}\hat{h}_1(\pi,\pi) &= e^{j\pi}\left\{\sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} a_1[n_0,n_1]\cos\left[(2n_0+1)\pi\right]+1\right\}\\ &= \sum_{n_0\in\mathbb{Z}}\sum_{n_1\in\mathbb{Z}} a_1[n_0,n_1]-1\\ &= -2.\end{aligned}$$

Thus, the Nyquist gain of H_1 is two. □

In the preceding discussion for filter banks with two lifting steps, it is assumed that the number of dual vanishing moments is no less than that of the primal ones (i.e., $\tilde{N} \geq N$). This is usually desired in image processing applications such as image coding, as the dual vanishing moments are more important than the primal ones for annihilating polynomials which are used to approximate images. If in some special cases, more primal vanishing moments are needed than dual ones, we can interchange the roles of the analysis and synthesis filters. This interchange, however, may have undesirable effects on the other filter bank properties such as the coding gain.

4.4.3 Frequency Response

For image coding, we desire analysis filters with good frequency selectivity. Since a lifting-based parametrization of quincunx filter banks is employed, we consider the relationship between the analysis filter frequency responses and the lifting filter coefficients.

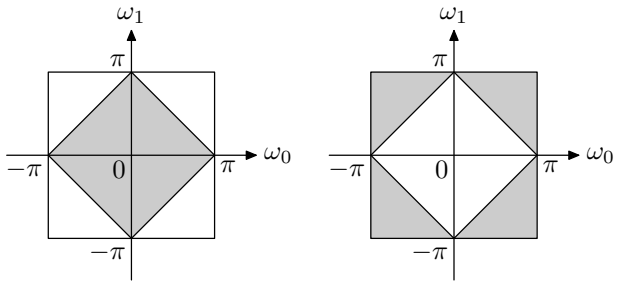
To measure the difference between a symmetric linear-phase filter H and an ideal filter H_d, we define the weighted frequency response error function of H as

$$e_h = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\left|\hat{h}_a(\boldsymbol{\omega}) - D\hat{h}_d(\boldsymbol{\omega})\right|^2 \mathrm{d}\boldsymbol{\omega}, \tag{4.33}$$

where $W(\boldsymbol{\omega})$ is a weighting function defined on $[-\pi,\,\pi)^2$, $\hat{h}_a(\boldsymbol{\omega})$ is the signed amplitude response of H as defined by (2.5), $\hat{h}_d(\boldsymbol{\omega})$ is the frequency response of the ideal filter H_d, and D is a scaling factor. In order for the filter H to approximate the ideal filter, the frequency response error function e_h is required to satisfy

$$e_h \leq \delta_h, \tag{4.34}$$

where δ_h is a prescribed upper bound on the error. Note that $\hat{h}_a(\boldsymbol{\omega})$ in (4.33) is possible to be negative. The error e_h will be large if $\hat{h}_a(\boldsymbol{\omega})$ changes sign in the passband. This, however, will not lead to any problem, as in this case, the filter does not have good frequency selectivity.



(a) (b)

Figure 4.4: Ideal frequency responses of quincunx filter banks for the (a) lowpass filters and (b) highpass filters, where the shaded and unshaded areas represent the passband and stopband, respectively.

For a quincunx filter bank with sampling matrix $\boldsymbol{M} = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, the shape of filter passband is not unique [20, 28]. Herein, in order to match the human visual system, we use diamond-shaped ideal passband/stopband for the analysis and synthesis filters [9]. Figure 4.4(a) illustrates the ideal lowpass filter frequency response given by

$$\hat{h}_{0d}(\boldsymbol{\omega}) = \begin{cases} 1 & \text{for } |\omega_0 \pm \omega_1| \leq \pi \\ 0 & \text{otherwise,} \end{cases} \tag{4.35}$$

and Figure 4.4(b) depicts the ideal highpass filter frequency response given by

$$\hat{h}_{1d}(\boldsymbol{\omega}) = \begin{cases} 1 & \text{for } |\omega_0 \pm \omega_1| \geq \pi, \text{ and } \omega_0, \omega_1 \in [-\pi, \pi) \\ 0 & \text{otherwise.} \end{cases} \tag{4.36}$$

The weighting function $W(\boldsymbol{\omega})$ is used to assign different weights for the passband, stopband, and transition band. For a quincunx highpass filter with a diamond-shaped stopband, the weighting function $W(\boldsymbol{\omega})$ is defined as

$$W(\boldsymbol{\omega}) = \begin{cases} 1 & \text{for passband } |\omega_0 \pm \omega_1| \geq \pi + \omega_p, \text{ and } \omega_0, \omega_1 \in [-\pi, \pi) \\ \gamma & \text{for stopband } |\omega_0 \pm \omega_1| \leq \omega_s \\ 0 & \text{otherwise (i.e., transition band),} \end{cases} \tag{4.37}$$

where $\gamma \geq 0$. By adjusting the value of γ, we can control the filter's performance in the stopband relative to the passband. In the case of highpass filters, for example, the weighting function is depicted in Figure 4.5. The weighting function for a quincunx lowpass filter is defined in a similar way (i.e., with the roles of passband and stopband interchanged in (4.37)).

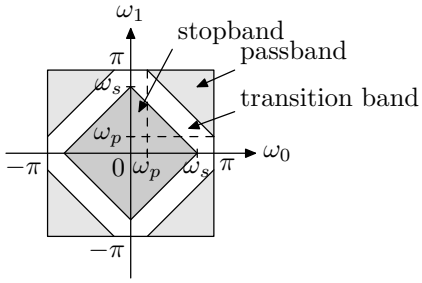


Figure 4.5: Weighting function for a highpass filter with diamond-shaped stopband.

Consider a type-1 filter bank constructed with two lifting filters A_1 and A_2 as the one shown in Figure 4.3. From (4.16), we obtain the frequency responses of the analysis filters as

$$\begin{bmatrix} \hat{h}_0(\boldsymbol{\omega}) \\ \hat{h}_1(\boldsymbol{\omega}) \end{bmatrix} = \begin{bmatrix} 1 & e^{-j\omega_0}\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ e^{j\omega_0}\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ e^{j\omega_0} \end{bmatrix} = \begin{bmatrix} 1+\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2+\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{v}_1^T\boldsymbol{E}_1\boldsymbol{x} \\ e^{j\omega_0}\left(1+\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1\right) \end{bmatrix}.$$

Then, the signed amplitude response $\hat{h}_{1a}(\boldsymbol{\omega})$ of H_1 is

$$\hat{h}_{1a}(\boldsymbol{\omega}) = 1+\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1.$$

The frequency response error function of the highpass analysis filter H_1 is computed as

$$e_{h_1} = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega}) \left|\hat{h}_{1a}(\boldsymbol{\omega}) - D\hat{h}_{1d}(\boldsymbol{\omega})\right|^2 \mathrm{d}\boldsymbol{\omega}, \tag{4.38}$$

where $W(\boldsymbol{\omega})$ is the weighting function defined in (4.37), $\hat{h}_{1d}(\boldsymbol{\omega})$ is the ideal frequency response of a quincunx highpass filter defined in (4.36), and the scaling factor D is chosen to be $D = 2$ in accordance with (4.29b). The frequency response error function in (4.38) can be expressed as the quadratic polynomial in the lifting filter coefficient vector $\boldsymbol{x}$

$$e_{h_1} = \boldsymbol{x}^T\boldsymbol{H}_x\boldsymbol{x}+\boldsymbol{x}^T\boldsymbol{s}_x+c_x, \tag{4.39}$$

where

$$\boldsymbol{H}_x = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{v}_1^T\boldsymbol{E}_1\mathrm{d}\boldsymbol{\omega}, \quad \boldsymbol{s}_x = \int_{[-\pi,\,\pi)^2} 2W(\boldsymbol{\omega})\boldsymbol{E}_1^T\boldsymbol{v}_1\left[1-2\hat{h}_{1d}(\boldsymbol{\omega})\right]\mathrm{d}\boldsymbol{\omega},$$
$$c_x = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\left[1-2\hat{h}_{1d}(\boldsymbol{\omega})\right]^2\mathrm{d}\boldsymbol{\omega},$$

and $\boldsymbol{H}_x$ is a positive semidefinite matrix. Substituting (4.39) into the constraint on the frequency response (4.34), we obtain an inequality involving $\boldsymbol{x}$ as

$$\boldsymbol{x}^T\boldsymbol{H}_x\boldsymbol{x}+\boldsymbol{x}^T\boldsymbol{s}_x+c_x-\delta_h \le 0.$$

For illustrative purposes, we now consider a simple example of the computation of the frequency response error function for the highpass analysis filter H_1. The lifting filters have the same support as those in Example 4.2.

Example 4.4 (Frequency response error function)**.** Consider a type-1 filter bank with two 2×2 lifting filters A_1 and A_2. The filter A_1 has two independent coefficients $a_1[0,-1]$ and $a_1[0,0]$, and A_2 also has two independent coefficients $a_2[1,0]$ and $a_2[1,1]$. The vector $\boldsymbol{x}$ and its associated parameters are given by

$$\boldsymbol{x} = \begin{bmatrix} a_1[0,-1] & a_1[0,0] & a_2[1,0] & a_2[1,1] \end{bmatrix}^T,$$

$$\boldsymbol{v}_1 = \boldsymbol{v}_2 = \begin{bmatrix} 2\cos\omega_1 \\ 2\cos\omega_0 \end{bmatrix}, \quad \boldsymbol{E}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{E}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The quantities $\boldsymbol{H}_x$, $\boldsymbol{s}_x$, and c_x in (4.39) are computed as

$$\boldsymbol{H}_x = \begin{bmatrix} 4\int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\cos^2\omega_1 \mathrm{d}\boldsymbol{\omega} & 4\int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\cos\omega_0\cos\omega_1 \mathrm{d}\boldsymbol{\omega} & 0 & 0 \\ 4\int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\cos\omega_0\cos\omega_1 \mathrm{d}\boldsymbol{\omega} & 4\int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\cos^2\omega_0 \mathrm{d}\boldsymbol{\omega} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{s}_x = \begin{bmatrix} 4\int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\left[1-2\hat{h}_{1d}(\boldsymbol{\omega})\right]\cos\omega_1 \mathrm{d}\boldsymbol{\omega} & 4\int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\left[1-2\hat{h}_{1d}(\boldsymbol{\omega})\right]\cos\omega_0 \mathrm{d}\boldsymbol{\omega} & 0 & 0 \end{bmatrix}^T,$$

$$\text{and } c_x = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\left[1-2\hat{h}_{1d}(\boldsymbol{\omega})\right]^2 \mathrm{d}\boldsymbol{\omega},$$

where $W(\boldsymbol{\omega})$ and $\hat{h}_{1d}(\boldsymbol{\omega})$ are defined in (4.37) and (4.36), respectively.

From (4.13), for a type-1 filter bank with two lifting steps, the frequency response $\hat{h}_1(\boldsymbol{\omega})$ of the highpass analysis filter H_1 is independent of the second lifting filter A_2. Therefore, the frequency response error function e_{h_1} in (4.39) only involves the filter coefficients $a_1[\boldsymbol{n}]$ of the first lifting filter A_1. This observation explains the zeros in $\boldsymbol{H}_x$ and $\boldsymbol{s}_x$. □

4.4.4 Design Problem Formulation

Consider a type-1 filter bank with two lifting steps as shown in Figure 4.3. The design of such a filter bank with all the desirable properties (i.e., PR, linear phase, high coding gain, certain vanishing moment properties, and good frequency selectivity) can be formulated as a constrained optimization problem. Our design employs the lifting-based parametrization introduced in Theorem 4.1. In this way, the PR and linear-phase conditions are automatically satisfied. We then maximize the coding gain subject to a set of constraints, where these constraints ensure that the desired vanishing moment and frequency selectivity conditions are

met. In what follows, we will show more precisely how this design problem can be formulated as a **second-order cone programming** (SOCP) problem.

In an SOCP problem, a linear function is minimized subject to a set of second-order cone constraints [41]. In other words, we have a problem of the form:

$$\begin{aligned} &\text{minimize} && \boldsymbol{b}^T\boldsymbol{x} \\ &\text{subject to:} && \left\|\boldsymbol{A}_i^T\boldsymbol{x}+\boldsymbol{c}_i\right\| \le \boldsymbol{b}_i^T\boldsymbol{x}+d_i \quad \text{for } i=1,\ldots,q, \end{aligned}$$

where $\boldsymbol{x}\in\mathbb{R}^n$ is the design vector containing n free variables, and $\boldsymbol{b}\in\mathbb{R}^n$, $\boldsymbol{A}_i\in\mathbb{R}^{n\times m_i}$, $\boldsymbol{c}_i\in\mathbb{R}^{m_i}$, $\boldsymbol{b}_i\in\mathbb{R}^n$, and $d_i\in\mathbb{R}$. The constraint $\left\|\boldsymbol{A}_i^T\boldsymbol{x}+\boldsymbol{c}_i\right\| \le \boldsymbol{b}_i^T\boldsymbol{x}+d_i$ is called a **second-order cone constraint**.

Consider a type-1 filter bank with two lifting filters A_1 and A_2, each having support size of $2l_{1,0}\times 2l_{1,1}$ and $2l_{2,0}\times 2l_{2,1}$, respectively. Let $\boldsymbol{a}_i$, $\boldsymbol{v}_i$, and $\boldsymbol{E}_i$ be as defined in Section 4.3.1, and let $\boldsymbol{x}$ denote the vector consisting of the $2l_{1,0}l_{1,1}+2l_{2,0}l_{2,1}$ independent lifting-filter coefficients defined in (4.14). As explained previously, in terms of the lifting filter coefficient vector $\boldsymbol{x}$, the constraint on vanishing moments is linear and the constraint on the frequency response of the highpass analysis filter is quadratic.

From Section 4.4.2, we know that in order for a filter bank to have N primal and $\tilde{N}$ dual vanishing moments, $\boldsymbol{x}$ needs to be the solution of a system of $\lceil\tilde{N}/2\rceil^2+\lceil N/2\rceil^2$ linear equations given by

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}. \tag{4.40}$$

In (4.40), $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ with rank r and $\boldsymbol{b}\in\mathbb{R}^{m\times 1}$, where $m=\lceil\tilde{N}/2\rceil^2+\lceil N/2\rceil^2$, $n=2l_{1,0}l_{1,1}+2l_{2,0}l_{2,1}$, and $r\le\min\{m,n\}$. The system is underdetermined when there are enough lifting filter coefficients such that $m<n$. In what follows, we assume that the system is underdetermined so that the feasible region of the design problem contains more than one point. Let the singular value decomposition (SVD) of $\boldsymbol{A}$ be $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$. All of the solutions to (4.40) can be parameterized as

$$\boldsymbol{x}=\underbrace{\boldsymbol{A}^+\boldsymbol{b}}_{\boldsymbol{x}_s}+\boldsymbol{V}_r\boldsymbol{\phi}=\boldsymbol{x}_s+\boldsymbol{V}_r\boldsymbol{\phi}, \tag{4.41}$$

where $\boldsymbol{A}^+$ is the Moore-Penrose pseudoinverse of $\boldsymbol{A}$, $\boldsymbol{V}_r=[\boldsymbol{v}_{r+1}\quad\boldsymbol{v}_{r+2}\quad\cdots\quad\boldsymbol{v}_n]$ is a matrix composed of the last $n-r$ columns of $\boldsymbol{V}$, and $\boldsymbol{\phi}$ is an arbitrary $(n-r)$-dimensional vector. Henceforth, we shall use $\boldsymbol{\phi}$ as the design vector instead of $\boldsymbol{x}$. Thus, the vanishing moment condition is enforced and the number of free variables involved is reduced from n to $n-r$.

