Chapter 1

Introduction and Contribution

After Grossman and Morlet introduced the *Continuous Wavelet Transform* in their seminal paper [3], wavelets unified or at least attracted large number of researchers in diverse fields such as mathematics, signal/image processing, computer vision, statistics etc. For this reason it experienced an explosive growth in the last decade. It is this fascination about wavelets which attracted me to work on this thesis. Though initially I was trying to work on *Wavelets and Applications* with no specific focus, later on the research got focused on *Wavelet Filter Banks and Digital Watermarking.* Thus the thesis consists of two parts, wavelet filter banks and Digital Watermarking.

Many results of wavelet theory were already in use before wavelet theory came to the attention of the signal processing community, in the form of quadrature mirror filters [35, 34] and pyramid algorithms [19, 14]. It was Burt and Adelson's pyramid algorithm [19] which inspired Stephane Mallat to develop *Multi-resolution Analysis* in conjunction with Yves Meyer. Wavelet theory introduced two new concepts to the filter bank community, *regularity* and *vanishing moments.* The theme of the first part of the thesis is to construct various filter banks with vanishing moments. The first part attacks three kind of filter banks, *M-band Bi-orthogonal Filter Banks, Double Density Filter banks and Its Generalizations* and *Filter Banks on the Hexagonal Lattice.*

Wavelet transform decomposes the frequencies into uniform subbands in the logarithmic scale. In the two band setting each low pass subband is subdivided repeatedly to achieve uniform subdivision in the logarithmic scale. The M-band wavelet transform can achieve finer subbands via decomposing each low pass subband into M components. The M-band bi-orthogonal wavelet transform is a natural extension to the M-band orthogonal wavelet transform. M-band bi-orthogonal wavelets are discussed in chapter 4. Like with classical approaches [16, 1, 38], our approach first designs scaling filters and then completes the filter bank to obtain wavelet filters. The design of scaling filters is similar to that of Daubechies et all. [1]. My contribution in this chapter is on filter bank completion with the constraints that each wavelet filter is linear phase symmetric and the analysis (synthesis) filters are of same size. We first show how *K-Regularity* of scaling filters transfer into vanishing moments of wavelet filters. Then we construct the filter banks with shortest symmetric analysis wavelet filters. However, the general problem remains unresolved.

The traditional discrete wavelet transform suffers *shift sensitivity* in that a shift in the input signal may cause unpredictable change in the transform coefficients. Removing shift sensitivity implies having *translation invariance.* Translation invariance is highly sought by computer vision and pattern recognition community. By dropping *critically sampled property* we can improve on the translation invariance of the wavelets. Such wavelets are more commonly referred to as *framelets.* One way of getting rid of critically sampled property is to add more subbands. This way, we lead into Double Density

Filter Banks (DDFB) and its generalizations. The double density filter banks are discussed in chapter 5. My contribution in this chapter is a factorization approach to construct wavelet filters. We first analyse how K-regularity of scaling filters transfers into vanishing moments of wavelet filters. The major steps of our approach is similar to that of Selesnick's [18], but we provide a more direct factorization approach for certain special cases of DDFB's and its generalizations. We identify that the determinant of the transfer polyphase matrix of wavelet filters is crucial in obtaining simpler factorization methods, and show when determinant of the transfer polyphase matrix of wavelet filters is a real number. Then we provide alternative factorization methods and number of example DDFB's explicitly solved. We also discuss how our approach generalizes into Multiple Density Filter Banks (MDFB's).

Traditionally discrete images are represented on a rectangular lattice. It has been shown that hexagonal lattice can optimally pack points and also has improved directional selectivity. Chapter 6 discusses two dimensional filter banks on the hexagonal lattice. Our initial goal of the chapter is to construct hexagonal wavelet filters from hexagonal scaling filters. Mainly due to the complexity of the two dimension, we were unable to obtain a systematic approach for the design of hexagonal wavelet filters. However, I have provided analysis of such filters with regularity.

Finally, we looked at an application of wavelet transform, *digital watermarking of images in wavelet domain.* When watermarking images we hide invisible signals in the images to improve the security of such images. These watermarks are usually random signals which are added to the image or some meaningful signals such as images. A particular watermarking algorithm is discussed in chapter 7. Our goal in the design of the watermarking algorithm is to achieve compression tolerance. Our approach is to embed each bit of a binary watermark image into a feature of the original image in wavelet domain. This feature is simply a rectangular area of the original image in wavelet domain. We further discuss algorithms to organize feature blocks and binary watermark images into a multi-resolution structure, which will enhance the compression tolerance.

The thesis is organized as follows. The foundational aspects are discussed in chapters 2 and 3 while original work and results are discussed in chapters 4, 5, 6, and 7. The chapter 2 is devoted to the discussion of foundational aspects of filter banks while chapter 3 is devoted to the discussion of foundational aspects of wavelet bases.

