THE NONREDUNDANT CONTOURLET TRANSFORM (NRCT): A MULTIRESOLUTION AND MULTIDIRECTION IMAGE REPRESENTATION WITH PERFECT RECONSTRUCTION PROPERTY

SHENQIU ZHANG

The Nonredundant Contourlet Transform (NRCT): A Multiresolution and Multidirection Image Representation with Perfect Reconstruction Property

by

© Shenqiu Zhang
Bachelor of Engineering

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Master of Engineering.

Department of Electrical and Computer Engineering
Memorial University of Newfoundland

April, 2008

ST. JOHN'S NEWFOUNDLAND

Abstract

Multiresolution and multidirection image representation has recently been an attractive research area, in which multiresolution corresponds to varying scale of structure in images, while multidirection deals with the oriented nature of image structure. Numerous new systems, such as the contourlet transform, have been developed. The contourlet transform has the benefit of efficiently capturing the oriented geometrical structures of images; however, it has the drawback of a 4/3 redundancy in its oversampling ratio. In order to eliminate the redundancy, this thesis proposes a progressive version of the contourlet transform which can be calculated with critical sampling. The new proposed image representation is called the nonredundant contourlet transform (NRCT), which is constructed with an efficient framework of filter banks. In addition to critical sampling, the proposed NRCT possesses many valuable properties including perfect reconstruction, sparse expression, multiresolution, and multidirection. Numerical experiments demonstrate that the novel NRCT has better peak signal-to-noise performance than the traditional contourlet transform. Moreover, for low ratios of retained coefficients, the NRCT outperforms the wavelet transform which is a standard method for the critically sampled representation of images.

After examining the computational complexity of the nonredundant contourlet transform, this thesis applies the NRCT to fingerprint image compression, since fingerprint images are examples of images with oriented structures. Based on an appropriately designed filter bank structure, the NRCT is easily compatible with the wavelet transform. Hence a new transform is created called the semi-NRCT, which takes the advantages of the directional selectivity of the NRCT and the lower complexity of the wavelet transform. Finally, this thesis proposes a new fingerprint image compression scheme based on the semi-NRCT. The semi-NRCT-based fingerprint image compression is compared with other transform-based compressions, for example the wavelet-based and the contourlet-based algorithms, and is shown to perform favorably.

Acknowledgements

I would like to express my sincerest gratitude to my supervisor Dr. Cecilia Moloney for her patience, kindness, guidance, and financial support. She always gave me constructive feedback when I sought advice. Her contribution to this thesis was truly invaluable. I am also grateful to thank Faculty of Engineering and Applied Science for providing me financial assistance to complete my master degree in Memorial University of Newfoundland. The help of Reza Shahidi in transferring the format of fingerprint images, which are obtained from the web site of National Institute of Standards and Technology, was much appreciated. I take this opportunity to thank my friends at Computer Engineering Research Labs (CERL) for their generous help and precious friendship. My appreciation also goes to my boyfriend Tianqi Wang, who suggested me to pursuit a master degree at Memorial University of Newfoundland. He was always there to encourage me and support me no matter what. Finally, my parents deserve much gratitude for their constant understanding and encouragement with love. I dedicate this thesis to them all.

Contents

List of Tables

List of Figures

List of Abbreviations

CDF	The filters created by Cohen, Daubechies and Feauveau	26
CS	Critical Sampling	48
CT	Contourlet Transform	3
DFB	Directional Filter Bank	8
DWT	Discrete Wavelet Transform	4
FIR	Finite Impulse Response	26
IIR	Infinite Impulse Response	27
LP	Laplacian pyramid	8
NRCT	Nonredundant Contourlet Transform	4
JPEG 2000	An wavelet-based image compression standard created by the Joint Photographic Experts Group committee in the year 2000	2
PR	Perfect Reconstruction	24
QMF	Quadrature Mirror Filter	27
SPR	Sheared-Parallelogram	35
WSQ	Wavelet/Scalar Quantization, a fingerprint image compression standard by the FBI	27
nuDFB	nonuniform Directional Filter Bank	24
uDFB	uniform Directional Filter Bank	23

List of Symbols

$\mathbf{M}$	A 2-D sampling matrix	32
$\mathbf{M}^T$	The transpose of $\mathbf{M}$	32
$\mathbf{M}^{-1}$	The inverse of $\mathbf{M}$	32
$\mathbf{M}^*$	The conjugate of $\mathbf{M}$	32
$\det(\mathbf{M})$	The determinant of $\mathbf{M}$	32
$\mathbf{I}_k$	A $k \times k$ identity matrix	32
$\Lambda(\mathbf{M})$	The lattice generated by $\mathbf{M}$ in Eq.(2.31)	33
$\mathbb{Z}^2$	A two-dimensional integer coordinate space	33
$\mathbf{n}$	The pixel index vector $\mathbf{n} = [n_1, n_2]^T$	33
$x(\mathbf{n})$	A discrete image	33
$X(\mathbf{z})$	The z-transform of $x(\mathbf{n})$	33
$\mathbf{z}$	The index vector in the z-domain $\mathbf{z} = [z_1, z_2]^T$	33
$\mathbf{z}^{\mathbf{n}}$	$\mathbf{z}^{\mathbf{n}} = z_1^{n_1} z_2^{n_2}$	33

Chapter 1

Introduction

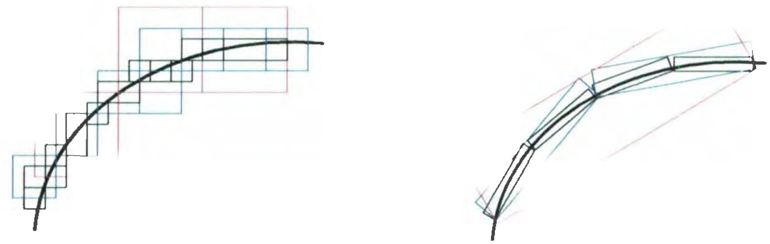
1.1 General

Multiresolution image representation, as implied by its name, analyzes images at multiple resolutions or scales or frequency bands. When a high-quality image is observed, its low resolution view can provide a rough impression of the image. When an image is to be transmitted as a bitstream over a bandwidth limited channel, a coarse version can be sent first. Then the details of the image can be enriched by the subsequent data at higher resolutions. This is a typical example of the application of progressive image transmission [1], which allows users to reconstruct or view the received image at increasing resolutions. Moreover, multiresolution representation is applied to many other areas as well, such as image compression and image denoising [2, 3].

A classical multiresolution approach is evident in the wavelet transform, which has been widely adopted in many fields. In contrast with the Fourier transform, which expresses an image as a sum of weighted sinusoids of various frequencies, the wavelet transform describes an image using basis functions of limited duration which vary in position and frequency. In other words, the basis functions of the Fourier transform are located in the frequency domain, while those of the wavelet transform are located in both the spatial and the frequency domains. If the basis elements of the wavelet transform are classified into the levels of a multiscale pyramid according to their frequency locations, then the wavelet transform can be viewed as a decorrelation in the frequency domain, where the coarser scales describe an approximate rendition of the image (corresponding to the low frequency components) and the finer scales represent its details (corresponding to the high frequency components) [4, 5]. Due to its advantages of multiscale representation and fast transformation, the wavelet transform is the foundation of many useful techniques in image processing. For example, the JPEG 2000 standard by the Joint Photographic Experts Group (JPEG) committee is a wavelet-based image coding standard [6].

As will be seen in the following chapter, the one-dimensional (1-D) wavelet transform is suitable for representing piecewise smooth signals in one dimension. However, the two-dimensional (2-D) separable wavelet transform, which implements the 1-D wavelet transform along one dimension, say the rows, and is then followed by the 1-D wavelet transform of the other dimension, the columns, has the limitation of directional selectivity [7]. Consider the case when a painter intends to draw a picture by using "wavelet"-style brushes. The picture is refined from coarse to fine by increasing resolution. However, the "wavelet"-brush strokes are rectangular or square shaped of various sizes, since the 2-D wavelet transform is a combination of the 1-D wavelet transform in two dimensions. Hence, by using such a restricted set of rectangular brush strokes, the printer requires many "dots" or "short-dashes" to capture a fine contour, as illustrated in Fig.1.1(a). Therefore, more powerful and efficient representations are required for digital image processing in many applications.

Recently, directional multiresolution image representation has attracted worldwide



(a) Wavelets (b) Contourlets

Figure 1.1: The successive refinement of a contour by using two different transforms: (a) the wavelet transform and (b) the contourlet transform. Such a contour, shown as a thick curve, may separate two smooth regions in an images. [Modified from a figure in [8]]

research interest. Numerous new systems have been proposed, such as curvelets, directionlets, contourlets, etc. The curvelet transform [9] was initially developed in the continuous domain. Therefore, it is difficult to directly implement the curvelet transform in the discrete domain. On the other hand, the directionlet transform [10] can be realized using a discrete filter bank; however, it only allows two major directions at each scale. The contourlet transform [8], which can be viewed as a discrete version of the curvelets, can offer different and flexible numbers of directions at each scale. Therefore, the contourlet transform is more suitable for tracking directional information of digital images in the discrete domain than either the curvelet transform or the directionlet transform. Consider a similar scenario, in which the painter uses "contourlet"-style brushes instead of "wavelet"-brushes. The "contourlet"-brushes can generate elongated rectangular-shaped strokes in a variety of directions. As illustrated in Fig.1.1(b), the contourlet transform can more effectively capture oriented geometrical structures in images.

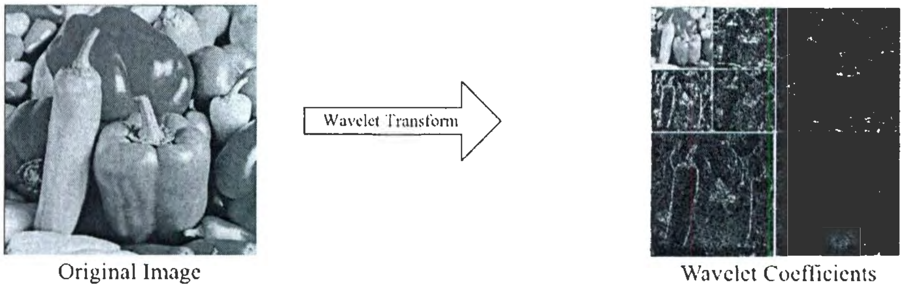
This thesis focuses on the contourlet transform (CT) and its application to image compression. The original contourlet transform by Do and Vetterli [8] has limited suitability for image compression because of its redundancy of oversampling. Thus the purpose of this thesis is to propose a nonredundant contourlet transform (NRCT) and implement it for image compression.

1.2 Motivation

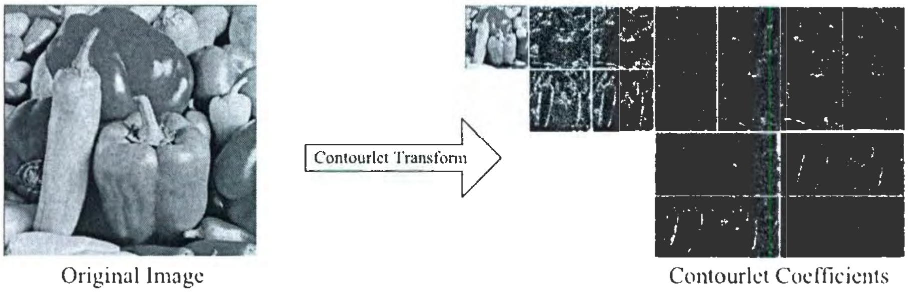
Image representation is a systemic "language" to express a set of basic building blocks that can be combined to form images. In the same way that an idea can be described in different languages, images can be represented by using the Fourier transform, the wavelet transform, the contourlet transform, or other transforms. For both images and natural languages, the efficiency and accuracy of different representations can vary widely.

For example, a French sentence might translate into a English sentence with fewer number of words, hypothetically. In this case, English is more efficient than French to represent this sentence. Moreover, if the original French sentence can be exactly translated back from the English sentence, hypothetically again, then English can be viewed as a compressed expression of French with perfect reconstruction property for this sentence. Similarly, for a given image, its original spatial expression, its Fourier transform, its wavelet transform, and its contourlet transform can be translated from one to another. The reason for seeking images in various "language" is that different transforms are suitable for different image processing tasks, such as denoising, compression and enhancement. Moreover, some types of images are more suitably expressed by one transform over another. To better understand this, an in-depth analysis of different image representation techniques is necessary.

For the discrete wavelet transform (DWT) , the multiresolution analysis by Mallat [4] can flexibly form an orthonormal basis, which guarantees that the DWT has the



(a) Wavelet transform



(b) Contourlet tranform

Figure 1.2: Examples of (a) the wavelet transform and (b) the contourlet transform on an original digital image "peppers", in which small coefficients in absolute value are colored toward black, while and large coefficients in absolute value are colored toward white. [Contrast boosted for display]

same number of coefficients as the number of pixels of the original image. Fig.1.2(a) shows an example of the wavelet transform on an image "peppers", in which small-magnitude coefficients are colored toward black. The total number of the wavelet coefficients is equal to the number of pixels of the original image. In contrast, as mentioned before, the contourlet transform [8] has a redundancy of oversampling, which means that the number of coefficients required by the contourlet transform is more than the number of pixels of the original image, as illustrated in Fig.1.2(b). For both the wavelet transform and the contourlet transform, there is a coarse and downsampled rendition of the original image (seen in Fig.1.2(a), (b) at the upper left corner of the coefficient configurations). Except for this coarse approximation, the remaining coefficients represent the increasingly finer details of the original image. Such an expression of images is known as a multiresolution representation.

If the original image is represented in digital form by a finite number set, say 8-bit resolution in gray-scale, with a range of possible pixel values from 0 to 255, then the resulting wavelet transform and contourlet transform both have their dynamic range of coefficient values beyond $[0, 255]$. However, most of the coefficients are concentrated around zero, as colored by black in Fig.1.2, and these smaller coefficients can be removed without significantly degrading the quality of the reconstructed image. Therefore, both the wavelet transform and the contourlet transform lend themselves well to be a sparse expression for images with the cost of small amounts of error. The slight degradation of the reconstructed image is usually imperceptible to human viewers. However, a drawback of the contourlet transform is that it uses more coefficients than the wavelet transform to express images. Therefore, in image compression, where nonredundant representation is a crucial requirement, the wavelet transform may be a preferred choice rather than the contourlet transform.

Because of its directional selectivity, which is illustrated in the previous section, the contourlet transform is widely used in many areas, such as image denoising [11, 12], image enhancement [11, 13], and feature extraction [14, 15]. However, to our best knowledge, the application of image compression using the contourlet transform does not exist due to its redundancy. In order to eliminate the redundancy of the contourlet transform, Lu and Do proposed a critically sampled version of the contourlet transform, namely the CRISP-contourlet transform [16]. Despite eliminating the redundant oversampling of the contourlet transform, the CRISP-contourlet transform is not a multiresolution representation [17], since its coefficients do not provide a coarse approximation of the original image. In other words, the CRISP-contourlet transform is not a "true" progressive rendition of the contourlet transform, although it removes the redundancy. Therefore, this thesis solves an open problem about devising a nonredundant contourlet transform (NRCT), and applies the proposed method to image compression. Due to its advantage of directional selectivity, the nonredundant contourlet transform can achieve an improved performance in image compression compared with the wavelet transform.

1.3 Problem Definition

The purpose of this thesis is to design a nonredundant contourlet transform (NRCT) with the following important properties: multiresolution, multidirection, perfect reconstruction, critical sampling, localization, and anisotropy [8]. The critical sampling implies the oversampling ratio of the whole system is zero, which indicates the nonredundancy of this system. The proposed transform also preserves many valuable properties of the contourlet transform, including efficient implementation using tree-structure filter banks, and a flexible number of directional subbands at each scale.

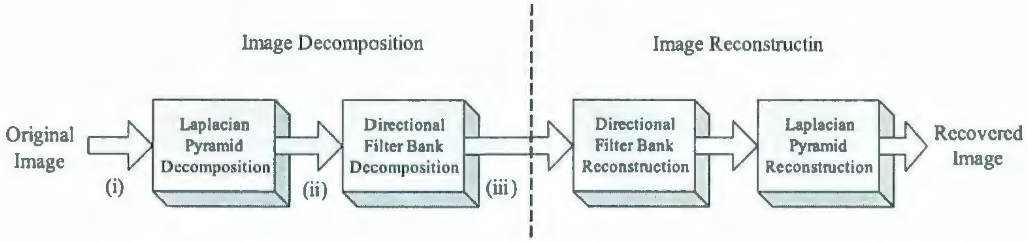
The proposed method possesses two important properties: critical sampling and perfect reconstruction, which are important in applications, such as image compression. Therefore, the goal of this thesis is to apply the proposed NRCT to image compression with a performance comparable to or superior to both the discrete wavelet transform and the original contourlet transform.

This thesis only considers image representation for gray-scale images. Since color can be represented as a variable combination of three primary colors, e.g. red (R), green (G) and blue (B) [7], color images can be decomposed into three RGB color planes, each of which can be processed by gray-scale image techniques. However, the decomposed images in RGB planes might have higher correlations with each other, such as the same objects and/or the same contours (although their intensity levels might be different). Therefore, color images can be represented more efficiently by reducing the correlations between color planes, than implementing gray-scale image techniques in the three RGB color planes independently. The correlations between RGB color planes can be reduced by converting colors from RGB format to other forms, such as HSI (hue, saturation, intensity) format [7].

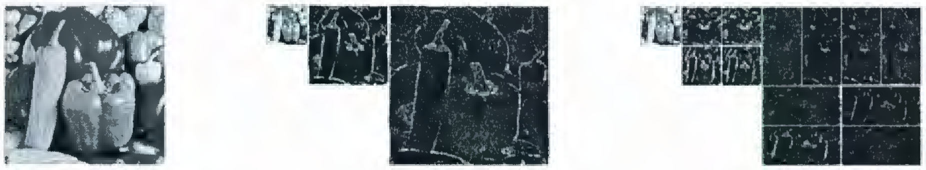
1.4 Approach to the Solution

The contourlet transform can be viewed as a combination of the Laplacian pyramid (LP) by Burt and Adelson [1], and the directional filter bank (DFB) by Bamberger and Smith [18]. The system scheme of the contourlet transform is shown in Fig.1.3, in which the contourlet decomposition is achieved by two steps: first, the multiresolution decomposition of images using the LP; second, the multidirection decomposition of the LP using the DFB. The data flow through this system is also illustrated in Fig.1.3, where the original image (i) is decomposed into a Laplacian pyramid (ii), which is then fed to the DFB to generate the contourlet transform (iii). The image reconstruction is the inverse procedure of the decomposition. As illustrated in Fig.1.3, the number of the coefficients in a Laplacian pyramid is greater than the number of pixels of the original image, which means the LP is a redundant image representation. Therefore, the redundancy of the contourlet transform is inherited from the Laplacian pyramid.

One method to eliminate the redundancy of the contourlet transform is to replace the LP with another critically sampled multiresolution expression, such as the wavelet transform. As illustrated in Fig.1.4, the proposed system implements the wavelet transform for multiresolution decomposition of images, followed by a multi-



The system scheme of the contourlet transform



(i) Original image (ii) Laplacian pyramid (iii) Contourlet Transform

Figure 1.3: Overall data flow for the contourlet decomposition/reconstruction, where images (i), (ii) and (iii) are data passing through the corresponding nodes in the system scheme of the contourlet transform, with an example showing the property of the "peppers" image by the contourlet transform.

direction decomposition. This system can be viewed as either a contourlet transform without redundancy or an extended wavelet transform by adding the multidirection decomposition/reconstruction modules. Hence one of the contributions of our work is to build a connection between the wavelet transform and the contourlet transform.

1.5 Outline of Thesis

This thesis is organized as follows. The first chapter is an introduction to image representations, especially the wavelet transform and the contourlet transform, which are illustrated in a high level manner. After presenting an overview of image rep-

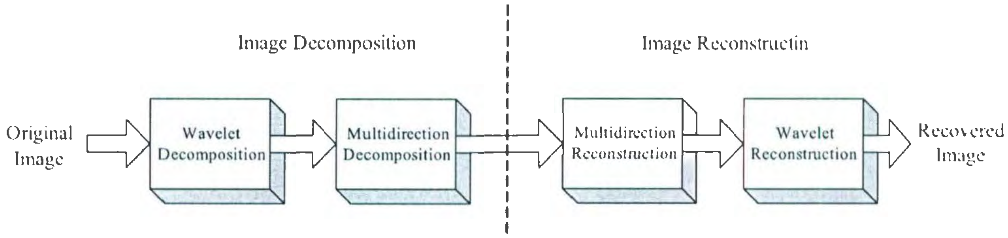


Figure 1.4: The system scheme of the proposed transformation

resentation techniques including mulresolution image representations, multidirection image representations, and multiresolution and multidirection image representations, Chapter 2 reviews a variety of uniform and nonuniform perfect reconstruction filter banks, which are adopted in our proposed system.

Chapter 3 proposes a new multiresolution and multidirection image representation with critical sampling and perfect reconstruction properties, called the nonredundant contourlet transform (NRCT). A complete comparison of the proposed NRCT with the wavelet transform and the traditional contourlet transform through computer simulation is also included in this chapter.

The computational complexity of the nonredundant contourlet transform is analyzed in Chapter 4. In Chapter 5, an application of fingerprint image compression using the proposed method is given. Moreover, the experimental results of the image compression algorithm using the NRCT against the wavelet transform and the traditional contourlet transform are also provided.

Finally, this thesis concludes in Chapter 6 with a summary of its methods, results and contributions, and a discussion of possible future work.

Chapter 2

Background

2.1 Image Representations

The purpose of image representation is to provide a suitable expression of an image for subsequent image processing tasks, such as compression and denoising. Still images (i.e. non-time-varying) are commonly described as two-dimensional signals, hence this chapter presents various popular 2-D transformation techniques. Some 2-D image representations, such as the separable 2-D wavelet transform, combine appropriate 1-D transformations in two dimensions. Therefore, an introduction to image representation starts from a review of 1-D transforms.

2.1.1 1-D Fourier Transform and Wavelet Transform

The most popular 1-D transformations are the Fourier transform and the wavelet transform, both of which are orthonormal transformations for signals of one dimension. Suppose a signal function $f(t)$ is analyzed as a linear combination of basis functions $\psi_k(t)$, as in

$$f(t) = \sum_k \alpha_k \psi_k(t), \tag{2.1}$$

where k is the index of the finite or infinite sum, α_k are weights or coefficients, and $\psi_k(t)$ are basis functions. If all the expression functions form an orthonormal basis, then α_k can be calculated as follows:

$$\alpha_k = \langle f(t), \psi_k(t) \rangle. \tag{2.2}$$

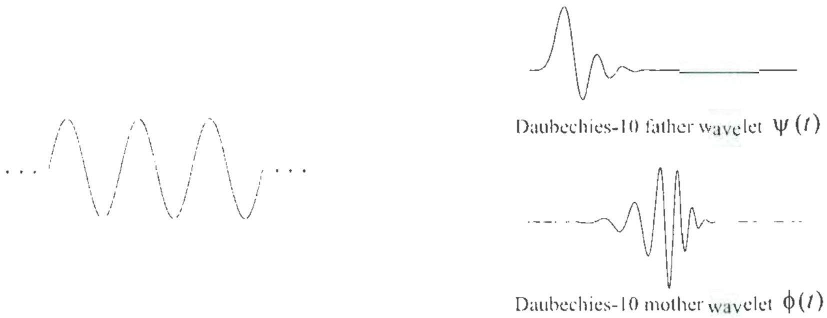
The basis functions $\{\psi_k(t)\}$ for the Fourier transform are sinusoid functions of various frequencies, while the basis functions for the wavelet transform, called wavelets, are scaling functions and wavelet functions.

Different wavelet transforms are based on different wavelet basis functions. Many canonical families of orthogonal wavelet basis functions exist, such as the Haar wavelets, the Daubechies wavelets and the Morlet wavelets [4]. Each wavelet family consists of various distinct sets of wavelet basis functions. For example, the Daubechies family wavelets are commonly denoted by "Daubechies-N", where N is the order. An example of the Daubechies-10 prototype wavelets, which contain a "father" wavelet $\phi(t)$ and a "mother" wavelet $\psi(t)$, is shown in Fig.2.1(b). From the "father" and "mother" wavelets, other wavelets, including scaling functions $\phi_{m,n}(t)$ and wavelet functions $\psi_{m,n}(t)$, are generated as follows [5]:

$$\phi_{m,n}(t) = \frac{1}{\sqrt{m}} \phi\left(\frac{t-n}{m}\right), \tag{2.3}$$

$$\psi_{m,n}(t) = \frac{1}{\sqrt{m}} \psi\left(\frac{t-n}{m}\right), \tag{2.4}$$

where m is a positive number and defines the scale of wavelets, while n defines the shift. Compared with the sinusoid function in Fig.2.1(a), which is smooth and symmetric with an infinite time duration, wavelets may be asymmetric and are fast-

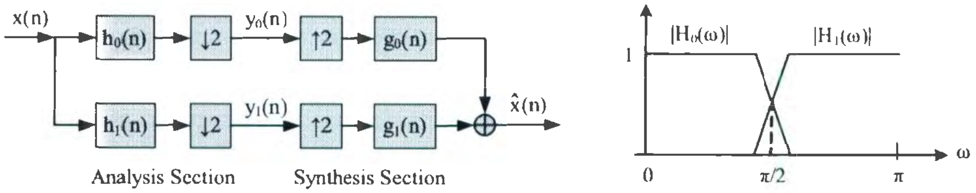


(a) Sinusoids (b) Wavelets

Figure 2.1: Basis functions: (a) Example of the sinusoid functions which form the Fourier basis. (b) The Daubechies-10 prototype wavelets which are extended to form the Daubechies-10 wavelet basis. These Daubechies-10 basis functions are generated by using the Matlab Wavelet Toolbox.

changing with limited durations. Therefore, wavelets can efficiently represent discontinuity in a signal function $f(t)$.

In practice, images are usually represented in the discrete domain. Therefore, the continuous transforms are necessarily extended to their corresponding discrete forms. The multiresolution analysis by Mallat [4] builds a connection between the wavelets in the continuous domain and the subband decomposition in the discrete domain. Moreover, the discrete wavelet transform can be easily implemented by Mallat's filter bank [19], which can be represented in an analysis/synthesis form as shown in Fig.2.2(a). A discrete signal $x(n)$ is decomposed into two subbands $y_0(n)$ and $y_1(n)$ by analysis filtering and 2-fold downsampling. The analysis filters $h_0(n)$ and $h_1(n)$ are half-band lowpass and highpass filters with the frequency responses shown, in magnitude form only, in Fig.2.2(b). $h_0(n)$ and $h_1(n)$ can be selected somewhat arbitrarily and often formally related, as discussed later. The synthesis filters $g_0(n)$ and $g_1(n)$ are the inverse of the analysis filters. In Fig.2.2(a), the 2-fold downsampling



(a) Analysis/synthesis filter bank (b) Frequency partition

Figure 2.2: (a) 2-channel analysis/synthesis filter bank, in which $h_0(n)$ and $h_1(n)$ are analysis filters and $g_0(n)$ and $g_1(n)$ are synthesis filters. (b) The frequency parition of the two-channel filter bank, where $|H_0(\omega)|$ and $|H_1(\omega)|$ are the magnitude responses of the analysis filters.

"↓ 2" retains the even samples of its input, while the 2-fold upsampling "↑ 2" replaces the downsampled values by zeros for subsequent interpolation to achieve the original signal rate.

The decomposed low frequency subband $y_0(n)$ corresponds to a coarse approximation of the image, while the high frequency subband $y_1(n)$ represents the details. If this two-band splitting is applied to the coarse scale iteratively, then the DWT provides a multiscale expression of signals.