The design objective is to maximize the coding gain G_{SBC} of an N-level octave-band quincunx filter bank, which is computed by (4.19) and can be expressed as a nonlinear function of the design vector $\boldsymbol{\phi}$. Let $G=-10\log_{10}G_{SBC}$. Then, the problem of maximizing G_{SBC} is equivalent to minimizing G. Although taking

the logarithm helps improve the numerical stability of the optimization algorithm and reduce the nonlinearity in G, direct minimization of G remains to be a very difficult task. Our design strategy is that, for a given parameter vector $\boldsymbol{\phi}$, we seek a small perturbation $\boldsymbol{\delta}_{\boldsymbol{\phi}}$ such that $G(\boldsymbol{\phi}+\boldsymbol{\delta}_{\boldsymbol{\phi}})$ is reduced relative to $G(\boldsymbol{\phi})$. Because $\left\|\boldsymbol{\delta}_{\boldsymbol{\phi}}\right\|$ is small, we can write the quadratic and linear approximations of $G(\boldsymbol{\phi}+\boldsymbol{\delta}_{\boldsymbol{\phi}})$ as

$$G(\boldsymbol{\phi}+\boldsymbol{\delta}_{\boldsymbol{\phi}}) \approx G(\boldsymbol{\phi}) + \boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{\phi}} + \frac{1}{2}\boldsymbol{\delta}_{\boldsymbol{\phi}}^T\boldsymbol{Q}\boldsymbol{\delta}_{\boldsymbol{\phi}} \quad \text{and} \tag{4.42}$$

$$G(\boldsymbol{\phi}+\boldsymbol{\delta}_{\boldsymbol{\phi}}) \approx G(\boldsymbol{\phi}) + \boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{\phi}}, \tag{4.43}$$

respectively, where $\boldsymbol{g}$ is the gradient of $G(\boldsymbol{\phi})$ and $\boldsymbol{Q}$ is the Hessian of $G(\boldsymbol{\phi})$ at the point $\boldsymbol{\phi}$. Having obtained such a $\boldsymbol{\delta}_{\boldsymbol{\phi}}$ (subject to some additional constraint to be described shortly), the parameter vector $\boldsymbol{\phi}$ is updated to $\boldsymbol{\phi}+\boldsymbol{\delta}_{\boldsymbol{\phi}}$. This iterative process continues until the reduction in G (i.e., $\left|G(\boldsymbol{\phi}+\boldsymbol{\delta}_{\boldsymbol{\phi}})-G(\boldsymbol{\phi})\right|$) becomes less than a prescribed tolerance ε.

Consider the constraint on the frequency response. In Section 4.4.3, we showed that for filter banks constructed with two lifting steps, the frequency response error function e_{h_1} of the highpass analysis filter H_1 is a quadratic polynomial in $\boldsymbol{x}$ as given by (4.39). Substituting (4.41) into (4.39), we have

$$e_{h_1} = \boldsymbol{\phi}^T\boldsymbol{H}_{\boldsymbol{\phi}}\boldsymbol{\phi} + \boldsymbol{\phi}^T\boldsymbol{s}_{\boldsymbol{\phi}} + c_{\boldsymbol{\phi}}, \tag{4.44}$$

where

$$\boldsymbol{H}_{\boldsymbol{\phi}} = \boldsymbol{V}_r^T\boldsymbol{H}_x\boldsymbol{V}_r, \quad \boldsymbol{s}_{\boldsymbol{\phi}} = \boldsymbol{V}_r^T\left(\boldsymbol{H}_x+\boldsymbol{H}_x^T\right)\boldsymbol{x}_s + \boldsymbol{V}_r^T\boldsymbol{s}_x,$$
$$c_{\boldsymbol{\phi}} = \boldsymbol{x}_s^T\boldsymbol{H}_x\boldsymbol{x}_s + \boldsymbol{x}_s^T\boldsymbol{s}_x + c_x,$$

and $\boldsymbol{H}_x$, $\boldsymbol{s}_x$, and c_x are given in (4.39). Note that from the above definition, it follows from the fact $\boldsymbol{H}_x$ is positive semidefinite that $\boldsymbol{H}_{\boldsymbol{\phi}}$ is also positive semidefinite. Further, let us replace $\boldsymbol{\phi}$ by $\boldsymbol{\phi}_k+\boldsymbol{\delta}_{\boldsymbol{\phi}}$ and let the SVD of $\boldsymbol{H}_{\boldsymbol{\phi}}$ be given by

$$\boldsymbol{H}_{\boldsymbol{\phi}} = \boldsymbol{U}_H\boldsymbol{\Sigma}\boldsymbol{V}_H^T.$$

Then, (4.44) can also be written as

$$e_{h_1} = \left\|\tilde{\boldsymbol{H}}_k\boldsymbol{\delta}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}_k\right\|^2 + \tilde{c}_k,$$

and the constraint (4.34) becomes the second-order cone constraint

$$\left\|\tilde{\boldsymbol{H}}_k\boldsymbol{\delta}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}_k\right\|^2 \le \delta_{h_1} - \tilde{c}_k, \tag{4.45}$$

where

$$\tilde{\boldsymbol{H}}_k = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{U}_H^T, \quad \tilde{\boldsymbol{s}}_k = \frac{1}{2}\tilde{\boldsymbol{H}}^{-T}\left(2\boldsymbol{H}_{\boldsymbol{\phi}}\boldsymbol{\phi}_k + \boldsymbol{s}_{\boldsymbol{\phi}}\right), \quad \text{and}$$
$$\tilde{c}_k = \boldsymbol{\phi}_k^T\boldsymbol{H}_{\boldsymbol{\phi}}\boldsymbol{\phi}_k + \boldsymbol{\phi}_k^T\boldsymbol{s}_{\boldsymbol{\phi}} + c_{\boldsymbol{\phi}} - \left\|\tilde{\boldsymbol{s}}_k\right\|^2.$$

Based on the preceding discussions, we now show how to employ the SOCP algorithm to solve the problem of maximizing the coding gain $G_{SBC}(\boldsymbol{x})$, or equivalently minimizing G, with the vanishing moment constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ as in (4.40) and the frequency response constraint $e_{h_1} \leq \delta_{h_1}$ as in (4.34). This problem can be formulated as the following iterative algorithm:

Algorithm 4.1 (Design algorithm for type-1 filter banks with two lifting steps)**.** This algorithm consists of the following steps:

Step 1. Compute $\boldsymbol{A}$ and $\boldsymbol{b}$ in (4.28) for the desired numbers of vanishing moments, and calculate $\boldsymbol{H}_{\boldsymbol{\phi}}$, $\boldsymbol{s}_{\boldsymbol{\phi}}$, and $c_{\boldsymbol{\phi}}$ in (4.44). Then, select an initial point $\boldsymbol{\phi}_0$. This point can be chosen randomly, or chosen to be a quincunx filter bank proposed in [6]. The vanishing moment condition is satisfied, and because of the way we choose the upper bound δ_{h_1} for the frequency response error function, which will be discussed later, $\boldsymbol{\phi}_0$ will not violate the frequency response constraint. In this way, the initial point is in the feasible region.

Step 2. For the kth iteration, at the point $\boldsymbol{\phi}_k$, compute the gradient $\boldsymbol{g}$ of $G(\boldsymbol{\phi})$ in (4.43), and calculate $\tilde{\boldsymbol{H}}_k$, $\tilde{\boldsymbol{s}}_k$, and $\tilde{c}_k$ in (4.45). Then, solve the SOCP problem given by:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{g}^T \boldsymbol{\delta}_{\boldsymbol{\phi}} \\ \text{subject to:} \quad & \left\| \tilde{\boldsymbol{H}}_k \boldsymbol{\delta}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}_k \right\| \leq \sqrt{\delta_{h_1} - \tilde{c}_k} \quad \text{and} \\ & \left\| \boldsymbol{\delta}_{\boldsymbol{\phi}} \right\| \leq \beta, \end{aligned} \tag{4.46}$$

where β is a given small value used to ensure that the solution is within the vicinity of $\boldsymbol{\phi}_k$. Then, update $\boldsymbol{\phi}_k$ by $\boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k + \gamma \boldsymbol{\delta}_{\boldsymbol{\phi}}$, where $\gamma = 1$ or is a scalar determined by a line-search step explained in more detail later. A number of software packages are available for solving SOCP problems. In our work, for example, we use the SeDuMi optimization package [42] to seek the optimal solution $\boldsymbol{\delta}_{\boldsymbol{\phi}}$.

Step 3. If $|G(\boldsymbol{\phi}_{k+1}) - G(\boldsymbol{\phi}_k)| < \varepsilon$, output $\boldsymbol{\phi}^* = \boldsymbol{\phi}_{k+1}$, compute $\boldsymbol{x}^* = \boldsymbol{x}_s + \boldsymbol{V}_r \boldsymbol{\phi}^*$, and stop. Otherwise, go to step 2.

The vector $\boldsymbol{x}^*$ output by the above algorithm is then the optimal solution to this problem. The filter bank constructed with the lifting filter coefficients $\boldsymbol{x}^*$ has high coding gain, good frequency selectivity, and the desired vanishing moment properties.

Two additional comments are now in order concerning the SOCP problem (4.46) in the second step of the above iterative algorithm. In particular, the choice of β is critical to the success of the algorithm. It should be chosen such that

$$\boldsymbol{g}^T \boldsymbol{\delta} \approx G(\boldsymbol{\phi} + \boldsymbol{\delta}) - G(\boldsymbol{\phi}) \quad \text{for} \quad \|\boldsymbol{\delta}\| = \beta.$$

If β is too large, the linear approximation (4.43) is less accurate, resulting in the linear term $\boldsymbol{g}^T \boldsymbol{\delta}_{\boldsymbol{\phi}}$ not correctly reflecting the actual reduction in G. If β is too small, the solution is restricted to an unnecessarily

small region around $\boldsymbol{\phi}_k$ with k being the number of iterations, causing points outside this region which may provide a greater reduction in G to be excluded. For this reason, we incorporate a line search in step 2 to find a better solution along the direction of $\boldsymbol{\delta}_{\boldsymbol{\phi}}$. We first evaluate G at N_0 equally spaced points between $\boldsymbol{\phi}_k$ and $\boldsymbol{\phi}_k + \alpha\boldsymbol{\delta}_{\boldsymbol{\phi}}$ along the direction of $\boldsymbol{\delta}_{\boldsymbol{\phi}}$ for some $\alpha \geq 1$, including the point $\boldsymbol{\phi}_k + \boldsymbol{\delta}_{\boldsymbol{\phi}}$. Then, we use the point $\boldsymbol{\phi}_k^*$ corresponding to the minimal G to select γ. By including a line search, in each iteration the reduction in G is as large as the reduction obtained without the line search. This makes the algorithm converge with less iterations. The choice of α depends on the choice of β. When β is large, we can choose $\alpha = 1$. When β is small, we can choose α to be one or greater. Note that a greater value of α may imply more evaluations of the coding gain function G in each iteration.

The second comment about step 2 concerns the choice of the upper bound δ_{h_1} of the frequency response error function in the SOCP problem (4.46). If δ_{h_1} is too small, the feasible region of the SOCP problem may be an empty set, especially for designs starting from a random initial point. Therefore, we choose δ_{h_1} to be a scaled version of the error function e_{h_1} evaluated at $\boldsymbol{\phi}_k$. That is, we select

$$\delta_{h_1} = d\left(\boldsymbol{\phi}_k^T \boldsymbol{H}_{\boldsymbol{\phi}} \boldsymbol{\phi}_k + \boldsymbol{\phi}_k^T \boldsymbol{s}_{\boldsymbol{\phi}} + c_{\boldsymbol{\phi}}\right),$$

where $0 < d \leq 1$ is a scaling factor. In this way, the error e_{h_1} is reduced after each iteration, and the frequency response of the highpass analysis filter H_1 improves gradually with each iteration.

4.4.5 Design Algorithm with Hessian

In Algorithm 4.1 of the preceding section, a linear approximation (4.43) of the coding gain function G is employed. This necessitates that the perturbation $\boldsymbol{\delta}_{\boldsymbol{\phi}}$ be located in a small region. For this design problem, we can instead use the quadratic approximation in (4.42). In this way, the approximation accuracy can be improved, and the solution can be sought in a larger region. Algorithm 4.1 can be adapted to utilize the quadratic approximation with some minor changes to the SOCP problem in each iteration. In step 2, we minimize $\boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{\phi}} + \frac{1}{2}\boldsymbol{\delta}_{\boldsymbol{\phi}}^T\boldsymbol{Q}\boldsymbol{\delta}_{\boldsymbol{\phi}}$ instead of $\boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{\phi}}$ in (4.46). That is, we seek a solution to the problem

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{\phi}} + \frac{1}{2}\boldsymbol{\delta}_{\boldsymbol{\phi}}^T\boldsymbol{Q}\boldsymbol{\delta}_{\boldsymbol{\phi}} \\ \text{subject to:} \quad & \left\|\tilde{\boldsymbol{H}}\boldsymbol{\delta}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}\right\| \leq \sqrt{\delta_{h_1} - \tilde{c}} \quad \text{and} \\ & \left\|\boldsymbol{\delta}_{\boldsymbol{\phi}}\right\| \leq \beta. \end{aligned} \tag{4.47}$$

Let the SVD of $\frac{1}{2}\boldsymbol{Q}$ be $\frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_{\boldsymbol{Q}}\boldsymbol{\Sigma}_{\boldsymbol{Q}}\boldsymbol{V}_{\boldsymbol{Q}}^T$. When $\boldsymbol{Q}$ is positive semidefinite, we can rewrite the objective function as

$$\boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{\phi}} + \frac{1}{2}\boldsymbol{\delta}_{\boldsymbol{\phi}}^T\boldsymbol{Q}\boldsymbol{\delta}_{\boldsymbol{\phi}} = \left\|\tilde{\boldsymbol{Q}}\boldsymbol{\delta}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}_{\boldsymbol{Q}}\right\|^2 + \tilde{c}_Q, \tag{4.48}$$

where

$$\tilde{\boldsymbol{Q}} = \boldsymbol{\Sigma}_{\boldsymbol{Q}}^{\frac{1}{2}} \boldsymbol{U}_{\boldsymbol{Q}}^{T}, \quad \tilde{\boldsymbol{s}}_{\boldsymbol{Q}} = \frac{1}{2} \tilde{\boldsymbol{Q}}^{-T} \boldsymbol{g}, \quad \text{and}$$

$$\tilde{c}_Q = -\tilde{\boldsymbol{s}}_{\boldsymbol{Q}}^{T} \tilde{\boldsymbol{s}}_{\boldsymbol{Q}}.$$

If we introduce another variable η to be the upper bound of $\left\| \tilde{\tilde{\boldsymbol{Q}}} \tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}_{\boldsymbol{Q}} \right\|$ and define $\tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} = \begin{bmatrix} \eta & \boldsymbol{\delta}_{\boldsymbol{\phi}} \end{bmatrix}^T$ and $\boldsymbol{b} = [1 \ 0 \ \cdots \ 0]^T$, then (4.47) becomes the SOCP problem

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{b}^T \tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} \\ \text{subject to:} \quad & \left\| \tilde{\tilde{\boldsymbol{Q}}} \tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}}_{\boldsymbol{Q}} \right\| \le \boldsymbol{b}^T \tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}}, \\ & \left\| \tilde{\tilde{\boldsymbol{H}}} \tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} + \tilde{\boldsymbol{s}} \right\| \le \sqrt{\delta_{h_1} - \tilde{c}}, \quad \text{and} \\ & \left\| \tilde{\boldsymbol{I}} \tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} \right\| \le \beta, \end{aligned}$$

where $\tilde{\tilde{\boldsymbol{Q}}} = \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{Q}} \end{bmatrix}$, $\tilde{\tilde{\boldsymbol{H}}} = \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{H}} \end{bmatrix}$, and $\tilde{\boldsymbol{I}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$. With the quadratic approximation, the algorithm reaches the optimal solution with fewer iterations than the linear-approximation case, but it takes longer for each iteration as the coding gain is evaluated many more times when computing the Hessian. An example illustrating the above statement is given as follows.

Example 4.5 (Design example with the Hessian matrix)**.** Two filter banks, EX1 and EX2, are designed using Algorithm 4.1 and the revised algorithm with the Hessian, respectively. These filter banks are optimized for two dual and two primal vanishing moments and maximal coding gain assuming an isotropic image model with correlation coefficient $\rho = 0.95$ and a one-level decomposition. For both of the design problems, the same initial point is used, which corresponds to the filter bank constructed using the method in [6] with four primal and four dual vanishing moments. Information about the optimization processes and results are summarized in Table 4.1. From this table, we see that very similar results are obtained for these two designs in terms of the coding gain. For the design with the quadratic approximation, the time used for each iteration is increased compared to the linear-approximation case, but the number of iterations is reduced greatly. □

Note that (4.48) holds only when $\boldsymbol{Q}$ is positive semidefinite and $\boldsymbol{Q}$ need not always be positive semidefinite. When $\boldsymbol{Q}$ is not positive semidefinite, we can always revert to the use of the linear approximation.

4.5 Design of Type-1 Filter Banks with More Than Two Lifting Steps

The strategy for the design of type-1 filter banks with more than two lifting steps is similar to the two-lifting-step case. When more lifting filters are involved, however, the relationships between the filter bank

Table 4.1: Comparison of algorithms with linear and quadratic approximations

Filter bank	EX1	EX2
Approximation	linear	quadratic
One-level isotropic coding gain (dB)	6.86	6.86
Number of evaluations of G per iteration	10	65
Average time per iteration	0.4	1.0
Number of iterations	41	5
Total time (seconds)	20.1	5.1

characteristics (i.e., coding gain, vanishing moment properties and frequency selectivity) and the lifting filter coefficients become more complicated. In this section, we consider how to formulate the design as an SOCP problem based on these relationships.

The computation of coding gains in this case is basically the same as the two-lifting-step case discussed in Section 4.4.1. For an N-level octave-band quincunx filter bank, the coding gain G_{SBC} is computed by (4.19), and G_{SBC} is a nonlinear function of the lifting filter coefficients.