In watermarking, my publications are [4, 5, 26]. In wavelet filters banks, two articles are in preparation, *Theory and Design of Biorthogonal M-band wavelets* and *Theory and Design of Double Density Filterbanks and framelets.*

Chapter 2

Filter Bank Theory

Dense sets of data are collected or sampled for speech, image and video are filtered to extract features, clean noise, compress etc. Commonly, filtering involves transforming the raw data by a convolution. The purpose of filtering may vary and can be to accentuate the desired properties at the expense of nuisance, decorrelate, or simply to express the data or signals in a domain or a language which simplify our purpose. After certain modification in the transform domain, we may want to represent the data or signals in the original form via a reverse transform. Thus filtering usually constitute a two stage process , a forward filtering (forward transform) and a reverse filtering (reverse transform).

Filtering of signals is better organized into filter banks. Thus filter banks are convolutional structures that have been used in sub-band coders for speech, image, and video signals. In a filter bank, a data sequence $x(n)$ is decomposed into M channels, called *sub-bands*, by convolving with sequences $\tilde{h}_i(n), i = 0, 1, ..., M-1$, called the analysis filters, then *down-sampled* (or *decimated*) by M_i on each channel, then *up-sampled* (or *expanded* by M_i on each channel, and then convolved with the sequences $h_i(n), i = 0, 1, ..., M-1$,

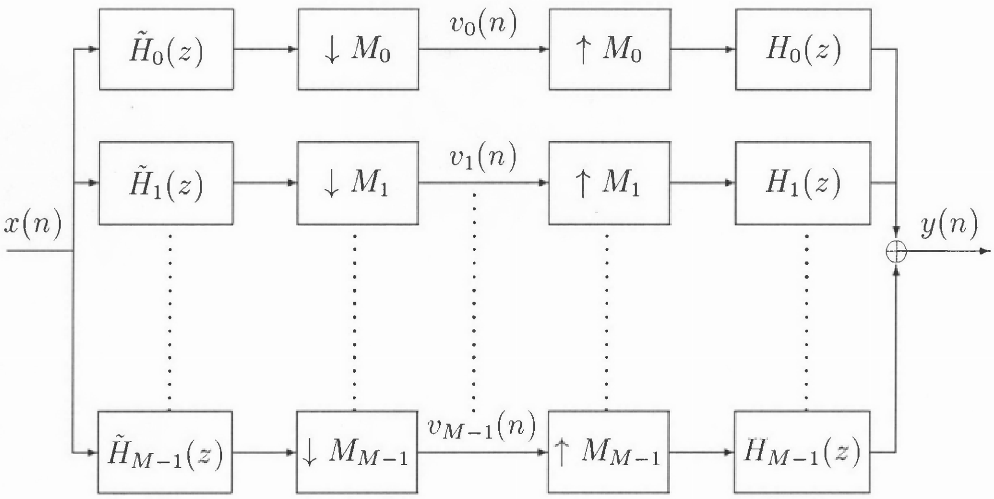

Figure 2.1: The general filter bank

called the synthesis filters, and finally recombined to give $y(n)$. Thus each channel or subband consists of a forward filter, a down-sampler, up-sampler and a reverse filtering stage. When $y(n) = x(n)$, or up to a delay, i.e. $y(n) = x(n-d)$ for some delay d, we say that the filter bank is a *perfect reconstruction filter bank.* When the number of data samples generated in all the channels after down-sampling is equal to the original number of data samples, we say that the filter bank is *maximally decimated* in which case we have $\sum_{i=0}^{M-1} \frac{1}{M_i} = 1$. When down-sampling factors are equal we say that the filter bank is *uniform.* The structure of the classical maximally decimated filter bank problem is given in Figure 2.1.

2.1 One Dimensional Filter Banks

The fundamental tool of analyzing discrete time signals is the z-transform [21]. The two-sided z-transform can be defined as

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n} \tag{2.1}$$

where $x(n)$ represent the value of the signal x at the time n. When we denote signals in time domain we represent them using lowercase letters while when we represent them in z-domain we denote them in uppercase letters. A similar convention is used to denote filters. The n^{th} coefficient of the filter h is denoted by $h(n)$ while the z-transform of the filter is denoted by $H(z)$. The difference between a filter and a signal is that the filter is a linear operator. When we want to denote a filter we alternatively use lower case letters or uppercase letters, i.e. h or H. Instead of solving the equations in time domain, we solve them in z-domain since z-domain simplifies many operators such as convolution ($H * X(z) = H(z)X(z)$), which are rather tedious in time domain. Thus in z-domain, the filtering of a signal x by the filter h is given by $H(z)X(z)$.