2.1.2 Multiresolution Representations

Image Pyramids

The most straightforward way to represent images in multiple resolutions is via a Gaussian pyramid [1] which consists of a series of images at various resolutions. In a $(J+1)$-level Gaussian pyramid, the finest level G_0 is the original image, each of other levels is an approximation of its next finer level, and G_J is the coarsest level. A one-level procedure to generate the Gaussian pyramid is illustrated in Fig.2.3, where the input image G_j at level l is lowpass filtered and 2×2 downsampled to produce

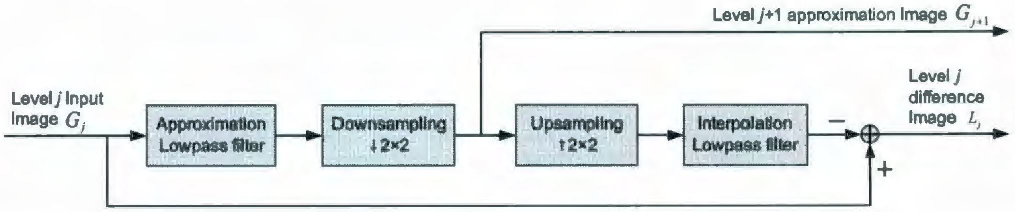


Figure 2.3: One-level structure to generate the Gaussian and the Laplacian pyramids

an approximation G_{j+1}, for $j = 0, \cdots, J-1$. If the approximation filter in Fig.2.3 is implemented by a Gaussian filter, then the corresponding pyramid is called the Gaussian pyramid.

An example of the Gaussian pyramid on an image "cameraman" is shown in Fig.2.4(a) (Note that the original "cameraman" image is G_0). Since the size of the output from the decimator "$\downarrow 2 \times 2$" is reduced by half in two dimensions compared with its input, the size of each level in the Gaussian pyramid is 1/4 of the size of its next finer level. If the size of the original image is $M \times N$, then the total number of coefficients in a $(J+1)$-level Gaussian pyramid, for $J > 0$, is

$$MN\left(1 + \frac{1}{4^1} + \frac{1}{4^2} + \cdots + \frac{1}{4^J}\right) = \frac{4}{3}MN\left(1 - \frac{1}{4^{J+1}}\right) \leq \frac{4}{3}MN. \qquad (2.5)$$

The above expression demonstrates the redundant oversampling ratio of up to 4/3 of the Gaussian pyramid. Note that the level G_0 is the original image containing all information of the other scales. Therefore, the Gaussian pyramid is a redundant representation with excessive information.

A more powerful image pyramid is the Laplacian pyramid (LP) by Burt and Adelson [1]. Except for the coarsest level L_J, which is equal to the same level of the Gaussian pyramid, the other Laplacian levels L_j $(j = 0, \cdots, J-1)$ are created by expanding the coarser Gaussian level G_{j+1} to the same size as G_j, then subtracting G_j by the expanded G_{j+1}. The procedure to generate the Laplacian pyramid from

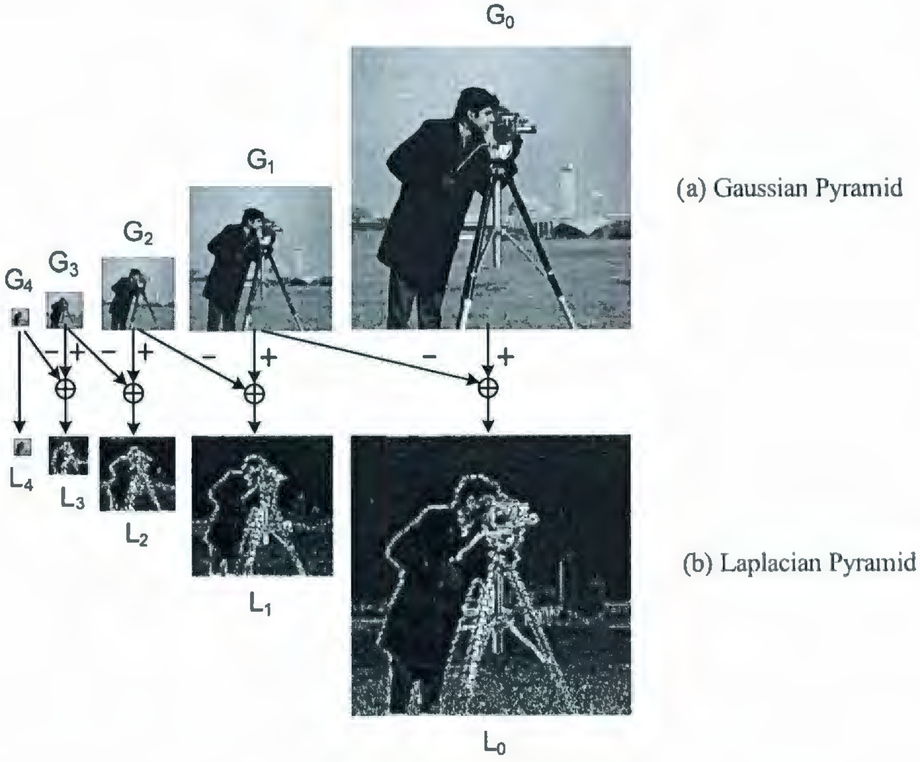


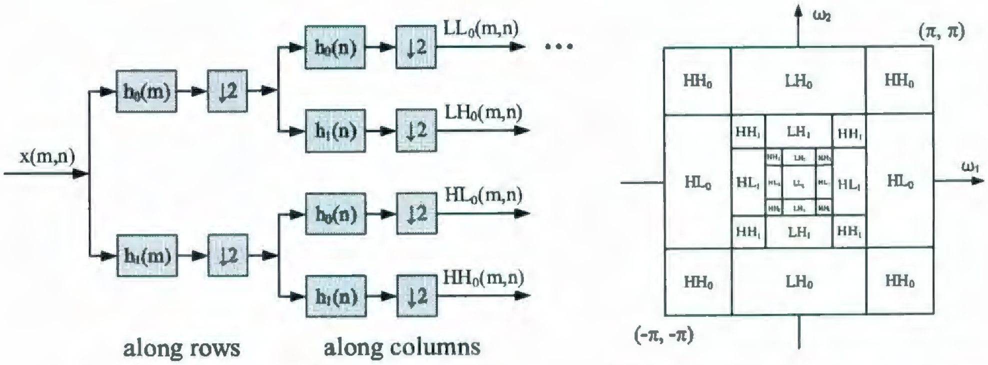
Figure 2.4: An example of image pyramids: (a) The Gaussian pyramid and (b) the Laplacian pyramid on the image "cameraman". Each Laplacian level is the difference between the corresponding and the barely finer levels of the Gaussian pyramid. The contrast of the Laplacian pyramid is boosted for display.

the Gaussian pyramid is illustrated in Fig.2.3.

An example of the Laplacian pyramid on the image "cameraman" is also shown in Fig.2.4(b), where the small coefficients in absolute value are colored towards black, while the large coefficients in absolute are colored gray to white. Because of the abundant number of small-absolute-value coefficients, the Laplacian pyramid can be easily applied in image compression, which is the initial purpose of designing the Laplacian pyramid [1]. However, the total number of the coefficients in the Laplacian pyramid is still greater than the number of pixels of the original image, with a redundant ratio of up to 4/3. Hence most of image compression applications choose a more powerful image representation, e.g. the wavelet transform, instead of the Laplacian pyramid.

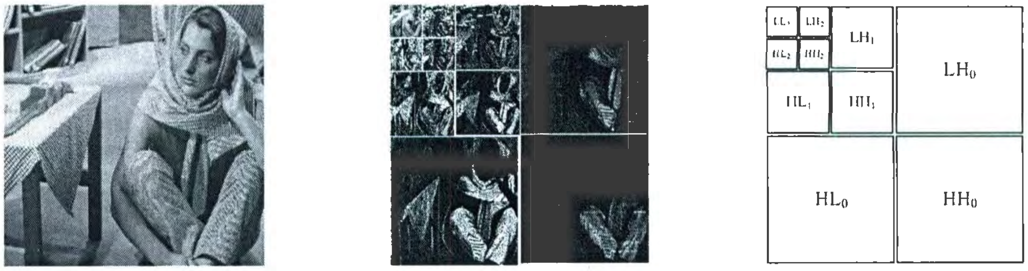
Wavelet Transform

A 2-D discrete wavelet transform (DWT) can be implemented by applying 1-D DWTs in two dimensions separately, as illustrated in Fig.2.5(a). This type of DWT is called the separable wavelet transform. The analysis filter bank in Fig.2.5(a) splits a discrete image $x(m,n)$ into four subbands: one coarse scale $LL_0(m,n)$ and three fine scales $LH_0(m,n)$, $HL_0(m,n)$ and $HH_0(m,n)$. If this analysis filter bank is iterated on the coarse subband, then the spectrum of the original image is divided by the wavelet transform as shown in Fig.2.5(b). Corresponding to this frequency partition, an example of a separable discrete wavelet transform, which is also a biorthogonal transform, on an image "barbara" is illustrated in Fig.2.6(b), while the original image "barbara" is shown in Fig.2.6(a). The original image is decomposed into four scales,



(a) 2-D DWT analysis filter bank (b) Frequency Partition of DWT

Figure 2.5: (a) The analysis filter bank of the 2-D separable wavelet transform. (b) The frequency partition of the 2-D discrete wavelet transform.



(a) Original image (b) Wavelet transform (c) Coefficient map

Figure 2.6: An example of the separable wavelet transform on the image "barbara". (a) The original image. (b) The wavelet transform of the original image, based on the biorthogonal wavelet transform with the CDF "9-7" filters [20]. [Contrast boosted for display] (c) The repacking map of the wavelet coefficient subbands, in which the subband indices correspond to that of the frequency partition shown in Fig.2.5(b).

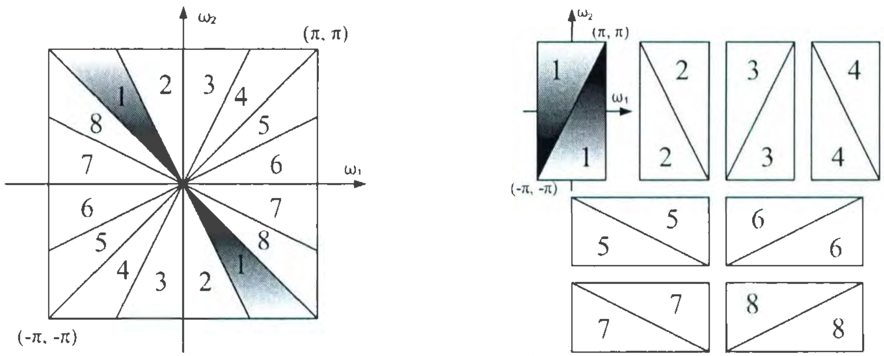
and the resulting wavelet coefficient subbands are repacked according to the pattern shown in Fig.2.6(c). In Fig.2.6(b), the coefficients close to zero are colored towards black, while the coefficients with large absolute value are colored towards white.

As shown in Fig.2.6(b), the wavelet transform provides a compact expression of images resulting in the same number of coefficients as the number of pixels of original image. Moreover, the wavelet transform is a sparse expansion of images due to the abundant number of small-absolute-value coefficients. Therefore, the above two features enable the wavelet transform to be widely applied in image compression.

2.1.3 Multidirection Representation

Directional Filter Bank

In [18], Bamberger and Smith proposed a directional filter bank (DFB), which can achieve the frequency partition illustrated in Fig.2.7(a). The DFB is implemented



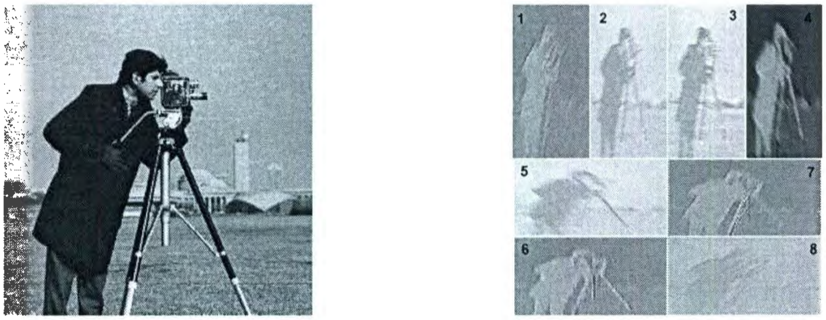
(a) Frequency partition of DFB (b) The downsampled subbands

Figure 2.7: The frequency mapping of the directional filter bank. (a) The frequency partition, where $2^3 = 8$ directional wedge-shaped subbands are shown as an example. (b) Subbands are downsampled by skewing and expanding to fill in their base bands, which are bounded by $(-\pi, -\pi)$ and (π, π) respectively.

by a binary tree structure consisting of two-band filter banks, hence the spectrum is divided into $N = 2^v$ wedge-shaped directional subbands. The first two levels of the DFB are built by fan filter banks to split the spectrum into four subbands, which are the bands numbered in Fig.2.7(a) as $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, and $\{7, 8\}$. Based on the above decomposition, a finer directional decomposition is implemented by iterative parallelogram filter banks (see [18, 21] for the detailed rule of filter bank construction). The fan and parallelogram filter banks will be investigated in Section 2.2.

An example of the DFB decomposition on the image "cameraman" is shown in Fig.2.8, in which subimage indices correspond to the subbands in Fig.2.7(a). As mentioned before, the decimation in DFB skews the frequency regions of the subbands to fill in the rectangular base band, hence the resulting coefficient subbands are rectangular in the spatial domain, as shown in Fig.2.8(b).

In addition to critical sampling and perfect reconstruction properties, the DFB has the advantage of directional selectivity which is important in the applications of



(a) Original image "cameraman" (b) Directional subbands

Figure 2.8: An example of the $N = 8$ DFB decomposition on the image "cameraman". (a) The original image. (b) Its decomposed directional subbands. The dynamic range of the DFB is shifted and scaled to be $[0, 255]$, then the coefficients are colored by black as 0 and white as 255.

image analysis. However, for most images, a major part of energy is located at the low frequency and DC area which is split N ways by the DFB. As a result, the dominant low-frequency and dc energy is distributed into all directional subbands. Thus the DFB can not provide a sparse expansion of images. This limitation is illustrated in Fig.2.8(b), where the DFB coefficients tend to occupy the entire dynamic range.

2.1.4 Multiresolution and Multidirection Representation

Contourlet Transform

By combining the LP and the DFB, Do and Vetterli proposed the contourlet transform (CT) [8], which decomposes the spectrum into trapezoid-shaped subbands, as shown in Fig.2.10(a). The generation of the contourlet transform is illustrated in Fig.2.9, where a one-level LP decomposes an input image into a downsampled lowpass subband and a highpass subband, then the highpass subband is decomposed

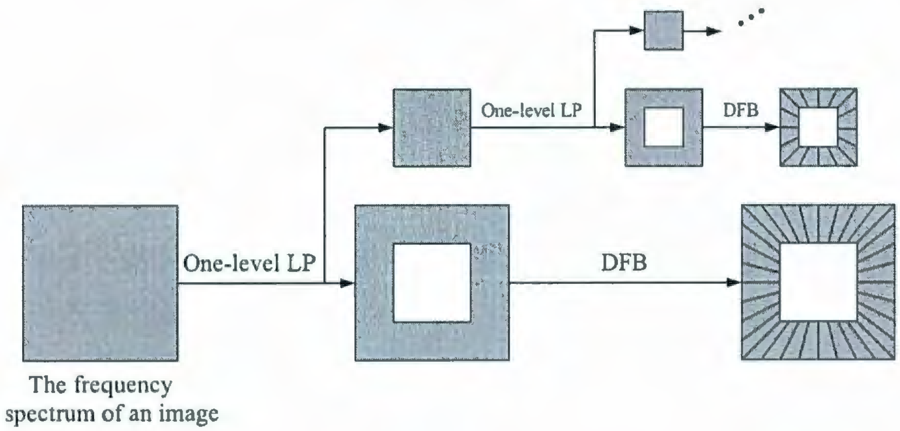


Figure 2.9: Illustration of the contourlet transform in the 2-D frequency domain, where the LP decomposes the frequency spectrum of an original image into multi-scales. Then each bandpass scale is decomposed into directional subbands by the DFB. [Based on a figure from [8]]

into various directional subbands by the DFB. To avoid the division of low frequency regions, at each stage the contourlet transform removes the low frequency component from the highpass subband before implementing the DFB decomposition. If this procedure is iterated on the lowpass subband, then a multiscale and multidirection decomposition of images is achieved. By taking the advantages of both the LP and the DFB, the contourlet transform can efficiently capture high frequency directional information in images, such as oriented edges.

A possible frequency partition by the contourlet transform is illustrated in Fig.2.10(a), by which the spectrum of an original image could be decomposed into four scales, which are then divided into one, four, four, and eight directional subbands from coarser scales to finer scales, respectively. Based on the frequency partition shown in Fig.2.10(a), the coefficient image resulting from the contourlet transform applied to

(a) Frequency partition of CT

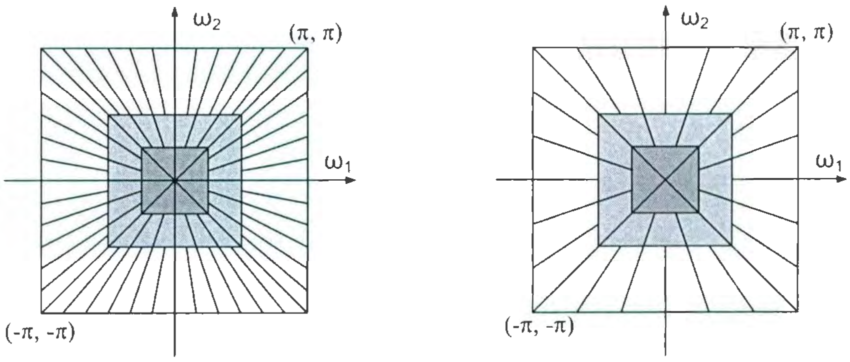
(b) Contourlet transform

Figure 2.10: Example of the contourlet transform. (a) Frequency partition, in which the spectrum of an image is decomposed into four pyramid levels (indicated by the three shades of gray and white), each of which are decomposed into one, four, four, and eight directional subbands, respectively. (b) The contourlet coefficients of the image "barbara". Small-absolute-value coefficients are colored towards black while large-absolute-value coefficients are towards white. [Contrast boosted for display]

the image "barbara" is shown in Fig.2.10(b), in which the coefficients with small absolute value are colored black. Due to the abundant number of small-absolute-value coefficients, the contourlet transform is a sparse expression of images.

Each subband of the contourlet transform, which consists of a pairwise trapezoid-shaped regions symmetric to the origin in Fig.2.10(a), corresponds to an oriented basis function. Therefore, the contourlet transform can offer basis functions oriented at 2^k different directions at each scale, where k is an arbitrary positive integer. Rich and flexible multiscale and oriented basis functions allow the contourlet transform to effectively represent smooth contours. However, the contourlet transform has the drawback of a 4/3 redundancy in its oversampling ratio, which comes from the Laplacian pyramid [8, 16].

In [16], Lu and Do proposed a critically sampled image representation, called the CRISP-contourlet transform, to eliminate the redundancy of the contourlet transform. The difference between the original contourlet transform and the CRISP-contourlet transform is that the main structure of the original contourlet transform uses a multiscale decomposition followed by a multidirection decomposition, while the CRISP-contourlet transform adds a four-band directional decomposition before the multiscale decomposition and then implements the subsequent directional division. Although the CRISP-contourlet transform removes the redundancy of oversampling, it divides the lowest frequency scale into four directional subbands, as shown in Fig.2.11(a). Recall that the low frequency area corresponds to a coarse approximation of the original image, e.g. the upper left corner of the coefficient configurations in Fig.1.2(b) and Fig.2.10(b). Division of the low frequency area splits the coarse approximation which is important in providing a rough impression of the original image, which is not expected by multiresolution representation. Therefore, the CRISP-contourlet transform is not a "true" progressive version of the contourlet transform, since it can not achieve the same frequency partition as the original contourlet transform.



(a) CRISP-contourlets (b) uniform DFB

Figure 2.11: Frequency partitions by (a) the CRISP-contourlet transform and (b) the uniform directional filter bank (uDFB).

Using the similar design method of the CRISP-contourlet transform, Nguyen and Oraintara proposed a critically sampled multiresolution and multidirection transform called the uniform directional filter bank (uDFB) [22], whose frequency partition is shown in Fig.2.11(b). In contrast to the CRISP-contourlet transform, the uDFB applies a two-band directional division before the multiscale decomposition. Unfortunately, the CRISP-contourlet transform and the uDFB both have the problem of directionally dividing the coarsest scale, as shown in Fig.2.11(a), (b). They also both leak low frequency components into high frequency directional subbands, due to their implementation of directional division before multiresolution decomposition [17]. In [22], the uDFB was extended to a nonuniform directional filter bank (nuDFB) to resolve these problems. However, the nuDFB does not preserve the perfect reconstruction property of the uDFB.

2.2 Perfect Reconstruction Filter Banks

As mentioned in the previous section, most discrete image representation systems, such as the DWT and the DFB, are constructed by digital filter banks. Therefore, this section will investigate the filter banks which can achieve perfect reconstruction (PR) property; this property requires that the original image can be exactly recovered from its decomposed subbands without errors. Moreover, all the filter banks investigated in this section also achieve the critical sampling property.

2.2.1 1-D Two-Channel PR Filter Banks

Two basic operations in filter banks are 2-fold decimation and 2-fold interpolation, as shown in Fig.2.12(a), (b). The z-transforms of the outputs from the downsampling

(a) 2-fold downsampling (b) 2-fold upsampling

Figure 2.12: Illustration of (a) the 2-fold downsampling and (b) the 2-fold upsampling in one dimension.

and upsampling blocks are, respectively, related to the original input $s(n)$ [23]:

$$S_d(z) = \frac{1}{2}\left[S(z^{\frac{1}{2}}) + S(-z^{\frac{1}{2}})\right], \tag{2.6}$$

$$S_u(z) = S(z^2), \tag{2.7}$$

where $S(z)$ is the z-transform of $s(n)$. The common structure of the 1-D two-channel filter bank is the analysis/synthesis form illustrated in Fig.2.2(a), in which the relationship between the reconstructed signal $\hat{x}(n)$ and the original signal $x(n)$ can be obtained, by using Eq.(2.6) and Eq.(2.7):

$$\hat{X}(z) = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z) + \frac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]X(-z), \tag{2.8}$$

where $X(-z)$ is an aliasing version of the original signal $X(z)$. Hence the constrain of aliasing cancellation [23] is

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0, \tag{2.9}$$

which can be satisfied with the following choice of filters:

$$G_0(z) = H_1(-z), \tag{2.10}$$

$$G_1(z) = -H_0(-z). \tag{2.11}$$

Recall that $H_0(z)$ and $H_1(z)$ are half-band lowpass and highpass filters with desired frequency responses in Fig.2.2(b). Therefore, by an appropriate relationship between

$H_0(z)$ and $H_1(z)$, which is discussed later, a two-channel filter bank without aliasing could be obtained by the choice of synthesis filters in Eq.(2.10) and Eq.(2.11). Thus the recovered signal without aliasing is expressed as follows:

$$\hat{X}(z) = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z). \tag{2.12}$$

CDF 9/7-tap Filter Bank

The filters by Cohen, Daubechies and Feauveau (CDF) [20] provide a family of filters designed under the condition of aliasing cancellation. The CDF filters are finite impulse response (FIR) filters with real-valued impulse responses. Table 2.1 shows an example of the CDF 9/7-tap analysis filters, in which the approximate values of the real impulse responses are presented. The lowpass filter $h_0(n)$ has 9 taps while the highpass filter $h_1(n)$ has 7 taps.

Table 2.1: The impulse response of the CDF 9/7-tap analysis filters [20, 24]

n	Analysis lowpass filter $h_0(n)$	Analysis highpass filter $h_1(n)$
0	0.6029490182363579	1.115087052456994
±1	0.2668641184428723	−0.5912717631142470
±2	−0.07822326652898785	−0.05754352622849957
±3	−0.01686411844287495	0.09127176311424948
±4	0.02674875741080976	0

The CDF analysis filters in Table 2.1 can be transferred to two causal filters $h_0(n-4)$ and $h_1(n-3)$ by introducing a time delay. Based on the requirement of aliasing cancellation in Eq.(2.10) and Eq.(2.11), the impulse responses of the CDF synthesis filters have the following relationships with the CDF analysis filters:

$$g_0(n-3) =(-1)^n h_1(n-3), \tag{2.13}$$

$$g_1(n-4) = -(-1)^n h_0(n-4), \tag{2.14}$$

which are simplified to be

$$g_0(n) = -(-1)^n h_1(n), \tag{2.15}$$

$$g_1(n) = -(-1)^n h_0(n), \tag{2.16}$$

The analysis/synthesis filter bank with the CDF 9/7-tap filters is called the CDF 9/7-tap filter bank, which can be easily implemented due to its tabulated coefficients. The CDF 9/7-tap filter bank almost has perfect reconstruction property if its real impulse responses are represented in a higher precision. Hence CDF filters are widely adopted in wavelet-based techniques, such as the JPEG2000 [6] and the FBI fingerprint image compression standard called wavelet/scalar quantization (WSQ) [25].

IIR QMF Polyphase Structure Filter Bank

Some applications, such as orthogonal wavelet transforms [4], require filter banks contain quadrature mirror filter (QMF) pairs. The strict QMF requirement is that $H_0(z)$ and $H_1(z)$ are mirror images of each other [26, 27], that is

$$H_1(z) = H_0(-z) \tag{2.17}$$

In general, it is easier to design FIR filters than infinite-impulse-response (IIR) filters. However, a two-channel FIR filter bank should meet all the requirements in Eq.(2.10), Eq.(2.11) and Eq.(2.17) in order to achieve both perfect reconstruction and QMF properties. In [23], the author demonstrated that a two-channel FIR filter bank achieves both PR and QMF properties if and only if the impulse response of the lowpass analysis filter $h_0(n)$ only contains two samples. In fact, a two-tap filter can not obtain as steep slopes of frequency response as expected. However, the practical filter $h_0(n)$ should be approximated to the ideal filter in Fig.2.2(b) as closely as possible. On the other hand, as shown below, a IIR filter bank can achieve both PR and QMF properties with its frequency response approximating to the ideal filter in Fig.2.2(b). Hence this thesis implements two-channel filter banks with the requirement of PR and QMF properties in IIR form.