4.5.1 Vanishing Moments

Compared to the two-lifting-step case, the vanishing moments condition changes considerably for a type-1 filter bank with at least three lifting filters, like the one shown in Figure 4.2. The condition is no longer linear with respect to the lifting filter coefficient vector $\boldsymbol{x}$. With the notations $\boldsymbol{a}_k$, $\boldsymbol{v}_k$, $\boldsymbol{x}$, and $\boldsymbol{E}_k$ introduced in Section 4.3.1, the frequency responses $\{\hat{h}_k(\boldsymbol{\omega})\}$ of the analysis filters are given by (4.16), and $\{\hat{h}_k(\boldsymbol{\omega})\}$ can each be expressed as a polynomial in $\boldsymbol{x}$.

In order for this filter bank to have $\tilde{N}$ dual vanishing moments, the frequency response $\hat{h}_1(\boldsymbol{\omega})$ of the highpass analysis filter should have a $\tilde{N}$th-order zero at $[0\ \ 0]^T$. Therefore,

$$\hat{h}_{1a}^{(\boldsymbol{m})}(0,0) = \left.\frac{\partial^{m_0+m_1}\hat{h}_{1a}(\omega_0,\omega_1)}{\partial\omega_0^{m_0}\partial\omega_1^{m_1}}\right|_{(0,0)} = 0 \tag{4.49}$$

for all $\boldsymbol{m} = [m_0\ \ m_1]^T$, $m_0+m_1 \in \mathbb{Z}_{even}$, and $m_0+m_1 < \tilde{N}$, where $\hat{h}_{1a}(\boldsymbol{\omega})$ is the signed amplitude response of H_1 as defined in (2.5). As H_1 has linear phase and $\hat{h}_1(\boldsymbol{\omega})$ can be viewed as a polynomial in $\boldsymbol{x}$, $\hat{h}_{1a}(\boldsymbol{\omega})$ and thus $\hat{h}_{1a}^{(\boldsymbol{m})}(0,0)$ can also be viewed as polynomials in $\boldsymbol{x}$. In this way, in order to have $\tilde{N}$ dual vanishing moments, the lifting-filter coefficients in $\boldsymbol{x}$ needs to satisfy $\left\lceil \tilde{N}/2 \right\rceil^2$ polynomial equations. Similarly, in order to have N

primal vanishing moments, the frequency response $\hat{h}_0^{(\boldsymbol{m})}(\boldsymbol{\omega})$ of the lowpass analysis filter H_0 should satisfy

$$\hat{h}_0^{(\boldsymbol{m})}(\pi,\pi) = \left.\frac{\partial^{m_0+m_1}\hat{h}_{0a}(\omega_0,\omega_1)}{\partial\omega_0^{m_0}\partial\omega_1^{m_1}}\right|_{(\pi,\pi)} = 0 \tag{4.50}$$

for all $\boldsymbol{m} = [m_0 \;\; m_1]^T$ such that $m_0+m_1 \in \mathbb{Z}_{even}$ and $m_0+m_1 < N$. It follows that $\boldsymbol{x}$ needs to satisfy $\lceil N/2 \rceil^2$ polynomial equations.

Below we give an example of a type-1 filter bank with more than two lifting steps. We show how to derive the condition for up to four dual and four primal vanishing moments.

Example 4.6 (Vanishing moment condition for a filter bank with three lifting steps)**.** Consider a type-1 filter bank with three lifting steps. The lifting filter coefficient vector $\boldsymbol{x}$ is defined in (4.15). From (4.16), the analysis filter frequency responses are

$$\hat{h}_0(\boldsymbol{\omega}) = 1 + \boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2 + \boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2 \quad \text{and} \tag{4.51}$$

$$\hat{h}_1(\boldsymbol{\omega}) = e^{j\omega_0}\left(\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1 + \boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3 + \boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3 + 1 + \boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3\right). \tag{4.52}$$

Recall that the nth elements of $\boldsymbol{v}_k$ assumes the form given in (4.6) and (4.10), and $n_0, n_1 \in \mathbb{Z}$ are defined in (4.4) and (4.8) for the odd- and even-indexed cases, respectively. We define vectors $\boldsymbol{n}_{k,0}$ and $\boldsymbol{n}_{k,1}$ to denote the coefficients of ω_0 and ω_1 in $\boldsymbol{v}_k$, respectively. Then, $\boldsymbol{n}_{k,0}$ and $\boldsymbol{n}_{k,1}$ each have $2l_{k,0}l_{k,1}$ elements. The nth element of $\boldsymbol{n}_{k,0}$ is n_0+n_1+1 or n_0+n_1-1, and the nth element of $\boldsymbol{n}_{k,1}$ is n_0-n_1. We further define $\boldsymbol{i}_k$ to be a $2l_{k,0}l_{k,1}$-dimensional column vector of all ones, and $\boldsymbol{u}_{k,d_0,d_1} = \boldsymbol{n}_{k,d_0} \circ \boldsymbol{n}_{k,d_1} \circ \boldsymbol{i}_k$ for $d_0, d_1 \in \{0,1\}$.

With the above notation in place, from (4.49), in order for the filter bank to have two dual vanishing moments, $\boldsymbol{x}$ should satisfy a third-order polynomial equation given by

$$8\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{i}_3 + 4\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2^T\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{i}_3 + 2\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{i}_1 + 2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{i}_3 + 1 = 0.$$

In order to have four dual vanishing moments, $\boldsymbol{x}$ needs to satisfy three more third-order equations

$$\begin{aligned}&8\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{u}_{1,d_0,d_1}\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{i}_3 + 8\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{u}_{2,d_0,d_1}\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{i}_3 + 8\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{u}_{3,d_0,d_1}\\ &+4\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{u}_{2,d_0,d_1}^T\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{i}_3 + 4\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2^T\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{u}_{3,d_0,d_1} + 2\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{u}_{1,d_0,d_1} + 2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{u}_{3,d_0,d_1} = 0,\end{aligned}$$

for $(d_0,d_1) = (0,0)$, $(0,1)$ and $(1,1)$. Similarly, in order for the filter bank to have two primal vanishing moments, $\boldsymbol{x}$ should satisfy the quadratic equation

$$4\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2 - 2\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2 + 1 = 0. \tag{4.53}$$

To have four primal vanishing moments, $\boldsymbol{x}$ needs to satisfy three additional quadratic equations

$$4\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{u}_{1,d_0,d_1}\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{i}_2 + 4\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{u}_{2,d_0,d_1} - 2\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{u}_{2,d_0,d_1} = 0,$$

for $(d_0,d_1)=(0,0)$, $(0,1)$ and $(1,1)$. □

Now we introduce another approach to enforce the condition on desired vanishing moments properties. Consider a type-1 filter bank with K lifting filters $\{\mathrm{A}_i\}$, where none of the transfer functions $\{A_i(\boldsymbol{z})\}$ are identically zero, except possibly $A_1(\boldsymbol{z})$. If the first $K-1$ lifting filters are fixed, to have a certain number of dual or primal vanishing moments, the constraint on the Kth lifting filter coefficients is a linear system. We denote the analysis filters constructed with the first l lifting filters as $\{\mathrm{H}_i^{(l)}\}$ for $i=0,1$. If $K\in\mathbb{Z}_{even}$, we have

$$\hat{h}_0^{(K)}(\boldsymbol{\omega})=\hat{h}_0^{(K-1)}(\boldsymbol{\omega}) \quad \text{and}$$
$$\hat{h}_1^{(K)}(\boldsymbol{\omega})=\hat{h}_1^{(K-1)}(\boldsymbol{\omega})+\hat{a}_K(\boldsymbol{M}^T\omega)\hat{h}_0^{(K-1)}(\boldsymbol{\omega})=\hat{h}_1^{(K-1)}(\boldsymbol{\omega})+e^{j\omega_0}\boldsymbol{a}_K^T\boldsymbol{v}_K\hat{h}_0^{(K-1)}(\boldsymbol{\omega}).$$

Therefore, the Kth lifting filter A_K can be used to impose $\tilde{N}$ dual vanishing moments to the filter bank. The signed amplitude response $\hat{h}_{1a}^{(K)}(\boldsymbol{\omega})$ of $\mathrm{H}_1^{(K)}$ needs to satisfy (4.49). Since $\hat{h}_{1a}^{(K)}(\boldsymbol{\omega})$ contains only the first-order term of $\boldsymbol{a}_K$, (4.49) becomes a set of $\lceil\tilde{N}/2\rceil^2$ linear equations of $\boldsymbol{a}_K$. Similarly, if $K\in\mathbb{Z}_{odd}$, we have

$$\hat{h}_1^{(K)}(\boldsymbol{\omega})=\hat{h}_1^{(K-1)}(\boldsymbol{\omega}) \quad \text{and}$$
$$\hat{h}_0^{(K)}(\boldsymbol{\omega})=\hat{h}_0^{(K-1)}(\boldsymbol{\omega})+\hat{a}_K(\boldsymbol{M}^T\omega)\hat{h}_1^{(K-1)}(\boldsymbol{\omega})=\hat{h}_0^{(K-1)}(\boldsymbol{\omega})+e^{-j\omega_0}\boldsymbol{a}_K^T\boldsymbol{v}_K\hat{h}_1^{(K-1)}(\boldsymbol{\omega}).$$

The Kth lifting filter A_K can provide N primal vanishing moments to the filter bank, if its independent coefficient vector $\boldsymbol{a}_K$ satisfies a linear system of $\lceil N/2\rceil^2$ equations. In summary, the last lifting filter A_K can be used to provide a certain number of dual or primal vanishing moments if its coefficients satisfy a set of linear equations. These results are useful in the suboptimal design algorithm proposed in Section 4.6, as well as in Section 4.5.3 for the choice of initial points.

4.5.2 Frequency Responses

Recall that in the two-lifting-step case, the frequency response constraint is defined in (4.33) and (4.34), and the constraint on the highpass analysis filter is a second-order cone. For type-1 filter banks with more than two lifting steps, we defined the frequency response constraint in a similar way. The frequency response error functions of the lowpass and highpass analysis filters, however, are at least fourth-order polynomials in the lifting filter coefficients. This is because the frequency responses of the analysis filters H_0 and H_1 are at least quadratic polynomials in the lifting filter coefficient vector $\boldsymbol{x}$ when more than two lifting filters are involved. Below is an example of the highpass analysis filter frequency response of a type-1 filter bank with three lifting steps.

Example 4.7 (Frequency response of the highpass analysis filter with three lifting steps)**.** Consider a filter bank with three lifting filters A_1, A_2, and A_3, where none of the transfer functions is identically zero. Using $\boldsymbol{x}$, $\boldsymbol{E}_k$, and $\boldsymbol{v}_k$ as defined in Section 4.3.1, from (4.52), the signed amplitude response $\hat{h}_{1a}(\boldsymbol{\omega})$ of the highpass analysis filter is a third-order polynomial in the lifting filter coefficient vector $\boldsymbol{x}$ given by

$$\hat{h}_{1a}(\boldsymbol{\omega}) = \boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1 + \boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3 + \boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3 + 1 + \boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3.$$

Therefore, it follows that the frequency response error function e_{h_1} of H_1 as defined in (4.33) is a sixth-order polynomial in $\boldsymbol{x}$. □

4.5.3 Design Problem Formulation

In the two-lifting-step case, we see that the vanishing moment condition is a linear system of equations and the frequency response constraint is a second-order cone in terms of the lifting filter coefficients. Thus, the design can be formulated as an SOCP problem. For filter banks with more than two lifting steps, the design problem becomes complicated as the constraints on vanishing moments and frequency responses become higher-order polynomials in the lifting filter coefficients. In order to use SOCP, the constraints on vanishing moments and the frequency response must be approximated by linear and quadratic constraints, respectively.

We deal with the coding gain $G_{SBC}(\boldsymbol{x})$ with the same strategy as in the two-lifting-step case. The linear approximation of G with $G(\boldsymbol{x}) = -10\log_{10} G_{SBC}(\boldsymbol{x})$ is given by

$$G(\boldsymbol{x}+\boldsymbol{\delta}_{\boldsymbol{x}}) = G(\boldsymbol{x}) + \boldsymbol{g}^T\boldsymbol{\delta}_{\boldsymbol{x}},$$

where $\boldsymbol{g}$ is the gradient of G at point $\boldsymbol{x}$. We iteratively seek a small perturbation $\boldsymbol{\delta}_{\boldsymbol{x}}$ in $\boldsymbol{x}$ such that $G(\boldsymbol{x}+\boldsymbol{\delta}_{\boldsymbol{x}})$ is reduced relative to $G(\boldsymbol{x})$ until the difference between $G(\boldsymbol{x}+\boldsymbol{\delta}_{\boldsymbol{x}})$ and $G(\boldsymbol{x})$ is less than a prescribed tolerance.

As discussed in Section 4.5.1, the constraint on vanishing moments is a set of polynomial equations in $\boldsymbol{x}$. We substitute $\boldsymbol{x}$ with $\boldsymbol{x}_k + \boldsymbol{\delta}_{\boldsymbol{x}}$. Provided that $\|\boldsymbol{\delta}_{\boldsymbol{x}}\|$ is small, the quadratic and higher-order terms in $\boldsymbol{\delta}_{\boldsymbol{x}}$ can be neglected, and these polynomial equations can be approximated by the linear system

$$\boldsymbol{A}_k\boldsymbol{\delta}_{\boldsymbol{x}} = \boldsymbol{b}_k. \tag{4.54}$$

In this way, the filter bank constructed with lifting filter coefficients $\boldsymbol{x}_k + \boldsymbol{\delta}_{\boldsymbol{x}}$ has the desired vanishing moment properties. Due to the problem formulation, the moments of interest are only guaranteed to be small, but not exactly zero. In practice, however, the moments are typically very close to zero, as will be illustrated by our design examples.

As explained above, we can approximate the vanishing moment conditions by linear equations. Below, we provide an example to illustrate the process of the linear approximation.

Example 4.8 (Linear approximation of the vanishing moment condition)**.** As shown in Example 4.6, in order for a filter bank with three lifting steps to have two primal vanishing moments, the lifting filter coefficient vector $\boldsymbol{x}$ needs to satisfy a quadratic equation (4.53). Replacing $\boldsymbol{x}$ with $\boldsymbol{x}_k + \boldsymbol{\delta}_{\boldsymbol{x}}$ and neglecting the quadratic term in $\boldsymbol{\delta}_{\boldsymbol{x}}$, we have that (4.53) is approximated by the linear equation in $\boldsymbol{\delta}_{\boldsymbol{x}}$ as given by

$$\left[4\boldsymbol{x}_k^T\left(\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{i}_2^T\boldsymbol{E}_2+\boldsymbol{E}_2^T\boldsymbol{i}_2\boldsymbol{i}_1^T\boldsymbol{E}_1\right)-2\boldsymbol{i}_2^T\boldsymbol{E}_2\right]\boldsymbol{\delta}_{\boldsymbol{x}}=-4\boldsymbol{x}_k^T\boldsymbol{E}_1^T\boldsymbol{i}_1\boldsymbol{i}_2^T\boldsymbol{E}_2\boldsymbol{x}_k+2\boldsymbol{x}_k^T\boldsymbol{E}_2^T\boldsymbol{i}_2-1.$$

□

Now we consider the frequency response of the highpass analysis filter H_1. The weighted error function e_{h_1} is defined in (4.33). In order to have good frequency selectivity, the function e_{h_1} must satisfy the constraint (4.34). From (2.28), $\hat{h}_{1a}(\boldsymbol{\omega})$ has at least a second-order term in $\boldsymbol{x}$. Therefore, e_{h_1} is at least a fourth-order polynomial in $\boldsymbol{x}$. Using a similar approach as above, we replace $\boldsymbol{x}$ by $\boldsymbol{x}_k + \boldsymbol{\delta}_{\boldsymbol{x}}$ in $\hat{h}_{1a}(\boldsymbol{\omega})$ with $\|\boldsymbol{\delta}_{\boldsymbol{x}}\|$ being small, and neglect the second- and higher-order terms in $\boldsymbol{\delta}_{\boldsymbol{x}}$. Now, $\hat{h}_{1a}(\boldsymbol{\omega})$ is approximated by a linear function of $\boldsymbol{\delta}_{\boldsymbol{x}}$. Using (4.33), a quadratic approximation of e_{h_1} is obtained as

$$e_{h_1}=\boldsymbol{\delta}_{\boldsymbol{x}}^T\boldsymbol{H}_k\boldsymbol{\delta}_{\boldsymbol{x}}+\boldsymbol{\delta}_{\boldsymbol{x}}^T\boldsymbol{s}_k+c_k,$$

where $\boldsymbol{H}_k$ is a symmetric positive semidefinite matrix, and $\boldsymbol{H}_k$, $\boldsymbol{s}_k$, and c_k are dependent on $\boldsymbol{x}_k$. Therefore, the constraint $e_{h_1} \le \delta_{h_1}$ can be expressed in the form of a second-order cone constraint as

$$\left\|\tilde{\boldsymbol{H}}_k\boldsymbol{\delta}_{\boldsymbol{x}}+\tilde{\boldsymbol{s}}_k\right\|^2 \le \delta_{h_1}-\tilde{c}_k. \tag{4.55}$$

Note that the approximation is not applied to e_{h_1}, but $\hat{h}_{1a}(\boldsymbol{\omega})$. In this way, the matrix $\boldsymbol{H}_k$ is guaranteed to be positive semidefinite, which allows for the form of a second-order cone as in (4.55).