It can be seen that z-transform converges uniformly for all $\mid z \mid = \mid z_i \mid$ if it converges uniformly for $z = z_i$. Thus, in general, the *Region Of Convergence* (ROC) of z-transform is an circular region in the z-plane. The *unit circle* plays a special role when we talk about z-transforms. The set of points such that $\mid z \mid = 1$ is defined as the unit circle. When ROC of the z-transform includes the unit circle, z-transform reduces to the discrete time Fourier transform which is defined as

$$X^f(\omega) = X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega}. \tag{2.2}$$

Figure 2.2: The decimator and expander

2.1.1 Decimators and Expanders

The frequently used operations in signal processing are *decimators* and *expanders*. An M-fold decimator (down-sampler) takes an input sequence $x(n)$ and produces the output sequence $y_D(n)$ defined as

$$y_D(n) = x(Mn) \tag{2.3}$$

where M is an integer. Thus the decimator retains every M^{th} sample. The M-fold expander (up-sampler) takes an input sequence $x(n)$ and produces the output sequence $y_E(n)$ defined as

$$y_E(n) = \begin{cases} x(n'), & \text{if } n = Mn' \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

The down-sampling by M and the up-sampling by M are symbolically represented as in figure 2.2.

We also use the operator notation, $[\downarrow M]$ for the decimator and $[\uparrow M]$ for the expander. It is evident that the decimator in general results in loss of information while expander does not cause loss of information. The decimators and expanders as building blocks leads to *multirate signal processing*, see Vaidyanathan [36]. The advent of *polyphase representation* [28, 37] lead to a great simplification of the theoretical results and efficient implementations.

2.1.2 Polyphase Representation

The essential idea of polyphase representation is to avoid unnecessary computation of the filter outputs. For example, it is not necessary to compute filter outputs of the analysis filter bank at all discrete time instants since only every Mth output is required. A given z-transform $X(z)$ can be written in the form of either [1],

$$X(z) = \sum_{k=0}^{M-1} z^{-k} X_k(z^M) \quad \text{Type 1 polyphase} \tag{2.5}$$

where $X_k(z) \stackrel{\text{def}}{=} \sum_n x(Mn+k)z^{-n}$, or

$$X(z) = \sum_{k=0}^{M-1} z^{k} X_k(z^M) \quad \text{Type 2 polyphase} \tag{2.6}$$

where X_k is alternatively defined as $X_k(z) \stackrel{\text{def}}{=} \sum_n x(Mn-k)z^{-n}$.

Lets $\tilde{H}_i(z)$ and $H_i(z)$ denote the z-transforms of the analysis and synthesis filters. Then we write $\tilde{H}_i(z)$ and $H_i(z)$ as

$$\tilde{H}_i(z) = \sum_{k=0}^{M-1} z^{-k} \tilde{H}_{i,k}(z^M), \tag{2.7}$$

$$H(z) = \sum_{k=0}^{M-1} z^{k} H_{i,k}(z^M) \tag{2.8}$$

[1]Note that the type 2 polyphase representation has been alternatively defined in [36] as

$$X(z) = \sum_{k=0}^{M-1} z^{-(M-1-k)} X_k(z^M)$$

where

$$\tilde{H}_{i,k}(z) \stackrel{\text{def}}{=} \sum_{n} \tilde{h}_i(Mn+k)z^{-n}, \tag{2.9}$$

$$H_{i,k}(z) \stackrel{\text{def}}{=} \sum_{n} h_i(Mn-k)z^{-n}. \tag{2.10}$$

Now we analyze perfect reconstruction constraints of the filter bank.

2.1.3 Perfect reconstruction Filter Banks

We consider perfect reconstruction constraints of uniform filter banks only, i.e. $M_i = M$ [36, 43]. From figure 2.1, we have

$$\begin{aligned} V_i(z) &= [\downarrow M]\tilde{H}_i(z)X(z) \\ &= [\downarrow M]\left[\sum_{k=0}^{M-1} z^{-k}\tilde{H}_{i,k}(z^M)\right]\left[\sum_{j=0}^{M-1} z^{j}X_j(z^M)\right] \\ &= \sum_{k=0}^{M-1} \tilde{H}_{i,k}(z)X_k(z) \end{aligned} \tag{2.11}$$

where X_j are type 2 polyphase components of $X(z)$ while $\tilde{H}_{i,k}$ are type 1 polyphase components of $\tilde{H}_i$. The last step follows from the fact that

$$[\downarrow M]z^k R(z^M) = \begin{cases} z^m R(z) & \text{for } k = mM \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

Similarly, for the synthesis bank we have

$$\begin{aligned}\sum_{k=0}^{M-1} z^k Y_k(z^M) = Y(z) &= \sum_{i=0}^{M-1} \{[\uparrow M]V_i(z)\} H_i(z) \\ &= \sum_{i=0}^{M-1} V_i(z^M) H_i(z) \\ &= \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} z^j V_i(z^M) H_{i,j}(z^M)\end{aligned} \tag{2.13}$$

where Y_k and $H_{i,j}$ are type 2 polyphase components of Y and H_i respectively. Equating like powers of z we get

$$Y_k(z) = \sum_{i=0}^{M-1} H_{i,k}(z) V_i(z). \tag{2.14}$$

We define type 1 polyphase component matrix of analysis filters and type 2 polyphase component matrix of synthesis filters as