The IIR QMF bank by Smith and Eddins [28] is a complete PR system with no aliasing, no frequency distortion and no phase distortion. The IIR QMF bank cancels the aliasing component in Eq.(2.8) with the following choices of synthesis filters [29]:

$$G_0(z) = \frac{2H_1(-z)}{[H_0(z)H_1(-z) - H_0(-z)H_1(z)]}, \tag{2.18}$$

$$G_1(z) = \frac{-2H_0(-z)}{[H_0(z)H_1(-z) - H_0(-z)H_1(z)]}. \tag{2.19}$$

If $H_0(z)$ is expressed in a polyphase form [23] as follows

$$H_0(z) = P_0(z^2) + z^{-1}P_1(z^2), \tag{2.20}$$

then $H_1(z)$ can be obtained according to the QMF requirement in Eq.(2.17):

$$H_1(z) = P_0(z^2) - z^{-1}P_1(z^2). \tag{2.21}$$

Submitting $H_0(z)$ and $H_1(z)$ to Eq.(2.18) and Eq.(2.19) leads to the polyphase expressions of $G_0(z)$ and $G_1(z)$:

$$G_0(z) = Q_0(z^2) + zQ_1(z^2), \tag{2.22}$$

$$G_1(z) = Q_0(z^2) - zQ_1(z^2), \tag{2.23}$$

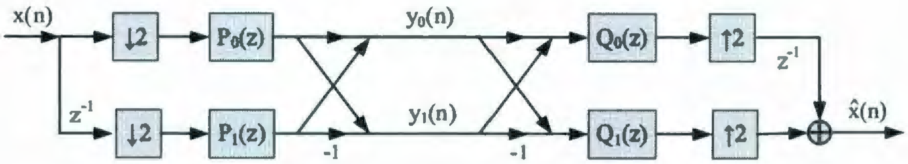


Figure 2.13: Two-channel polyphase structure filter bank

where

$$Q_0(z) = \frac{1}{P_0(z)}, \tag{2.24}$$

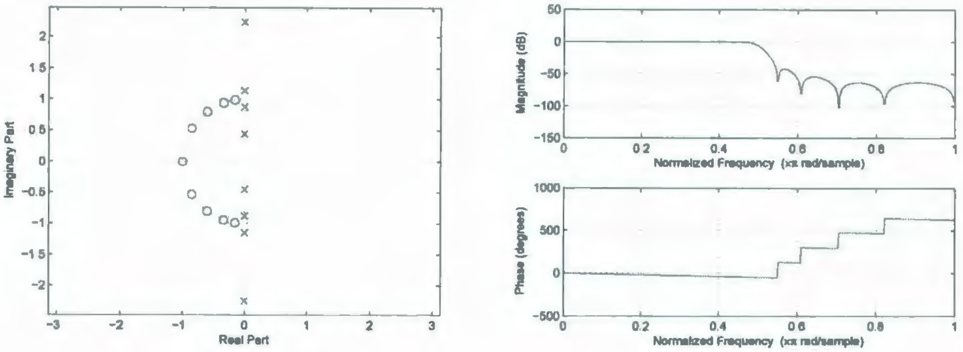
$$Q_1(z) = \frac{1}{P_1(z)}. \tag{2.25}$$

Based on the above discussion, the IIR QMF bank can be implemented by a polyphase structure shown in Fig.2.13, in which filtering operates on downsampled data, and hence at a lower data rate than the original input signal. Therefore, the polyphase structure saves almost 50% in computational cost compared with the more straightforward analysis/synthesis structure of Fig.2.2(a) [18].

According to Eq.(2.21)-(2.25), $H_1(z)$, $G_0(z)$ and $G_1(z)$ can be determined by $H_0(z)$, hence only $H_0(z)$ needs to be designed. In [28], Smith and Eddins proposed the design of $H_0(z)$ by constraining its poles on the imaginary axis in complex conjugate reciprocal quads and placing its zeros at $z = -1$ and on the unit circle in complex conjugate pairs. For example, the 8-order IIR filter $H_0(z)$ is expressed as follows [21]:

$$H_0(z) = K\frac{(1+z^{-1})S(\omega_1)S(\omega_2)S(\omega_3)S(\omega_4)}{(1+\alpha_1^2 z^{-2})\left(1+\frac{1}{\alpha_1^2}z^{-2}\right)(1+\alpha_2^2 z^{-2})\left(1+\frac{1}{\alpha_2^2}z^{-2}\right)}, \tag{2.26}$$

where $S(\omega_i) = (1 - 2\cos\omega_i z^{-1} + z^{-2})$ for $i = 1,2,3,4$. $H_0(z)$ has nine zeros which occur at -1, $e^{\pm j\omega_1}$, $e^{\pm j\omega_2}$, $e^{\pm j\omega_3}$, $e^{\pm j\omega_4}$, and eight poles located at $\pm j\alpha_1$, $\pm j/\alpha_1$, $\pm j\alpha_2$, $\pm j/\alpha_2$ with a ninth pole at ∞. Hence $H_0(z)$ is a non-causal filter. The causality of filters is insignificant for still-image processing, in which images are not time-varying.



(a) Zero-pole plane (b) Frequency response

Figure 2.14: IIR QMF 8-order filter $H_0(z)$: (a) its zero-pole plane, and (b) its log magnitude response and its phase response. [Generated from the parameters in [21]]

The zero-pole plane of $H_0(z)$ is illustrated in Fig.2.14(a), where a dashed circle represents the unit circle. The parameters K, α_1, α_2 and ω_i $(i = 1, 2, 3, 4)$ are optimized by the modified Powell's direction set method [30, 27]. The optimized parameters of the 8-order IIR filter bank are given in [21] as that $K = 0.20146905$, $\alpha_1 = 2.24245241$, $\alpha_2 = 1.14369193$, $\omega_1 = 2.57994607$, $\omega_2 = 2.21432487$, $\omega_3 = 1.91248705$, and $\omega_4 = 1.72625369$. With the frequency response shown in Fig.2.14(b), the designed $H_0(z)$ can be expressed with its polyphase terms $P_0(z)$ and $P_1(z)$, with which the practical filter bank is implemented in the polyphase structure shown in Fig.2.13.

IIR filtering assumes that an input image is known over an infinite time duration, which is not usual in practice for image processing (i.e. images are known only over observation windows). Fortunately, this problem can be resolved by periodically replicating input signals. The convolution of a periodic signal (with period N) with an aperiodic signal, which is the infinite impulse response in this case, is also a periodic signal with period N [28]. Therefore, the input signal is periodically extended, and only one period of the filtered signal needs to be retained. The detailed solution is presented in [28], and here a summary is presented for completeness.

With the restriction of the location of its zeros and poles, as illustrated in Fig.2.14(a), the designed IIR analysis filter $H_0(z)$ generates the polyphase filters $P_0(z)$, $P_1(z)$, $Q_0(z)$, and $Q_1(z)$ without poles on the unit circle of zero-pole planes. Hence these polyphase filters can be expressed in a general form:

$$P(z) = K + \sum_{i=1}^{Q_1} \frac{C_i}{1-\beta_i z^{-1}} + \sum_{j=1}^{Q_2} \frac{D_j}{1-\gamma_j z^{-1}}, \tag{2.27}$$

where $|\beta_i| < 1$ and $|\gamma_j| > 1$ represent poles inside and outside the unit circle, respectively. The total number of poles inside the unit circle is Q_1, while the total number of poles outside the unit circle is Q_2. Therefore, the region of convergence of $P(z)$ is $\max_{i\in\{1,\cdots,Q_1\}} \beta_i < z < \min_{j\in\{1,\cdots,Q_2\}} \gamma_j$, which implies that $P(z)$ is stable but not causal. However, the requirement of causality can be ignored in still-image processing, as a whole image is generally available while filtering. Therefore, the filter $P(z)$ is determined by the partial fraction expression coefficients K, C_i, β_i, D_j, and γ_j. The impulse response of $P(z)$ is denoted by $p(n)$.

As mentioned before, the input signal $x(n)$ is periodically extended to yield an infinite-length signal $\tilde{x}(n) = x(n \mod N)$, where "$n \mod N$" is the modulo operation by calculating the remainder of division of n by N. IIR filtering the periodic signal $\tilde{x}(n)$ can be implemented by the convolution of $\tilde{x}(n)$ with the impulse response $p(n)$, which generates another periodic signal $\tilde{y}(n)$ as follows:

$$\begin{aligned} \tilde{y}(n) &= \tilde{x}(n) \otimes p(n) \\ &= K\tilde{x}(n) + \sum_{k=0}^{N-1} \tilde{x}(n-k)h(k), \end{aligned} \tag{2.28}$$

where (see [28] for details)

$$h(k) = \sum_{i=1}^{Q_1} \frac{C_i \beta_i^k}{1-\beta_i^N} - \sum_{j=1}^{Q_2} \frac{D_j \gamma_j^{k-N}}{1-\gamma_j^{-N}}. \tag{2.29}$$

$h(k)$ can be calculated in advance, based on the given parameters C_i, β_i, D_j, and γ_j of $P(z)$, and $h(n)$ has the same length as $x(n)$. Without periodically replicating $x(n)$, the linear convolution in Eq.(2.29) can be replaced by a circular convolution as follows:

$$y(n) = Kx(n) + x(n) \circledast h(n), \tag{2.30}$$

where $\circledast$ denotes the circular convolution and $x(n)$ is the original discrete input signal in finite time duration.

2.2.2 2-D Two-Channel PR Filter Banks

Some 2-D filter banks can be implemented by 1-D filter banks in two dimensions separately, e.g. the structure of the 2-D DWT in Fig.2.5(a). This type of filter banks is called the separable filter bank, which, when the filters approximate ideal frequency selective filters, could generate decomposed subbands with rectangular frequency regions, such as the frequency partition of the 2-D DWT in Fig.2.5(b). In contrast, nonseparable filter banks could achieve more complex shaped frequency decompositions. In this section, several 2-D nonseparable filter banks with perfect reconstruction property are introduced. As this topic relies heavily on mathematical notations, a preface on notation is provided before the main discussion. These notations can also be viewed in the List of Symbols.

Notations

Vectors are denoted by bold-faced lower case letters while matrices are denoted by bold-faced upper case letters. The notations $\mathbf{M}^T$, $\mathbf{M}^{-1}$ and $\mathbf{M}^*$ denote the transpose, the inverse and the conjugate of a matrix $\mathbf{M}$. The notation $\det(\mathbf{M})$ denotes the determinant of the matrix $\mathbf{M}$. The symbol $\mathbf{I}_k$ denotes a $k \times k$ identity matrix.

The lattice $\Lambda(\mathbf{M})$ generated by a nonsingular matrix $\mathbf{M}$ is defined as

$$\Lambda(\mathbf{M}) = \left\{\mathbf{Mk} | \mathbf{k} \in \mathbb{Z}^2\right\}, \tag{2.31}$$

where $\mathbb{Z}^2$ denotes a two-dimensional integer coordinate space. Decimation with the sampling matrix $\mathbf{M}$ only outputs samples on the lattice $\Lambda(\mathbf{M})$.

For example, if $\mathbf{M}$ is one of the following quincunx matrices:

$$\mathbf{Q}_1 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{Q}_2 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \tag{2.32}$$

then the quincunx lattice is represented by solid dots in Fig.2.15. The quincunx downsampling only retains half the samples of the original image, due to $\det(\mathbf{Q}_1) = \det(\mathbf{Q}_2) = 2$. An example of the quincunx decimation on the image "cameraman" is illustrated in Fig.2.16, in which the decimated images are the downsampled and rotated versions of the input image.

A discrete image is described as $x(\mathbf{n})$, where $\mathbf{n} = [n_1, n_2]^T$ is the index of pixels. The z-transform of $x(\mathbf{n})$ is defined as

$$X(\mathbf{z}) = \sum_{\mathbf{n} \in \mathbb{Z}^2} x(\mathbf{n}) \mathbf{z}^{-\mathbf{n}}, \tag{2.33}$$

where $\mathbf{z} = [z_1, z_2]^T$ and $\mathbf{z}^{\mathbf{n}} = z_1^{n_1} z_2^{n_2}$.

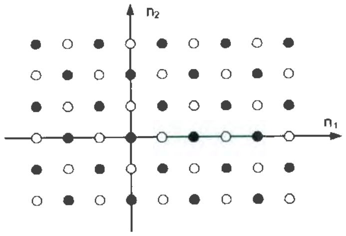


Figure 2.15: The lattice generated by the quincunx matrices, in which the solid dots represent the samples on the lattice.

(a) Original image (b) Downsampled by $\mathbf{Q}_1$ (c) Downsampled by $\mathbf{Q}_2$

Figure 2.16: Example of quincunx downsampling on the image "cameraman". (a) Original image. (b) and (c) Quincunx downsampled image by $\mathbf{Q}_1$ and $\mathbf{Q}_2$, respectively.

Analysis/Synthesis Filter Bank

A typical two-channel analysis/synthesis filter bank is shown in Fig.2.17, in which $H_0(\mathbf{z})$, $H_1(\mathbf{z})$ are 2-D analysis filters and $G_0(\mathbf{z})$, $G_1(\mathbf{z})$ are 2-D synthesis filters. The system has perfect reconstruction property if $\hat{x}(\mathbf{n}) = kx(\mathbf{n}-\mathbf{c})$, where k is a constant number and $\mathbf{c}$ is a constant vector [31]. On the other hand, the system is critically sampled if the total number of samples of the decomposed coefficient images $y_0(\mathbf{n})$ and $y_1(\mathbf{n})$ is equal to the number of pixels of the original image $x(\mathbf{n})$.

In a general two-channel filter bank, downsampling matrices for both the lowpass and the highpass channels are usually the same, that is the matrix $\mathbf{M}$ in Fig.2.17. Therefore, $y_0(\mathbf{n})$ and $y_1(\mathbf{n})$ contain the same number of coefficients, which is 50% of the number of pixels of the original image $x(\mathbf{n})$ if this 2-channel filter bank requires

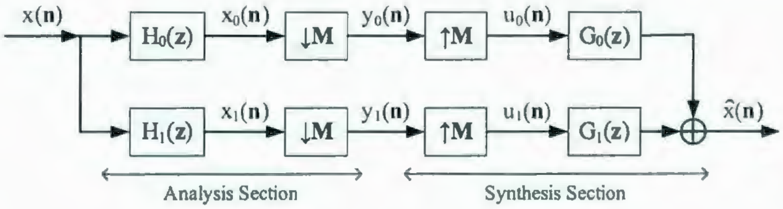
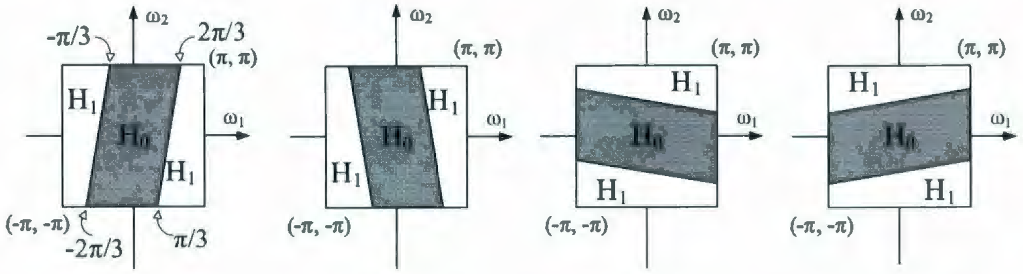


Figure 2.17: Two-channel analysis/synthesis filter bank



(a) SPR-I (b) SPR-II (c) SPR-III (d) SPR-IV

Figure 2.18: Four types of 2-D SPR support configurations. H_0 and H_1 indicate the passband regions of the ideal analysis lowpass and highpass filters, respectively.

critical sampling. The 50% samples retained by the decimator $\mathbf{M}$ implies that its determinant equals 2 (i.e. $\det(\mathbf{M}) = 2$).

A two-channel filter bank is usually defined by the supports of the analysis filters $H_0(\mathbf{z})$, $H_1(\mathbf{z})$, which are ideal bandpass filters, in theory, or carefully selected approximations, in practice. In this thesis, the two-channel analysis/synthesis structure is used to generate the filter bank with sheared-parallelogram (SPR) shaped supports, as illustrated in Fig.2.18. The gray and the white regions, respectively, represent the support of the ideal filters $H_0(\mathbf{z})$ and $H_1(\mathbf{z})$. In common cases, the synthesis filters have the same supports as the analysis filters. Therefore, only the supports of the analysis filters are shown in Fig.2.18. In practice, the analysis and synthesis filters can be designed using the transformation of variables technique [32] to achieve perfect reconstruction property. The detailed design of the SPR type-I filter bank will be described below.

The supports of the SPR type-I analysis filters are shown in Fig.2.18(a), in which the parallelogram shaped supports are fixed by four vertices $(\omega_1, \omega_2) = (-\frac{\pi}{3}, \pi)$, $(\frac{2\pi}{3}, \pi)$, $(-\frac{2\pi}{3}, -\pi)$, and $(\frac{\pi}{3}, -\pi)$. The name "sheared" comes from the fact that the four fixing vertices are not the midpoint or the vertices of the borders of baseband.

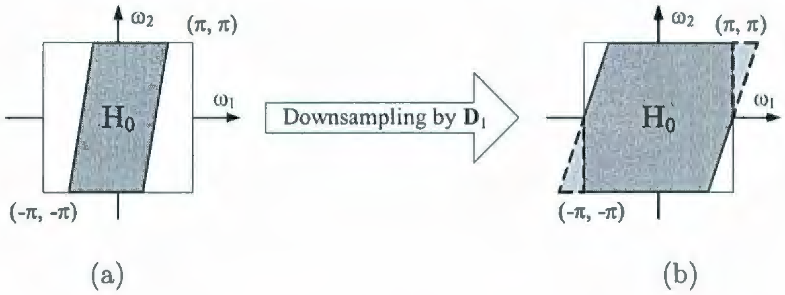


Figure 2.19: Downsampling the SPR type-I filtered region in 2-D by the sampling matrix $\mathbf{D}_1$. (a) The SPR-I lowpass filtered region H_0. (b) The region H_0 after downsampling. Overlapping appears between the periodic replicas of the baseband.

Due to the symmetry among the four SPR supports, the other three filter banks (SPR-II, III, IV) have similar fixing points.

The sampling matrices in the SPR filter banks are the diagonal matrices as follows:

$$\mathbf{D}_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{D}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}. \tag{2.34}$$

With $\mathbf{D}_1$ or $\mathbf{D}_2$, images are decimated only by rows or columns, respectively. The sampling matrix corresponding to the sheared-parallelogram type-I filter bank is $\mathbf{D}_1$. The original SPR shaped region, as shown in Fig.2.19(a), overlaps onto the periodic replicas of the $2\pi \times 2\pi$ baseband after downsampling. As illustrated in Fig.2.19(b), the lighter gray regions represent downsampled frequencies which extend beyond 2π. Due to the 2π periodicity of the discrete frequency space and the characteristics of parallelogram, the downsampled signal is not distorted by overlapping.

Since decimation by $\mathbf{D}_1$ only operates downsampling in one dimension, the z-transforms of the $\mathbf{D}_1$-fold decimation and the $\mathbf{D}_1$-fold interpolation have the similar relationships as the one-dimensional case in Eq.(2.6) and Eq.(2.7):

$$\begin{aligned} S_d(z_1, z_2) =& \frac{1}{2}\left[S(z_1^{\frac{1}{2}}, z_2) + S(-z_1^{\frac{1}{2}}, z_2)\right], & (2.35)\\ S_u(z_1, z_2) =& S(z_1^2, z_2), & (2.36)\end{aligned}$$

where $S(z_1, z_2)$, $S_d(z_1, z_2)$ and $S_u(z_1, z_2)$ respectively represent the z-transforms of the input image, the $\mathbf{D}_1$-fold downsampled image, and the $\mathbf{D}_1$-fold upsampled image.

Based on the above generic relationship of the diagonal downsampling and upsampling, the input/output of the SPR-I filter bank has the following relationship:

$$\begin{aligned}\hat{X}(z_1, z_2) = & \frac{1}{2}\left[H_0(z_1, z_2)G_0(z_1, z_2) + H_1(z_1, z_2)G_1(z_1, z_2)\right]X(z_1, z_2) + \\ & \frac{1}{2}\left[H_0(-z_1, z_2)G_0(z_1, z_2) + H_1(-z_1, z_2)G_1(z_1, z_2)\right]X(-z_1, z_2), \quad (2.37)\end{aligned}$$

where $X(-z_1, z_2)$ describes the aliasing component, which can be eliminated with the following choice of highpass filters [32]:

$$\begin{aligned} H_1(z_1, z_2) =& z_1^{-1}G_0(-z_1, z_2), & (2.38)\\ G_1(z_1, z_2) -& z_1 H_0(-z_1, z_2). & (2.39)\end{aligned}$$

After eliminating the aliasing component, the analysis/synthesis filter bank is a time-invariant system with the following transfer function:

$$\begin{aligned} T(z_1, z_2) &= \frac{1}{2}\left[H_0(z_1, z_2)G_0(z_1, z_2) + H_1(z_1, z_2)G_1(z_1, z_2)\right] \\ &= \frac{1}{2}\left[H_0(z_1, z_2)G_0(z_1, z_2) + H_0(-z_1, z_2)G_0(-z_1, z_2)\right] \quad (2.40)\\ &= \frac{1}{2}\left[E(z_1, z_2) + E(-z_1, z_2)\right], \end{aligned}$$

where $E(z_1, z_2) = H_0(z_1, z_2)G_0(z_1, z_2)$. If $E(z_1, z_2)$ satisfies the following constraint:

$$E(z_1, z_2) + E(-z_1, z_2) = 2, \quad (2.41)$$

then a PR 2-channel filter bank is achieved. Moreover $H_0(z_1, z_2)$ and $G_0(z_1, z_2)$ can be determined by factorizing $E(z_1, z_2)$. Therefore, the design of an analysis/synthesis filter bank is reduced to the design of a filter $E(z_1, z_2)$ under the condition of Eq.(2.41).

$E(z_1, z_2)$ can be designed using the transformation of variables technique [32] with the following steps. The first step is to design a 2-D FIR lowpass filter $R(z_1, z_2)$ with a sheared-parallelogram shaped passband by windowing [33]. The impulse response of the ideal SPR-I lowpass filter without windowing is given by [16]:

$$r_{SPR-I}(n_1, n_2) = \mathrm{sinc}\left(\frac{n_1\pi}{2}\right)\mathrm{sinc}\left(n_2\pi + \frac{n_1\pi}{6}\right), \tag{2.42}$$

where $\mathrm{sinc}(x) - \frac{\sin(x)}{x}$. Thus the windowed FIR SPR-I lowpass filter is expressed as follows:

$$r(n_1, n_2) = r_{SPR-I}(n_1, n_2)w(n_1, n_2), \tag{2.43}$$

in which $w(n_1, n_2)$ is the window function with the following form:

$$w(n_1, n_2) = m\left(n_1\right) m\left(2n_2 + \frac{n_1}{3}\right), \tag{2.44}$$

where $m(n)$ is a common 1-D window function, such as the Hamming window, the Chebyshev window, or the Kaiser window [33]. Therefore, it can be proved that $R(z_1, z_2)$, which is the z-transform of $r(n_1, n_2)$, satisfies the following constraint:

$$R(-z_1, z_2) = -R(z_1, z_2). \tag{2.45}$$

Proof. Given that

$$r(n_1, n_2) = \mathrm{sinc}\left(\frac{n_1\pi}{2}\right)\mathrm{sinc}\left(n_2\pi + \frac{n_1\pi}{6}\right) m\left(n_1\right) m\left(2n_2 + \frac{n_1}{3}\right). \tag{2.46}$$

Thus $R(z_1, z_2)$ is the z-transform of $r(n_1, n_2)$, as follows:

$$R(z_1, z_2) = \sum_{n_1}\sum_{n_2} r(n_1, n_2) z_1^{-n_1} z_2^{-n_2}. \tag{2.47}$$

Since $\mathrm{sinc}(x) = \frac{\sin(x)}{x}$, which implies that

$$r(n_1, n_2) = \begin{cases} 0 & \text{if } n_1 = \text{even}, \\ \text{arbitrary} & \text{if } n_1 = \text{odd}. \end{cases} \tag{2.48}$$

Therefore,

$$R(-z_1, z_2) = \sum_{n_1=\text{odd}} \sum_{n_2} r(n_1, n_2)(-z_1)^{-n_1} z_2^{-n_2} = -R(z_1, z_2) \tag{2.49}$$

□

After the 2-D FIR filter $R(z_1, z_2)$ is generated, the second step is to produce a QMF pair based on $R(z_1, z_2)$. Let $J = R(z_1, z_2)$, find a 1-D transformation with the form of $E_T(J) = 1 + a_1 J + a_3 J^3 + a_5 J^5 + \cdots$, which implies that

$$E_T(J) + E_T(-J) = 2, \tag{2.50}$$

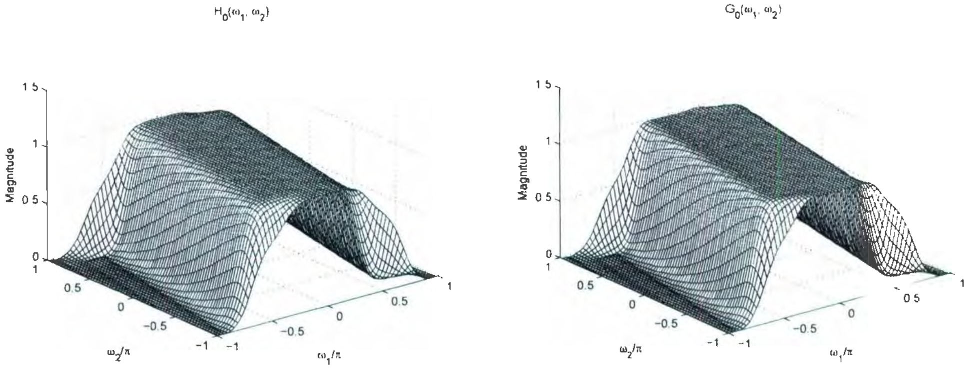
where $E_T(J) = H_T(J)G_T(J)$, $H_T(J)$ and $G_T(J)$ are the designed analysis QMF pair.

Tay and Kingsbury recommended various choices of $E_T(J)$ (i.e. $H_T(J)$ and $G_T(J)$) in [32]. With the balance between the sharpness of roll-off region and the ripple characteristics in passband and stopband, the factors of $E_T(J)$ are chosen as follows [32]:

$$H_T(J) = -\frac{1}{5}(J+1)\left(J - \frac{7}{2}\right), \tag{2.51}$$

$$G_T(J) = -\frac{3}{35}(J+1)\left(J^2 + \frac{3}{2}J - \frac{25}{3}\right). \tag{2.52}$$

After submitting the previous designed 2-D FIR filter $R(z_1, z_2)$ to the variable J in Eq.(2.51) and Eq.(2.52), $H_T(R(z_1, z_2))$ and $G_T(R(z_1, z_2))$ are the designed analysis and synthesis lowpass filters $H_0(z_1, z_2)$ and $G_0(z_1, z_2)$ for the SPR type-I filter bank. If the length of the window function $m(n)$ is set to be 15, then the designed $H_0(z_1, z_2)$ and $G_0(z_1, z_2)$ have the 2-D magnitude responses shown in Fig.2.20.



(a) (b)

Figure 2.20: 2-D magnitude responses of the FIR SPR-I filters: (a) the analysis lowpass filter $H_0(z_1, z_2)$ and (b) the synthesis lowpass filter $G_0(z_1, z_2)$

The transformation of variables technique can generate two-channel filter banks with numerous shapes of supports, but with a high computational complexity. Therefore, the following section introduces a ladder structure filter bank, which has the advantages of flexible implementation and lower computational complexity in addition to the perfect reconstruction property.

Ladder Structure Filter Bank

The primal ladder-structure filter bank was proposed in one dimension by Phoong et al [34]. The ladder structure achieves PR property due to its structure. Therefore, images can be exactly recovered from the decomposed subbands even though the impulse responses of the filters are represented in a limited precision. In [31], the 1-D ladder structure was extended to a 2-D case, which will be investigated below.