As introduced above, for a filter bank with more than two lifting steps, the frequency response error functions can be approximated by quadratic functions of the lifting filter coefficients. We illustrate the approximation process by the example below.

Example 4.9 (Approximation of the frequency response error function)**.** Consider a type-1 filter bank with three lifting filters A_1, A_2, and A_3. From Example 4.7, we know that the signed amplitude response $\hat{h}_{1a}(\boldsymbol{\omega})$ of the highpass analysis filter is a cubic polynomial in the lifting filter coefficient vector $\boldsymbol{x}$ given by

$$\hat{h}_{1a}(\boldsymbol{\omega})=\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1+\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3+\boldsymbol{x}^T\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3+1+\boldsymbol{x}^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}^T\boldsymbol{E}_3^T\boldsymbol{v}_3.$$

Replacing $\boldsymbol{x}$ by $\boldsymbol{x}_k + \boldsymbol{\delta}_{\boldsymbol{x}}$ and neglecting the quadratic and higher-order terms in $\boldsymbol{\delta}_{\boldsymbol{x}}$, $\hat{h}_{1a}(\boldsymbol{\omega})$ is approximated by the linear function of $\boldsymbol{\delta}_{\boldsymbol{x}}$ given by

$$\hat{h}_{1a}(\boldsymbol{\omega})=\boldsymbol{\delta}_{\boldsymbol{x}}^T\boldsymbol{u}_k+l_k,$$

where

$$\boldsymbol{u}_k = \boldsymbol{E}_1^T\boldsymbol{v}_1 + \boldsymbol{E}_3^T\boldsymbol{v}_3 + \left(\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{v}_3^T\boldsymbol{E}_3 + \boldsymbol{E}_3^T\boldsymbol{v}_3\boldsymbol{v}_2^T\boldsymbol{E}_2\right)\boldsymbol{x}_k + \boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}_k^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}_k^T\boldsymbol{E}_3^T\boldsymbol{v}_3$$
$$+ \boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}_k^T\boldsymbol{E}_3^T\boldsymbol{v}_3\boldsymbol{x}_k^T\boldsymbol{E}_1^T\boldsymbol{v}_1 + \boldsymbol{E}_3^T\boldsymbol{v}_3\boldsymbol{x}_k^T\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}_k^T\boldsymbol{E}_2^T\boldsymbol{v}_2 \quad \text{and}$$
$$l_k = \boldsymbol{x}_k^T\boldsymbol{E}_1^T\boldsymbol{v}_1 + \boldsymbol{x}_k^T\boldsymbol{E}_3^T\boldsymbol{v}_3 + \boldsymbol{x}_k^T\boldsymbol{E}_1^T\boldsymbol{v}_1\boldsymbol{x}_k^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}_k^T\boldsymbol{E}_3^T\boldsymbol{v}_3 + 1 + \boldsymbol{x}_k^T\boldsymbol{E}_2^T\boldsymbol{v}_2\boldsymbol{x}_k^T\boldsymbol{E}_3^T\boldsymbol{v}_3.$$

Then, the frequency response error function e_{h_1} is a quadratic function of $\boldsymbol{\delta_x}$ given by

$$e_{h_1} = \boldsymbol{\delta}_{\boldsymbol{x}}^T\boldsymbol{H}_k\boldsymbol{\delta_x} + \boldsymbol{\delta}_{\boldsymbol{x}}^T\boldsymbol{s}_k + c_k, \tag{4.56}$$

where

$$\boldsymbol{H}_k = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\boldsymbol{u}_k\boldsymbol{u}_k^T\,\mathrm{d}\boldsymbol{\omega}, \quad \boldsymbol{s}_k = \int_{[-\pi,\,\pi)^2} 2W(\boldsymbol{\omega})\boldsymbol{u}_1\left[l_k - D\hat{h}_{1d}(\boldsymbol{\omega})\right]\mathrm{d}\boldsymbol{\omega}, \quad \text{and}$$
$$c_k = \int_{[-\pi,\,\pi)^2} W(\boldsymbol{\omega})\left[l_k - D\hat{h}_{1d}(\boldsymbol{\omega})\right]^2\mathrm{d}\boldsymbol{\omega}.$$

Since $\boldsymbol{H}_k$ is positive semidefinite, the constraint on the frequency response of H_1 can be rewritten as a second order cone constraint as (4.55). Note that for filter banks with four lifting filters, the frequency response error function for H_1 assumes the same form as in (4.56). □

The approximation method illustrated by the preceding example can also be used to control the frequency response of the lowpass analysis filter H_0 for filter banks with two or more lifting steps. For example, with two lifting steps, the analysis lowpass filter frequency response $\hat{h}_0(\boldsymbol{\omega})$ is a quadratic polynomial in the design vector $\boldsymbol{\phi}$. We can replace $\boldsymbol{\phi}$ by $\boldsymbol{\phi}_k + \boldsymbol{\delta_\phi}$ in $\hat{h}_0(\boldsymbol{\omega})$ and keep only the constant and first-order terms. Then, the error function e_{h_0} computed with this linear approximation of $\hat{h}_0(\boldsymbol{\omega})$ becomes a quadratic function of $\boldsymbol{\delta_\phi}$, and the constraint $e_{h_0} \le \delta_{h_0}$ can be expressed as a second-order cone in $\boldsymbol{\delta_\phi}$.

Based on the preceding approximation methods of the vanishing moment condition and frequency response constraint, the design of filter banks with more than two lifting steps can be formulated as an iterative SOCP problem. To solve this design problem, we use a scheme similar to Algorithm 4.1. Let K be the number of lifting steps. The new algorithm is given below.

Algorithm 4.2 (Design algorithm for type-1 filter banks with more than two lifting steps)**.** This algorithm is comprised of the following steps:

Step 1. Select an initial point $\boldsymbol{x}_0$ such that the resulting filter bank has the desired number of vanishing moments. We can choose the first two lifting filters using the method proposed for the two-lifting-step case, and then set the other $K-2$ filter coefficients to be all zeros. Alternatively, we can randomly select the coefficients of the first $K-2$ filters, and then use the last two lifting filters to provide dual and primal

vanishing moments. In this way, the filter bank constructed with the initial point $\boldsymbol{x}_0$ has the desired number of vanishing moments. Moreover, since the upper bound δ_{h_1} for the frequency response error function is chosen in the same way as in Algorithm 4.1, the frequency response constraint will not be violated. Therefore, $\boldsymbol{x}_0$ is inside the feasible region.

Step 2. For the kth iteration, at the point $\boldsymbol{x}_k$, compute the gradient $\boldsymbol{g}$ of $G(\boldsymbol{x})$, $\boldsymbol{A}_k$ and $\boldsymbol{b}_k$ in (4.54), and $\tilde{\boldsymbol{H}}_k$, $\tilde{\mathbf{s}}_k$, and $\tilde{c}_k$ in (4.55). Then, solve the SOCP problem:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{g}^T \boldsymbol{\delta}_{\boldsymbol{x}} \\ \text{subject to:} \quad & \boldsymbol{A}_k \boldsymbol{\delta}_{\boldsymbol{x}} = \boldsymbol{b}_k, \\ & \left\| \tilde{\boldsymbol{H}}_k \boldsymbol{\delta}_{\boldsymbol{x}} + \tilde{\mathbf{s}}_k \right\| \leq \sqrt{\delta_{h_1} - \tilde{c}_k}, \quad \text{and} \\ & \|\boldsymbol{\delta}_{\boldsymbol{x}}\| \leq \beta. \end{aligned} \tag{4.57}$$

The linear constraint $\boldsymbol{A}_k \boldsymbol{\delta}_{\boldsymbol{x}} = \boldsymbol{b}_k$ can be parameterized as in Algorithm 4.1 to reduce the number of design variables, or be approximated by the second-order cone $\|\boldsymbol{A}_k \boldsymbol{\delta}_{\boldsymbol{x}} - \boldsymbol{b}_k\| \leq \varepsilon_\delta$ with ε_δ being a prescribed tolerance. Then, we can use the optimal solution $\boldsymbol{\delta}_{\boldsymbol{x}}$ to update $\boldsymbol{x}_k$ by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\delta}_{\boldsymbol{x}}$. We can also incorporate a line search into this process to improve the efficiency of the algorithm.

Step 3. If $|G(\boldsymbol{x}_{k+1}) - G(\boldsymbol{x}_k)| < \varepsilon$, then output $\boldsymbol{x}^* = \boldsymbol{x}_{k+1}$ and stop. Otherwise, go to step 2.

Upon termination of the above algorithm, the output $\boldsymbol{x}^*$ will correspond to a filter bank with all of the desired properties. In step 2, we deal with the constant δ_{h_1} in the same way as in Algorithm 4.1 and it is chosen to be a scaled version of the error function evaluated at the point $\boldsymbol{x}_k$. We use a variable scaling factor D in the frequency response error function (4.33) since the Nyquist gain of H_1 is not fixed in this case. For the kth iteration, we choose D to be the Nyquist gain of the highpass analysis filter obtained from the previous iteration (i.e., $D = \hat{h}_{1a}(\pi, \pi)$ with $\hat{h}_{1a}(\boldsymbol{\omega})$ being the signed amplitude response of H_1 obtained from the $(k-1)$th iteration).

Due to the linear approximation (4.54), the moments associated with the desired vanishing moment conditions are only guaranteed to be small but not necessarily zero. An adjustment step can be applied after step 3 to further reduce the moments in question at the expense of a slight decrease in the coding gain. This step is formulated as follows. Let $\{\Gamma_i(\boldsymbol{x})\} = 0$ be the set of polynomial equations that the lifting filter coefficient vector $\boldsymbol{x}$ needs to satisfy to achieve N primal and $\tilde{N}$ dual vanishing moments. When $\|\boldsymbol{\delta}_{\boldsymbol{x}}\|$ is small, the linear approximation of $\Gamma_i(\boldsymbol{x}^* + \boldsymbol{\delta}_{\boldsymbol{x}})$ is obtained by

$$\Gamma_i(\boldsymbol{x}^* + \boldsymbol{\delta}_{\boldsymbol{x}}) = \Gamma_i(\boldsymbol{x}^*) + \boldsymbol{g}_i^T \boldsymbol{\delta}_{\boldsymbol{x}},$$

where $\boldsymbol{g}_i$ is the gradient of Γ_i at the point $\boldsymbol{x}^*$. This adjustment process can then be formulated as the following

optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_i \left[\Gamma_i(\boldsymbol{x}^*) + \boldsymbol{g}_i^T \boldsymbol{\delta}_{\boldsymbol{x}}\right]^2 \\ \text{subject to:} \quad & \|\boldsymbol{\delta}_{\boldsymbol{x}}\| \le \beta_a, \end{aligned} \tag{4.58}$$

where β_a is a prescribed small value. The objective function of (4.58) can be rewritten as

$$\sum_i \left(\Gamma_i(\boldsymbol{x}^*) + \boldsymbol{g}_i^T \boldsymbol{\delta}_{\boldsymbol{x}}\right)^2 = \boldsymbol{\delta}_{\boldsymbol{x}}^T \left(\sum_i \boldsymbol{g}_i \boldsymbol{g}_i^T\right) \boldsymbol{\delta}_{\boldsymbol{x}} + \boldsymbol{\delta}_{\boldsymbol{x}}^T \left[2\sum_i \Gamma_i(\boldsymbol{x}^*)\boldsymbol{g}_i\right] + \sum_i \Gamma_i^2(\boldsymbol{x}^*).$$

Since $\sum_i \boldsymbol{g}_i \boldsymbol{g}_i^T$ is positive semidefinite, the objective function can be expressed in the form $\left\|\tilde{\boldsymbol{H}}_{\boldsymbol{\delta}} \boldsymbol{\delta}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{\delta}}\right\|^2 + \tilde{c}_{\boldsymbol{\delta}}$. If we introduce another variable η to be the upper bound of the term $\left\|\tilde{\boldsymbol{H}}_{\boldsymbol{\delta}} \boldsymbol{\delta}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{\delta}}\right\|$, the problem in (4.58) becomes

$$\begin{aligned} \text{minimize} \quad & \eta \\ \text{subject to:} \quad & \left\|\tilde{\boldsymbol{H}}_{\boldsymbol{\delta}} \boldsymbol{\delta}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{\delta}}\right\| \le \eta \quad \text{and} \\ & \|\boldsymbol{\delta}_{\boldsymbol{x}}\| \le \beta_a. \end{aligned}$$

The above problem is equivalent to the SOCP problem

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{b}^T \tilde{\boldsymbol{\delta}}_{\boldsymbol{x}} \\ \text{subject to:} \quad & \left\|\tilde{\tilde{\boldsymbol{H}}}_{\boldsymbol{\delta}} \boldsymbol{\delta}_{\boldsymbol{x}} + \tilde{\boldsymbol{s}}_{\boldsymbol{\delta}}\right\| \le \boldsymbol{b}^T \tilde{\boldsymbol{\delta}}_{\boldsymbol{x}} \quad \text{and} \\ & \left\|\tilde{\boldsymbol{I}} \boldsymbol{\delta}_{\boldsymbol{x}}\right\| \le \beta_a, \end{aligned}$$

where $\tilde{\boldsymbol{\delta}}_{\boldsymbol{\phi}} = \begin{bmatrix} \eta & \boldsymbol{\delta}_{\boldsymbol{\phi}} \end{bmatrix}^T$, $\boldsymbol{b} = [1 \;\; 0 \;\; \cdots \;\; 0]^T$, $\tilde{\tilde{\boldsymbol{H}}}_{\boldsymbol{\delta}} = \begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{H}}_{\boldsymbol{\delta}} \end{bmatrix}$, and $\tilde{\boldsymbol{I}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$.

Similar to the two-lifting-step case, for the design of filter banks with more than two lifting steps, we can also use the quadratic approximation of the coding gain function G given by

$$G(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{x}}) \approx G(\boldsymbol{x}) + \boldsymbol{g}^T \boldsymbol{\delta}_{\boldsymbol{x}} + \frac{1}{2} \boldsymbol{\delta}_{\boldsymbol{x}}^T \boldsymbol{Q} \boldsymbol{\delta}_{\boldsymbol{x}}.$$

A similar change can be made to the SOCP problem (4.57) in step 2 in Algorithm 4.2.

4.6 Suboptimal Design Algorithm

For filter banks with more than two lifting steps, Algorithm 4.2 can only guarantee nearly vanishing moments due to the approximation in the design process. In this section, we introduce a suboptimal design method to obtain exact vanishing moments for filter banks with more than two lifting steps. Let K be the number

of lifting steps. The strategy is to first design a filter bank with $K-2$ lifting steps without any vanishing moments. Then, we design two additional lifting filters in order to achieve desired numbers of dual and primal vanishing moments. In this case, as discussed at the end of Section 4.5.1, the constraint on vanishing moments is linear and exact vanishing moments can be achieved. The procedure for the design of a filter bank with K lifting steps with $\tilde{N}$ dual and N primal vanishing moments is given by the following:

Algorithm 4.3 (Suboptimal design algorithm for filter banks with more than two lifting steps)**.** This algorithm consists of the following steps:

Step 1. Design a filter bank constructed with $K-2$ lifting steps with high coding gain and good frequency response, but without any vanishing moments.

Step 2. Design the $(K-1)$th lifting filter A_{K-1}. If $K \in \mathbb{Z}_{even}$, A_{K-1} is designed to provide $\tilde{N}$ dual vanishing moments. The condition is a linear system of $\lceil \tilde{N}/2 \rceil^2$ equations in the filter coefficients of A_{K-1}. Therefore, A_{K-1} should have at least $\lceil \tilde{N}/2 \rceil^2$ independent coefficients. If $K \in \mathbb{Z}_{odd}$, A_{K-1} is designed to provide N primal vanishing moments and must have at least $\lceil N/2 \rceil^2$ independent coefficients.

Step 3. Design the Kth lifting filter A_K. Similar to the previous step, for $K \in \mathbb{Z}_{even}$ and $K \in \mathbb{Z}_{odd}$, A_K is designed to provide N primal and $\tilde{N}$ dual vanishing moments, respectively. This filter must have at least $\lceil N/2 \rceil^2$ or $\lceil \tilde{N}/2 \rceil^2$ independent coefficients.

For step 1, we use Algorithm 4.1 or 4.2 with the constraint on vanishing moments being completely removed. For steps 2 and 3, since the constraint is linear in the design vector, we employ an algorithm similar to Algorithm 4.1. In this way, the resulting filter banks have desired number of (exact) vanishing moments.