$$\tilde{H}_p(z) = \begin{bmatrix} \tilde{H}_{0,0}(z) & \tilde{H}_{0,1}(z) & \dots & \tilde{H}_{0,M-1}(z) \\ \tilde{H}_{1,0}(z) & \tilde{H}_{1,1}(z) & \dots & \tilde{H}_{1,M-1}(z) \\ \vdots & \vdots & \dots & \vdots \\ \tilde{H}_{M-1,0}(z) & \tilde{H}_{M-1,0}(z) & \dots & \tilde{H}_{M-1,M-1}(z) \end{bmatrix}, \tag{2.15}$$

$$H_p(z) = \begin{bmatrix} H_{0,0}(z) & H_{0,1}(z) & \dots & H_{0,M-1}(z) \\ H_{1,0}(z) & H_{1,1}(z) & \dots & H_{1,M-1}(z) \\ \vdots & \vdots & \dots & \vdots \\ H_{M-1,0}(z) & H_{M-1,0}(z) & \dots & H_{M-1,M-1}(z) \end{bmatrix}. \tag{2.16}$$

We can write equations 2.11 and 2.14 in matrix form as

$$\mathbf{v}(z) = \tilde{H}_p(z)\mathbf{x}_p(z), \tag{2.17}$$

$$\mathbf{y}_p(z) = H_p^T(z)\mathbf{v}(z) \tag{2.18}$$

where

$$\mathbf{x}_p(z) = [X_0(z), X_1(z), ..., X_{M-1}(z)]^T,$$

$$\mathbf{y}_p(z) = [Y_0(z), Y_1(z), ..., Y_{M-1}(z)]^T,$$

$$\mathbf{v}(z) = [V_0(z), V_1(z), ..., V_{M-1}(z)]^T.$$

Now we arrive at

$$\mathbf{y}_p(z) = H_p^T(z)\tilde{H}_p(z)\mathbf{x}_p(z). \tag{2.19}$$

Since $[\downarrow M]z^{-k}X(z) = X_k(z)$, for $0 \leq k \leq M-1$, we can express equation 2.19 in the graphical form as in the figure 2.3. When $Y(z) = X(z)$, it must be that

$$H_p^T(z)\tilde{H}_p(z) = I. \tag{2.20}$$

The most general form of $H_p^T(z)\tilde{H}_p(z)$ has the form

$$H_p^T(z)\tilde{H}_p(z) = cz^{-m}\begin{bmatrix} \mathbf{0}_{r\text{X}M-r} & zI_{r\text{X}r} \\ I_{M-r\text{X}M-r} & \mathbf{0}_{M-r\text{X}r} \end{bmatrix} \tag{2.21}$$

for some integer r with $0 \leq r \leq M-1$, some integer m, and some constant $c \neq 0$ [36]. Under this condition the reconstructed signal is $y(n) = cx(n-d)$, where $d = Mm - r$. The result can be easily seen by first writing both $X(z)$ and $Y(z)$ in terms type 2 polyphase representation and then multiplying $X(z)$ by cz^{-d} and finally equating polyphase components of $Y(z)$ and $cz^{-d}X(z)$.

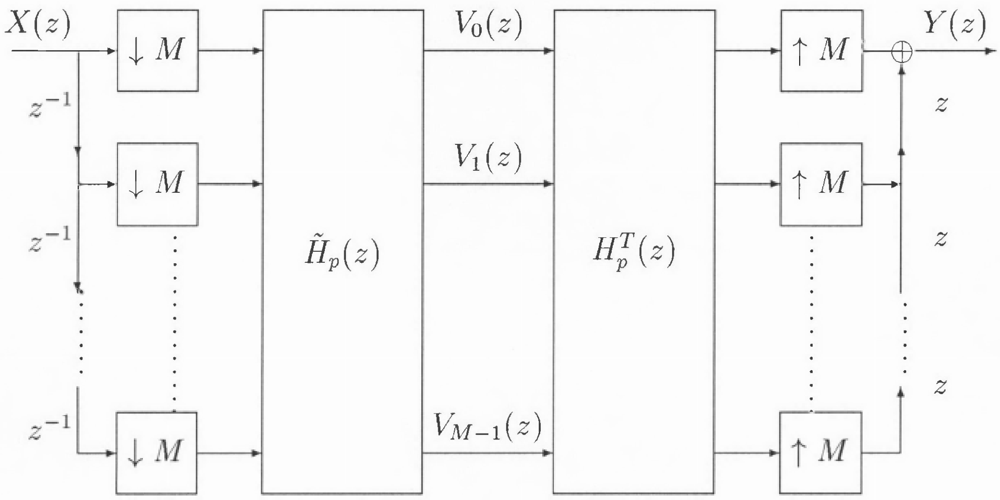

Figure 2.3: The polyphase form of uniform filter bank

We are interested in the case where $H_p^T(z)\tilde{H}_p(z) = I$. From here on-wards, when we say perfect reconstruction, we mean $Y(z) = X(z)$.

The above polyphase analysis is very useful in designing and implementing filter banks. The following fundamental result is also useful in deriving certain properties of filter banks, see Gopinath and Burrus [43].