The ladder structure comes from the polyphase representation of a 2-D 2-channel filter bank which is presented in Fig.2.21, in which $E(\mathbf{z})$ and $R(\mathbf{z})$ are analysis and

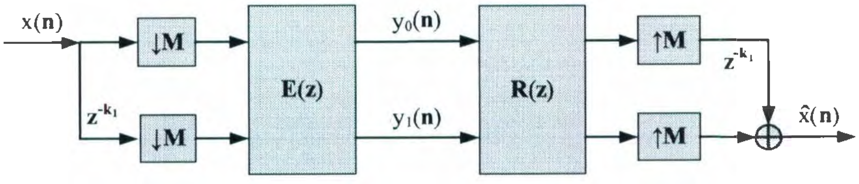


Figure 2.21: The polyphase representation of a 2-D two-channel filter bank

synthesis polyphase processors, and can be represented in a matrix form as follows

$$\mathbf{E}(\mathbf{z}) = \begin{pmatrix} E_{00}(\mathbf{z}) & E_{01}(\mathbf{z}) \\ E_{10}(\mathbf{z}) & E_{11}(\mathbf{z}) \end{pmatrix}, \qquad \mathbf{R}(\mathbf{z}) = \begin{pmatrix} R_{00}(\mathbf{z}) & R_{01}(\mathbf{z}) \\ R_{10}(\mathbf{z}) & R_{11}(\mathbf{z}) \end{pmatrix}. \tag{2.53}$$

The condition of perfection reconstruction is given by

$$\mathbf{R}(\mathbf{z}) = \mathbf{E}^{-1}(\mathbf{z}). \tag{2.54}$$

Comparing the analysis/synthesis system in Fig.2.17 with the polyphase system in Fig.2.21, the former implements decimation after filtering, while the latter down-samples input images before filtering. In fact, the polyphase representation can be identified with a corresponding analysis/synthesis system based on the Noble identities of decimators and interpolators illustrated in Fig.2.22. The Noble identities are valuable in multirate image processing, since they allow the movement of decimators and interpolators across linear filters.

With the help of the Noble identities, the polyphase representation has their corresponding analysis and synthesis filters expressed as follows:

$$\begin{pmatrix} H_0(\mathbf{z}) \\ H_1(\mathbf{z}) \end{pmatrix} = \begin{pmatrix} E_{00}(\mathbf{z}^{\mathbf{M}}) & E_{01}(\mathbf{z}^{\mathbf{M}}) \\ E_{10}(\mathbf{z}^{\mathbf{M}}) & E_{11}(\mathbf{z}^{\mathbf{M}}) \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{z}^{-\mathbf{k}_1} \end{pmatrix} \tag{2.55}$$

$$\begin{pmatrix} G_0(\mathbf{z}) & G_1(\mathbf{z}) \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{-\mathbf{k}_1} & 1 \end{pmatrix} \begin{pmatrix} R_{00}(\mathbf{z}^{\mathbf{M}}) & R_{01}(\mathbf{z}^{\mathbf{M}}) \\ R_{10}(\mathbf{z}^{\mathbf{M}}) & R_{11}(\mathbf{z}^{\mathbf{M}}) \end{pmatrix}, \tag{2.56}$$

(a) Noble identity of decimators

(b) Noble identity of interpolators

Figure 2.22: Noble identities for multirate systems. (a) Equivalent cascades with decimators. (b) Equivalent cascades with interpolators

where $\mathbf{k}_1 = [1,0]^T$ is a non-zero coset vector [35]. In general, a 2-D two-channel filter bank has two coset vectors $\mathbf{k}_0 = [0,0]^T$ and $\mathbf{k}_1$, which are, respectively, corresponded to the lattice $\Lambda(\mathbf{M})$ and its shifted version by 1 along one dimension (e.g. the quincunx lattice composed of the solid dots in Fig.2.15 and the remaining circles in Fig.2.15 forming a shifted version of the quincunx lattice).

The ladder structure filter bank is a specific scenario of the polyphase representation with the following choice of $\mathbf{E}(\mathbf{z})$ and $\mathbf{R}(\mathbf{z})$ [34, 31]:

$$\mathbf{E}(\mathbf{z}) = \begin{pmatrix} \frac{1}{2} & 0 \\ -\frac{1}{2}\mathbf{z}^{-\mathbf{d}}\beta(\mathbf{z}) & 1 \end{pmatrix} \begin{pmatrix} 1 & \beta(\mathbf{z}) \\ 0 & 1 \end{pmatrix} \tag{2.57}$$

$$\mathbf{R}(\mathbf{z}) = \begin{pmatrix} 1 & -\beta(\mathbf{z}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{1}{2}\mathbf{z}^{-\mathbf{d}}\beta(\mathbf{z}) & \frac{1}{2} \end{pmatrix}. \tag{2.58}$$

It can be proved that $\mathbf{E}(\mathbf{z})\mathbf{R}(\mathbf{z}) = \frac{1}{4}\mathbf{I}_2$. Hence the system meets perfect reconstruction property regardless of $\beta(\mathbf{z})$ and $\mathbf{d}$. According to Eq.(2.57) and Eq.(2.58), $\mathbf{E}(\mathbf{z})$ and $\mathbf{R}(\mathbf{z})$ can be efficiently implemented with a ladder form shown in Fig.2.23.

With different $\beta(\mathbf{z})$, $\mathbf{d}$ and $\mathbf{M}$, the ladder structure can generate filter banks with numerous support configurations, such as the separable, diamond, quadrant, and parallelogram supports in Fig.2.24. The parameters $\mathbf{M}$, $\beta(\mathbf{z})$ and $\mathbf{d}$ for different supports are shown in Table 2.2, in which $\mathbf{D}_1$ and $\mathbf{D}_2$ are the diagonal matrices in

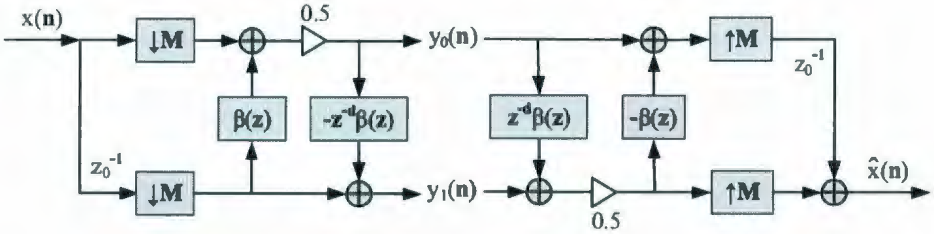


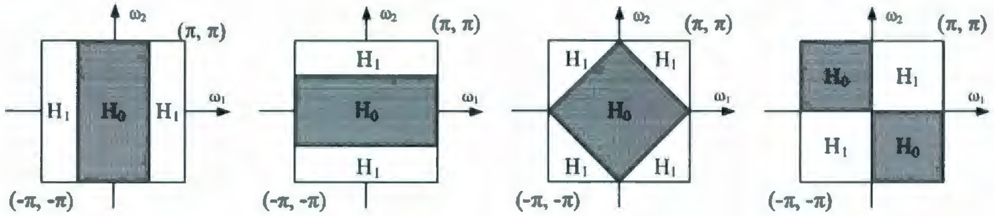
Figure 2.23: The ladder structure of two-channel filter bank

Table 2.2: The parameters $\mathbf{M}$, $\beta(\mathbf{z})$ and $\mathbf{d}$ of numerous ladder structure filter banks

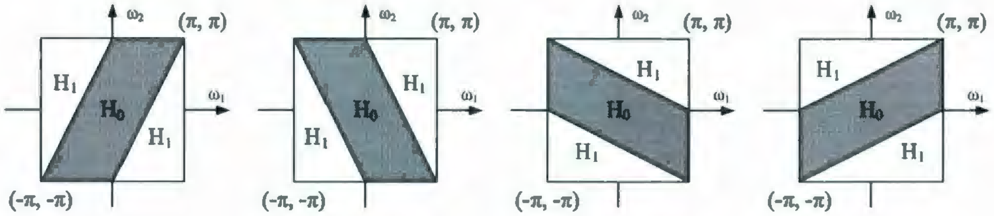
Filter Bank Name	$\mathbf{M}$	$\beta(\mathbf{z})$	$\mathbf{d}$
(a) Separable-I	$\mathbf{D}_1$	$\alpha(z_1)$	$\mathbf{d}_{10}$
(b) Separable-II	$\mathbf{D}_2$	$\alpha(z_2)$	$\mathbf{d}_{01}$
(c) Diamond	$\mathbf{Q}_1$ or $\mathbf{Q}_2$	$\alpha(z_1)\alpha(z_2)$	$\mathbf{d}_{11}$
(d) Quadrant	$\mathbf{D}_1$	$-z_2^{-1}\alpha(-z_1)\alpha(-z_2^2)$	$\mathbf{d}_{10}$
	$\mathbf{D}_2$	$-z_1^{-1}\alpha(-z_1^2)\alpha(-z_2)$	$\mathbf{d}_{01}$
(e) Parallel-I	$\mathbf{P}_1$	$\alpha(z_1)\alpha(z_2)$	$\mathbf{d}_{11}$
(f) Parallel-II	$\mathbf{P}_2$		
(g) Parallel-III	$\mathbf{P}_3$		
(h) Parallel-IV	$\mathbf{P}_4$		

Eq.(2.34), and $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are the quincunx matrices in Eq.(2.32). Four parallelogram matrices $\mathbf{P}_i$ $(i = 1, 2, 3, 4)$ and three alternative vectors of $\mathbf{d}$ are shown as follows:

$$\begin{aligned} \mathbf{P}_1 &= \begin{pmatrix} 2 & 0 \\ -1 & 1 \end{pmatrix}, \qquad \mathbf{P}_2 = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} \\ \mathbf{P}_3 &= \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}, \qquad \mathbf{P}_4 = \begin{pmatrix} 1 & -1 \\ 0 & 2 \end{pmatrix}, \end{aligned} \tag{2.59}$$



(a) Separable-I (b) Separable-II (c) Diamond (d) Quadrant



(e) Parallel-I (f) Parallel-II (g) Parallel-III (h) Parallel-IV

Figure 2.24: 2-D analysis filter support configurations. (a)-(b) Two types of the separable filter bank (Separable-I, II). (c) The diamond filter bank. (d) The quadrant filter bank. (e)-(h) Four types of the parallelogram filter bank (Parallel-I, II, III, IV).

$$\mathbf{d}_{01} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{d}_{10} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{d}_{11} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{2.60}$$

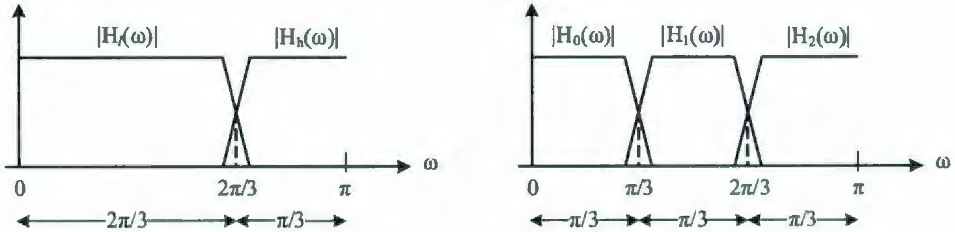
In Table 2.2, a basic operation is the 1-D filter $\alpha(z)$, which is a type II FIR lowpass filter. The design of $\alpha(z)$ can control the frequency characteristics of the resulting 2-D filters, including the roll-off sharpness and the ripples in passband and stopband. An appropriate $\alpha(z)$ of length 12 is recommended in [36], in which the right half of $\alpha(z)$ is $[0.6300, -0.1930, 0.0972, -0.0526, 0.0272, -0.0144]$ and the left half is a mirror image of the right half. If the order of the ladder structure filter bank is defined by the length of $\alpha(z)$, then the 12-order ladder structure is used in this thesis to generate filter banks with the supports shown in Fig.2.24. Note that all these supports can be implemented with a generic structure in Fig.2.23.

In fact, the separable supports in Fig.2.24(a) and (b) can be implemented by the CDF 9/7-tap filter bank, the IIR QMF bank, or the ladder structure. Among these systems, the ladder structure can achieve higher order filter responses with lower computational complexity. However, the CDF filters are adopted in many wavelet-based applications, while the IIR QMF bank possesses QMF property. Therefore, this thesis implements the ladder structure for the separable supports, except for the comparison between the proposed NRCT and the DWT, which uses the CDF 9/7-tap filters, and when the QMF property is required, by the IIR QMF bank.

2.2.3 1-D Nonuniform (2/3, 1/3) Filter Bank

The previous sections investigate various uniform two-channel PR filter banks. This section introduces a nonuniform (2/3, 1/3) filter bank with the 1-D frequency partition shown in Fig.2.25(a), where the bandwidth of the lowpass filter $H_l(\omega)$ is $2\pi/3$ and the highpass filter $H_h(\omega)$ as $\pi/3$, hence its name is the nonuniform (2/3, 1/3) filter bank [37]. In contrast, a uniform 3-channel frequency splitting is illustrated in 2.25(b), in which the lowpass, bandpass and highpass subbands have equal bandwidth of $\pi/3$.

In [38], Kovačević and Vetterli developed the design of such a critically sampled nonuniform (2/3, 1/3) filter bank with perfect reconstruction property. The procedure is illustrated in Fig.2.26, in which the first stage is a uniform 3-channel decomposition, and the second stage combines the lowpass and bandpass outputs from the 3-channel decomposition to form $y_l(n)$, while the highpass output constructs $y_h(n)$ directly. $y_l(n)$ and $y_h(n)$ are the two decomposed subbands by the nonuniform (2/3, 1/3) filter bank. The corresponding nonuniform analysis lowpass and highpass filters are



(a) Nonuniform (2/3, 1/3) filter bank (b) Uniform 3-channel filter bank

Figure 2.25: 1-D frequency division by (a) the nonuniform (2/3, 1/3) filter bank, where the cutoff frequency is $2\pi/3$, and (b) the uniform 3-channel filter bank, in which the spectrum is decomposed into lowpass, bandpass and highpass subbands equally.

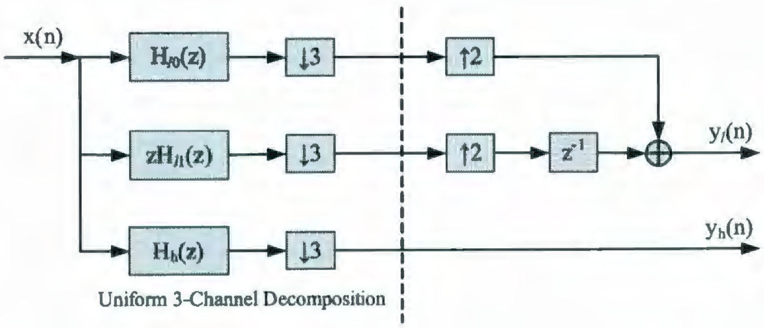


Figure 2.26: The analysis section of the nonuniform (2/3, 1/3) filter bank

expressed as follows:

$$H_l(z) = H_{l0}(z^2) + z^2 H_{l1}(z^2) z^{-3} = H_{l0}(z^2) + z^{-1} H_{l1}(z^2) \tag{2.61}$$

$$H_h(z) = H_h(z), \tag{2.62}$$

where the polyphase components of $H_l(z)$, i.e. $H_{l0}(z)$ and $H_{l1}(z)$, are the lowpass and the delayed bandpass filters in Fig.2.26. $H_h(z)$ is the analysis highpass filter of the uniform 3-channel filter bank. The uniform 3-channel filter bank in Fig.2.26 can be constructed by a lattice structure [39] to obtain the desired $H_{l0}(z)$, $H_{l1}(z)$ and $H_h(z)$.

Lattice Structure Filter Banks

As illustrated in Fig.2.27, the lattice structure [39] can be viewed as a cascade of lossless matrices $\mathbf{K}_i$ for $i = 1, 2, \cdots, L$. Hence the cascaded system is lossless, which guarantees the perfect reconstruction property of the system [39, 40]. The lossless matrix $\mathbf{K}_i$ is chosen to be a sequence of planar rotations, as follow [39]:

$$\mathbf{K}_i = \begin{pmatrix} \cos\theta_{1,i} & \sin\theta_{1,i} & 0 \\ \sin\theta_{1,i} & -\cos\theta_{1,i} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{2,i} & \sin\theta_{2,i} \\ 0 & \sin\theta_{2,i} & -\cos\theta_{2,i} \end{pmatrix}, \tag{2.63}$$

which can be efficiently implemented by a lattice structure illustrated in Fig.2.28.

As shown in Fig.2.27 and Fig.2.28, the uniform 3-channel lattice structure filter bank is determined by the parameters $\theta_{1,i}$ and $\theta_{2,i}$, for $i = 1, 2, \cdots, L$. These

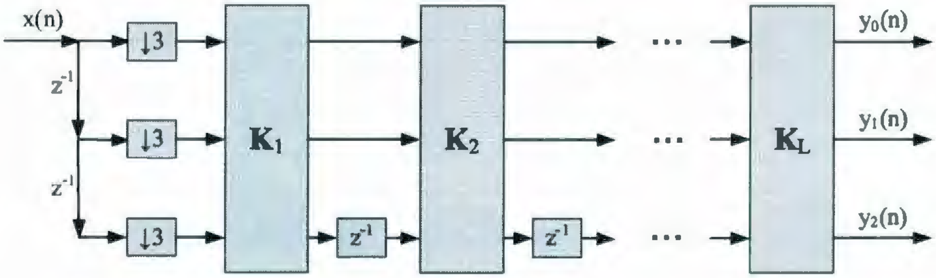


Figure 2.27: The analysis section of the uniform 3-channel filter bank

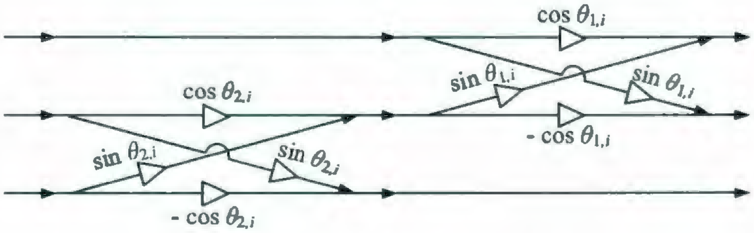


Figure 2.28: The lattice structure of $\mathbf{K}_i$

parameters are optimized by minimizing error functions for $H_l(z)$ and $H_h(z)$, defined as the differences between the expected filter responses and the designed results. The optimized parameters of the PR nonuniform (2/3, 1/3) filter bank with $L = 5$ are given in [38], as shown in Table 2.3. Although the lattice structure filter bank requires higher design complexity to optimize the parameters, the designed system can be easily implemented once the optimum parameters are obtained.

Table 2.3: The parameters of the PR nonuniform (2/3, 1/3) filter bank [38]

$\theta_{1,1}$	1.848559	$\theta_{2,1}$	-2.004359
$\theta_{1,2}$	0.580876	$\theta_{2,2}$	-1.752920
$\theta_{1,3}$	1.199016	$\theta_{2,3}$	2.672730
$\theta_{1,4}$	-1.385383	$\theta_{2,4}$	-1.574727
$\theta_{1,5}$	3.005676	$\theta_{2,5}$	1.957543

2.2.4 Summary of Filter Banks

Numerous PR filter banks have been introduced in this Section 2.2. A summary is presented in Table 2.4, in which "PR" represents perfect reconstruction property, while "CS" denotes critical sampling property. All the filter banks investigated possess these two properties.

The last column in Table 2.4 lists some types of supports which can be realized by the corresponding filter banks. This thesis considers the comprehensive characteristics of all these filter bank structures, and then decides the optimum structure for a certain support, as shown in the "supports" column of Table 2.4. Except for the separable-I and II supports, which can be realized by the CDF 9/7-tap filter bank, the IIR QMF bank or the ladder structure filter bank, this thesis implements the other supports by

Table 2.4: The summary of PR filter banks

Filter bank	Dimension	Properties	Supports
CDF 9/7-tap	1-D	PR/CS	Separable-I,II
IIR QMF polyphase	1-D	PR/CS	Separable-I,II
Analysis/synthesis & variable transformation	2-D	PR/CS	SPR-I, II, III, IV
Ladder structure	2-D	PR/CS	Separable-I,II, Diamond, Quadrant, Parallel-I,II,III,IV
Lattice structure	1-D	PR /CS	Nonuniform (2/3, 1/3)

the given filter bank structures listed in Table 2.4.

2.3 Summary of Chapter

This chapter has investigated a variety of image representation techniques including image pyramids, the wavelet transform, the directional filter bank, and the contourlet transform. The limitation of the contourlet transform is a 4/3 redundancy in its oversampling ratio. To overcome this issue, a nonredundant contourlet transform (NRCT) will be proposed in Chapter 3. This chapter has also investigated the design of numerous uniform and nonuniform filter banks with perfect reconstruction and critical sampling properties. These filter banks are the fundamental structures for building the nonredundant contourlet transform.

Chapter 3

Nonredundant Contourlet Transform (NRCT)

The motivation of this thesis is the elimination of the redundant oversampling of the contourlet transform. In [41], Zhang and Moloney analyzed the feasibility to eliminate the redundancy and proposed a redundancy reduction scheme. This chapter proposes a new multiresolution and multidirection image representation with perfect reconstruction and critical sampling properties. The proposed algorithm achieves the same frequency partition as the contourlet transform, while eliminating its redundancy. Hence the new image representation is called the nonredundant contourlet transform (NRCT).

The proposed NRCT is constructed by a framework of filter banks. Numerous non-ideal filter banks with perfect reconstruction and critical sampling properties have been investigated in Section 2.2. This chapter focuses on the configuration of these filter banks to construct the NRCT system. Hence the filter banks are simply represented by block diagrams showing the ideal filter supports and decimation matrices. These ideal filters are realized by non-ideal filters, with passbands approxi-

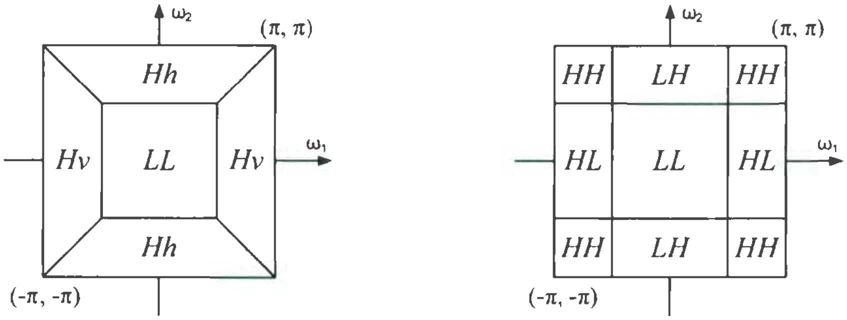
(a) Diamond filter bank (b) Quadrant filter bank

Figure 3.1: Examples of 2-channel filter bank (analysis section): (a) diamond filter bank, (b) quadrant filter bank. Ideal filter supports are indicated by black as passband and white as stopband.

mating the supports of the ideal filters used in this chapter, designed by the methods presented in Section 2.2. For example, the diamond and quadrant filter banks are illustrated in Fig.3.1(a) and (b), respectively, where black regions indicate the filter passbands and white indicates the stopbands of the ideal filters. Note that only the analysis sections are presented, when the synthesis sections are the inverse procedure of the analysis sections. The horizontal and vertical directions of the square filter spectrum, respectively, represent the ω_1 and ω_2 directions, which are omitted in the illustration in Fig.3.1.

3.1 2-Directional Wavelet Transform

A key element in the structure of the nonredundant contourlet transform is the 2-directional wavelet transform, which has the frequency partition depicted in Fig.3.2(a). This transform maximally decimates an input image into one lowpass subband (*LL*), one horizontal highpass subband (*Hh*) and one vertical highpass subband (*Hv*). By contrast, the discrete wavelet transform (DWT) decomposes an image into one coarser scale (*LL*) and three finer scales (*LH*, *HL*, *HH*), respectively, corresponding to the horizontal, vertical and diagonal directional information, as illustrated in Fig.3.2(b).



(a) 2-directional wavelet transform (b) Wavelet transform

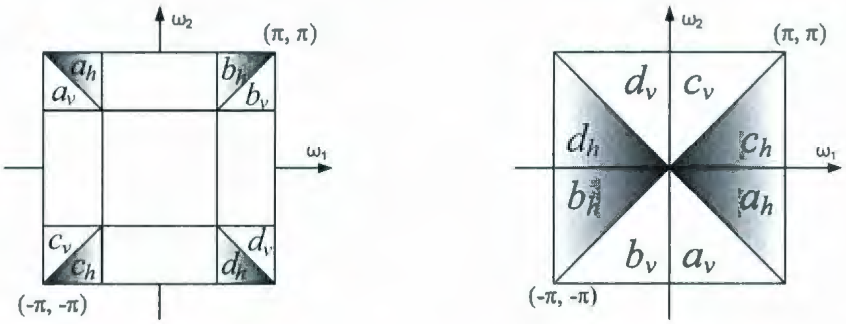
Figure 3.2: 2-D frequency division by (a) the 2-directional wavelet transform, and (b) the separable wavelet transform.

Therefore, the main idea for generating the 2-directional wavelet transform is to first implement the traditional wavelet transform, and then combine the three wavelet highpass subbands (*LH*, *HL*, *HH*) to form *Hh* and *Hv*.

Note that *Hh* and *Hv* split the DWT diagonal highpass subband *HH*, as indicated in Fig.3.2(a) and (b). Hence the separation of *HH* to generate two appropriate subbands, called the *h*-*v* decomposition of *HH*, is a necessary stage of the 2-directional wavelet transform.

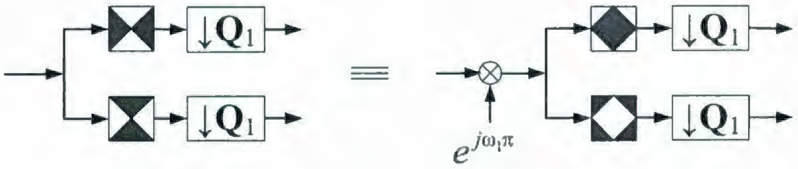
h-*v* Decomposition of *HH*

As illustrated in Fig.3.3(a), the frequency regions of *HH* are located at the four corners of the spectrum of an input image, as indicated by the letters "*a*", "*b*", "*c*", and "*d*". Each corner is further divided into two parts, which are indicated by the subscripts "*h*" and "*v*" to refer to the horizontal and vertical information in the corresponding half-corners, respectively. If the four frequency regions a_h, b_h, c_h, and d_h, colored by a gray gradient in Fig.3.3(a), are collected into a single subband *Dh*, then *Dh* can be combined with *LH* to form *Hh*. In this manner, *HH* is decomposed into two



(a) Input image (b) *HH* subband

Figure 3.3: The frequency mapping of the *HH* subband. (a) The frequency regions of *HH* in the spectrum of the input image. (b) The frequency contents of *HH* after decimation as in Fig.3.5



(a) (b)

Figure 3.4: The identity between (a) the fan filter bank and (b) a frequency shifted version of the diamond filter bank along the ω_2 dimension by π.

subbands *Dh* and *Dv*, with e.g. *Dv* representing the frequency components in the regions a_v, b_v, c_v, d_v. This procedure is called the *h-v* decomposition of *HH*.

Decimation operations in the DWT (i.e. the 1-D 2-fold downsampling along rows and columns in Fig.2.5) stretch the original frequency regions of *HH*, as shown in Fig.3.3(b). Therefore, a natural choice to separate *Dh* from *Dv* is to use a 2-channel filter bank with ideal fan supports [18], as shown in Fig.3.4(a). In fact, the fan filter bank can be implemented by first modulating the input image by π along either the ω_1 or ω_2 frequency dimension, and then passing the modulated image through a

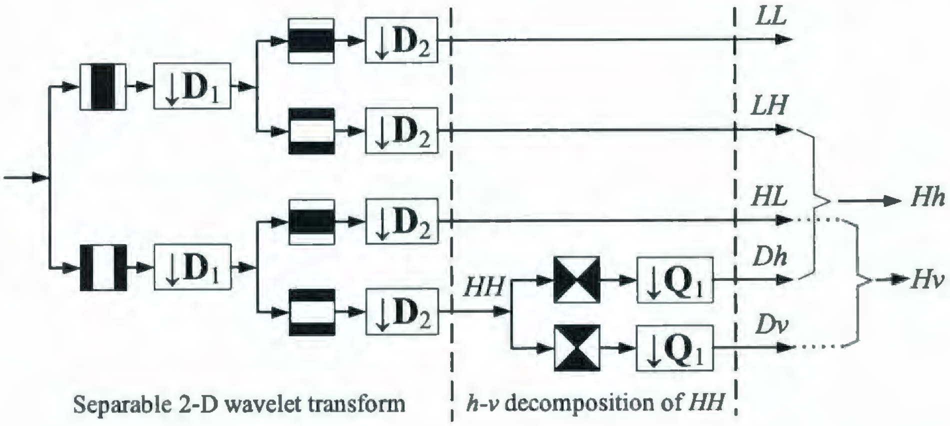


Figure 3.5: The filter bank structure (analysis section) of the separable 2-D wavelet transform and the *h-v* decomposition of *HH*.

diamond filter bank, as illustrated in Fig.3.4(b). The modulation along the ω_1 or ω_2 dimension causes the reversal of the order of two channels in the resulting fan filter bank. Therefore, the fan filter bank can be designed using the same method as the diamond filter banks, which has been investigated in Section 2.2.2. The decimation matrix for the fan filter bank is the quincunx matrix $\mathbf{Q}_1$, as shown in Eq.(2.32). Therefore, the *h-v* decomposition of *HH* is accomplished by the fan filter bank.