4.7 Design Examples

In order to demonstrate the effectiveness of our proposed design methods, we now present several examples of filter banks constructed using our methods. For all of the design examples in this section, the optimization is carried out for maximal coding gain assuming an isotropic image model with correlation coefficient $\rho = 0.95$ and a six-level wavelet decomposition.

Several type-1 filter banks, identified by names OPT1 to OPT7, were designed using our proposed algorithms. For comparison purposes, we also consider a type-2 filter bank, referred to as TYPE2, and three filter banks, referred to by the names KS1, KS2, and 9/7, produced by methods previously proposed by others. The KS1 and KS2 filter banks are quincunx filter banks constructed using the method in [6]. The 9/7 one is the well-known separable 9/7 filter bank [40], with four primal and four dual vanishing moments. Some of the

Table 4.2: Filter bank comparison

Filter banks	Support of lifting filters†	Analysis filters		Coding gain(dB)		Vanishing moments		
		Lowpass	Highpass	Iso.	Sep.	$\tilde{N}$	N	Max.
OPT1	6×6, 6×6	13×13	7×7	12.06	13.59	2	2	
OPT2	6×6, 6×6	13×13	7×7	12.02	13.38	4	4	
OPT3	4×4, 4×4, 4×4	9×9	13×13	12.23	13.26	2	2	10^{-12}
OPT4	4×4, 4×4, 2×2, 2×2	13×13	11×11	12.21	13.07	2	2	10^{-10}
OPT5	4×4, 4×4, 2×2, 2×2	13×13	11×11	12.14	12.90	4	4	10^{-11}
OPT6	4×4, 4×4, 4×4, 4×4	17×17	13×13	12.23	13.02	2	2	10^{-8}
OPT7	2×2, 2×2, 4×4, 4×4	13×13	9×9	12.16	13.08	2	2	
TYPE2	1×1, 1×1, 5×5, 5×5	18×17	10×9	12.03	12.33	1	1	
KS1	4×4, 4×4	9×9	5×5	11.94	13.08	4	4	
KS2	6×6, 6×6	13×13	7×7	11.95	13.64	6	6	
9/7	2, 2, 2, 2	9	7	12.09	14.88	4	4	

†all of these filter banks employ lifting filters with diamond-shaped support, except TYPE2 and 9/7

important characteristics of the various filter banks are shown in Table 4.2, where the columns in order correspond to the filter bank names, support sizes of the lifting filters, support sizes of the lowpass and highpass analysis filters, six-level isotropic and separable coding gains (three-level for the 9/7 separable filter bank), numbers of dual and primal vanishing moments. Some of the filter banks are designed using Algorithm 4.2, and thus these filter banks have nearly vanishing moments. The maximal order of the moments in question for these filter banks are shown in the rightmost column in Table 4.2. The frequency responses and the scaling and wavelet functions associated with the first eight filter banks are illustrated in Figures 4.9 to 4.24. From these figures, we see that the optimal filter banks have good diamond-shaped frequency responses, and result in smooth scaling and wavelet functions. For the optimally-designed filter banks OPT1 to OPT7, the lifting filter coefficient vectors $\{\boldsymbol{a}_i\}$ as defined in (4.3) and (4.7) are given in Figures 4.6, 4.7, and 4.8.

The first two filter banks, referred to as OPT1 and OPT2, were designed using Algorithm 4.1 with two lifting steps. Exact vanishing moments are achieved through the linear constraint (4.40). Comparing OPT2 with OPT1, the degrees of freedom in the design process are reduced as the number of vanishing moments increases. The filter bank obtained with more vanishing moments has a slightly lower coding gain, but smoother frequency responses and associated scaling and wavelet functions as shown in Figures 4.9, 4.11, 4.10, and 4.12.

$$\boldsymbol{a}_1 = \begin{bmatrix} -0.0159198316 \\ 0.0570315087 \\ -0.3319070666 \\ -0.3336501890 \\ 0.0596966372 \\ -0.0177016160 \\ 0 \\ -0.0002158944 \\ 0.0584826734 \\ 0.0590711965 \\ -0.0014144431 \\ 0 \\ 0 \\ 0 \\ -0.0171945340 \\ -0.0162784411 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} 0.0141419383 \\ -0.0475750610 \\ 0.1826552865 \\ 0.1839773572 \\ -0.0501021101 \\ 0.0165757568 \\ 0 \\ 0.0073072183 \\ -0.0487234955 \\ -0.0488388947 \\ 0.0082567802 \\ 0 \\ 0 \\ 0 \\ 0.0165064152 \\ 0.0158188087 \\ 0 \\ 0 \end{bmatrix}$$

(a)

$$\boldsymbol{a}_1 = \begin{bmatrix} -0.0047050039 \\ 0.0436166542 \\ -0.3299574722 \\ -0.3296130946 \\ 0.0439752219 \\ -0.0041661713 \\ 0 \\ 0.0010341357 \\ 0.0441100617 \\ 0.0443930881 \\ 0.0004875138 \\ 0 \\ 0 \\ 0 \\ -0.0045830938 \\ -0.0045918392 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} 0.0075680097 \\ -0.0394587620 \\ 0.1877307031 \\ 0.1878726283 \\ -0.0406636719 \\ 0.0059764243 \\ 0 \\ 0.0032763980 \\ -0.0404818414 \\ -0.0406702876 \\ 0.0049627337 \\ 0 \\ 0 \\ 0 \\ 0.0072619281 \\ 0.0066257377 \\ 0 \\ 0 \end{bmatrix}$$

(b)

$$\boldsymbol{a}_1 = \begin{bmatrix} 0.0121916538 \\ -0.2252324567 \\ -0.2244562781 \\ 0.0131716139 \\ 0 \\ 0.0123383222 \\ 0.0125969226 \\ 0 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} -0.0412467652 \\ 0.2230448713 \\ 0.2234323639 \\ -0.0423652185 \\ 0 \\ -0.0429058837 \\ -0.0419932594 \\ 0 \end{bmatrix}, \boldsymbol{a}_3 = \begin{bmatrix} 0.0312090846 \\ -0.1065049947 \\ -0.1060172665 \\ 0.0301113988 \\ 0 \\ 0.0289842780 \\ 0.0317300494 \\ 0 \end{bmatrix}$$

(c)

Figure 4.6: Lifting filter coefficients for (a) OPT1, (b) OPT2, and (c) OPT3.

$$\boldsymbol{a}_1 = \begin{bmatrix} 0.0634983772 \\ -0.1474840240 \\ -0.2023765008 \\ 0.0294352099 \\ 0 \\ 0.0622324334 \\ 0.0202133422 \\ 0 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} -0.0451377582 \\ 0.0687594491 \\ 0.1518386544 \\ -0.0326419204 \\ 0 \\ -0.0460766038 \\ -0.0240443429 \\ 0 \end{bmatrix}, \boldsymbol{a}_3 = \begin{bmatrix} -0.2321916679 \\ -0.0651787971 \end{bmatrix}, \boldsymbol{a}_4 = \begin{bmatrix} 0.2012955400 \\ 0.0186944256 \end{bmatrix}$$

(d)

$$\boldsymbol{a}_1 = \begin{bmatrix} 0.0329298151 \\ -0.1520002090 \\ -0.2751639042 \\ 0.0315136289 \\ 0 \\ 0.0309640004 \\ 0.0334794436 \\ 0 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} -0.0151174693 \\ 0.0139147688 \\ 0.2249881489 \\ -0.0167397231 \\ 0 \\ -0.0139749233 \\ -0.0178822691 \\ 0 \end{bmatrix}, \boldsymbol{a}_3 = \begin{bmatrix} -0.1465265538 \\ -0.0302127013 \end{bmatrix}, \boldsymbol{a}_4 = \begin{bmatrix} 0.1594273047 \\ -0.0337055904 \end{bmatrix}$$

(e)

$$\boldsymbol{a}_1 = \begin{bmatrix} 0.0158791169 \\ -0.1702110236 \\ -0.2046008841 \\ 0.0315609365 \\ 0 \\ 0.0048148721 \\ 0.0290791393 \\ 0 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} -0.0378150856 \\ 0.1135420442 \\ 0.1761653854 \\ -0.0667691340 \\ 0 \\ -0.0283790185 \\ -0.0677823867 \\ 0 \end{bmatrix}, \boldsymbol{a}_3 = \begin{bmatrix} 0.0370390890 \\ -0.1907458715 \\ -0.0889679014 \\ 0.0022815831 \\ 0 \\ 0.0447592730 \\ 0.0032500222 \\ 0 \end{bmatrix}, \boldsymbol{a}_4 = \begin{bmatrix} -0.0101968155 \\ 0.1399326973 \\ 0.0221084949 \\ 0.0314718430 \\ 0 \\ -0.0216477949 \\ 0.0309273079 \\ 0 \end{bmatrix}$$

(f)

Figure 4.7: Lifting filter coefficients for (d) OPT4, (e) OPT5, and (f) OPT6.

$$\boldsymbol{a}_1 = \begin{bmatrix} -0.2540932200 \\ -0.2540932200 \end{bmatrix}, \boldsymbol{a}_2 = \begin{bmatrix} 0.1433256025 \\ 0.1433256025 \end{bmatrix}, \boldsymbol{a}_3 = \begin{bmatrix} 0.0421949206 \\ -0.0804671468 \\ -0.0800681185 \\ 0.0422375734 \\ 0 \\ 0.0421271748 \\ 0.0422396075 \\ 0 \end{bmatrix}, \boldsymbol{a}_4 = \begin{bmatrix} -0.0401286998 \\ 0.0604879275 \\ 0.0604248855 \\ -0.0398294167 \\ 0 \\ -0.0396522148 \\ -0.0399342818 \\ 0 \end{bmatrix}$$

(g)

Figure 4.8: Lifting filter coefficients for OPT7.

The next four filter banks, referred to as OPT3 to OPT6, were designed using Algorithm 4.2 with either three or four lifting steps. In this case, as the vanishing moment condition is approximated by a linear system (4.54), the moments in question are small but not precisely zero. For these design examples, however, the zeroth and second (if applicable) moments are near vanishing on the order of 10^{-8} to 10^{-13}, which is small enough to be considered as zero for all practical purposes. The first and third moments are automatically zero due to the linear-phase property as previously discussed in Section 4.5.1. The filter bank OPT5 has the same lifting filter supports as OPT4, but is designed to have more dual and primal vanishing moments. From Figures 4.15, 4.17, 4.16, and 4.18, we see that, compared to OPT4, OPT5 has better frequency responses and leads to smoother scaling and wavelet functions, but the coding gain of OPT5 is about 0.07 dB lower than that of OPT4.

The filter bank OPT7 was designed using the suboptimal method Algorithm 4.3, where the four lifting filters are designed using three separate steps. We see that OPT7 has a lower coding gain than filter banks obtained with Algorithm 4.2 where all of the lifting filters are jointly optimized. In this case, however, the moments in question are exactly vanishing.

The TYPE2 filter bank is an example of a type-2 filter bank. Recall that for a type-2 filter bank, the highpass analysis filter H_1 has antisymmetry. This leads to the frequency response $\hat{h}_1(\boldsymbol{\omega})$ of H_1 being zero at $[0 \;\; 0]^T$ and $[0 \;\; \pm\pi]^T$, and at least one dual and one primal vanishing moments. This type-2 filter bank has a high coding gain but poor frequency responses as the lowpass filters cannot preserve the vertical high frequencies, as illustrated in Figure 4.23.

From Table 4.2, clearly, the optimal designs, OPT1 to OPT7, have higher isotropic coding gains than the

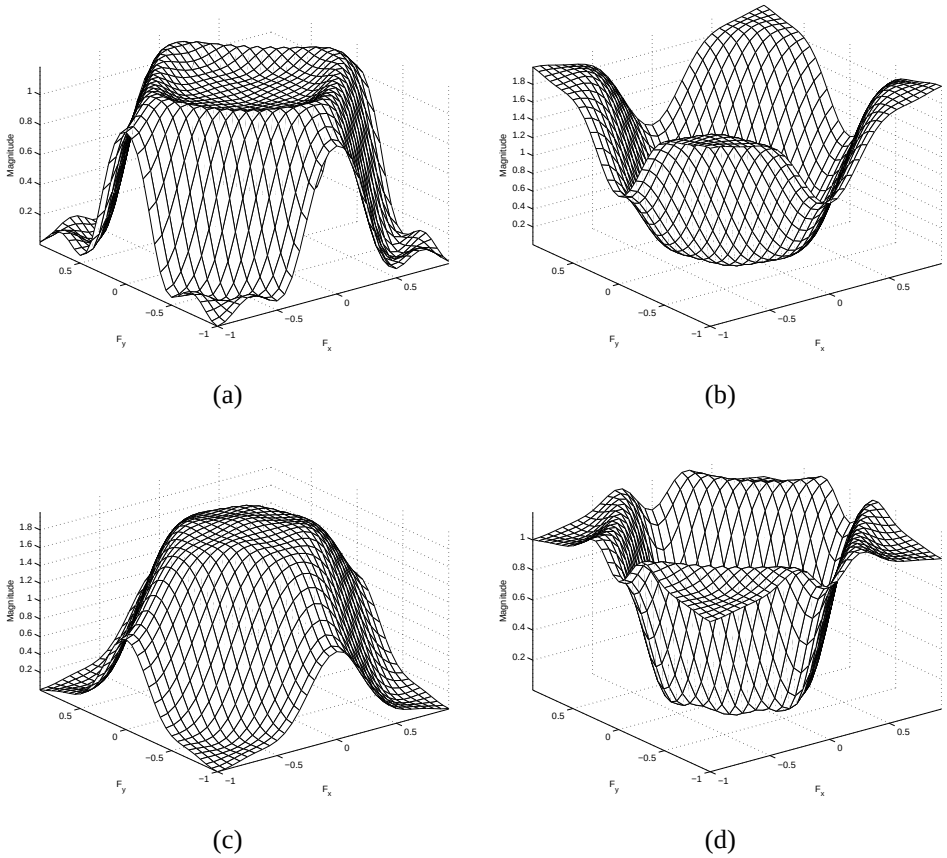
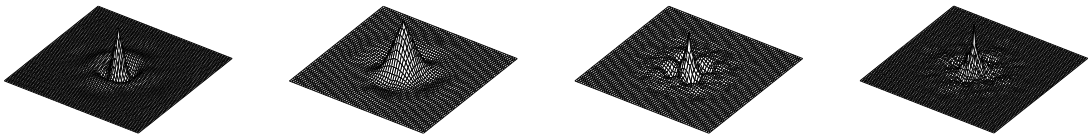


Figure 4.9: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters of OPT1.

quincunx KS1 and KS2 filter banks. Furthermore, the designs with three and four lifting steps also have a higher coding gain than the 9/7 filter bank, which is very impressive considering that the 9/7 filter bank is well known for its high coding gain.