Lemma 1 *Let $\tilde{H}_i$ $i \in \{0, ..., M-1\}$ denote the analysis filter bank. Assume that for all i, $H_i^f(\omega) < \infty$. Then there exists a synthesis filter bank with filters H_i $i \in \{0, ..., M-1\}$ if and only if $\tilde{H}_p(z)$ has full rank for all z on the unit circle (i.e. $z = e^{j\omega}$), or equivalently if $det\tilde{H}_p(e^{j\omega}) \neq 0$ for all ω.*

Proof: If $\tilde{H}_p(z)$ has full rank for all z on the unit circle, then it has full rank for all ω, is hence invertible and therefore $\tilde{H}_p^{-1}(e^{j\omega})$ is well defined. Take $H_p(e^{j\omega}) = \tilde{H}_p^{-T}(e^{j\omega})$. From $H_p(e^{\omega})$, the numbers $h_{i,j}(n)$ and hence $H_i(n)$ can be obtained. ∇

2.1.4 Orthogonality Vs. Bi-orthogonality

Unitary filter banks are a special case of perfect reconstruction filter banks where synthesis (or analysis) filters are determined from analysis (or synthesis) filters via time reversal: $h(n)_i = \tilde{h}_i(-n)$. In this case, we have $H_p(z) = \tilde{H}_p(z)$ where $H_p(z)$ is paraunitary (or unitary on the unit circle) such that $H_p^T(z^{-1})H_p(z) = I$ [35, 34, 36]. We alternatively use the terms, *unitary*, *paraunitary*, or *orthogonal* to refer to such filter banks.

When the filter bank is not orthogonal, we say the filter bank is *bi-orthogonal.* Further, the equation 2.20 implies the following:

$$\begin{aligned} &\tilde{H}_p(z)H_p^T(z) = I \\ &\Rightarrow \sum_{k=0}^{M-1} \tilde{H}_{i,k}(z)H_{j,k}(z) = \delta(i-j) \text{ for } i,j = 0,1,...,M-1 \\ &\Rightarrow [\downarrow M]\tilde{H}_i(z)H_j(z) = \delta(i-j) \text{ for } i,j = 0,1,...,M-1. \end{aligned} \tag{2.22}$$

The above equations are sometimes known as the bi-orthogonality constraints. They also imply that the product filters $\tilde{H}_i(z)H_i(z)$ are M-band interpolating. We say a filter ,$R(z)$, is M-band interpolating if $r(Ml) = \delta(l)$.

The equations 2.22 also implies that the M-shifts of the time reversed h_i, are orthogonal to $\tilde{h}_i$. This is a direct consequence of $[\downarrow M]\tilde{H}_i(z)H_i(z) = 1$ in time domain

$$\sum_k \tilde{h}_i(k)h_i(Ml-k) = \delta(l). \tag{2.23}$$

2.1.5 FIR Perfect Reconstruction Filter banks

A filter h is said to be *realizable* if convolution with the sequence $h(n)$ can be implemented, Gopinath and Burrus [43]. Finite Impulse Response (FIR) filters are a special class of realizable filters. A filter is said to be FIR if $H(z)$ is a Laurent polynomial. Most of the filter bank theory has dealt with

FIR filters. FIR filter bank theory includes all orthogonal and bi-orthogonal wavelet bases with compactly supported wavelets.

Given a set FIR analysis filters (or synthesis filters), FIR synthesis filters (or analysis filters) does not always exists. The following result characterize FIR filter banks.

Lemma 2 *h_i and $\tilde{h}_i$ form an FIR filter bank if and only if $det(H_p(z))$ or $det(\tilde{H}_p(z))$ is of the form cz^k for some integer k and constant $c (\neq 0)$.*

Proof: From equation 2.20, $H_p^T(z) = \tilde{H}_p^{-1}(z)$ and hence

$$H_p^T(z) = \frac{1}{\det \tilde{H}_p(z)} \mathrm{adj}(\tilde{H}_p(z)). \tag{2.24}$$

Since the adjoint of a Laurent polynomial matrix is a Laurent polynomial matrix, the above equation holds with $H_p^T(z)$ a Laurent polynomial if and only if $\det(\tilde{H}_p(z)) = cz^k$ for some integer k. ∇

2.2 Two-Dimensional Filter Banks

For simplicity of notations we consider only two dimensional signals though the following results on sampling, down-sampling and up-sampling generalizes to more than two dimensions. We are concerned with two dimensional discrete time images of the form $x(\mathbf{t})$ where x is sampled at discrete locations $\mathbf{t} = [t_1, t_2]^T$. The geometry of these discrete points can be chosen in variety of ways. We will discuss two special cases, rectangular sampling and hexagonal sampling. In rectangular sampling, the values of t_1 and t_2 are integer multiples of T_1, T_2 respectively. This sampling points can be described by $\mathbf{Vn}$ where $\mathbf{V}$ is a diagonal matrix with T_1 and T_2 along the diagonal and $\mathbf{n}$ is an integer vector. The columns of $\mathbf{V}$ forms a rectangle, and hence the name rectangular sampling. In general $\mathbf{V}$ can be any real valued matrix which we call the sampling matrix. For example, a special case of hexagonal sampling is given by the sampling matrix:

$$\mathbf{V} = \begin{bmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{bmatrix}.$$