The filter bank structure of the discrete wavelet transform followed by the *h-v* decomposition of *HH* is presented in Fig.3.5. The separated *Dh* and *Dv* are further combined, respectively, with *LH* and *HL* to form *Hh* and *Hv*, respectively. Due to the symmetry between *Dh* (*LH*) and *Dv* (*HL*), this thesis focuses on the combination of *Dh* with *LH*.

Resampling *Dh*

In Fig.3.5, *Dh* is the output from the quincunx decimator $\mathbf{Q}_1$. As illustrated in Fig.2.16, the quincunx decimator rotates and compresses the input image in the spa-

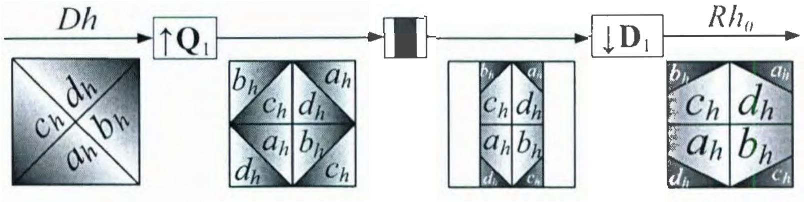
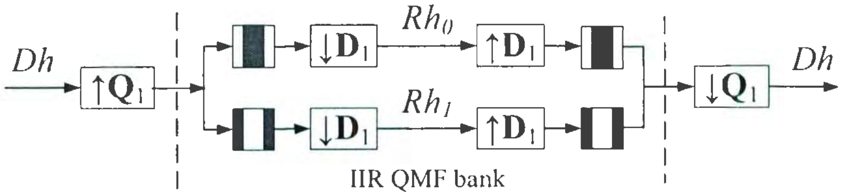


Figure 3.6: The procedure for resampling Dh, with corresponding frequency partition maps

tial domain. In fact, the spatial rotation and compression correspond to the frequency rotation and expansion in the frequency domain, e.g. the frequency components of Dh (colored by a gray gradient in Fig.3.3(b)) are rotated and expanded after the quincunx downsampling, to fill in the $2\pi \times 2\pi$ baseband as the frequency representation of Dh in Fig.3.6. Hence Dh should be resampled to inverse the rotation by the quincunx downsampling before being combined with LH.

A natural choice to inverse the quincunx rotation is a quincunx upsampling with the same sampling matrix $\mathbf{Q}_1$. Hence Dh first passes through a quincunx interpolator by $\mathbf{Q}_1$ to counteract the quincunx rotation, as illustrated in Fig.3.6. The quincunx interpolation repeats and rotates the frequency regions of Dh to fill in the baseband. The upsampled signal is then ideal lowpass filtered and downsampled along one dimension to generate a signal Rh_0. Note that the quincunx interpolator $\mathbf{Q}_1$ doubles the number of coefficients, while the diagonal decimator $\mathbf{D}_1$ reduces the coefficients by half. Therefore, the resampled Rh_0 has the same number of coefficients as Dh.

On the reconstruction side of the overall system, Dh should be exactly recovered from Rh_0. Hence the resampling procedure in Fig.3.6 is extended to a complete filter bank structure in Fig.3.7(a), in which Dh is decomposed into two signals Rh_0 and Rh_1, and should be also exactly recovered from them. If the IIR QMF bank in the interior of Fig.3.7(a) has perfect reconstruction, the overall filter bank of Fig.3.7(a)



(a) The filter bank structure of resampling

(b) Rh_0

(c) Rh_1

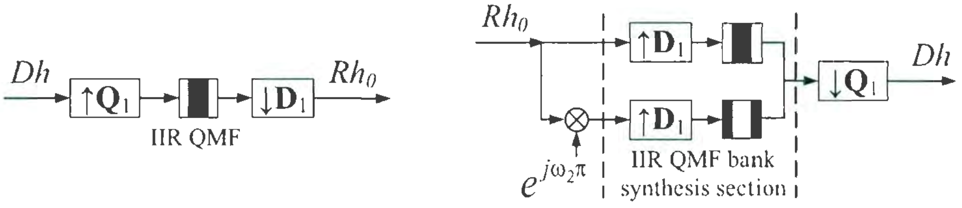
Figure 3.7: The resampling of Dh and its inverse procedure. (a) The filter bank for resampling. (b) The frequency map of Rh_0. (c) The frequency map of Rh_1.

will also have perfect reconstruction. However, the critical resampling of Dh only allows the same number of coefficients to be retained. Therefore, Rh_1 should be exactly determined by Rh_0.

Based on the frequency contents of Rh_0 and Rh_1 shown in Fig.3.7(b) and (c), Rh_1 can be obtained by frequency shifting Rh_0 by π along the ω_2 dimension, if the lowpass and highpass analysis filters in Fig.3.7(a) consist of a quadrature mirror filter (QMF) pair. The strict QMF requirement is that $H_0(\boldsymbol{\omega})$ and $H_1(\boldsymbol{\omega})$ are mirror images of each other, that is, in this case

$$H_1(\omega_1, \omega_2) = H_0(\omega_1, \omega_2 + \pi). \tag{3.1}$$

Fortunately, the QMF property in Eq.(3.1) can be achieved by the IIR QMF bank [28] with its inputs periodically replicated. Therefore, Rh_1 is discarded in the analysis section shown in Fig.3.8(a), and is reconstructed by modulating Rh_0 by π along the ω_2 dimension in the synthesis section illustrated in Fig.3.8(b). The practical filter bank structures of resampling Dh and its inverse procedure are presented in Fig.3.8,



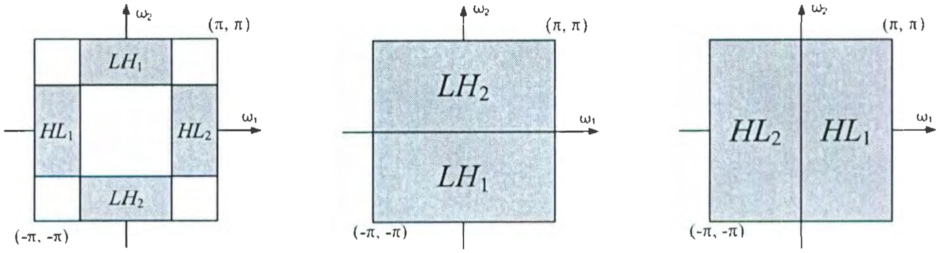
(a) Analysis section (b) Synthesis section

Figure 3.8: The filter bank structure for resampling Dh used in the practical system. (a) The resampling from Dh to Rh_0. (b) The resampling from Rh_0 to Dh.

in which the analysis section processes the resampling from Dh to Rh_0, while the synthesis section resamples Rh_0 back to Dh perfectly.

Combination between LH and Rh_0

As illustrated in Fig.3.9, decimation in the discrete wavelet transform, i.e. the 1-D 2-fold downsampling in Fig.2.5, rearranges the frequency regions of LH and HL. The decimated LH subband is then combined with the resampled component Rh_0 to produce the Hh subband in Fig.3.2(a).



(a) Input signal (b) LH subband (c) HL subband

Figure 3.9: The frequency mapping of the LH and HL subbands. (a) The spectrum of the input signal, in which the frequency regions of LH and HL subbands are colored gray. (b) The frequency contents of LH after decimation. (c) The frequency contents of HL after decimation.

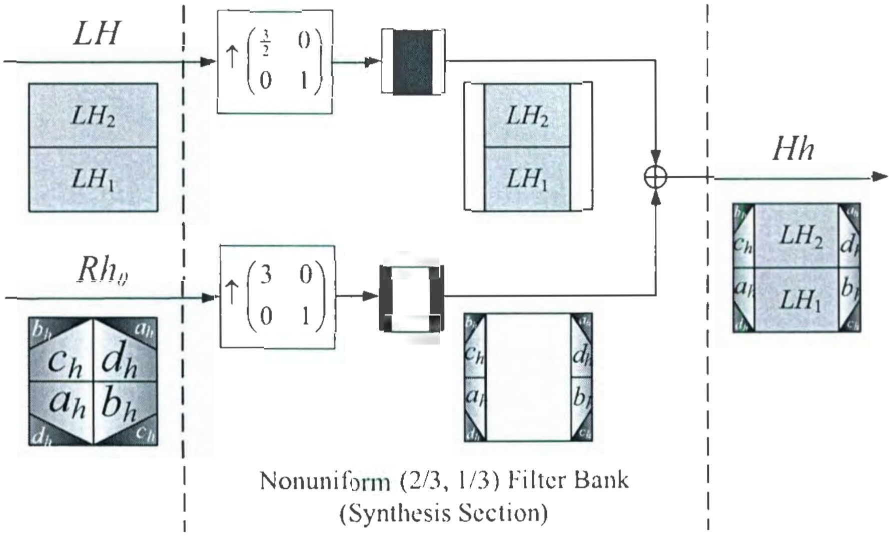


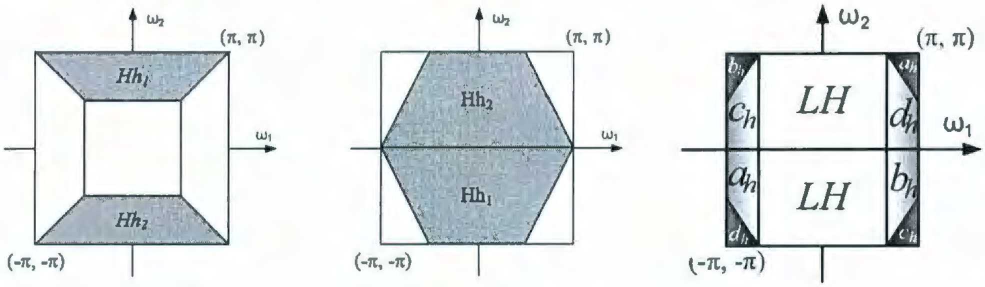
Figure 3.10: LH and Rh_0 are combined using the synthesis section of the nonuniform (2/3, 1/3) filter bank to produce the Hh subband, with their frequency partition maps. The nonuniform (2/3, 1/3) combination is illustrated in synthesis form.

The resampled Rh_0 is rectangular shaped both spatially as coefficients and in its frequency domain representation. Moreover, Rh_0 has the same number of spatial rows as the LH subband. In frequency, Rh_0 and LH are meant to be directly combined along the ω_1 dimension. The combination procedure with the corresponding spectrum maps is illustrated in Fig.3.10, in which LH and Rh_0 are fed into the lowpass and highpass channels of the synthesis section of the nonuniform (2/3, 1/3) filter bank [38], respectively, to generate a combined signal Hh.

In Fig.3.10, the nonuniform (2/3, 1/3) filter bank is illustrated in a synthesis form with diagonal upsampling matrices, one of which contains a fractional sampling factor $\frac{3}{2}$. In practice, the nonuniform (2/3, 1/3) filter bank can be implemented using the lattice structure [38, 39] in Section 2.2.3, with 3-fold upsampling and 2-fold downsampling in one dimension, say the ω_1 dimension. Fig.2.26 illustrates the analysis section of the nonuniform (2/3, 1/3) filter bank. Hence, the synthesis section in Fig.3.10 is the inverse procedure of the analysis section [39]. As shown in the output from the synthesis section of the nonuniform (2/3, 1/3) filter bank in Fig.3.10, *LH* and Rh_0 (i.e. *Dh*) share the 2-D spectrum of *Hh*.

Characteristics of *Hh*

The previous algorithm provides the procedure to generate *Hh*; the frequency content of *Hh* is analyzed next. As illustrated in Fig.3.11(a), *Hh* corresponds to two pairwise trapezoidal regions in the image spectrum of the original image. If the frequency regions of *Hh* are decimated by 2 in ω_2 dimension, then these two trapezoidal regions map to a hexagon as illustrated in Fig.3.11(b). By contrast, the 2-directional wavelet transform maximally decimates the frequency of *Hh* to fill the full baseband as in Fig.3.11(c). Now a question arises, whether rearranging the frequency components of *Hh* from a hexagon to a rectangle contaminates its representation in the spatial



(a) Input signal (b) Downsample along ω_2 (b) *Hh* subband

Figure 3.11: The frequency mapping of the *Hh* subband. (a) The frequency regions of *Hh* in the spectrum of the input image. (b) The frequency regions of *Hh* downsampled by 2 along ω_2 dimension. (c) The frequency contents of *Hh*, in which the pairwise hexagon-shaped frequency regions are stretched to a rectangular region by the 2-directional wavelet transform.

domain? The answer is negative, as is clear from the procedure developed in Section 3.1; the following offers a more theoretical explanation of why the frequencies are preserved, although rearranged.

Suppose a 2-D continuous signal $s_c(\mathbf{t}), \mathbf{t} \in \mathbb{R}^2$, has its Fourier transform $S_c(\mathbf{f})$ with the hexagon-shaped spectrum illustrated in Fig.3.12(a), where $\mathbb{R}^2$ indicates a 2-D real coordinate space. If this band-limited continuous signal is sampled under the 2-D Nyquist condition [7] by a sampling lattice $\Lambda(\mathbf{U}_H)$, to generate a discrete signal:

$$s(\mathbf{t}) = s_c(\mathbf{t}), \qquad \mathbf{t} \in \Lambda(\mathbf{U}_H), \tag{3.2}$$

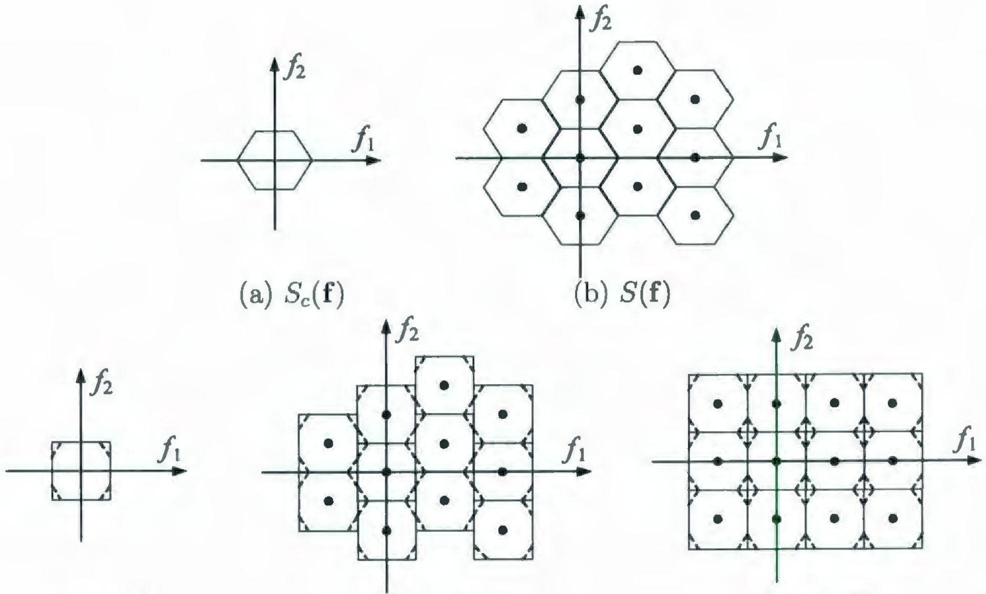
then the Fourier transform of this sampled signal $s(\mathbf{t})$ is the sum of the period replications of the hexagon-shaped spectrum of the original continuous signal $s_c(\mathbf{t})$ [35], as follows

$$S(\mathbf{f}) = \frac{1}{\det(\mathbf{U}_H)} \sum_{\mathbf{r} \in \Lambda(\mathbf{R}_H)} S_c(\mathbf{f} + \mathbf{r}), \tag{3.3}$$

where the matrix $\mathbf{R}_H$ determines the periodicity lattice in the frequency domain, which is illustrated by the dark dots in Fig.3.12(b). The spatial and frequency lattices are related by their matrices $\mathbf{U}_H$ and $\mathbf{R}_H$ [35], with the following relationship:

$$\mathbf{U}_H = \mathbf{R}_H^{-T}, \tag{3.4}$$

As illustrated in Fig.3.11(b), the *Hh* subband compose a hexagon-shaped frequency region, hence which can only be critically downsampled by the sampling matrix $\mathbf{U}_H$, with the frequency pattern $S(\mathbf{f})$ illustrated in Fig.3.12(b). However, traditional image sampling operates on a regular array of points in the spatial domain, which implies a rectangular periodicity lattice in the frequency domain, based on Eq.(3.4). It is clear that the lattice $\Lambda(\mathbf{R}_H)$ in Fig.3.12(b) is not a rectangular lattice. Hence it is difficult to maximally decimate such a hexagon-shaped frequency region of



(a) $S_c(\mathbf{f})$ (b) $S(\mathbf{f})$

(c) $R_c(\mathbf{f})$ (d) $R_1(\mathbf{f})$ (e) $R_2(\mathbf{f})$

Figure 3.12: Illustration of critical sampling *Hh*. (a) The hexagon-shaped spectrum of a continuous signal $s_c(\mathbf{t})$. (b) The spectrum of $s_c(\mathbf{t})$ critically sampled with a lattice of $\mathbf{U}_H$. (c) The rectangular region of the spectrum support of a continuous signal $r_c(\mathbf{t})$. (d) The spectrum of $r_c(\mathbf{t})$ critically sampled with a lattice of $\mathbf{U}_H$. (e) The spectrum of $r_c(\mathbf{t})$ critically sampled with a lattice generated by $\mathbf{U}_R$.

Hh in a direct way. The proposed 2-directional wavelet transform critically downsample *Hh* by moving frequencies within the hexagon, with resulting frequency pattern $R_2(\mathbf{f})$ as illustrated in Fig.3.12(e). The following analysis will show why $R_2(\mathbf{f})$ and $S(\mathbf{f})$ can represent similar contents in the spatial domain.

In the frequency view of a discrete signal, there are many possible choices for the baseband of a frequency lattice. For example, the spectrums of $S(\mathbf{f})$ in Fig.3.12(b) and $R_1(\mathbf{f})$ in Fig.3.12(d) could represent the Fourier transform of a same discrete signal with different basebands, i.e. $S(\mathbf{f}) = R_1(\mathbf{f})$. If $R_1(\mathbf{f})$ is generated by downsampling a continuous signal $R_c(\mathbf{f})$ with the frequency pattern illustrated in Fig.3.12(c)) under the 2-D Nyquist condition [7], then the original continuous signal $r_c(\mathbf{t})$, which is the inverse 2-D Fourier transform of $R_c(\mathbf{f})$, can be exactly reconstructed by lowpass-filtering the discrete signal $r_1(\mathbf{t})$.

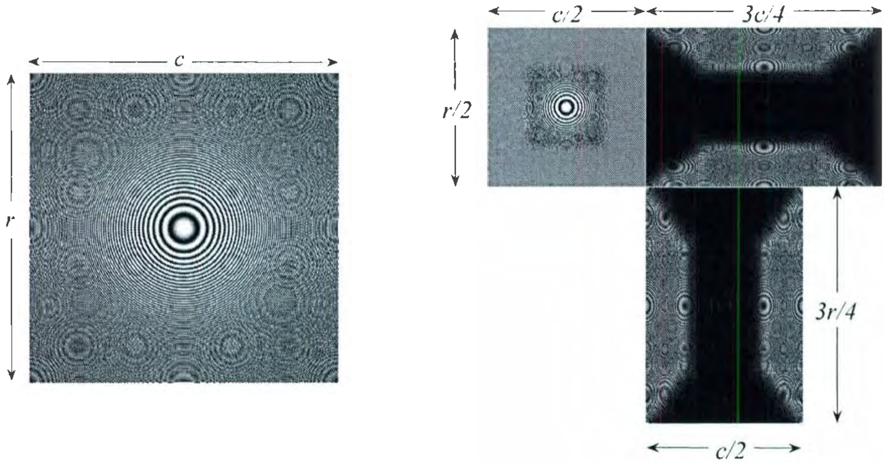
By using a rectangular sampling lattice by $\mathbf{U}_R$, $r_c(\mathbf{t})$ can also be critically sampled under the 2-D Nyquist condition to generate a discrete signal $r_2(\mathbf{t})$ with the frequency pattern in Fig.3.12(e). The continuous signal $r_c(\mathbf{t})$ can also be interpolated from $r_2(\mathbf{t})$ without errors if $r_c(\mathbf{t})$ is bandwidth limited. In other words, $r_1(\mathbf{t})$ and $r_2(\mathbf{t})$ are two critically-sampled versions of $r_c(\mathbf{t})$ with different lattices determined by $\mathbf{U}_H$ and $\mathbf{U}_R$, respectively. Most natural images contain contents which change smoothly in the spatial domain. Hence $r_1(\mathbf{t})$ and $r_2(\mathbf{t})$ present almost the same contents in the spatial domain. Recall that $s(\mathbf{t}) = r_1(\mathbf{t})$ (i.e. $S(\mathbf{f}) = R_1(\mathbf{f})$). Therefore, the proposed 2-directional wavelet transform offers an approach to critically sample Hh (with the frequency pattern $R_2(\mathbf{f})$ in Fig.3.12(e)), while preserving its contents in the spatial domain.

Results of the 2-Directional Wavelet Transform

Fig.3.13(a) shows a synthetic image "zoneplate", in which black is 0 and white is 255. The image "zoneplate" for a (512 × 512) sized image is expressed by the following 2-D discrete function:

$$f(x,y) = \frac{255}{2}\cos\left(\frac{\pi}{512}(x-256)^2 + \frac{\pi}{512}(y-256)^2\right) + \frac{255}{2}, \tag{3.5}$$

where the gray intensities have the dynamic range of $[0, 255]$ and $1 \leq x, y \leq 512$. The image "zoneplate" is good at examining the frequency partition performance of a transform, since it provides the same distribution in both the spatial and the frequency domains.



(a) Original image "zoneplate" (b) 2-directional wavelet transform

Figure 3.13: Example of the 2-directional wavelet transform on "zoneplate". (a) The original image. (b) Its 2-directional wavelet transform, in which the subimages from up to down, from left to right are LL, Hh and Hv subbands. For LL, small coefficients are colored black while large coefficients are colored white. For Hh and Hv, coefficients are shown in absolute value; the small-magnitude coefficients are colored towards black and the large-magnitude coefficients are colored towards white.

Fig.3.13(b) shows the 2-directional wavelet transform (after one iteration) on "zoneplate", in which the left-top corner is the LL subband, while the right-top and right-bottom corners are the highpass subbands Hh and Hv, respectively.

The critical sampling of the 2-directional wavelet transform is also illustrated in Fig.3.13. If the original image is of size $r \times c$, then the decomposed LL, Hh and Hv subbands are of size $\frac{r}{2} \times \frac{c}{2}$, $\frac{r}{2} \times \frac{3c}{4}$ and $\frac{3r}{4} \times \frac{c}{2}$, respectively. Hence the total number of the 2-directional wavelet coefficients is

$$\frac{rc}{4} + \frac{3rc}{8} + \frac{3rc}{8} = rc, \tag{3.6}$$

which equals the number of pixels of the original image. Note that the ratio of the area

(a) Original image "peppers" (b) 2-directional wavelet transform

Figure 3.14: Example of the 2-directional wavelet transform on a natural image. (a) The original image "peppers". (b) Its 2-directional wavelet transform, in which the small coefficients in absolute value are colored towards black.

of each highpass subband to the original image is $\frac{3}{8}$. Previous research considered the maximal decimation of such originally trapezoidal spectral areas by any integer matrix to be impossible [16]. However, the proposed 2-directional wavelet transform offers a practical structure of filter banks to rearrange trapezoid-shaped frequency regions, for example Hh, by critical sampling with integer matrices, to one rectangular shaped coefficient region.

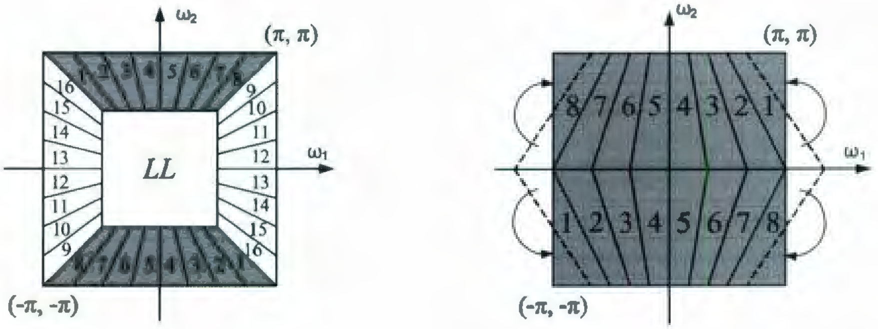
An example of the 2-directional wavelet transform on a natural image "peppers" is presented in Fig.3.14, in which LL is an approximation of the original image, while Hh and Hv, respectively, represent horizontal and vertical details around the peppers. Except for the approximation subband in the upper left corner, the other two detail subbands are scaled to make their underlying structures more visible.

3.2 Finer Directional Decomposition

The first stage of the nonredundant contourlet transform is the 2-directional wavelet transform proposed before. In order to obtain the desired frequency partition of the contourlet transform, shown in Fig.2.10, the *Hh* and *Hv* subbands should be further directionally decomposed. Due to the symmetry between *Hh* and *Hv*, this section focuses on the decomposition of *Hh*.

Fig.3.15(a) shows a possible frequency division by the NRCT, in which the high frequency scale is decomposed into 16 directional subbands. The first eight subbands, colored grey in Fig.3.15(a), are mapped to the spectrum of *Hh* in Fig.3.15(b). As illustrated in Fig.3.11(c), the hexagon-shaped frequency region of *Hh* is rearranged to a rectangle, hence frequency shifts occur within the subbands 1 and 8, as illustrated in Fig.3.15(b).

An inspection of the *Hh* spectrum in Fig.3.15(b) reveals that the separation of



(a) Input signal

(b) *Hh* subband

Figure 3.15: The frequency mapping of the directional decomposition of the *Hh* subband. (a) A possible frequency division of the input signal, in which the highpass scale is decomposed into 16 directional subbands. (b) The first eight high frequency directional subbands are mapped to the spectrum of the *Hh* subband by the proposed 2-directional wavelet transform.