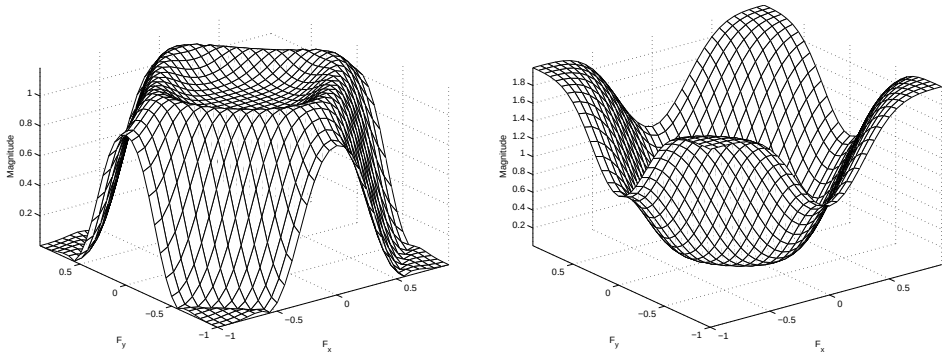
4.8 Image Coding Results and Analysis

In order to further demonstrate the utility of our new filter banks, they were employed in the embedded lossy/lossless image coder of [43]. This coder can be used with either nonseparable or separable filter banks based on the lifting framework. Reversible integer-to-integer versions of filter banks are employed. For the most part, the JPEG-2000 test images [44] were used in our experiments. Using each of the filter banks

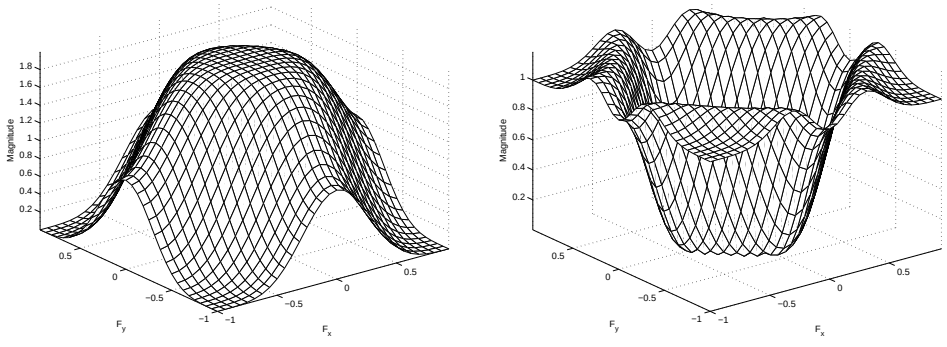


(a) (b) (c) (d)

Figure 4.10: Scaling and wavelet functions for OPT1. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

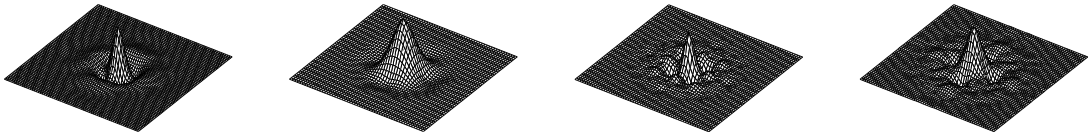


(a) (b)



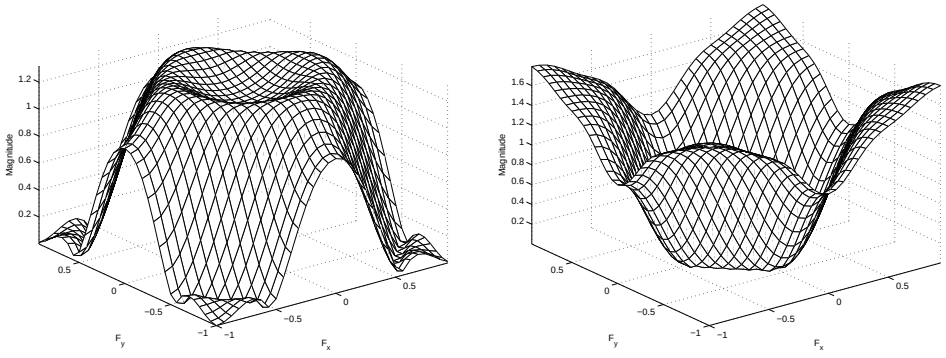
(c) (d)

Figure 4.11: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for OPT2.

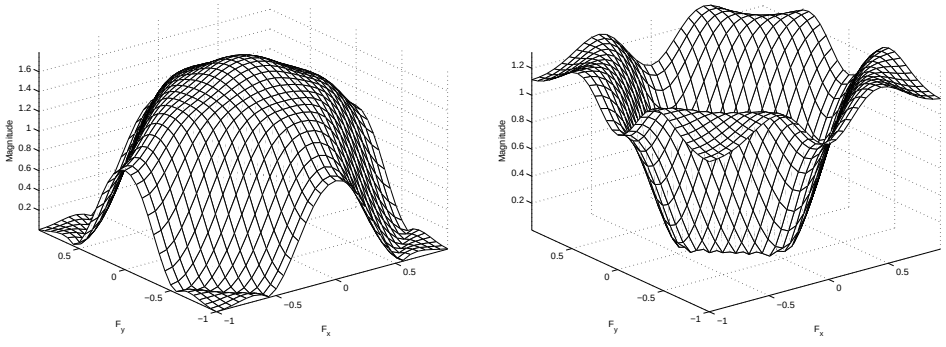


(a) (b) (c) (d)

Figure 4.12: Scaling and wavelet functions for OPT2. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

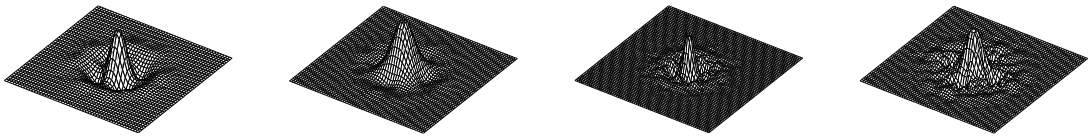


(a) (b)



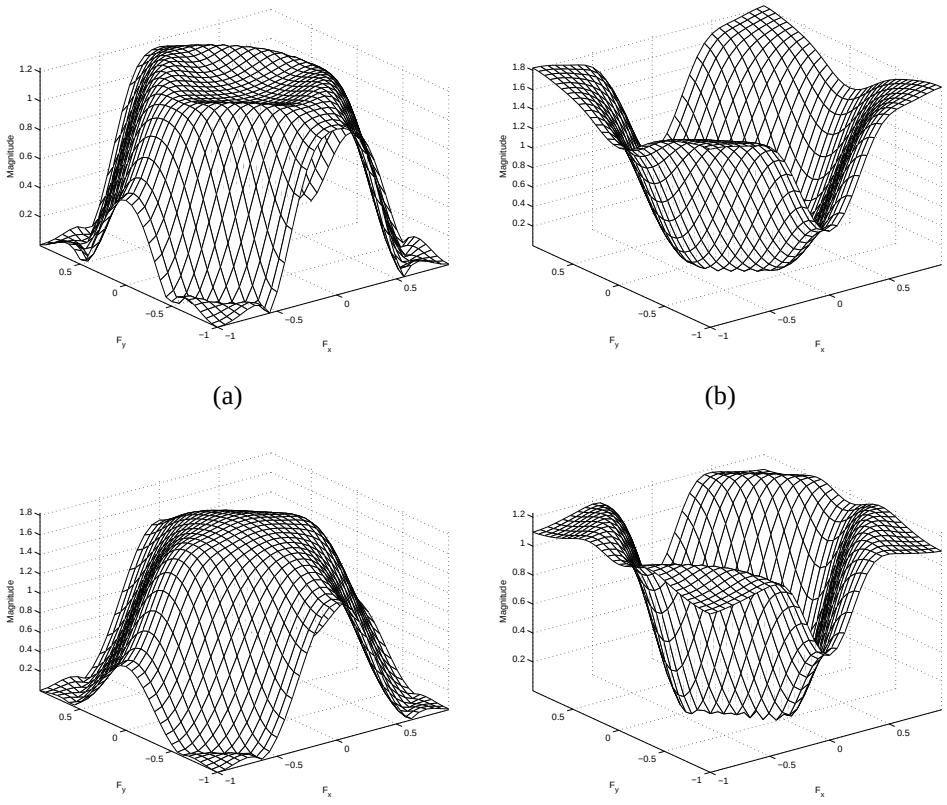
(c) (d)

Figure 4.13: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters of OPT3.



(a) (b) (c) (d)

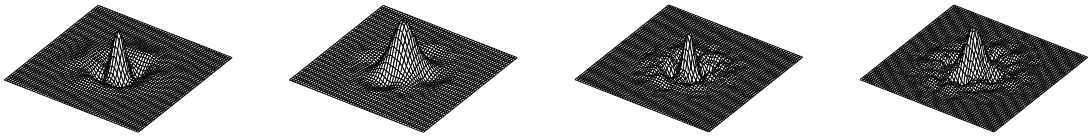
Figure 4.14: Scaling and wavelet functions for OPT3. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.



(a) (b)

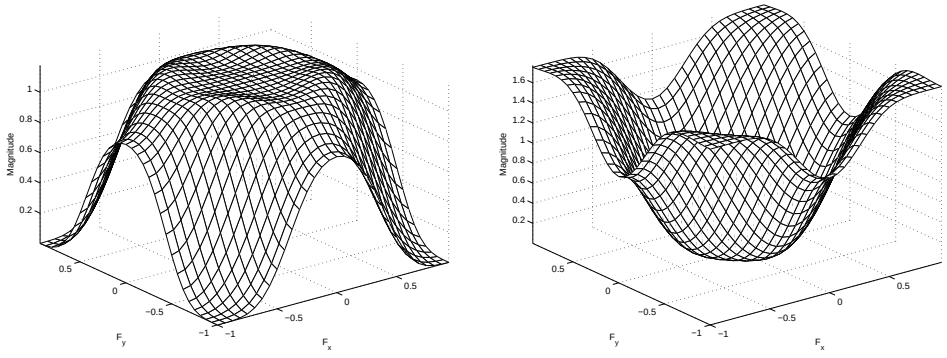
(c) (d)

Figure 4.15: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for OPT4.

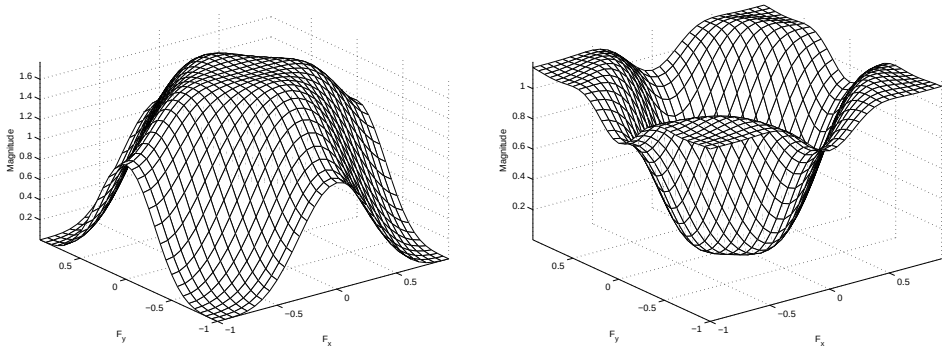


(a) (b) (c) (d)

Figure 4.16: Scaling and wavelet functions for OPT4. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

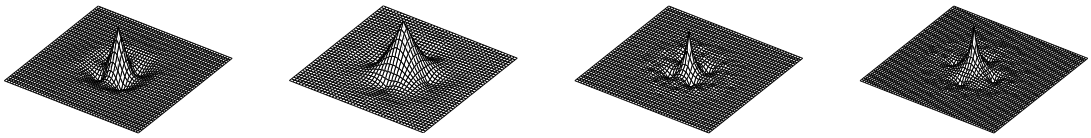


(a) (b)



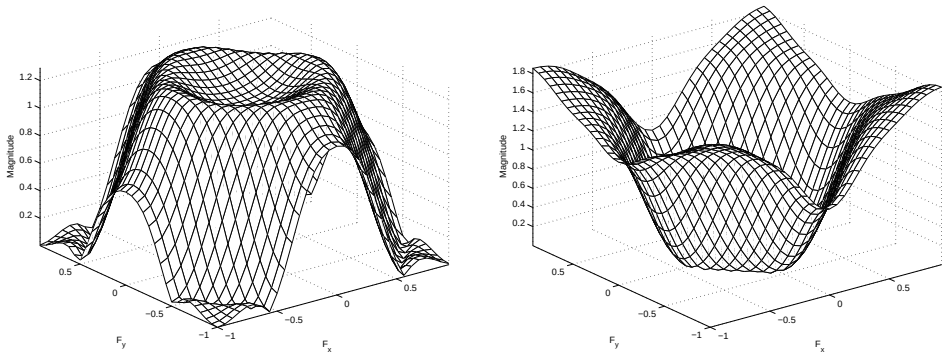
(c) (d)

Figure 4.17: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for OPT5.

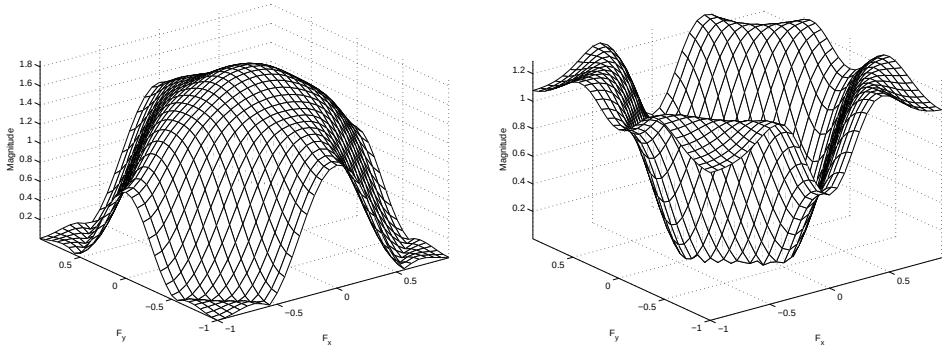


(a) (b) (c) (d)

Figure 4.18: Scaling and wavelet functions for OPT5. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

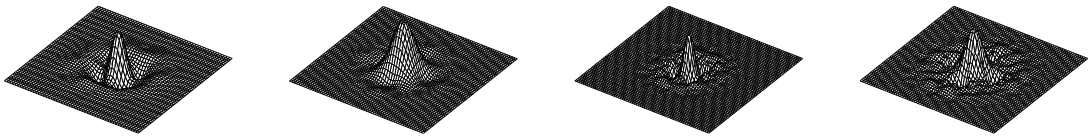


(a) (b)



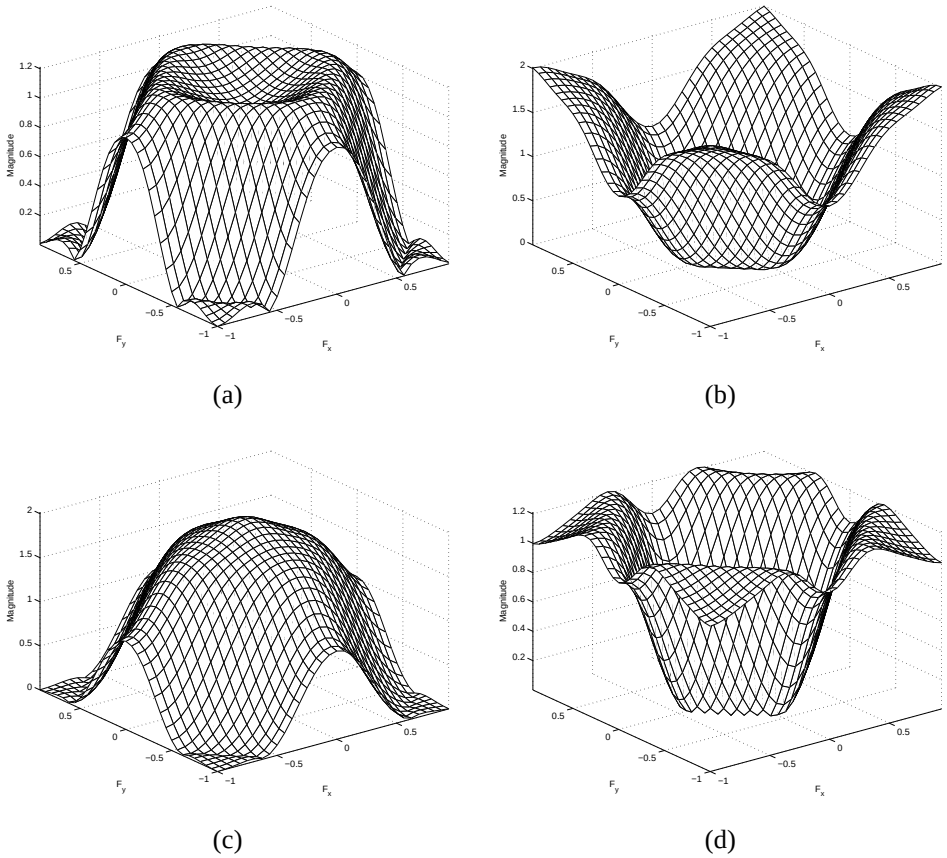
(c) (d)

Figure 4.19: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for OPT6.



(a) (b) (c) (d)

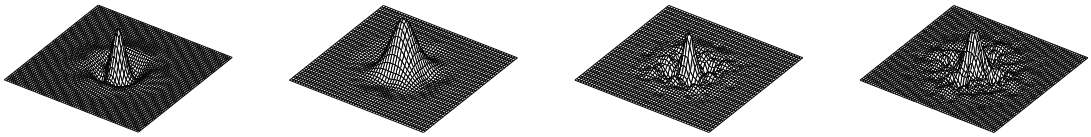
Figure 4.20: Scaling and wavelet functions for OPT6. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.