Down-sampling is parameterized by a nonsingular *down-sampling matrix* $\mathbf{K}$ with integer entries. Down-sampling along the horizontal direction and vertical direction by two results in the down-sampling matrix $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$. Down-sampling by $\mathbf{K}$ results in $\det(\mathbf{K})$ number of sub-latices known as cosets. For each of these cosets we associate *polyphase shift vectors* $\mathbf{k}_i^r$ defined by

$$\{\mathbf{k}_i^r : \left(\mathbf{K}^{-1}\right)^T \mathbf{k}_i^r \in [0,1)^2, \mathbf{k}_i^r \in \mathbb{Z}^2\}. \tag{2.25}$$

Thus for $\mathbf{K} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, polyphase shift vectors are given by $[0,0]^T, [1,0]^T$, $[1,1]^T$, and $[0,1]^T$. Note the polyphase shift vectors given above are for the rectangular lattice where $\mathbf{V} = \mathbf{I}$. For a more general sampling lattice $\mathbf{V}$, the polyphase shift vectors, $\mathbf{k}_i$, are given by the mapping

$$\mathbf{k}_i = \left(\mathbf{V}^{-1}\right)^T \mathbf{k}_i^r. \tag{2.26}$$

The frequency domain counterparts of polyphase shift vectors, the *down-sampling modulation vectors* $\tilde{\mathbf{k}}_i$ are defined as $2\pi \left(\mathbf{K}^{-1}\right)^T \mathbf{k}_i$. For the rectangular sampling lattice down-sampling modulation vectors are given by $[0,0]^T, [\pi,0]^T, [\pi,\pi]^T$ and $[0,\pi]^T$. For the hexagonal lattice, down sampling modulation vectors are given by $[0,\ 0]$, $[\frac{2\pi}{\sqrt{3}}, 0]^T$, $[\frac{\pi}{\sqrt{3}}, \pi]^T$, and $[-\frac{\pi}{\sqrt{3}}, \pi]^T$.

The Z-transform in two dimensions is defined as

$$X(\mathbf{z}) = \sum_{\mathbf{k}} x(\mathbf{k})\mathbf{z}^{-\mathbf{k}} \tag{2.27}$$

where $\mathbf{z} = [z_0 \;\; z_1]^T$ and $\mathbf{k} = [k_0 \;\; k_1]^T$ $\mathbf{z}^{-\mathbf{k}} = z_0^{-k_0} z_1^{-k_1}$. This summation does not, in general, converge for arbitrary $\mathbf{z}$, see Lim [44]. If it converges for all z_l of the form

$$z_l = e^{j\omega_l}, l \in \{0, 1\}, \omega_l \in \mathcal{R}, \tag{2.28}$$

it reduces to the Fourier transform $X^f(\omega)$, which is defined by

$$X^f(\omega) = \sum_{\mathbf{k}} x(\mathbf{k})e^{-\omega^T\mathbf{k}} \tag{2.29}$$

where $\omega^T = [\omega_0 \;\; \omega_1]^T$.

We can establish the following relation for the down-sampling of $x(n)$ by $\mathbf{K}$ to obtain $y(n)$, Vaidyanathan [36, page 583]:

$$Y^f(\omega) = \frac{1}{|\mathbf{K}|} \sum_{\mathbf{k} \in \mathcal{N}(\mathbf{K}^T)} X^f(\mathbf{K}^{-T}(\omega - 2\pi\mathbf{k})) \tag{2.30}$$

where $\mathcal{N}(\mathbf{K}^T)$ is the set of polyphase shift vectors. The up-sampling of $x(n)$ by $\mathbf{K}$ to obtain $y(n)$ can be represented as

$$Y^f(\omega) = X^f(\mathbf{K}^T\omega). \tag{2.31}$$

2.2.1 Perfect Reconstruction Filter Banks

The conditions for perfect reconstruction in Z-transform domain can be easily generalized for multiple dimensions. In this section we analyze the perfect reconstruction conditions in Fourier domain. By applying convolution with analysis filter, then applying the down-sampling operator followed by up-sampling and convolution with synthesis filter, and finally summing up for all the subbands we get the following condition for the perfect reconstruction:

$$\frac{1}{|\mathbf{K}|}\sum_{k=0}^{|\mathbf{K}|-1} H_k^f(\omega)\sum_{i=0}^{|\mathbf{K}|-1} \tilde{H}_i^f(\omega+\tilde{\mathbf{k}}_i)X^f(\omega+\tilde{\mathbf{k}}_i) = X^f(\omega) \tag{2.32}$$

where $\tilde{\mathbf{k}}_i$ are the down sampling modulation vectors. In the above equation, the terms $X^f(\omega+\tilde{\mathbf{k}}_i)$ $(i \in \{1..|\mathbf{K}|-1\})$ are known as *aliasing terms.* For perfect reconstruction, it must be that the aliasing terms are canceled. Thus we require

$$\begin{array}{c} \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega+\tilde{\mathbf{k}}_1) = 0, \\ \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega+\tilde{\mathbf{k}}_2) = 0, \\ \vdots \\ \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega+\tilde{\mathbf{k}}_{|\mathbf{K}|-1}) = 0. \end{array} \tag{2.33}$$

After aliasing cancellation the remaining term must be $X^f(\omega)$. Thus we require

$$H_0^f(\omega)\tilde{H}_0^f(\omega) + H_1^f(\omega)\tilde{H}_1^f(\omega) + \ldots H_{|\mathbf{K}|-1}^f(\omega)\tilde{H}_{|\mathbf{K}|-1}^f(\omega) = |\mathbf{K}| \tag{2.34}$$

We gather the above results into the following lemma.