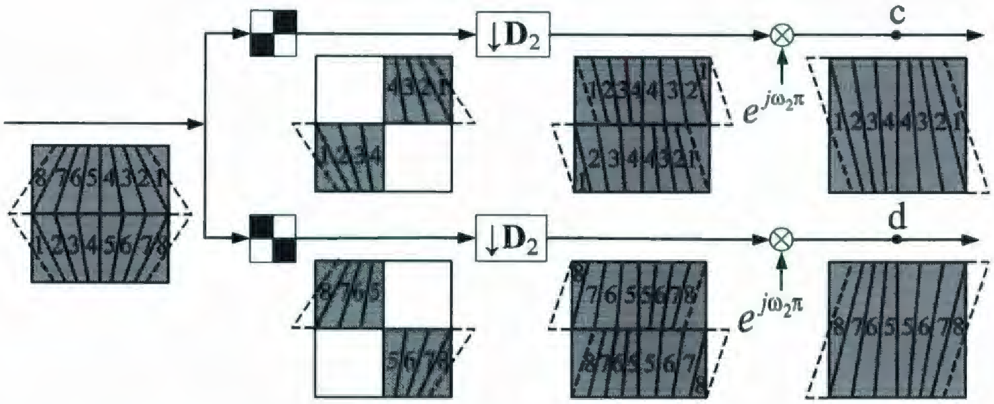


Figure 3.16: The filter bank structure for splitting *Hh* into two groups $\{1,2,3,4\}$ and $\{5,6,7,8\}$, with corresponding frequency partition maps.

subbands $\{1,2,3,4\}$ from $\{5,6,7,8\}$ can be achieved by a filter bank with quadrant support [31]. The block diagram with frequency partition maps is illustrated in Fig.3.16, in which the decimation matrix following the quadrant filters is $\mathbf{D_2}$. The decimated signals are modulated by π along the ω_2 dimension, to output coefficient images with frequency partitions more suitable to the subsequent decomposition by the filter banks of Fig.3.17. The decomposed spectra $\{1,2,3,4\}$ represent the output at "c", and $\{5,6,7,8\}$ the output at "d", as shown in Fig.3.16.

The next stage is to further separate components $\{1,2\}$ from $\{3,4\}$ in branch "c" and separate $\{5,6\}$ from $\{7,8\}$ in branch "d". By using the sheared-parallelogram (SPR) filter bank [16], the separation is achieved by the first stage of the block diagram in Fig.3.17. The frequency regions of the SPR downsampled signals, e.g. the subband $\{1,2\}$, overlap onto the periodic replicas of the $2\pi \times 2\pi$ baseband. However, these SPR-decomposed subbands are not distorted by overlapping, due to the 2π periodicity of the discrete frequency space and the characteristics of parallelograms. Recall that the 2-directional wavelet transform rearranges the frequencies inside the subbands 1

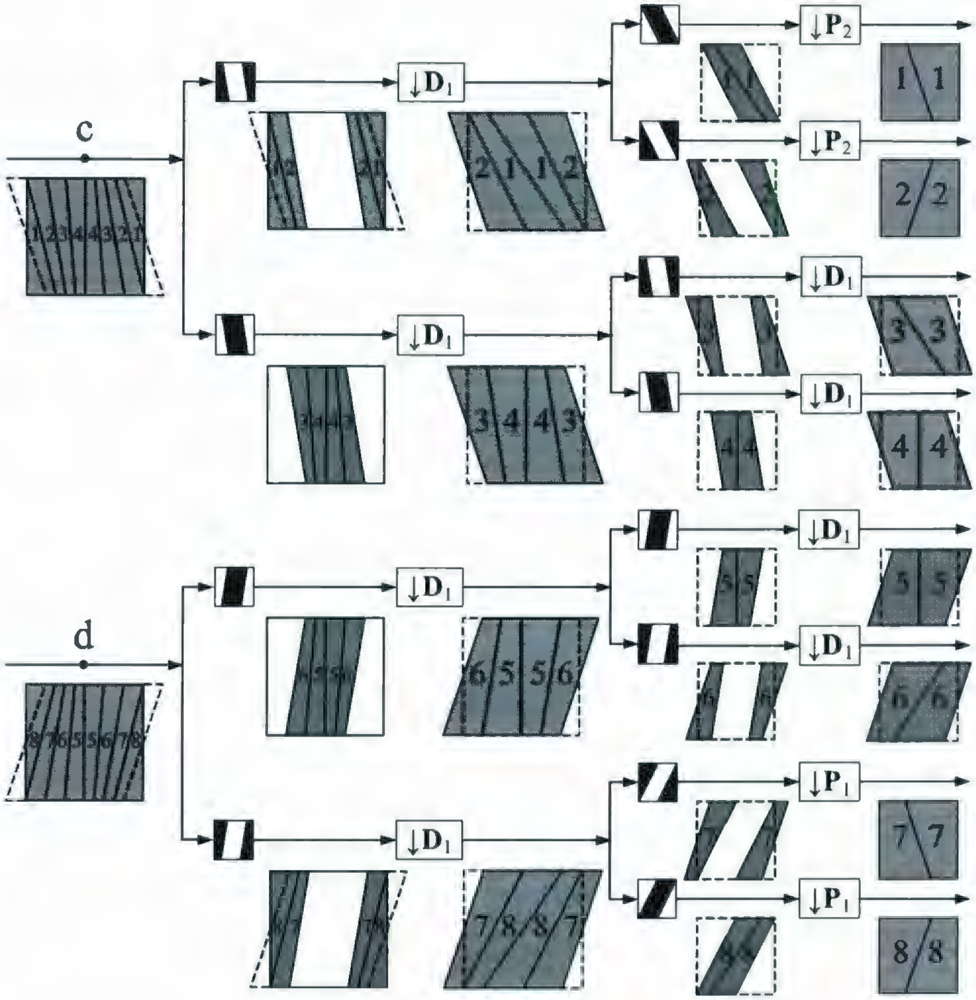


Figure 3.17: The diagram for further decomposing *Hh* into eight subbands.

and 8. It is interesting to note that these frequencies move back automatically after implementing the sheared-parallelogram filter banks.

An analysis of the output from the sheared-parallelogram decomposition (the first stage of Fig.3.17) reveals that the next stage to generate eight individual subbands is achieved by the cooperation of SPR and parallelogram filter banks, as shown in the second stage of Fig.3.17. The decimation matrices following the parallelogram filters are the parallelogram matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, as given by Eq.(2.59), with which the spectrum of subbands 1, 2, 7, and 8 are skewed to rectangles, as shown in Fig.3.17. By contrast, the SPR filters only requires diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ for row or column downsampling.

Following the decomposition illustrated above, the proposed NRCT can decompose the highpass frequency scale into 2^n directional subbands. The rule of the configuration of filter banks is presented below.

Rule of the Tree Structure Filter Bank

Note that the decomposition of *Hh* with the quadrant filter bank generates two outputs "c" and "d" with their characteristic frequency representation. Then the subsequent separation is constructed by iteratively linking six basic filter bank blocks according to the expansion rules shown in Fig.3.18, as follows:

- The type "a" output is followed by a SPR-I-HL filter bank, which generates one type "d" output and one type "e" output.
- The type "b" output is followed by a Parallel-II filter bank, which generates one type "f" output and one type "a" output.
- The type "c" output is followed by a SPR-II-HL filter bank, which generates one type "b" output and one type "c" output.

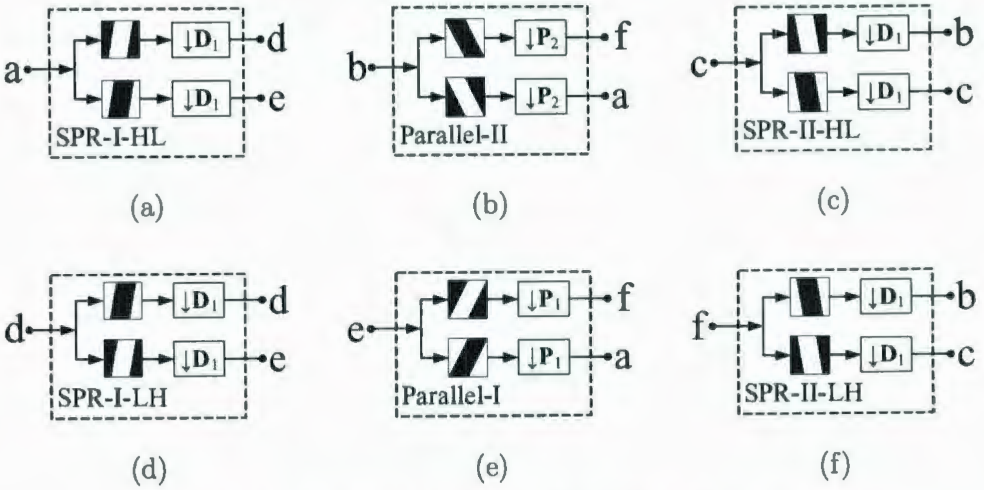


Figure 3.18: The block diagram of six basic filter banks.

- The type "d" output is followed by a SPR-I-LH filter bank, which generates one type "d" output and one type "e" output.
- The type "e" output is followed by a Parallel-I filter bank, which generates one type "f" output and one type "a" output.
- The type "f" output is followed by a SPR-II-LH filter bank, which generates one type "b" output and one type "c" output.

These six basic blocks are grouped into two classes: the sheared-parallelogram (SPR) filter bank and the parallelogram (Parallel) filter bank. In fact, Fig.3.18 only has four different cases, since the SPR-I-HL filter bank in Fig.3.18(a) and the SPR-I-LH filter bank in Fig.3.18(d) are the same, except for the order of output channels. Similarly, the SPR-II-HL filter bank in Fig.3.18(c) reverses the order of output channels of the SPR-II-LH filter bank in Fig.3.18(f).

As an example to show how the rule is applied, the tree-structure of filter banks to decompose *Hh* into eight individual components is shown in Fig.3.19. The first stage implements quincunx filter banks followed by modulators to generate one type

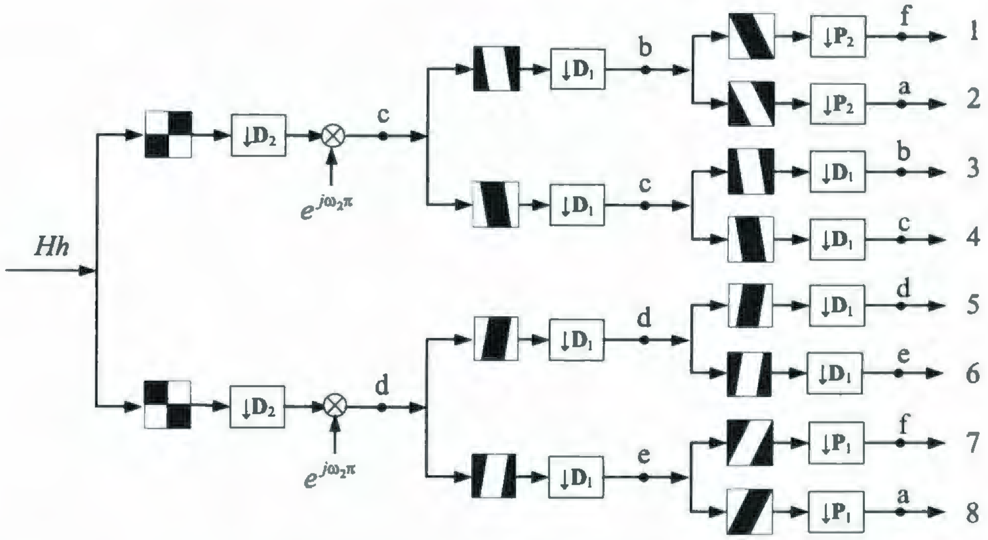


Figure 3.19: The binary tree structure of filter bank to achieve the frequency decomposition of *Hh* shown in Fig.3.15(a)

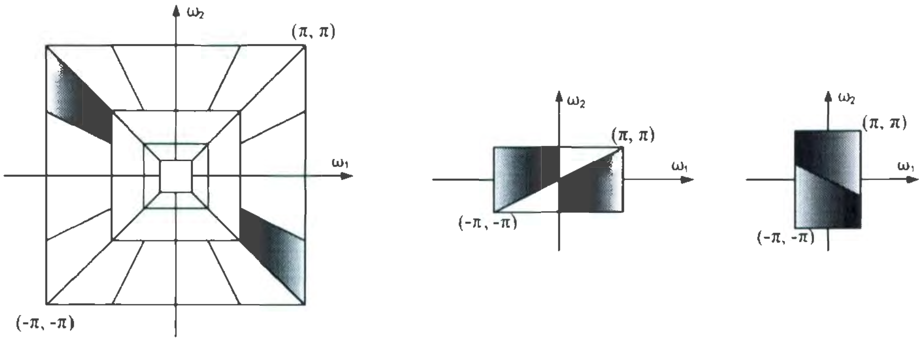
"c" and one type "d" output. Then the subsequent stages append basic two-channel filter banks according to the above rules. Moreover, the separated eight bands can be further decomposed based on the output types.

Using the techniques investigated in Section 2.2, the quadrant filter bank and the six basic filter-bank blocks can be designed to achieve critical sampling and perfect reconstruction properties. These properties are also possessed by the 2-directional wavelet transform. Therefore, the algorithm outlined above, which implements the 2-directional wavelet transform, and then directionally decomposes the higher frequency bands, also has these two valuable properties.

3.3 Multiscale and Directional Decomposition

The previous algorithm decomposes the spectrum of an input image into one lowpass subband (LL) and several directional highpass subbands. If the basic algorithm is iterated on the coarse scale, then a multiscale and multidirection image representation is achieved. The new image representation is called the nonredundant contourlet transform (NRCT).

The proposed NRCT can obtain the same frequency decomposition, illustrated in Fig.3.20(a), as the traditional contourlet transform. Moreover, the NRCT is a progressive version of the contourlet transform with critical sampling. The comparison between the traditional contourlet transform and the nonredundant contourlet trans-



(a) Frequency Partition (b) Traditional CT (c) NRCT

Figure 3.20: The comparison between the traditional and the nonredundant contourlet transforms. (a) A possible frequency partition (with four scales), which can be achieved by both transforms. A certain directional subband with pairwise trapezoid-shaped frequency regions is colored by a gray gradient. (b) The decomposed subband colored in (a) by the traditional contourlet transform (CT), whose redundancy is implied by the empty frequency regions colored white. (c) The subband colored in (a) is critically sampled by the nonredundant contourlet transform.

form is illustrated in Fig.3.20, in which a certain directional subband is decimated by both the traditional contourlet transform and the NRCT. Recall that the traditional contourlet applies the directional filter bank on the finer levels of a Laplacian pyramid. Hence, as illustrated in Fig.3.20(b), the directional subband decomposed by the traditional contourlet transform leaves $\frac{1}{4}$ of the frequency domain empty, in theory, if ideal filters are used to generate this subband. By contrast, the NRCT implements the multiscale decomposition using the 2-directional wavelet transform instead of the Laplacian pyramid. The 2-directional wavelet transform is an image representation with the critical sampling property. Therefore, the directional subband is maximally decimated by the NRCT, as illustrated in Fig.3.20(c).

An example of the NRCT on "barbara" is shown in Fig.3.21, in which small coefficients in absolute value are colored towards black while large coefficients in absolute

(a) Original image "barbara" (b) Nonredundant contourlet transform

Figure 3.21: Example of the NRCT on the image "barbara" of size 512×512. (a) The original image. (b) Its nonredundant contourlet transform, in which the coefficients with small absolute value are colored towards black while the coefficients with large absolute value are colored towards white.

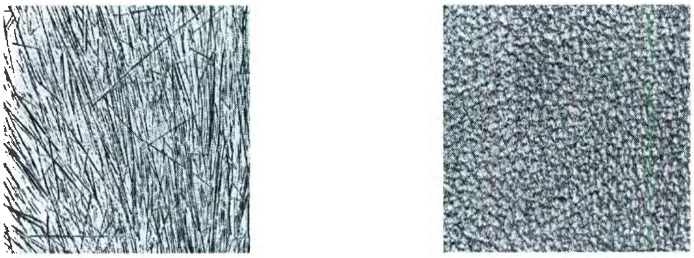
value are colored towards white. This NRCT example fits the spectral decomposition of Fig.3.20(a) (same as Fig.2.10(a)). The total number of NRCT coefficients is equal to the number of pixels of the original image. By contrast, the traditional contourlet transform on "barbara" with the same frequency partition is presented in Fig.2.10(b). It is clear that both transforms represent similar image content in the spectral domain, while the nonredundant contourlet transform saves almost $\frac{1}{4}$ coefficients of the traditional contourlet transform. Moreover, with abundant coefficients with small absolute value, the proposed NRCT has the potential for applications to image compression.

3.4 Experimental Results

3.4.1 Test Set

To evaluate the performance of the proposed NRCT, a test set of six gray-scale images as shown in Fig.3.22 is used. The names of these images are `peppers`, `barbara`, `D15`, `D92`, `fingerprint`, and `zoneplate`. These images are chosen as they manifest a range of image oriented structure. `peppers` is a piecewise smooth natural image with significant blocks of contents and `barbara` is a natural image with some fine texture on the table cloth and the barbara's pants and kerchief. Both `D15` and `D92` are texture patterns from the Brodatz collection, where `D15` is a directional texture while `D92` is a texture with less directionality. `fingerprint` is a natural image consisting of abundant oriented ridges. `zoneplate` is a synthetic image with the formula given in Eq.(3.5). They are all 8-bit gray-scale images of size 512×512.

(a) `Peppers` (b) `Barbara`



(c) `D15` (d) `D92`

(e) `Fingerprint` (f) `Zoneplate`

Figure 3.22: Test set of six gray-scale images of size 512 × 512.

3.4.2 Lossless Reconstruction Results

This section provides lossless reconstruction results as verification of the perfect reconstruction property of the three transforms, i.e. the wavelet transform, the contourlet transform, and the nonredundant contourlet transform, which are realized using non-ideal filter banks as reviewed in Section 2.2. For all the three transforms, the multiscale decomposition is implemented by the CDF "9/7-tap" filters [20] (see also Table 2.1).

Test images are represented by the three transforms, respectively; then the coefficients of these transforms pass through their corresponding inverse procedures to reconstruct the original images, respectively. The qualities of the reconstructed images are evaluated by the peak signal-to-noise ratio (PSNR). For a gray-scale 8 bits per pixel (bpp) image, the PSNR is defined as follows

$$PSNR = 20\log_{10}\frac{255}{RMSE}\ \text{dB}, \tag{3.7}$$

with root mean square error (RMSE) defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}[x(i,j)-r(i,j)]^2}, \tag{3.8}$$

where $x(i,j)$ and $r(i,j)$ are the original and reconstructed images respectively, and $N = n_1 \times n_2$ is the number of pixels of the original image.

In theory, all the three transforms can perfectly reconstruct the original images, i.e. PSNR$= +\infty$; however, this is not achievable using precision-limited computers. Table 3.1 illustrates the experimental PSNR comparison among the wavelet transform, the contourlet transform, and the NRCT, when the original images are reconstructed without loss of any coefficients (all programs run by Matlab). In Table 3.1, all the reconstructed images achieve a PSNR around 270 dB, which is a very high PSNR in practice, and as such may be considered to be perfect reconstruction.

Table 3.1: PSNR comparison for three different transforms when all coefficients are retained to reconstruct the original images (in dB)

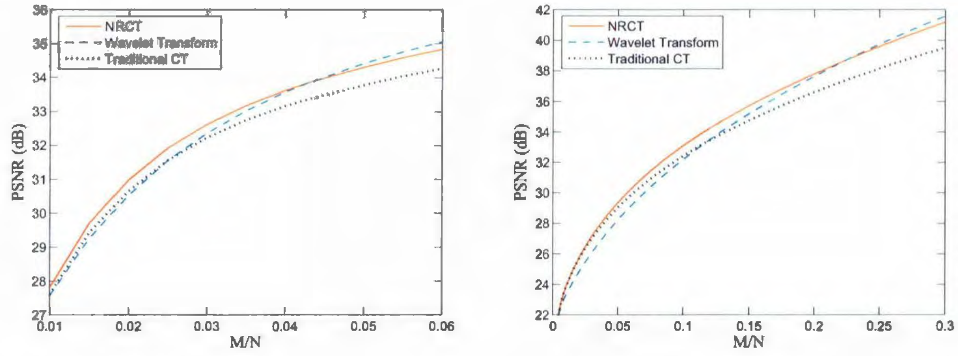
	`peppers`	`barbara`	`D15`	`D92`	`fingerprint`	`zoneplate`
DWT	271.4536	271.6328	269.5848	271.0881	270.1703	270.9097
CT	271.4939	271.6918	269.9374	271.3527	270.4156	271.5302
NRCT	271.4540	271.6345	269.5898	271.0931	270.1740	270.9230

3.4.3 Nonlinear Approximation

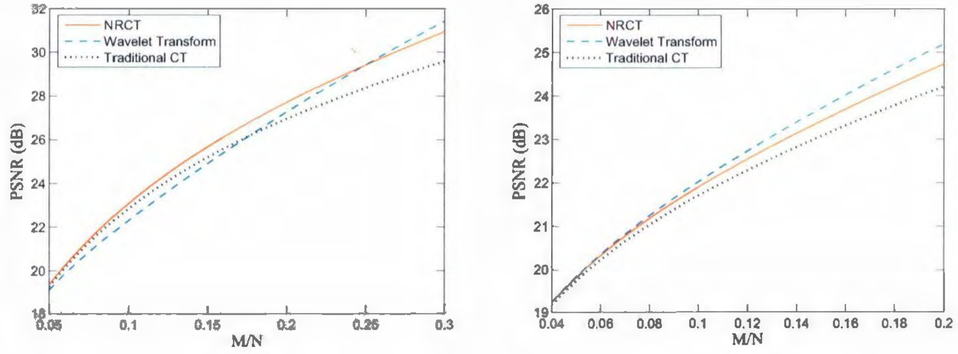
The performance of the NRCT for image approximation is compared with that of the discrete wavelet transform (DWT) and the contourlet transform (CT). In the experiments, images are decomposed into six scales. All three transformations (the wavelet transform, the traditional contourlet transform and the NRCT) share the same directional decomposition on these scales except for the finest one. In other words, both the traditional and the nonredundant contourlet transforms implement the wavelet transform in their coarser scales, each of which is decomposed into three directional subbands. The difference is that the wavelet transform splits the finest scale into three rectangular subbands, while both the traditional contourlet transform and the NRCT separate the finest scale into 32 directional subbands.

Nonlinear approximations are obtained by first applying each transform on the original image, retaining the M most-significant coefficients according to their absolute value, and then reconstructing the image from these M coefficients. If the original image consists of N pixels, M/N measures the ratio of coefficients retained. The quality of the reconstructed image is evaluated by PSNR which is defined in Eq.(3.7) and Eq.(3.8).

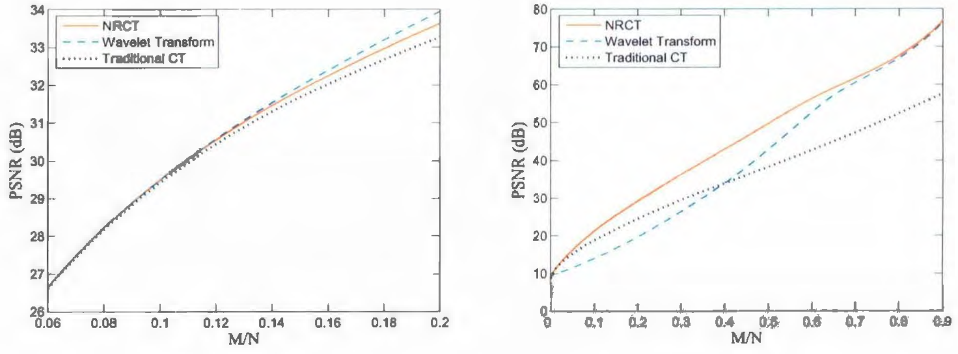
Fig.3.23 shows the PSNR comparison of the nonlinear approximation among the



(a) `Peppers` (b) `Barbara`



(c) `D15` (d) `D92`



(e) `Fingerprint` (f) `Zoneplate`

Figure 3.23: Nonlinear approximation results on the test images with six-level multiscale decomposition. Both the traditional and the nonredundant contourlet transforms decompose the finest scale into 32 directional subbands.

wavelet transform, the traditional contourlet transform and the NRCT. First compare the curves of the NRCT (solid lines) and the traditional contourlet transform (dotted lines). The proposed NRCT possesses most of the important properties of the traditional contourlet transform, including perfect reconstruction, localization [1], multiresolution, directionality, and critical sampling [8], while eliminating the redundant oversampling of the contourlet transform. Therefore, the PSNR performance of the NRCT is always better than the traditional contourlet transform, as illustrated in Fig.3.23. Moreover, both the DWT and the NRCT can exactly recover the original image for $M/N = 1$, but the CT can not.

For images `peppers`, `barbara`, `D15`, and `zoneplate`, the NRCT outperforms the wavelet transform for low ratios M/N, which are approximately lower than 0.04 for `peppers`, 0.22 for `barbara`, 0.25 for `D15`, and 0.8 for `zoneplate`. Since the NRCT is a progressive version of the traditional contourlet transform, the NRCT has a wider improvement range over the wavelet transform than the traditional contourlet transform. The NRCT presents an outstanding performance for `zoneplate` which includes abundant oriented circles with a wide frequency dynamic range. By comparing the NRCT with the wavelet transform, the peak enhancement value of PSNR is about 10 dB for `zoneplate` at $M/N = 0.3$. For the photographic natural images with finer texture, such as `barbara` and `D15`, the NRCT has an approximate 1 dB of PSNR enhancement compared with the wavelet transform at $M/N = 0.14$. Therefore, the experiments demonstrate the potential of the NRCT for image compression, especially for images with significant directional information.

As illustrated in Fig.3.23(d) and (e), the PSNR performance of the NRCT is worse than the wavelet transform for `D92` and `fingerprint`. `D92` contains insignificant

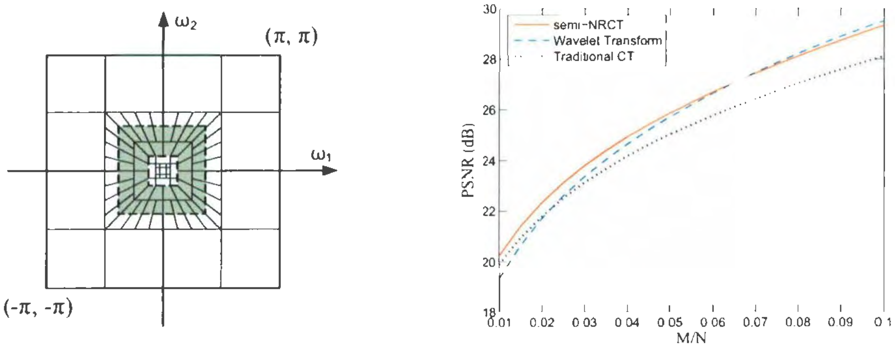
---

[1] The localization means that the basic elements in the representation should be located in both spatial and frequency domain.

directionality, hence it is not a favored image for the NRCT which has the advantage of tracking the directional information in images. On the other hand, the reason why the NRCT performs worse than the wavelet transform for `fingerprint` in Fig.3.23(e) is that the previous decomposition patten does not match the frequency characteristic of `fingerprint`.

Examination of typical fingerprint power spectra by the FBI showed that most spectral energy in fingerprint images is located in the frequency range from about $\pi/8$ to $3\pi/8$ [42], as illustrated by the gray region in Fig.3.24(a). Hence a suitable frequency partition scheme should focus on the directional information in this frequency region. Fortunately, the NRCT has directional selectivity and is compatible with the wavelet transform. Therefore, a new transform scheme with the frequency partition shown in Fig.3.24(a) is called the semi-NRCT, in which the second and the third finest scales are implemented by the NRCT, and the remaining scales from the wavelet transform.

If the `fingerprint` image is decomposed into six scales, and the second and the third finest scales are decomposed into 32 directional subbands respectively, then the corresponding PSNR performance by the semi-NRCT is illustrated as the solid curve in Fig.3.24(b). The PSNR curves of the six-level wavelet transform and the contourlet transform on the `fingerprint` image are also presented for comparison, where the contourlet transform is implemented with the frequency partition shown in Fig.3.24(a). As shown in Fig.3.24(b), the PSNR performance of the semi-NRCT is always better than the contourlet transform, and it outperforms the DWT for low ratios M/N approximately less than 0.06. The PSNR comparison among the three transforms on `fingerprint` demonstrates the potential of the semi-NRCT in fingerprint image compression, hence Chapter 5 proposes a fingerprint image compression scheme based on the semi-NRCT.