(a) (b)

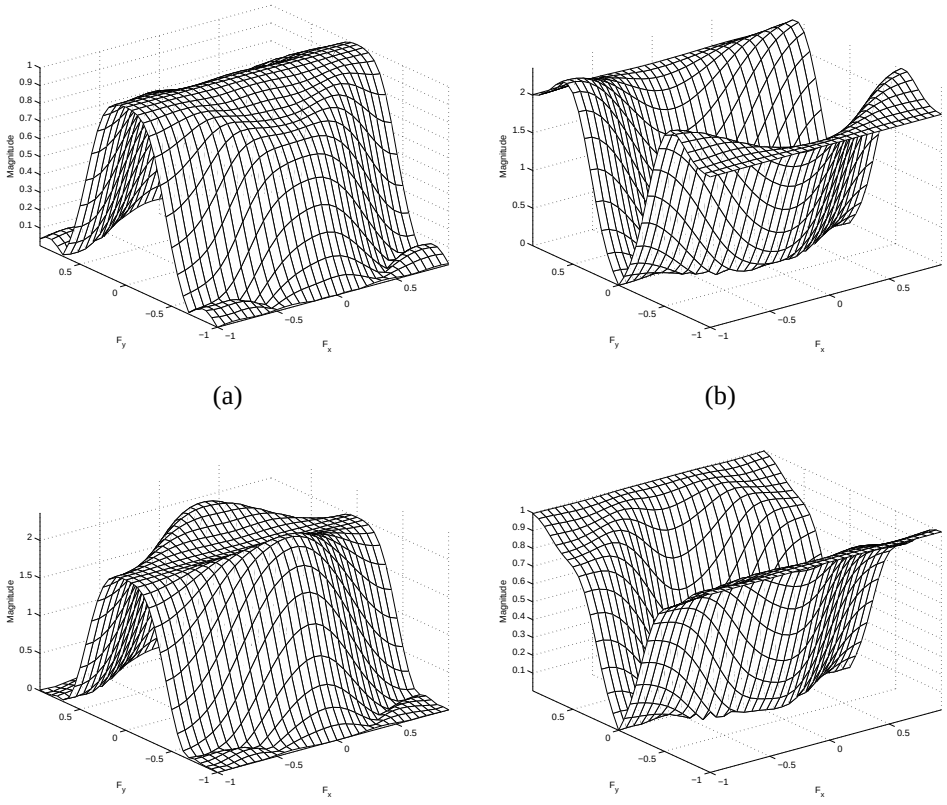
(c) (d)

Figure 4.21: Frequency responses of (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for OPT7.



(a) (b) (c) (d)

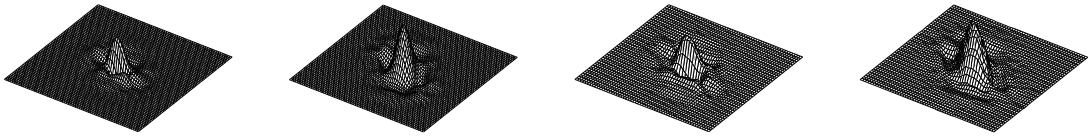
Figure 4.22: Scaling and wavelet functions associated with OPT7. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.



(a) (b)

c) (d)

Figure 4.23: Frequency responses of the (a) lowpass analysis, (b) highpass analysis, (c) lowpass synthesis, and (d) highpass synthesis filters for the type-2 filter bank.



(a) (b) (c) (d)

Figure 4.24: Scaling and wavelet functions for the type-2 filter bank. The (a) primal wavelet, (b) primal scaling, (c) dual wavelet, and (d) dual scaling functions.

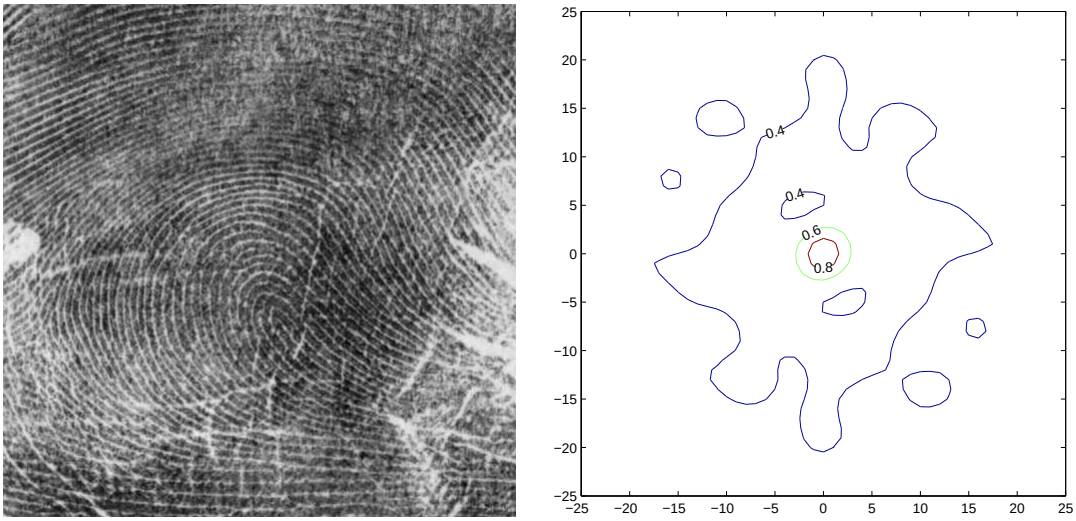
Table 4.3: Test images

Image	Size	Bits/sample	Model	Description
`finger`	512×512	8	isotropic	fingerprint
`sar2`	800×800	12	isotropic	synthetic aperture radar
`gold`	720×576	8	separable	houses

in Table 4.2, the test images were coded in a lossy manner at various bit rates, and then decoded. The error between reconstructed and original images were measured in terms of PSNR. In the cases of quincunx and separable filter banks, six and three levels of decomposition were employed, respectively.

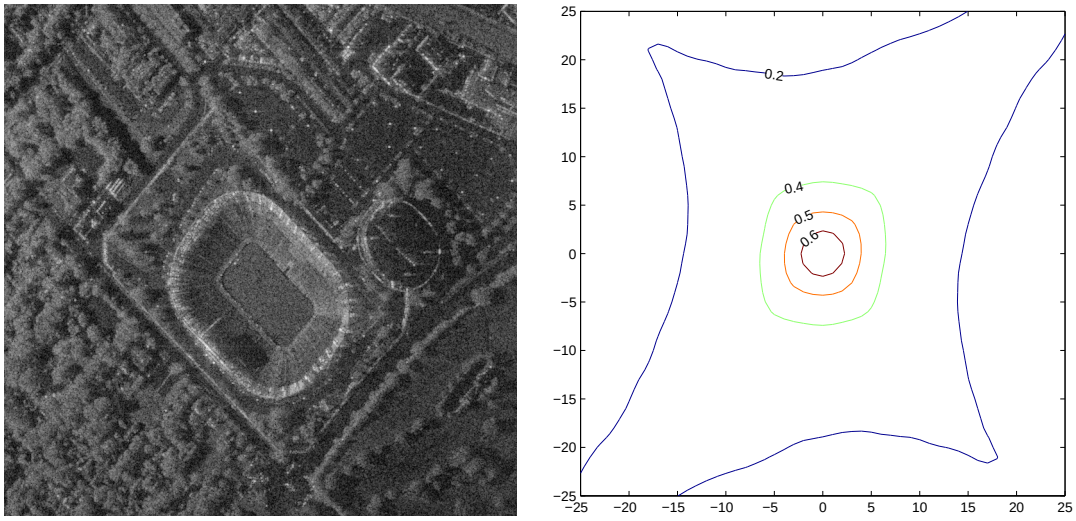
We present coding results for three images, namely the `finger`, `sar2`, and `gold` images. Information about each of these images is provided in Table 4.3. The original images and the contour plots of their normalized autocorrelation functions are illustrated in Figures 4.25, 4.26, and 4.27. We see that `finger` and `sar2` are images more isotropic in nature, while `gold` is more separable.

Since our filter banks are designed assuming an isotropic image model, we first discuss coding results for the `finger` and `sar2` images shown in Tables 4.4 and 4.5, respectively. Obviously, the optimal filter banks perform very well, consistently outperforming the KS filter banks. For the `finger` image, the OPT3 design outperforms the 9/7 filter bank, except at the lowest bit rate. For the `sar2` image, our optimal designs OPT1 to OPT6 achieve better results than the 9/7 filter bank in most cases. This is a very encouraging result, as the 9/7 filter bank is generally held to be one of the very best in the literature. The lossy reconstructed images for `finger` at the compression ratio of 32:1 using OPT1, OPT3, KS2, and 9/7 are shown in Figure 4.28. It is apparent from the figures that the reconstructed images associated with the optimal filter banks have good subjective quality. Now we consider the coding results for the `gold` image shown in Table 4.6. This image is more separable than isotropic in nature, as demonstrated by the contours of its normalized autocorrelation function plotted in Figure 4.27(b). From Table 4.6, we see that in most cases the optimally-designed filter



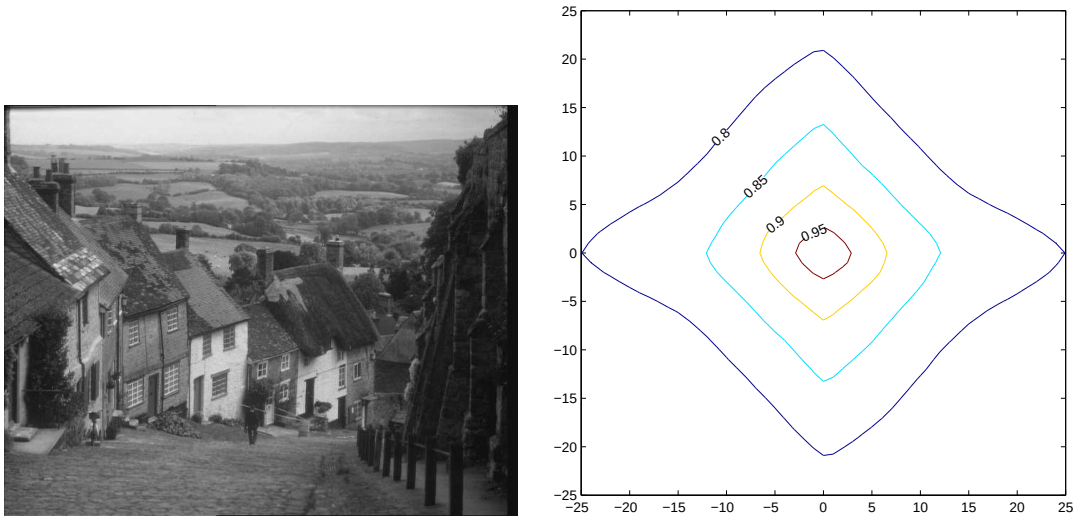
(a) (b)

Figure 4.25: (a) The `finger` image and (b) the contour plots of its normalized autocorrelation function.



(a) (b)

Figure 4.26: (a) The `sar2` image and (b) the contour plots of its normalized autocorrelation function.



(a) (b)

Figure 4.27: (a) The `gold` image and (b) the contour plots of its normalized autocorrelation function.

banks, OPT1 to OPT7, outperform the KS1 and KS2 quincunx filter banks. It is worth noting that, although we only present results for three test images herein, the optimal filter banks proposed above consistently provide better coding performance than the KS1 and KS2 filter banks for other JPEG-2000 test images in most cases. In particular, OPT1, OPT3, and OPT4 consistently outperform the previously proposed quincunx filter bank for the JPEG-2000 test images in approximately 80% cases, and OPT3 has the best overall performance among the optimally-designed filter banks.

Our experimental results in image coding show that the coding gain, frequency selectivity, vanishing moments, and the scaling and wavelet functions all have effects on the coding performance of the associated filter bank. High coding gains are important for a filter bank to achieve good coding performance. Although the coding results highly depend on the particular images, a filter bank with a high isotropic coding gain normally leads to better results in lossy coding for isotropic images. The separable coding gain is important for images that are more separable than isotropic. For example, the 9/7 filter bank has a separable coding gain of 14.88 dB, which is 1.5 to 2.2 dB higher than that of the other filter banks in Table 4.2. This filter bank outperforms the other filter banks for the separable `gold` image except at the lowest bit rate. If the analysis and synthesis filters of our optimal designs or the KS filter banks are interchanged, the resulting filter banks still have good frequency responses and smooth scaling and wavelet functions, but the coding gains are lowered by 1 to 2 dB compared to the original filter banks. When employed in the image coder, these filter banks yield low PSNR and poor subjective quality of the reconstructed images. An example of

Table 4.4: Lossy compression results for the `finger` image

Filter bank	PSNR (dB) at various CR†			
	128	64	32	16
OPT1	19.52	21.28	**24.15**	27.46
OPT2	19.41	21.18	24.04	27.38
OPT3	19.60	**21.50**	24.05	**27.49**
OPT4	19.68	21.36	24.07	27.46
OPT5	19.54	21.18	23.87	27.21
OPT6	19.69	21.42	24.12	27.48
OPT7	19.47	21.22	24.01	27.48
TYPE2	19.25	21.10	23.58	26.56
KS1	19.24	20.99	23.78	27.14
KS2	19.27	21.07	24.02	27.32
9/7	**19.70**	21.48	24.00	27.47

†compression ratio

Table 4.5: Lossy compression results for the `sar2` image

Filter bank	PSNR (dB) at various CR†			
	128	64	32	16
OPT1	22.63	23.47	24.68	26.63
OPT2	22.60	23.43	24.63	26.58
OPT3	**22.69**	23.53	**24.78**	**26.75**
OPT4	22.67	**23.54**	24.73	26.70
OPT5	22.63	23.49	24.69	26.72
OPT6	22.65	23.53	24.76	26.74
OPT7	22.59	23.47	24.67	26.68
TYPE2	22.43	23.36	24.56	26.50
KS1	22.51	23.32	24.50	26.51
KS2	22.56	23.38	24.55	26.51
9/7	22.64	23.51	24.65	26.60

†compression ratio

Table 4.6: Lossy compression results for the `gold` image

Filter bank	PSNR (dB) at various CR†			
	128	64	32	16
OPT1	**27.14**	28.84	30.82	33.27
OPT2	27.06	28.80	30.76	33.22
OPT3	**27.14**	28.87	30.91	33.32
OPT4	27.02	28.83	30.81	33.23
OPT5	26.90	28.67	30.74	33.13
OPT6	27.11	28.82	30.85	33.26
OPT7	27.04	28.81	30.80	33.18
TYPE2	26.65	28.41	30.40	32.80
KS1	26.98	28.66	30.66	33.12
KS2	27.00	28.75	30.72	33.19
9/7	27.03	**29.03**	**31.25**	**33.78**

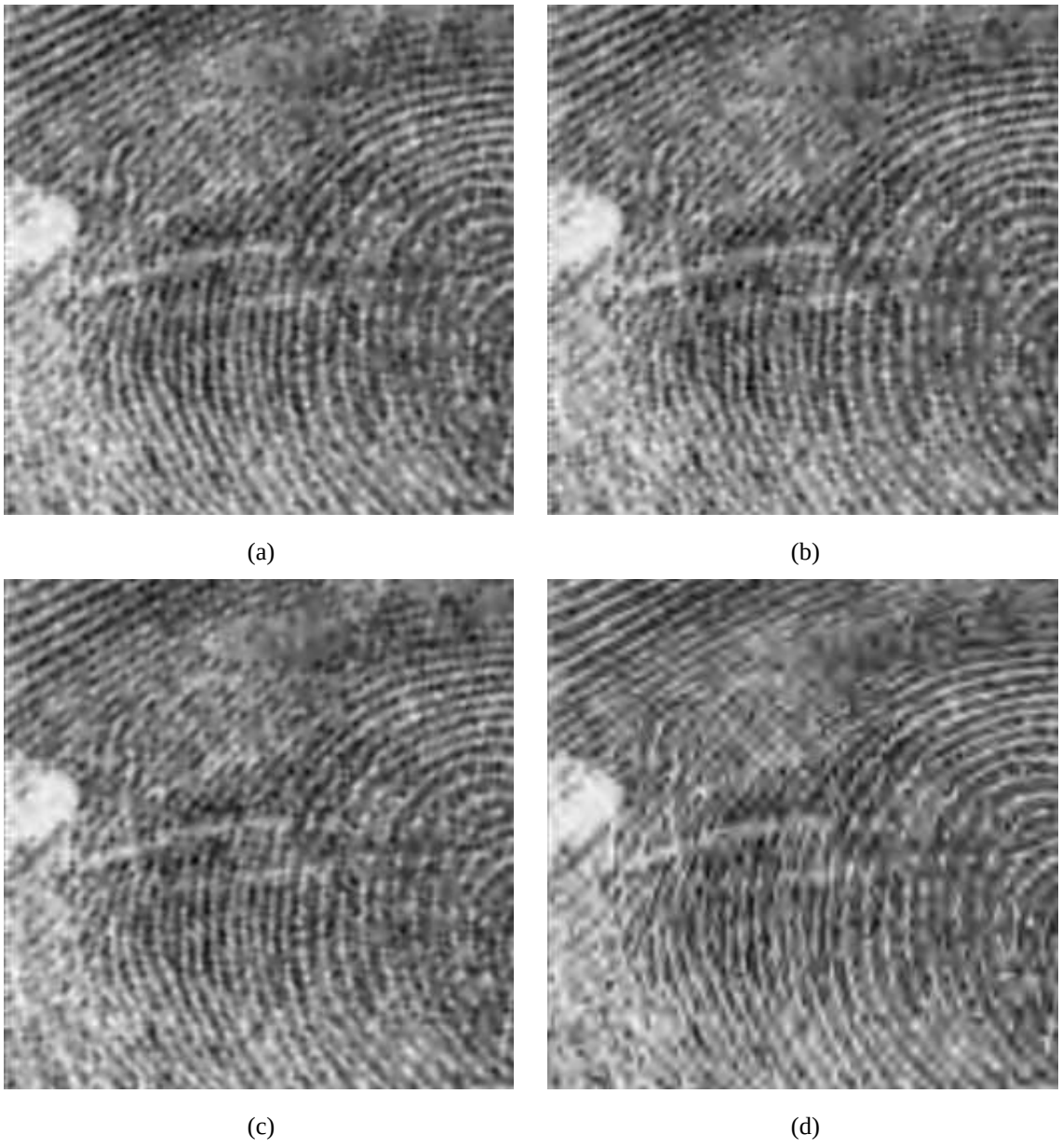
†compression ratio

the reconstructed image illustrating the above statement is shown in Figure 4.29(a), where the filter bank is derived from KS2 with interchanged analysis and synthesis filters.