Lemma 3 *The analysis filters $\tilde{H}_0, ..., \tilde{H}_{|\mathbf{K}|-1}$, and the synthesis filters $H_0, ...,$ $H_{|\mathbf{K}|-1}$ has perfect reconstruction if and only if it satisfies*

$$\begin{array}{c} \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega+\tilde{\mathbf{k}}_1) = 0, \\ \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega+\tilde{\mathbf{k}}_2) = 0, \\ \vdots \\ \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega+\tilde{\mathbf{k}}_{|\mathbf{K}|-1}) = 0, \\ \sum_{l=0}^{|\mathbf{K}|-1} H_l^f(\omega)\tilde{H}_l^f(\omega) = |\mathbf{K}|. \end{array} \tag{2.35}$$

We can write the equation 2.35 in the matrix form:

$$\tilde{\mathbf{H}}_a^f(\omega)\mathbf{h}(\omega) = \mathbf{t} \tag{2.36}$$

where

$$\tilde{\mathbf{H}}_a(\omega) = \left| \begin{array}{llll} \tilde{H}_0^f(\omega) & \tilde{H}_1^f(\omega) & \ldots & \tilde{H}_{|\mathbf{K}|-1}^f(\omega) \\ \tilde{H}_0^f(\omega+\tilde{\mathbf{k}}_1) & \tilde{H}_1^f(\omega+\tilde{\mathbf{k}}_1) & \ldots & \tilde{H}_{|\mathbf{K}|-1}^f(\omega+\tilde{\mathbf{k}}_1) \\ \vdots & \vdots & & \vdots \\ \tilde{H}_0^f(\omega+\tilde{\mathbf{k}}_{|\mathbf{K}|-1}) & \tilde{H}_1^f(\omega+\tilde{\mathbf{k}}_{|\mathbf{K}|-1}) & \ldots & \tilde{H}_{|\mathbf{K}|-1}^f(\omega+\tilde{\mathbf{k}}_{|\mathbf{K}|-1}) \end{array} \right|, \tag{2.37}$$

$\mathbf{h}(\omega) = [H_0^f(\omega) \ldots H_{|\mathbf{K}|-1}^f(\omega)]^T$ and $\mathbf{t} = [|\mathbf{K}|\ 0 \ldots 0]^T$. The matrix $\tilde{\mathbf{H}}_a(\omega)$ is known as the Alias Component (AC) matrix. The AC matrix is related to the polyphase component matrix in the sense that all theoretical conclusions obtained from use of one of these matrices can also be obtained from the other. In particular, in one-dimensional M-band setting, the two matrices are related via

$$\tilde{\mathbf{H}}_a(\omega) = W^* D(e^{j\omega})\tilde{H}_p^T(e^{j\omega M}) \tag{2.38}$$

where $D(z) = \text{diag}(1z^{-1} \ldots z^{-(M-1)})$ and W^* is the conjugate of the DFT matrix defined by $[W]_{km} = w^{km}$ where $w = e^{-2\pi j/M}$. Thus the invertibility of the polyphase matrix on the unit circle given in the Lemma 1 implies the invertibility of the AC matrix for all ω.

2.2.2 Polyphase Representation

Each analysis filter $\tilde{H}_l$ can be written in the type 1 polyphase form as

$$\tilde{H}_l(\mathbf{z}) = \sum_{l=0}^{|\mathbf{K}|-1} \mathbf{z}^{-\mathbf{k}_l} \tilde{H}_{l,m}(\mathbf{z}^{\mathbf{K}^T}) \tag{2.39}$$

where

$$\tilde{H}_{l,m}(\mathbf{z}) = \sum_{\mathbf{n}} \tilde{h}_l(\mathbf{K}\mathbf{n} + \mathbf{k}_l)\mathbf{z}^{-\mathbf{n}}. \tag{2.40}$$