(a) The frequency support (b) The PSNR performance

Figure 3.24: Nonlinear approximation of `fingerprint` with the semi-NRCT. (a) The frequency partition by the semi-NRCT, in which the gray region represents the frequency range of $[\pi/8, 3\pi/8]$. The second and the third finest scales are decomposed into 32 directional subbands, respectively. (b) The PSNR performance comparison among the semi-NRCT, the wavelet transform and the contourlet transform.

A detailed comparison of a zoomed portion of "barbara" is presented in Fig.3.25. Both the traditional and the nonredundant contourlet transforms preserve more details in `barbara` (see the stripes on her right pant) than the wavelet transform. However, the NRCT can perfectly recover the original image for $M/N = 1$ but the contourlet transform can not. Therefore, the NRCT, as a progressive version of the contourlet transform without the redundancy of oversampling, can more efficiently represent directional structures in images than the wavelet transform.

3.5 Summary

This chapter proposes a critically sampled contourlet transform with perfect reconstruction, namely the nonredundant contourlet transform (NRCT). The NRCT is generated by first implementing the 2-directional wavelet transform, and then direc-

(a) Original Image

(b) NRCT: PSNR = 25.70 dB

(c) DWT: PSNR = 24.78 dB

(d) CT: PSNR = 25.67 dB

Figure 3.25: Detailed comparison of nonlinear approximation on "barbara". For each transform, the image is reconstructed from 5243 most-significant coefficients ($M/N = 0.02$). (a) The original image `barbara` of size 512×512. The reconstructed images by (b) the nonredundant contourlet transform, (c) the wavelet transform, and (d) the traditional contourlet transform, each with resulting PSNR.

tionally decomposing the higher frequency bands.

The NRCT is constructed by perfect reconstruction filter banks, hence it is also a perfect reconstruction system. Moreover, the NRCT offers a critically sampled representation of images by using the same number of coefficients as the number of pixels of the original image. In addition to these two important properties, e.g. perfect reconstruction and critical sampling, the NRCT possesses other useful properties including multiresolution, directionality, localization, flexible number of directions at each scale, and efficient configuration of filter banks. In other words, the proposed NRCT preserves almost all the advantages of the traditional contourlet transform while eliminating its redundancy of oversampling. Therefore, the PSNR performance of the NRCT is always better than the traditional contourlet transform in the experiments of nonlinear approximation. The NRCT obtains these advantages by paying a cost of higher computational complexity. Hence Chapter 4 will investigate its computational complexity.

Due to its directionality, the NRCT outperforms the wavelet transform for lower ratios M/N. If $M/N = 1$, both the wavelet transform and the NRCT can exactly recover the original image. Moreover, the nonredundant contourlet transform is an extension of the DWT, based on an appropriately designed filter bank structure. Therefore, the NRCT is easily compatible with the prevalent wavelet-based techniques. This chapter proposes a combination of the NRCT and the wavelet transform, called the semi-NRCT.

As mentioned before, the NRCT is suitable in tracking and efficiently coding the oriented texture of ridges of fingerprint images. Moreover, the semi-NRCT inherits the advantages of the directional selectivity of the NRCT and the lower complexity of the wavelet transform. Therefore, Chapter 5 will proposes a fingerprint image compression technique using the semi-NRCT.

Chapter 4

Computational Complexity

The previous chapter proposed a novel image representation, namely the nonredundant contourlet transform, which can decompose an image into multiresolution and multidirection subbands with critical sampling and perfect reconstruction properties. This chapter investigates the computational complexity of the NRCT, and compares it with the wavelet transform and the traditional contourlet transform.

For an individual system, such as a filter bank and the wavelet transform, the computational complexity is examined by the number of real multiplications and real additions required for each input sample, respectively, denoted by R_M and R_A. If the system has critical sampling property, which means the number of output samples is equal to the number of input samples, then the computational complexity can be simply represented as the number of real multiplications/additions per sample.

The computational complexity can also be evaluated by the number of complex multiplications C_M and the number of complex additions C_A for each input sample. As a complex multiplication takes four real multiplications and two real additions, while a complex addition needs two real additions [33], hence C_M and C_A can be easily transferred to R_M and R_A.

4.1 Computational Complexity of Convolution

Both linear and circular convolutions are basic operations in digital image processing. As illustrated in Chapter 3, the NRCT system implements circular convolution for the IIR QMF bank [28], while linear convolution for the remaining blocks. Moreover, some 2-D operations in the proposed system, such as the separable filter bank, are constructed by 1-D filters. Therefore, this section reviews the computational complexity of the 1-D and 2-D convolutions.

Time-domain convolution is more efficient than frequency-domain convolution for lower-order filters [33]. In the proposed system, all FIR filters except for that used in the IIR QMF bank have their impulse responses with fewer samples. Hence the linear convolutions with these FIR filters are implemented in the time domain. On the other hand, the IIR QMF bank replaces its IIR linear convolutions with FIR circular convolutions (see Eq.(2.28) and Eq.(2.30)); and the corresponding FIR filter $h(k)$ in Eq.(2.29) has the same length as the input signal. For this specific case when the input signal and the filter are approximately with the same length, frequency-based convolution processes more efficiently than time-based, as discussed later.

4.1.1 1-D Convolution

The convolution of two 1-D discrete functions can be directly calculated in the time domain, or implemented in the frequency domain by processing the inverse Fourier transform on the product of the Fourier transforms of these two functions.

Time-domain Linear Convolution

The direct way to implement the convolution between an input signal $x(n)$ and the impulse response of a filter $f(n)$ is to calculate the amount of overlap between $x(n)$

and a reversed version of $f(n)$. Assuming that $f(n)$ is FIR and consists of L real samples, the convolution requires at most L real multiplications and $(L-1)$ real additions for each output sample. If the additional computational complexity due to data loading, scaling and padding is insignificant and can be ignored, then the computational complexity per output sample of the time-domain linear convolution is shown as follows:

$$\begin{aligned} R_M &= L \\ R_A &= L-1. \end{aligned} \tag{4.1}$$

FFT-based 1-D Circular Convolution

Circular convolution in the time domain is equal to the point-wise multiplication in the frequency domain [33]. Therefore, the convolution can be implemented by first processing the discrete Fourier transform on the real input signal $x(n)$ (with the length of N), calculating the complex multiplication of the Fourier transform of $x(n)$ and the transfer function of the filter $f(n)$, and then processing the inverse Fourier transform on the complex product. If the Fourier transform and inverse Fourier transform are implemented by the fast Fourier transform (FFT) algorithm, then the above frequency-domain convolution is called the FFT-based convolution. In the circular convolution for the IIR QMF bank in Eq.(2.30), the size of input signals is equal to the length of the replaced FIR filters. Hence the FFT operations are implemented on resolution N.

As a typical radix-2 FFT of a complex N-point signal [1] takes $\left(\frac{N}{2}\log_2 N\right)$ complex multiplications and $(N\log_2 N)$ complex additions [33]. Moreover, the FFT of a N-point real signal can be implemented by a complex $\frac{N}{2}$-point FFT plus $\frac{N}{2}$ extra complex

---

[1] The radix-2 FFT requires that the number of input points should be 2^n. If K is not a power of 2, the input signal is simply padded with zeros to a length of $N' = 2^{\lceil \log_2 N \rceil}$.

multiplications and $2N$ extra complex additions (see Eq.(6.2.11) and (6.2.12) in [43]). This implies that the real N-point FFT takes $\frac{N}{4}(\log_2 N + 1)$ complex multiplications and $\frac{N}{2}(\log_2 N + 3)$ complex additions.

As mentioned before, the FFT-based convolution requires a real N-point FFT on the input signal, N extra complex multiplications (assuming the FFT of $f(n)$ is precalculated), and a complex N-point inverse FFT. Therefore, the FFT-based convolution requires

$$\begin{aligned} C_M &= \frac{\frac{N}{2}\log_2 N + N + \frac{N}{4}(\log_2 N + 1)}{N} = \frac{3}{4}\log_2 N + \frac{5}{4} \\ C_A &= \frac{N\log_2 N + \frac{N}{2}(\log_2 N + 3)}{N} = \frac{3}{2}\log_2 N + \frac{3}{2} \end{aligned} \tag{4.2}$$

complex multiplications and complex additions per output sample, respectively, for each sample. Since a complex multiplication takes four real multiplications and two real additions, while a complex addition requires two real additions, the number of real multiplications and real additions per output sample required by the FFT-based convolution are shown as follows:

$$\begin{aligned} R_M &= 6\log_2 N + 8 \\ R_A &= 3\log_2 N + 3. \end{aligned} \tag{4.3}$$

As shown in Eq.(4.3), the complexity of the FFT-based circular convolution involves the size of the input signal on a logarithmic scale, while the complexity of the time-domain linear covolution contains the length of the impulse response (see Eq.(4.1)). In fact, the circular convolution can be realized using the linear convolution by periodically extending the input image; and the linear convolution can be implemented with the circular convolution by padding outside of the original image with an appropriate number of zeros. Therefore, the FFT-based convolution is more efficient than the time-domain convolution when the input signal $x(n)$ is a shorter sequence, and/or the filter $f(n)$ is a higher order filter. On the other hand, if the length of the filter is much less than the length of the input signal ($L \ll N$), then the time-domain convolution is a more efficient algorithm than the FFT-based convolution.

4.1.2 2-D Linear Convolution

The separable 2-D convolution can be easily implemented by cascading 1-D convolutions in two dimensions. Hence our focus of the 2-D convolution is the nonseparable linear convolution between a digital image and the impulse response of a 2-D FIR filter. (The proposed NRCT only implements circular convolution for 1-D filtering in the IIR QMF bank, hence the 2-D circular convolution are not investigated in this section.)

Spatial Linear Convolution

Assume that a discrete image of size $N_1 \times N_2$ is convolved with a 2-D filter of size $L_1 \times L_2$. The spatial convolution is processed by moving a reversed version of the filter from point to point in the image. For each point, the sum of products of the filter coefficients with the corresponding pixels under the filter is calculated. Therefore, the spatial linear convolution requires

$$\begin{aligned} R_M &= L_1 L_2 \\ R_A &= L_1 L_2 - 1 \end{aligned} \tag{4.4}$$

real multiplications and real additions, respectively, for each sample.

FFT-based 2-D Linear Convolution

The 2-D linear convolution can be implemented in the frequency domain as well. First, the input image is transferred to its discrete Fourier transform, which is then multiplied with the transfer function of the 2-D filter. After that, the inverse Fourier transform of the product is calculated to output the filtered image. Both the Fourier transform and its inverse can be implemented by the 2-D FFT algorithm. If the size $N_1 \times N_2$ of the input image is much greater than the size $L_1 \times L_2$ of the 2-D filter, that is $N_1 \gg L_1$ and $N_2 \gg L_2$, then the 2-D FFT operations are implemented on resolution $(N_1 + L_1 - 1) \times (N_2 + L_2 - 1)$, approximated to $N_1 \times N_2$.

The 2-D FFT of a complex signal of size $N_1 \times N_2$ can be implemented by N_1 1-D complex N_2-point FFTs in one dimension, and N_2 1-D complex N_1-point FFTs in another dimension. Therefore, the complex $(N_1 \times N_2)$-point FFT takes $\left(\frac{N_1 N_2}{2} \log_2(N_1 N_2)\right)$ complex multiplications and $(N_1 N_2 \log_2(N_1 N_2))$ complex additions.

When the input image is real, the 2-D FFT can be implemented by 1-D real FFTs in one dimension followed by 1-D complex FFTs in another dimension. Assume that $N_1 \geq N_2$, a more efficient way for the real $(N_1 \times N_2)$-point FFT is N_2 1-D real N_1-point FFTs plus N_1 1-D complex N_2-point FFTs, which takes $\frac{N_1 N_2}{4}(\log_2(N_1 N_2^2) + 1)$ complex multiplications and $\frac{N_1 N_2}{2}(\log_2(N_1 N_2^2) + 3)$ complex additions. Therefore, the 2-D FFT-based linear convolution requires a real $(N_1 \times N_2)$-point FFT, $(N_1 N_2)$ extra complex multiplications, and a complex $(N_1 \times N_2)$-point FFT, which takes

$$C_M = \frac{\frac{N_1 N_2}{4}(\log_2(N_1 N_2^2) + 1) + N_1 N_2 + \frac{N_1 N_2}{2}\log_2(N_1 N_2)}{N_1 N_2} = \frac{3}{4}\log_2 N_1 + \log_2 N_2 + \frac{5}{4}$$

$$C_A = \frac{\frac{N_1 N_2}{2}(\log_2(N_1 N_2^2) + 3) + N_1 N_2 \log_2(N_1 N_2)}{N_1 N_2} = \frac{3}{2}\log_2 N_1 + 2\log_2 N_2 + \frac{3}{2} \tag{4.5}$$

complex multiplications and complex additions, respectively, for each sample.

As known that a complex multiplication requires four real multiplications and two real additions, and a complex additions needs two real additions, hence the number of real multiplications/additions per sample required by the FFT-based 2-D linear convolution is shown as follows, for $N_1 \geq N_2$:

$$\begin{aligned} R_M &= 6\log_2 N_1 + 8\log_2 N_2 + 8 \\ R_A &= 3\log_2 N_1 + 4\log_2 N_2 + 3. \end{aligned} \tag{4.6}$$

4.2 Computational Complexity of Filter Banks

The proposed nonredundant contourlet transform is constructed by a framework of filter banks. This section investigates the computational complexities of various filter banks used in the NRCT. If a general filter bank has a symmetric structure, then the synthesis section has the same computational complexity as the analysis section [44]. Therefore, the following investigation focuses on the computational complexity of the analysis sections for such case.

4.2.1 1-D CDF 9/7-tap Filter Bank

The 1-D analysis/synthesis filter bank is illustrated in Fig.2.2(a), in which the input image is lowpass and highpass filtered and decimated to generate two decomposed subbands. If the downsampling and upsampling operations do not significantly increase the computational complexity, then the main computation of this filter bank is the two lowpass and highpass filters.

The analysis/synthesis filter bank can be constructed with the CDF 9/7-tap filters [20] shown in Table 2.1. In this case, according to Eq.(4.1), the CDF 9-tap lowpass filter takes 9 real multiplications and 8 real additions for each sample, and the CDF

7-tap highpass filter requires 7 real multiplications and 6 real additions for each sample. Therefore, the two analysis filters requires 16 real multiplications and 14 real additions for each input sample. However, the decimators followed by these two filters discard half the filtered samples, which need not be calculated in the previous filtering procedure. Therefore, the analysis section of the CDF 9/7-tap filter bank takes 8 real multiplications and 7 real additions for each input sample.

4.2.2 1-D IIR Polyphase Filter Bank

The configuration of the polyphase filter bank is illustrated in Fig.2.13, in which the original signal of length N is separated into two half-length signals before passing through filters. Hence the filtering operations are implemented at lower data rates, which saves approximately 50% of the computations compared with the corresponding analysis/synthesis structure [18].

In order to achieve a quadrature mirror filter bank, the filters in the polyphase structure should be IIR filters [23]. As mentioned in Section 2.2.1, IIR filtering a signal can be implemented by the circular convolution of the signal with a corresponding FIR filter which has the same size as the original signal, as illustrated in Eq.(2.30). Therefore, IIR filtering a $\frac{N}{2}$-sample signal requires a $\frac{N}{2}$-point FFT-based convolution plus $\frac{N}{2}$ extra real additions (see Eq.(2.30)). This takes $\frac{N}{2}(6\log_2\frac{N}{2}+8)$ real multiplications and $\left(\frac{N}{2}(3\log_2\frac{N}{2}+3)+\frac{N}{2}\right)$ real additions.

The analysis section of the polyphase filter bank contains two IIR filtering operations with $\frac{N}{2}$-point signals and $2\cdot\frac{N}{2}$ extra real additions. Therefore, the analysis section of the 1-D IIR polyphase filter bank takes

$$\begin{aligned} R_M &= \frac{2\cdot\frac{N}{2}(6\log_2 N+2)}{N} = 6\log_2 N+2 \\ R_A &= \frac{2\cdot\frac{N}{2}(3\log_2 N+1)+N}{N} = 3\log_2 N+2 \end{aligned} \tag{4.7}$$

real multiplications and real additions, respectively, for each sample.

4.2.3 2-D Analysis/Synthesis SPR Filter Bank

The framework of the 2-D analysis/synthesis filter bank is illustrated in Fig.2.17, in which the input image is lowpass and highpass filtered and decimated to output two decomposed subbands. The computation of the analysis section is concentrated in the two analysis filters. If the filtering is implemented based on the FFT-based convolution, then according to Eq.(4.6), the analysis section of the filter bank requires

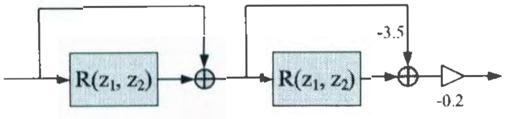
$$\begin{aligned} R_M &= 12\log_2 N_1 + 16\log_2 N_2 + 16 \\ R_A &= 6\log_2 N_1 + 8\log_2 N_2 + 6 \end{aligned} \tag{4.8}$$

real multiplications and real additions, respectively, for each sample.

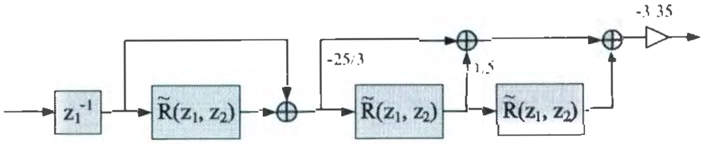
On the other hand, filtering can be processed in the spatial domain as well. Recall that the analysis filters are designed based on the transformation of variables technique [32]. If the designed filters have their passband regions approximating the sheared-parallelogram (SPR) supports shown in Fig.2.18, then the corresponding filter bank is called the SPR filter bank. For example, the analysis lowpass and highpass filters of the SPR type-I filter bank are designed as follows (see Eq.(2.51) and Eq.(2.52)):

$$\begin{aligned} H_0(z_1, z_2) &= -\,0.2\,(R(z_1, z_2) + 1)\,(R(z_1, z_2) - 3.5) \\ H_1(z_1, z_2) &= -\frac{3}{35} z_1^{-1} \left(\tilde{R}(z_1, z_2) + 1\right)\left(\tilde{R}(z_1, z_2)^2 + 1.5\tilde{R}(z_1, z_2) - \frac{25}{3}\right), \end{aligned} \tag{4.9}$$

where $\tilde{R}(z_1, z_2) = R(-z_1, z_2)$ and $R(z_1, z_2)$ is a 2-D FIR filter. These filters can be implemented with the efficient structures in Fig.4.1. The filter $H_0(z_1, z_2)$ is composed of two stages, and each stage contains a filter $R(z_1, z_2)$, a real adder and a scalar multiply. Note that the decimator following $H_0(z_1, z_2)$ in Fig.2.17 discards half the



(a) $H_0(z_1, z_2)$



(b) $H_1(z_1, z_2)$

Figure 4.1: Efficient implementation of analysis filters (a) $H_0(z_1, z_2)$ and (b) $H_1(z_1, z_2)$, where $R(z_1, z_2)$ and $\tilde{R}(z_1, z_2)$ are 2-D FIR filters. [Modified from [30]]

samples, hence the second stage of Fig.4.1(a) only needs to calculate the samples retained by the decimator which follows $H_0(z_1, z_2)$. If $R(z_1, z_2)$ is of size $K_1 \times K_2$, then filtering with $H_0(z_1, z_2)$ followed by a decimator requires $(1.5K_1K_2)$ real multiplications and $(1.5(K_1K_2 - 1) + 1.5K_1K_2)$ real additions for each sample. Similarly, filtering with $H_1(z_1, z_2)$ and downsampling need $(2.5K_1K_2)$ real multiplications and $(2.5(K_1K_2 - 1) + 2.5K_1K_2)$ real additions for each sample. Therefore, the total complexity is $(4K_1K_2)$ real multiplications per sample plus $(4K_1K_2)$ real additions per sample. The 2-D filter $R(z_1, z_2)$ is of size 15×15 in the experiments, hence the corresponding SPR filter bank takes 900 real multiplications and 900 real additions for each input sample.

When the SPR analysis/synthesis filter bank is applied to an image of normal size, say 1024×1024, the analysis section with the FFT-based convolution takes 296 real multiplications and 146 real additions for each sample, according to Eq.(4.8). However, the filter bank based on the spatial convolution requires 900 real multiplications and 900 real additions for each sample. Therefore, the SPR filter bank with the FFT-based convolution is more efficient in practice.

4.2.4 2-D Ladder Structure Filter Bank

The generic ladder structure is illustrated in Fig.2.23, in which the input image is 2-D decimated and filtered with a ladder structure of filters. The key block in the ladder strucutre is the filter $\beta(\mathbf{z})$, which is a cascade of 1-D filters $\alpha(z)$ in two dimensions. If $\alpha(z)$ is a FIR filter with length L, then filtering with $\beta(\mathbf{z})$ takes $2L$ real multiplications and $2(L-1)$ real additions for each sample.

The decimated images with $\frac{N_1N_2}{2}$ samples are fed to the ladder structure, which consists of two filters $\beta(\mathbf{z})$ and $\left(2 \cdot \frac{N_1N_2}{2}\right)$ extra real additions. Therefore, the analysis section of the ladder-structure filter bank takes

$$\begin{aligned} R_M &= \frac{2 \cdot \frac{N_1N_2}{2} \cdot 2L}{N_1N_2} = 2L \\ R_A &= \frac{2 \cdot \frac{N_1N_2}{2} \cdot 2(L-1) + 2 \cdot \frac{N_1N_2}{2}}{N_1N_2} = 2L-1 \end{aligned} \tag{4.10}$$

real multiplications and real additions, respectively, for each input sample.

In the experiments, $\alpha(z)$ is a 1-D FIR type II filter with 12 samples. Therefore, the corresponding ladder-structure filter bank takes 24 real multiplications and 23 real additions for each input sample.

4.2.5 1-D Nonuniform (2/3, 1/3) Lattice Filter Bank

The block diagram of the analysis section of nonuniform (2/3, 1/3) filter bank is illustrated in Fig.2.26, in which the lowpass and bandpass outputs from the uniform 3-channel filter bank are combined to form the nonuniform lowpass subband, while the highpass output from the uniform 3-channel filter bank directly constructs the nonuniform highpass subband. The procedure to combine the uniform lowpass and bandpass subbands involves two interpolators, one delay operation, and one multiplexer, all of which do not significantly increase the computational complexity. Therefore, the computational complexity of the nonuniform (2/3, 1/3) filter bank approximately equals that of the uniform 3-channel filter bank.

Fig.2.27 illustrates the configuration of the uniform 3-channel filter bank, in which the lossless matrices $\mathbf{K}_i$, $i = 1, \cdots, L$, constitute the main computation of the filter bank. As shown in Fig.2.28, each $\mathbf{K}_i$ is applied in a lattice structure, which contains two stages, and each stage needs four real multiplications and two real additions. Therefore, for every three input samples, the analysis section requires $(4 \cdot 2 \cdot L)$ real multiplications and $(2 \cdot 2 \cdot L)$ real additions. This implies that the analysis section requires $\frac{8L}{3}$ real multiplications and $\frac{4L}{3}$ real additions for each input sample.

In the experiments of this thesis, the nonuniform (2/3, 1/3) lattice-structure filter bank contains 5 levels of lossless matrices with parameters shown in Table 2.3. In this case, the filter bank requires $\frac{40}{3}$ real multiplications and $\frac{20}{3}$ real additions for each input sample.

4.2.6 Summary

The computational complexity of these five types of filter banks is summarized in

Table 4.1: Summary of the computational complexity of the filter banks in this thesis

Filter Banks	R_M	R_A
1-D CDF 9/7-tap	8	7
1-D IIR polyphase	$6\log_2 N + 2$	$3\log_2 N + 2$
2-D analysis/synthesis	$12\log_2 N_1 + 16\log_2 N_2 + 16$	$6\log_2 N_1 + 8\log_2 N_2 + 6$
2-D ladder structure	24	23
1-D lattice structure	$\frac{40}{3}$	$\frac{20}{3}$

Table 4.1, in which the second and the third columns, respectively, represent the number of real multiplications and real additions required for each sample. The 2-D analysis/synthesis filter based on the frequency-domain convolution is the most complex system among these five filter banks, due to its nonseparable structure of filters.

The following section will investigate the computational complexity of various image representations, which are constructed with the filter banks introduced in the present section.

4.3 Image Representations

The computational complexities of the wavelet transform, of the traditional contourlet transform and of the nonredundant contourlet transform is investigated in this section. These transforms have a major procedure in common, namely the iteration of an elementary one-level transform. If the complexity of an one-level transform is C operations/input sample, then the total complexity of a L-level transform is given by [44]

$$C_{\text{total}} = C + \frac{C}{4} + \frac{C}{4^2} + \cdots + \frac{C}{4^{L-1}} < \frac{4}{3}C. \tag{4.11}$$

Therefore, this section focuses on the computational complexity of one-level transforms, which can be simply multiplied by a factor $\frac{4}{3}$ to produce the upper bound of complexity for the corresponding transforms with irrespective levels.

4.3.1 Wavelet Transform

The wavelet transform can be constructed by a cascade of the CDF 9/7-tap filter bank in two dimensions. Therefore, the corresponding one-level wavelet transform requires

$$R_M = \frac{N_2 \cdot N_1 \cdot 8 + N_1 \cdot N_2 \cdot 8}{N_1 N_2} = 16$$

$$R_A = \frac{N_2 \cdot N_1 \cdot 7 + N_1 \cdot N_2 \cdot 7}{N_1 N_2} = 14 \tag{4.12}$$

real multiplications and real additions, respectively, for each input sample.

4.3.2 Contourlet Transform

In the one-level contourlet transform, the original image is decomposed into a downsampled lowpass scale and a highpass scale using the Laplacian pyramid [1], then the highpass scale is further separated into multiple directional subbands by the directional filter bank (DFB) [18]. Since the Laplacian pyramid and the DFB are two independent stages, their computational complexities add to produce the total complexity of the one-level contourlet transform.

The one-level structure of the Laplacian pyramid is illustrated in Fig.2.3, which consists of three stages: first, derive an approximation of the original image; then predict the original image based on the approximation; and finally calculate the difference between the original image and the prediction. Actually, the LL subband of the wavelet transform is an approximated version of the original image, hence the first stage can be implemented using the same procedure as the generation of LL in the wavelet transform. As shown in Fig.2.5(a), LL is generated by lowpass filtering and downsampling the original image in two dimension separately. If the lowpass filter is the CDF 9-tap filter, then the generation of LL takes $\left(\frac{9}{2} + \frac{9}{4} = 6.75\right)$ real multiplications and $\left(\frac{8}{2} + \frac{8}{4} = 6\right)$ real additions for each input sample. In the second stage, the approximation is upsampled and filtered with the CDF 7-tap lowpass filter to produce a predictive version of the original image. This stage requires $\left(\frac{7}{4} + \frac{7}{2} = 5.25\right)$ real multiplications and $\left(\frac{6}{4}+\frac{6}{2}=4.5\right)$ real additions for each output sample. Moreover, the last stage calculating the difference between the original image and the prediction takes 1 extra real addition for each sample. Therefore, the one-level Laplacian pyramid requires $(6.75+5.25=12)$ real multiplications and $(6+4.5+1=11.5)$ real additions for each input sample.