The primal scaling and wavelet functions affect the subjective quality of the reconstructed image. In the case of lossy coding, the shape of these functions may appear as artifacts in reconstructed images. Figure 4.29 shows part of the lossy reconstructed images using various filter banks, including the Haar-like one from Example 3.3 discussed in Section 3.6.2. The scaling functions associated with the Haar-like filter bank have the shape of a parallelogram as illustrated in Figure 3.10. The parallelogram-shaped artifacts are clearly visible in the reconstructed image depicted in Figure 4.29(b).

The frequency selectivity is also important for a filter bank's coding performance. This is demonstrated by the coding results of the TYPE2 filter bank. Recall that this filter bank has a high coding gain of 12.03 dB, which is very close to that of OPT1 and OPT2, but the lowpass analysis filter of TYPE2 does not have a desirable diamond-shaped passband and cannot preserve high frequencies in the vertical direction. Tables 4.4, 4.5 and 4.6 show that this filter bank performs worse than OPT1 and OPT2 in all cases with 0.1 dB to 0.9 dB lower PSNR. Furthermore, comparing Figures 4.29(c) and (d), the image reconstructed using the type-2 filter bank contains less details than the image reconstructed using OPT2.

Therefore, in order to have high performance in image coding, a filter bank should have high coding gain,



(a) (b)

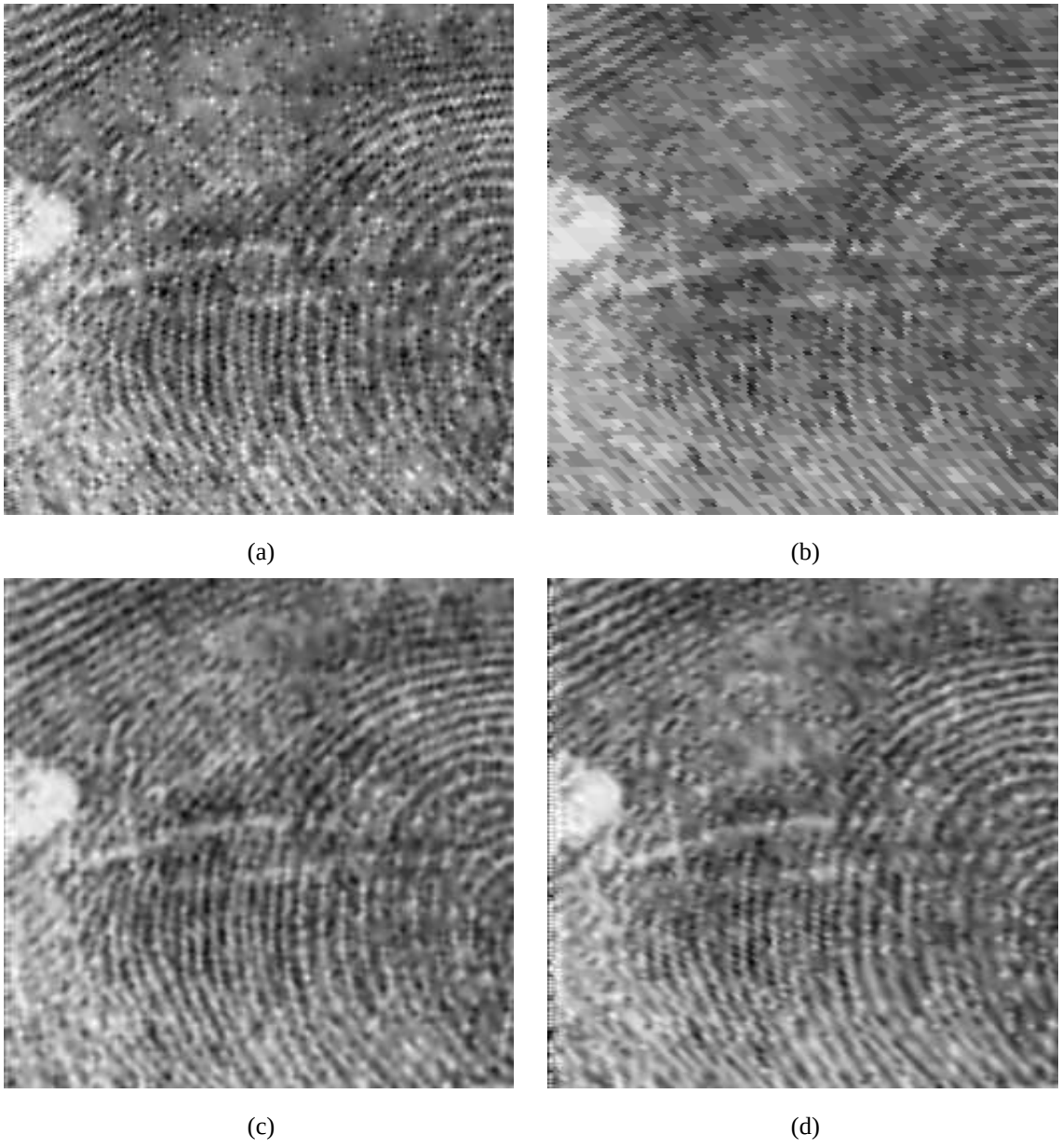
(c) (d)

Figure 4.28: Part of the reconstructed images for `finger` at the compression ratio 32 using (a) OPT1, (b) OPT3, (c) KS2, and (d) 9/7 filter banks.

smooth scaling and wavelet functions, and good frequency responses. Our optimal filter banks have all of these properties, and have proven to be very efficient for image coding.

4.9 Summary

In this chapter, we have presented our new lifting-based parameterization for three types of linear-phase PR quincunx filter banks and examined how the parametrization can be used to build filter banks with quadrantal centrosymmetry, which are compatible with the symmetric extension algorithms introduced in Chapter 3. Then, we have proposed several new optimization-based methods for the design of high-performance quincunx filter banks for the application of image coding. In our new design methods, the lifting parametrization is employed, and the coding gain is maximized subject to constraints on the vanishing moment and frequency response properties. In this way, these algorithms yield linear-phase PR quincunx systems with high coding gain, good analysis/synthesis filter frequency selectivity, and certain vanishing moment properties. Finally, we have presented several examples of filter banks designed with our method, and demonstrated by experimental results that these design examples work well for image coding.



(a) (b)

(c) (d)

Figure 4.29: Part of the reconstructed images for `finger` at compression ratio 32 using (a) KS2 with interchanged analysis and synthesis filters, (b) the Haar-like, (c) OPT2, and (d) TYPE2 filter banks.

Chapter 5

Conclusions and Future Research

5.1 Conclusions

In this thesis, we have studied the design and application of quincunx filter banks. In particular, we have introduced symmetric extension algorithms for quincunx filter banks and proposed design methods for constructing high-performance quincunx filter banks for the application of image coding.

In the 1D case, the symmetric extension technique is a commonly used solution to the boundary problem when finite-length sequences are processed. We have shown how this technique can be extended to the 2D nonseparable quincunx case. We have examined three types of quadrantally-centrosymmetric quincunx filter banks, and proposed an algorithm for each type to build nonexpansive transforms of 2D sequences defined on arbitrary rectangular regions. The type-1 algorithm can further be applied in an octave-band decomposition if the analysis filters satisfy certain additional conditions. These schemes are potentially useful in applications that process finite-extent sequences using quincunx filter banks.

Filter banks are highly effective for image coding applications. We have proposed three new optimization-based techniques for the design of quincunx filter banks for image coding. The proposed design techniques are summarized in Algorithms 4.1, 4.2, and 4.3 and yield linear-phase PR systems with high coding gain, good analysis/synthesis filter frequency responses, and certain dual and primal vanishing moment properties. In our design algorithms, a parametrization of quincunx filter banks based on the lifting scheme is employed to structurally impose the PR and linear phase properties. Then, the coding gain is maximized subject to a set of constraints on vanishing moments and frequency selectivity. Algorithm 4.1 is used to design filter banks with two lifting steps. Algorithms 4.2 and 4.3 both work for filter banks with more than two lifting steps. In

Algorithm 4.2, all of the lifting filter coefficients are jointly optimized, while Algorithm 4.3 is a suboptimal one which utilizes the first two algorithms to optimize the lifting filter coefficients in three separate steps. It is observed that although Algorithm 4.3 yields filter banks with exact vanishing moments, in general, this algorithm does not work as well as Algorithm 4.2. Another important observation is that Algorithm 4.2 yields the best filter banks among all of our designs. Design examples of filter banks with all the desirable properties were presented for each of the three techniques. These optimal filter banks were employed in an image coder and their coding performance was compared to that of some existing quincunx and separable filter banks. The experimental results show that our new filter banks outperform the previously proposed quincunx filter banks for the set of JPEG-2000 test images in most cases, and sometimes these designs are even able to outperform the 9/7 filter bank, which is considered to be one of the very best in the literature. In particular, the OPT1, OPT3, and OPT4 filter banks consistently outperform the previously proposed quincunx filter banks for the JPEG-2000 test images in approximately 80% of the cases. These results demonstrate the effectiveness of our new design techniques.

5.2 Future Research

In this thesis, we focus exclusively on quincunx filter banks, which are the simplest multidimensional nonseparable filter banks. Thus, more work can be done for the general multidimensional case. For example, symmetric extension algorithms for the face-centered-orthorhombic lattice (an extension of the quincunx lattice to the three-dimensional (3D) case) would be useful for dealing with finite-extent 3D sequences. Another example of future research in this regard is the construction of linear-phase multidimensional, multichannel, nonseparable filter banks based on the lifting structure. In particular, it would be helpful to parameterize the lifting filters such that the resulting analysis and synthesis filters have desired linear-phase properties.

In our optimal design algorithms for linear-phase PR quincunx filter banks, we assume either an isotropic or a separable image model when maximizing the coding gain, while most images are not completely separable or isotropic in nature. Therefore, if the separable and isotropic coding gains could be optimized jointly, the resulting optimal filter bank may achieve better performance in image coding.

In the filter bank design algorithms, we also used a diamond-shaped passband for the ideal frequency response. With the quincunx sampling matrix, however, the diamond shape is not the only possibility. Although this kind of passband matches the human visual system and is desirable in image coding, in some cases, a hexagonal- or fan-shaped passband may be more suitable. Thus, it would be useful to design quincunx filter banks for different passband shapes.

The type-2 and type-3 filter banks introduced in Section 4.3.2 have antisymmetric highpass filters. As mentioned previously, it is impossible for these filter banks to have good diamond-shaped frequency responses, thus limiting their utility for image coding. This said, however, such filter banks may be advantageous in other applications which employ hexagonal- or fan-shaped passband/stopband. Therefore, algorithms for the design of type-2 and type-3 filter banks can be helpful for these applications.

Bibliography

[1] R. Ansari, H. Gaggioni, and D. G. LeGall, "HDTV coding using a nonrectangular subband decomposition," in *Proceedings of SPIE Visual Communications and Image Processing*, Jan. 1988, pp. 821–824.

[2] 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. 29–41, Jan. 1991.

[3] J. Kovačević and M. Vetterli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbb{R}^n$," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 533–555, 1992.

[4] J. Kovačević and M. Vetterli, "Nonseparable two- and three-dimensional wavelets," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1269–1273, May 1995.

[5] A. Gouze, M. Antonini, and M. Barlaud, "Quincunx lifting scheme for lossy image compression," in *Proceedings of IEEE International Conference on Image Processing*, Sept. 2000, vol. 1, pp. 665–668.

[6] J. Kovačević and W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 480–496, Mar. 2000.

[7] S. C. Chan, K. S. Pun, and K. L. Ho, "On the design and implementation of a class of multiplierless two-channel 1D and 2D nonseparable PR FIR filterbanks," in *Proceedings of IEEE International Conference on Image Processing*, Oct. 2001, vol. 2, pp. 241–244.

[8] D. Van De Ville, T. Blu, and M. Unser, "On the multidimensional extension of the quincunx subsampling matrix," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 112–115, Feb. 2005.

[9] M. Vetterli, J. Kovačević, and D. LeGall, "Perfect reconstruction filter banks for HDTV representation and coding," *Signal Processing: Image Communication*, vol. 3, pp. 349–364, 1990.

[10] M. Vetterli and J Kovačević, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[11] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2002.

[12] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge, Wellesley, MA, 1996.

[13] M. J. T. Smith and S. L. Eddins, "Subband coding images with octave band tree structures," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1987, pp. 1378–1381.

[14] M. J. T. Smith and S. L. Eddins, "Analysis/synthesis techniques for subband image coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 8, pp. 1446–1456, Aug. 1990.

[15] S. A. Martucci, "Signal extension and noncausal filtering for subband coding of images," in *Proceedings of SPIE Visual Communications and Image Processing*, Nov. 1991, vol. 1605, pp. 137–148.

[16] C. M. Brislawn, "Preservation of subband symmetry in multirate signal coding," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 3046–3050, Dec. 1995.

[17] J. H. McClellan, "The design of two-dimensional digital filters by transformation," in *Proc. 7th Annual Princeton Conf. Info. Sci. and Syst.*, Princeton, NJ, Mar. 1973, pp. 247–251.

[18] 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, 1993.

[19] T. A. C. M. Kalker and I. A. Shah, "A group theoretic approach to multidimensional filter banks: Theory and applications," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1392–1405, 1996.

[20] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Upper Saddle River, NJ, 1993.

[21] 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.

[22] 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, Mar. 1995.

[23] K. S. C. Pun and T. Q. Nguyen, "A novel and efficient design of multidimensional PR two-channel filter banks with hourglass-shaped passband support," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 345–348, Mar. 2004.

[24] 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, 1990.

[25] T. D. Tran, R. L. de Queiroz, and T. Q. Nguyen, "Linear-phase perfect reconstruction filter bank: Lattice structure, design, and application in image coding," *IEEE Transactions on Signal Processing*, vol. 48, no. 1, pp. 133–147, Jan. 2000.

[26] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, pp. 186–200, 1996.

[27] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," .

[28] T. Cooklev, A. Nishihara, T. Yoshida, and M. Sablatash, "Multidimensional two-channel linear phase FIR filter banks and wavelet bases with vanishing moments," *Journal of Multidimensional Systems and Signal Processing*, vol. 9, pp. 39–76, Jan. 1998.

[29] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1984.

[30] S. G. Mallat, "A theory of multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[31] O. Rioul, "A discrete-time multiresolution theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 8, pp. 2591–2606, Aug. 1993.

[32] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 2091–2110, Dec. 1989.

[33] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.

[34] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," in *Proceedings of SPIE Visual Communications and Image Processing*, Nov. 1991, vol. 1605, pp. 95–106.

[35] E. A. B. da Silva, *Wavelet Transforms for Image Coding*, Ph.D. thesis, Department of Electronic Systems Engineering, University of Essex, UK, 1995.

[36] Y. Chen, M. D. Adams, and W.-S. Lu, "Symmetric extension for quincunx filter banks," in *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Aug. 2005, pp. 542–545.

[37] Y. Chen, M. D. Adams, and W.-S. Lu, "Symmetric extension for two-channel quincunx filter banks," in *Proceedings of IEEE International Conference on Image Processing*, Sept. 2005, vol. 1, pp. 461–464.

[38] S. Coulombe and E. Dubois, "Linear phase and symmetries for multidimensional FIR filters over lattices," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 45, no. 4, pp. 473–481, Apr. 1998.

[39] Y. Chen, M. D. Adams, and W.-S. Lu, "Design of optimal quincunx filter banks for image coding," in *Proceedings of IEEE International Symposium on Circuits and Systems*, May 2006, to appear.

[40] *ISO/IEC 15444-1: Information technology—JPEG 2000 image coding system—Part 1: Core coding system*, 2000.

[41] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and Applications*, vol. 284, pp. 193–228, Nov. 1998.

[42] J.F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.

[43] M. D. Adams, "ELEC 545 project: A wavelet-based lossy/lossless image compression system," Dept. of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, Apr. 1999.

[44] "JPEG-2000 test images," ISO/IEC JTC 1/SC 29/WG 1 N 545, July 1997.

UNIVERSITY OF VICTORIA PARTIAL COPYRIGHT LICENSE

Title of Thesis:

Design and Application of Quincunx Filter Banks

Author _____

Yi Chen

Signed: May 15, 2006