The polyphase matrix of a maximally decimated filter bank is a $M \times M$ matrix $\tilde{\mathbf{H}}_p(\mathbf{z})$ with one row for each filter and one column for each polyphase component. Thus

$$\tilde{\mathbf{H}}_p(\mathbf{z}) = \begin{vmatrix} \tilde{H}_{0,0}(\mathbf{z}) & \tilde{H}_{0,1}(\mathbf{z}) & \dots & \tilde{H}_{0,|\mathbf{K}|-1}(\mathbf{z}) \\ \tilde{H}_{1,0}(\mathbf{z}) & \tilde{H}_{1,1}(\mathbf{z}) & \dots & \tilde{H}_{1,|\mathbf{K}|-1}(\mathbf{z}) \\ \vdots & \vdots & & \vdots \\ \tilde{H}_{|\mathbf{K}|-1,0}(\mathbf{z}) & \tilde{H}_{|\mathbf{K}|-1,1}(\mathbf{z}) & \dots & \tilde{H}_{|\mathbf{K}|-1,|\mathbf{K}|-1}(\mathbf{z}) \end{vmatrix}. \tag{2.41}$$

Similarly, we define type 2 polyphase form for synthesis filters as

$$H_l(\mathbf{z}) = \sum_{l=0}^{|\mathbf{K}|-1} \mathbf{z}^{\mathbf{k}_l} H_{l,m}(\mathbf{z}^{\mathbf{K}^T}). \tag{2.42}$$

Now define the type 2 polyphase matrix for the synthesis filters as

$$\mathbf{H}_p(\mathbf{z}) = \begin{vmatrix} H_{0,0}(\mathbf{z}) & H_{0,1}(\mathbf{z}) & \dots & H_{0,|\mathbf{K}|-1}(\mathbf{z}) \\ H_{1,0}(\mathbf{z}) & H_{1,1}(\mathbf{z}) & \dots & H_{1,|\mathbf{K}|-1}(\mathbf{z}) \\ \vdots & \vdots & & \vdots \\ H_{|\mathbf{K}|-1,0}(\mathbf{z}) & H_{|\mathbf{K}|-1,1}(\mathbf{z}) & \dots & H_{|\mathbf{K}|-1,|\mathbf{K}|-1}(\mathbf{z}) \end{vmatrix}. \tag{2.43}$$

Now the conditions for perfect reconstruction can be expressed in terms of $\tilde{\mathbf{H}}_p(\mathbf{z})$ and $\mathbf{H}_p(\mathbf{z})$.

Lemma 4 *The analysis filters $\tilde{H}_0, \dots, \tilde{H}_{|\mathbf{K}|-1}$, and the synthesis filters $H_0, \dots,$ $H_{|\mathbf{K}|-1}$ are perfect reconstruction filters if and only if they satisfy*

$$\mathbf{H}_p^T(\mathbf{z})\tilde{\mathbf{H}}_p(\mathbf{z}) = I. \tag{2.44}$$

As with the one dimensional case, given a set of FIR analysis filters, FIR synthesis filters do not always exists. The following is the generalization of one dimensional FIR filter banks.

Lemma 5 *$\tilde{H}_l$ and H_l ($l \in \{0, ..., |\mathbf{K}| - 1\}$) form an FIR filter bank if and only if the determinant of $\tilde{\mathbf{H}}_a(\mathbf{z})$ or $\mathbf{H}_a(\mathbf{z})$ is of the form $C\mathbf{z}^{\mathbf{c}}$ for some integer $\mathbf{c}$ and a constant $C(\neq 0)$.*

The following lemma shows that the product filters, $\tilde{H}_i H_i$, are interpolatory.

Lemma 6 *Let $\tilde{h}_i$ and h_i form an FIR filter bank and $|\tilde{\mathbf{H}}_a(\mathbf{z})| = C\mathbf{z}^{\mathbf{c}}$. Then*

$$\sum_{k=0}^{|\mathbf{K}|-1} \tilde{H}_i^f(\omega + \tilde{\mathbf{k}}_k) H_i^f(\omega + \tilde{\mathbf{k}}_k) = Ce^{-j\omega^T \mathbf{c}} \quad \textit{for} \quad i = 0, ..., |\mathbf{K}| - 1. \tag{2.45}$$

Proof: It can be seen that

$$H_i^f(\omega) = \frac{-M_i(\omega)}{Ce^{-j\omega^T \mathbf{c}}} \tag{2.46}$$

where $M_i(\omega)$ is the minor associated with the $H_i(\omega)$ of the system of equations 2.35. Now constructing the determinant of the system along the column corresponding to $H_i^f(\omega)$, we get

$$\sum_{k=0}^{|\mathbf{K}|-1} \tilde{H}_i^f(\omega + \tilde{\mathbf{k}}_k) M_i(\omega + \tilde{\mathbf{k}}_k) = Ce^{-j\omega^T \mathbf{c}} \quad \text{for} \quad i = 0, ..., |\mathbf{K}| - 1. \tag{2.47}$$

from which we deduce 2.45. ∇

When $\tilde{H}_i = H_i^*$, we say that the filter bank is orthogonal or paraunitary, otherwise it is said to be bi-orthogonal. Most of the filter bank theory developed is concerned with paraunitary filter banks due to the availability of factorization methods for unitary matrices. Given the analysis filters, the bi-orthogonal synthesis filters H_i could be obtained via equation 2.47 but such is not popular since it can yield large filters. The equation 2.45 is known as the bi-orthogonal constraint.