The difference image from the Laplacian pyramid is further decomposed using the directional filter bank. The DFB is constructed with a tree structure of the diamond filter bank and the parallelogram filter bank, both of which can be applied in the ladder structure. Therefore, the 2^k-band DFB, which is a k-stage tree of filter banks, requires $24k$ real multiplications and $23k$ real additions for each input sample.

Combining the Laplacian pyramid and the DFB, the one-level contourlet transform with 2^k directional subbands in its highpass scale requires the number of real multiplications/additions for each sample as follows:

$$\begin{aligned} R_M(k) &= 12+24k \\ R_A(k) &= 11.5+23k \end{aligned} \tag{4.13}$$

4.3.3 Nonredundant Contourlet Transform

The purpose of this section is to investigate the computational complexity of the one-level nonredundant contourlet transform. The complexity is recorded by two functions $R_M(k)$ and $R_A(k)$, which, respectively, represent the number of real multiplications and real additions per input sample required by the one-level NRCT with 2^k highpass directional subbands.

The first stage of the nonredundant contourlet transform is the 2-directional wavelet transform, which is the specific case of the NRCT with $k=1$. Assume that an original image has a size of $N_1 \times N_2$. The original image is first decomposed into *LL*, *LH*, *HL*, and *HH* subbands using the wavelet transform, then the *HH* subband of size $\frac{N_1}{2} \times \frac{N_2}{2}$ is further separated into two components *Dh* and *Dv* using the ladder-structure fan filter bank. After that, *Dh* and *Dv* pass through the quincunx interpolator and the IIR QMF polyphase filter bank to produce two resampled signals. Due to the quincunx upsampling, the images fed to the IIR polyphase filter bank are of size $\frac{N_1}{2} \times \frac{N_2}{2}$. The resampled signals are further combined, respectively, with *LH* and *HL* to generate *Hh* and *Hv*, respectively, using the synthesis section of the nonuniform (2/3, 1/3) filter bank. Note that the total number of samples of *Hh* and *Hv* is $\frac{3N_1N_2}{4}$. Therefore, the one-level 2-directional wavelet transform, which can be viewed as the one-level contourlet transform with 2^1 highpass directional subbands, has the computational complexity as follows:

$$\begin{aligned}
R_M(1) &= \frac{N_1N_2 \cdot 16 + \frac{N_1N_2}{4} \cdot 24 + \frac{N_1N_2}{4} \cdot \left[\left(6\log_2\frac{N_1}{2} + 2\right) + \left(6\log_2\frac{N_2}{2} + 2\right)\right] + \frac{3N_1N_2}{4} \cdot \frac{40}{3}}{N_1N_2} \\
&= 30 + 1.5\log_2(N_1N_2) \\
\\
R_A(1) &= \frac{N_1N_2 \cdot 14 + \frac{N_1N_2}{4} \cdot 23 + \frac{N_1N_2}{4} \cdot \left[\left(3\log_2\frac{N_1}{2} + 2\right) + \left(3\log_2\frac{N_2}{2} + 2\right)\right] + \frac{3N_1N_2}{4} \cdot \frac{20}{3}}{N_1N_2} \\
&= 24.25 + 0.75\log_2(N_1N_2).
\end{aligned} \tag{4.14}$$

In the second stage, the *Hh* and *Hv* subbands are further decomposed by the quadrant filter bank, as illustrated in Fig.3.16. The quadrant filter bank can be implemented in the ladder structure. As mentioned before, the total number of samples of *Hh* and *Hv* is $\frac{3}{4}$ of the number of pixels of the original image. Therefore, the one-level NRCT with 2^2 highpass directional subbands requires the number of real multiplications/additions per sample as follows:

$$\begin{aligned}
R_M(2) &= R_M(1) + \frac{3}{4} \cdot 24 = 48 + 1.5\log_2(N_1N_2) \\
R_A(2) &= R_A(1) + \frac{3}{4} \cdot 23 = 41.5 + 0.75\log_2(N_1N_2).
\end{aligned} \tag{4.15}$$

Table 4.2: The computational complexity of the one-level NRCT

k	$m_S(k)$	$m_P(k)$	2^k	Computational Complexity $R_M(k)$ and $R_A(k)$
1			2	$R_M(1) = 1.5\log_2 N_1 + 1.5\log_2 N_2 + 30$
				$R_A(1) = 0.75\log_2 N_1 + 0.75\log_2 N_2 + 24.25$
2			4	$R_M(2) = 1.5\log_2 N_1 + 1.5\log_2 N_2 + 48$
				$R_A(2) = 0.75\log_2 N_1 + 0.75\log_2 N_2 + 41.5$
3	4	0	8	$R_M(3) = 10.5\log_2 N_1 + 13.5\log_2 N_2 + 60$
				$R_A(3) = 5.25\log_2 N_1 + 6.75\log_2 N_2 + 46$
4	4	4	16	$R_M(4) = 15\log_2 N_1 + 19.5\log_2 N_2 + 75$
				$R_A(4) = 7.5\log_2 N_1 + 9.75\log_2 N_2 + 56.88$
5	12	4	32	$R_M(5) = 21.75\log_2 N_1 + 28.5\log_2 N_2 + 88.5$
				$R_A(5) = 10.88\log_2 N_1 + 14.25\log_2 N_2 + 64.56$
6	20	12	64	$R_M(6) = 27.38\log_2 N_1 + 36\log_2 N_2 + 102.75$
				$R_A(6) = 13.69\log_2 N_1 + 18\log_2 N_2 + 73.84$
7	44	20	128	$R_M(7) = 33.56\log_2 N_1 + 44.25\log_2 N_2 + 116.63$
				$R_A(7) = 16.78\log_2 N_1 + 22.13\log_2 N_2 + 82.33$

The following stages double the number of directional subbands using the sheared-parallelogram (SPR) and parallelogram filter banks. Let $m_S(k)$ and $m_P(k)$, respectively, denote the number of SPR filter banks and parallelogram filter banks contained in stage-k, for $k \geq 3$. Table 4.2 presents $m_S(k)$ and $m_P(k)$ for $k = 3, \cdots, 7$. For each stage, $m_S(k) + m_P(k) = 2^{k-1}$, since 2^{k-1} two-channel filter banks can output 2^k decomposed subbands. Note that the six basic filter banks in Fig.3.18, including four SPR filter banks and two parallelogram filter banks, share equal possibility to construct the stage-k when k is a larger number. Therefore, $m_S(k)$ approximately equals $\left(\frac{2}{3} \cdot 2^{k-1}\right)$ and $m_P(k)$ is almost $\left(\frac{1}{3} \cdot 2^{k-1}\right)$ for a larger number of k.

The sheared-parallelogram filter bank can be constructed with the 2-D analysis/synthesis structure, hence one stage consisting of SPR filter banks requires

$$\begin{aligned} s_M &= \frac{3}{4}\cdot(12\log_2 N_1 + 16\log_2 N_2 + 16) = 9\log_2 N_1 + 12\log_2 N_2 + 12 \\ s_A &= \frac{3}{4}\cdot(6\log_2 N_1 + 8\log_2 N_2 + 6) = 4.5\log_2 N_1 + 6\log_2 N_2 + 4.5 \end{aligned} \tag{4.16}$$

real multiplications and real additions, respectively, for each input sample. Similarly, the parallelogram filter bank can be implemented in the ladder structure, hence if one stage is composed of parallelogram filter banks, then this stage needs

$$\begin{aligned} p_M &= \frac{3}{4}\cdot 24 = 18 \\ p_A &= \frac{3}{4}\cdot 23 = 17.25 \end{aligned} \tag{4.17}$$

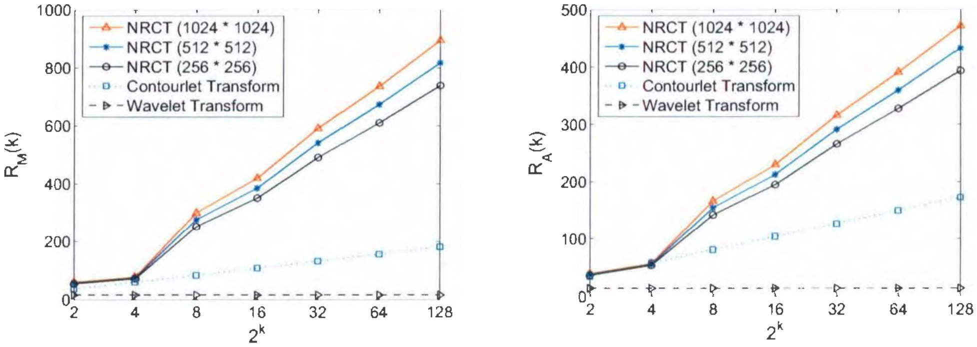
real multiplications and real additions, respectively, for each input sample. If one stage consists of SPR filter banks and parallelogram filter banks, then its computational complexity is the sum of the weighted complexities increased by these two types of filter banks, as follows:

$$\begin{aligned} r_M(k) &= \frac{m_S(k)}{2^{k-1}}\cdot s_M + \frac{m_P(k)}{2^{k-1}}\cdot p_M \\ r_A(k) &= \frac{m_S(k)}{2^{k-1}}\cdot s_A + \frac{m_P(k)}{2^{k-1}}\cdot p_A \end{aligned} \tag{4.18}$$

Therefore, the computational complexity $R_M(k)$ and $R_A(k)$ are the total complexity of the first k stages, that is

$$\begin{aligned} R_M(k) &= R_M(2) + \sum_{i=3}^{k} r_M(i) \qquad \text{for } k \geq 3 \\ R_A(k) &= R_A(2) + \sum_{i=3}^{k} r_A(i) \qquad \text{for } k \geq 3 \end{aligned} \tag{4.19}$$

Table 4.2 presents the computational complexity $R_M(k)$ and $R_A(k)$ for $k = 1, \cdots, 7$, which are calculated based on Eq.(4.16)-(4.19). When the one-level nonredundant contourlet transform is applied on images of size 256×256, 512×512 and



(a) $R_M(k)$ (b) $R_A(k)$

Figure 4.2: Comparison of the computational complexity of the wavelet transform, the contourlet transform and the NRCT. The NRCT is applied on images of size 256×256, 512×512 and 1024×1024. (a) The real multiplications per sample $R_M(k)$ as a function of 2^k, where 2^k is displayed on the scale of $\log_2(\cdot)$. (b) The real additions per sample $R_A(k)$ as a function of 2^k. In fact, the wavelet transform is not a multidirectional decomposition, hence its complexity is two constant R_M and R_A.

1024×1024, the corresponding complexity with regard to the number of highpass directional subbands 2^k is illustrated in Fig.4.2.

Assume that the computational complexity of the one-level NRCT is approximately represented as two linear functions, as follows:

$$\begin{aligned} R_M(k) &\approx b_M \cdot k + c_M \\ R_A(k) &\approx b_A \cdot k + c_A. \end{aligned} \tag{4.20}$$

As mentioned before, $m_S(k) = \frac{2}{3} \cdot 2^{k-1}$ and $m_P(k) = \frac{1}{3} \cdot 2^{k-1}$ when k is a larger number. Therefore, the slopes b_M and b_A are approximately

$$\begin{aligned} b_M &= \frac{2}{3} s_M + \frac{1}{3} p_M = 6 \log_2 N_1 + 8 \log_2 N_2 + 14 \\ b_A &= \frac{2}{3} s_A + \frac{1}{3} p_A = 3 \log_2 N_1 + 4 \log_2 N_2 + 8.75 \end{aligned} \tag{4.21}$$

According to Eq.(4.20) and Eq.(4.21), the number of real multiplications/additions per sample required by the one-level NRCT is approximately linear in k, as illustrated in Fig.4.2. Hence if the NRCT decomposes the highpass scale of an original image into $m = 2^k$ directional subbands, then the computational complexity (i.e. the number of operations per input sample) of this procedure has an upper bound $\mathcal{O}(\log_2 m)$.

These three transforms, the wavelet transform, the contourlet transform and the nonredundant contourlet transform, are simulated in Matlab. Their running times

Table 4.3: Comparison of running-times (in seconds) for Matlab implementations of the three one-level transforms, with three sizes of images

	2^k	256×256	512×512	1024×1024
Wavelet Transform		0.0642	0.2979	1.2291
Contourlet Transform	2	0.1135	0.7001	3.1026
	4	0.1301	0.8265	4.0951
	8	0.1818	0.9718	5.1325
	16	0.2172	1.0909	5.9662
	32	0.2737	1.2402	6.5740
	64	0.3646	1.4454	7.4687
	128	0.5117	1.7475	8.1259
Nonredundant Contourlet Transform (NRCT)	2	0.1463	0.7142	4.2459
	4	0.1728	0.8181	5.0841
	8	0.5626	2.1232	10.4287
	16	0.8064	2.8576	13.2778
	32	1.2076	4.0551	17.1892
	64	1.6892	5.2323	21.0139
	128	2.5055	6.9063	25.7591

for a complete procedure of decomposing an original image into subbands and reconstructing the image from the subbands are compared in Table 4.3, in which the running time is displayed in seconds. Note that the simulations are executed in Matlab (based on Intel Pentium 4 CPU 3.01 GHz and 2.00 G of RAM), and hence not optimized. All the times are for the calculation of a one-level transform, with the contourlet transform and the NRCT decomposing the highpass scale into 2^k directional subbands, for various values of k.

Three different sizes of images are examined, composing 256×256, 512×512 and 1024×1024. The pixel resolutions of test images are similar with many popular digital display formats, such as the Extended Graphics Array (XGA) [45] introduced in 1990 by IBM. The XGA is a display standard of 1024×768 pixel resolution, and is widely used in personal computers and projection systems. It is clear from Table 4.3 that, despite its relatively higher time complexity than that of the wavelet transform and the contourlet transform, the nonredundant contourlet transform can be processed in an acceptable amount of time in practice, e.g. a one-level 16-directional-subband NRCT of a 1024×1024 image only takes 13.2778 seconds in Matlab, which is not an optimized implementation.

4.3.4 Summary of Image Representation Complexities

The computational complexities of the wavelet transform, the contourlet transform, and the nonredundant contourlet transform are compared in Fig.4.2 and Table 4.3. In theory, the one-level wavelet transform requires a constant number of operations/sample, while the computational complexity of the contourlet transform and the NRCT are approximately linear in $\log_2 m$, where m is the number of directional subbands in the highpass scale. Based on Table 4.3, the time complexity of the nonredundant contourlet transform is tolerable in practice.

The traditional contourlet transform and the nonredundant contourlet transform have similar complexity for $k = 2$ or 4. However, when the highpass scale is decomposed into more than 4 directional subbands, the computational complexity of the NRCT significantly increases due to its built-in sheared-parallelogram filter bank. Therefore, the computational complexity of the NRCT can be significantly reduced if the sheared-parallelogram filter bank can be implemented in a more efficient structure.

4.4 Summary of Chapter

This chapter investigates the computational complexity of various filter banks and three image representations: the wavelet transform, the contourlet transform, and the nonredundant contourlet transform. Although the NRCT has the higher computational complexity than the wavelet transform and the contourlet transform, it has the advantage of directional selectivity compared with the wavelet transform, and it overcomes the redundant oversampling of the contourlet transform. In fact, the time complexity of the nonredundant contourlet transform is tolerable in practice. Therefore, the nonredundant contourlet transform has its potential in compressing images with significant directional information, such as fingerprint images. The following chapter will propose a fingerprint image compression scheme using the nonredundant contourlet transform.

Chapter 5

Fingerprint Image Compression

In fingerprint identification systems for criminal investigation, fingerprint images must be stored and transmitted, now usually in digital form. Efficient image compression is an essential step before storing and transmitting digital fingerprint images, due to the increasing number of fingerprint records.

The Federal Bureau of Investigation (FBI) has a fingerprint database of approximately 200 million inked fingerprint cards [42]. Each inked card consists of about 39 square inches which would be scanned at a resolution of 500 pixels per inch, with 8 bits per pixel [42], yielding approximately 10 megabytes of uncompressed digital data per card. Furthermore, the FBI receives 50,000 new cards on each work day; these records are used to check criminal backgrounds [42]. Hence the FBI requires a storage of about 2000 terabytes and a transmission per work day of about 500 gigabytes, if the digitized fingerprint images are uncompressed. The wavelet/scalar quantization (WSQ) specification [25] has been developed by the FBI for the compression of its fingerprint database. Essential characteristics of fingerprint images are the many directional curves and delicate features. However, the 2-D wavelet transform, which is used in the WSQ, combines the 1-D wavelet transform in two dimensions, and, as noted in Section 2.1.2, thus has a limitation in its ability to efficiently capture directional structure of images.

Chapter 3 has proposed a nonredundant contourlet transform (NRCT), which can efficiently track directional information in images. Since the NRCT is compatible with the wavelet transform, a new transform can be proposed by flexibly combining the NRCT and the wavelet transform. This new transform, called the semi-NRCT, has the advantages of the directional selectivity of the NRCT and the lower computation complexity of the wavelet transform. A possible frequency partition by the semi-NRCT is illustrated in Fig.5.1 (the same as Fig.3.24(a)), in which the frequency region corresponding to fingerprint ridges is separated into various directional subbands. Moreover, since the NRCT and the wavelet transform possess critical sampling and perfect reconstruction properties, so does the semi-NRCT. In fact, these properties are important in applications such as image compression. Therefore, this chapter proposes a new fingerprint image compression scheme using the semi-NRCT, and compares its performance with other transform-based compressions, including the wavelet transform and the traditional contourlet transform.

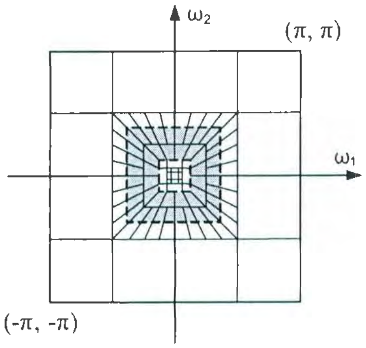


Figure 5.1: A possible frequency partition by the semi-NRCT, in which most energy of a fingerprint image is located in the frequency range of $[\pi/8, 3\pi/8]$ colored by gray.

5.1 Semi-NRCT-based Compression

Transform-based compression, for a general invertible transform, is illustrated in Fig.5.2. At the transform-based encoder, the source image is first represented by a transform, the coefficients of which are then quantized and entropy coded to generate compressed image data. The transform-based decoder reverses the above procedure by passing the compressed data through an entropy decoder, a dequantizer, and the inverse transform to reconstruct an approximation of the original image. In this diagram, the entropy encoding is a lossless data compression, while the quantizer causes quantization errors which lead to lossy image compression. However, the entropy can be substantially reduced by quantization, which results in a higher compression ratio for the overall system.

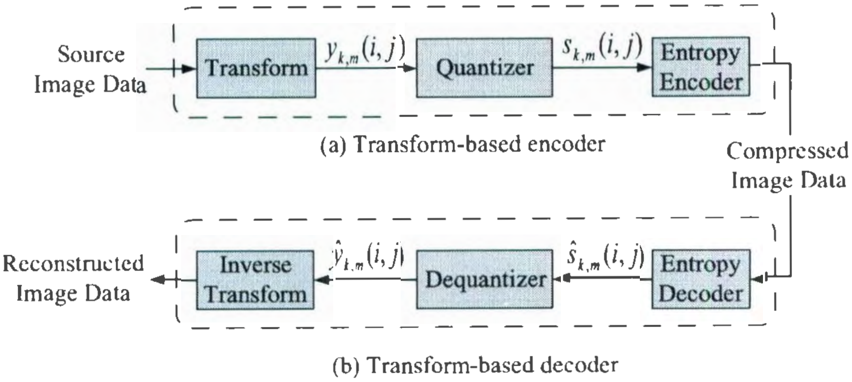


Figure 5.2: The diagram of transform-based fingerprint image compression. (a) transform-based encoder and (b) transform-based decoder.

If the transform block and its inverse in Fig.5.2 are the semi-NRCT and its inverse, respectively, and if the input/output images are of fingerprints, then the corresponding compression scheme is called the semi-NRCT-based fingerprint image compression. Let $y_{k,m}(i,j)$, where (i,j) denotes the indices of the matrix, $k = 0, 1, \cdots, L-1$, and $m = 0, 1, \cdots, D_k - 1$, denote the directional multiscale subbands in the semi-NRCT, where L is the number of scales and D_k is the number of directional subbands at scale k.

As mentioned in Section 3.4.3, most frequency components of a fingerprint image are located in the frequency region of $[\pi/8, 3\pi/8]$, which is colored by gray in Fig.5.1. Hence an efficient representation for fingerprint images should only implement directional decomposition on this frequency area. On the other hand, most energy of an image could be concentrated to fewer coefficients when the image is decomposed into more scales. However, the performance is decreased when the number of scales is over-increased. Based on numerous experimental tests, the proposed semi-NRCT fingerprint image compression decomposes an original image into six scales, and the second and the third finest scales, which occupy the gray region in Fig.5.1, are decomposed into 32 directional subbands respectively, that is $L = 5$, $D_0 = D_1 = D_2 = D_5 = 3$ and $D_3 = D_4 = 32$.

5.1.1 Uniform Scalar Quantization

The coefficient subbands of the semi-NRCT are quantized to reduce their entropies, as illustrated in Fig.5.3, which is the characteristic of the uniform scalar quantizer for an individual subband $y_{k,m}(i, j)$. The quantization characteristic is determined by two crucial parameters: the width of zero-bin $Z_{k,m}$ and the width of other nonzero bins $Q_{k,m}$ [42]. As illustrated in Fig.5.3, real coefficients lying on the zero-bin are truncated to zero by the quantizer. In addition, real coefficients falling within nonzero bins are represented by the middle point of their corresponding bins.

The quantization at the semi-NRCT-based encoder section outputs a quantizer bin indices matrix $s_{k,m}(i, j)$, which is dequantized to real numbers by the decoder

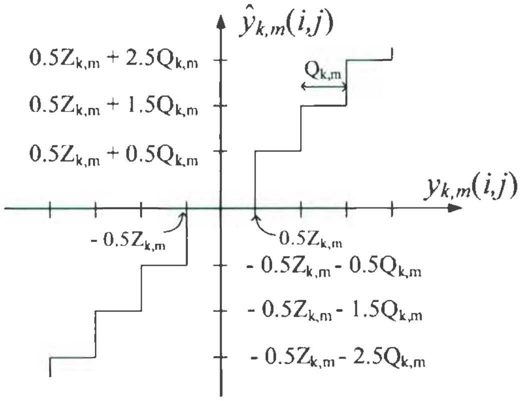


Figure 5.3: Subband quantization characteristics. The deadzone size is $Z_{k,m}$, and the step size of other bins is $Q_{k,m}$.

section. Moreover, the lossless property of the entropy encoder and decoder leads to $\hat{s}_{k,m}(i,j) = s_{k,m}(i,j)$. Therefore, the mathematical expressions of quantization and dequantization are given by [25]:

$$s_{k,m}(i,j) = \begin{cases} \left\lfloor \frac{y_{k,m}(i,j) - \frac{Z_{k,m}}{2}}{Q_{k,m}} \right\rfloor + 1, & y_{k,m}(i,j) > \frac{Z_{k,m}}{2} \\ 0, & -\frac{Z_{k,m}}{2} \leq y_{k,m}(i,j) \leq \frac{Z_{k,m}}{2} \\ \left\lceil \frac{y_{k,m}(i,j) + \frac{Z_{k,m}}{2}}{Q_{k,m}} \right\rceil - 1 & y_{k,m}(i,j) < -\frac{Z_{k,m}}{2}, \end{cases} \tag{5.1}$$

$$\hat{y}_{k,m}(i,j) = \begin{cases} \left(s_{k,m}(i,j) - 0.5\right) Q_{k,m} + \frac{Z_{k,m}}{2}, & s_{k,m}(i,j) > 0 \\ 0, & s_{k,m}(i,j) = 0 \\ \left(s_{k,m}(i,j) + 0.5\right) Q_{k,m} - \frac{Z_{k,m}}{2}, & s_{k,m}(i,j) < 0 \end{cases} \tag{5.2}$$

where $\lfloor x \rfloor$ is the largest integer not larger than x, and $\lceil x \rceil$ is the smallest integer not smaller than x.

5.1.2 Entropy

For an individual subband, the minimum number of bits per sample required to encode the quantizer bin indices matrix $s_{k,m}(i,j)$ is equal to the entropy of its coefficient value distribution [46]. Assume that the coefficients of $s_{k,m}(i,j)$ are represented as a set $A_{k,m}$. Let $f_{k,m}(u)$ represents the probability of the occurrence of value u in the matrix $s_{k,m}(i,j)$. Thus the entropy of this matrix is expressed as follows

$$H_{k,m} = -\sum_{u \in A_{k,m}} f_{k,m}(u) \log_2 f_{k,m}(u). \tag{5.3}$$

The minimum data rate $H_{k,m}$ (bits/sample) for encoding $s_{k,m}(i,j)$ can be approached in practice by techniques such as variable-length code [1, 46].

If the original image has a size of $N_1 \times N_2$ and each decomposed subband $s_{k,m}(i,j)$ contains $M_{k,m}$ samples, then the total bit rate for the semi-NRCT-based compressed image is calculated as follows:

$$H = \frac{1}{N_1 \times N_2} \sum_{k=0}^{L-1} \sum_{m=0}^{D_k-1} H_{k,m} \times M_{k,m} \quad \text{(bits/pixel)}. \tag{5.4}$$

The entropy of each subband is substantially reduced by quantization, hence the data rate of the overall semi-NRCT-based compression system is also reduced.

5.1.3 Bit Allocation

Since the minimum data rate (bits/pixel) for each subband is equal to its entropy, the total bit budget should be allocated to subbands according to their entropies. The entropy of each quantized coefficient subband is determined by the characteristic of the corresponding subband quantizer, that is the bin widths $Z_{k,m}$ and $Q_{k,m}$. Therefore, the procedure of bit allocation is the design of the zero-bin widths $Z_{k,m}$ and the nonzero bin widths $Q_{k,m}$, for $k = 0, 1, \cdots, L$, and $m = 0, 1, \cdots, D_k - 1$.

In order to enhance the transmission efficiency, the finest diagonal subband ($k = L-1, m = 2$) is discarded when the compressed data are transmitted, since this band contains little information for most fingerprint images [25]. Therefore, the bin width $Q_{L-1,2}$ is set to zero to signify that all the elements in this subband are quantized to zero. Based on the adaptive bit allocation [1] method developed by the FBI [25], the zero-bin width $Z_{k,m}$ and the nonzero bin width $Q_{k,m}$ are chosen as follows:

$$Q_{k,m} = \begin{cases} \frac{1}{\gamma} & k = 0 \\ \frac{10}{\gamma \ln\left(\sigma_{k,m}^2\right)} & k = 1, \cdots, L-2 \text{ and } k = L-1, m = 0, 1 \\ 0 & k = L-1, m = 2 \end{cases} \tag{5.5}$$

$$Z_{k,m} = 1.2 Q_{k,m}, \tag{5.6}$$

where $\sigma_{k,m}^2$ is the variance of the (k,m)-subband. $\gamma = Q_{0,m}^{-1}$ is the reciprocal of the step size of the quantizer for the coarsest scale. For other scales, $Q_{k,m}$ is proportionate to $Q_{0,m}$ with a weight dependent on the variance $\sigma_{k,m}^2$. Moreover, the zero-bin width $Z_{k,m}$ is chosen in terms of $Q_{k,m}$. As shown in Eq.(5.5) and Eq.(5.6), $Q_{k,m}$ and $Z_{k,m}$ are determined by a single parameter γ. Therefore, the performance of the overall system, including the bit rate and the quality of reconstructed images, is also controlled by the parameter γ.

---

[1] In the original adaptive bit allocation method by the FBI, the second branch of Eq.(5.5) is given by $10/\left(\gamma A_{k,m} \ln\left(\sigma_{k,m}^2\right)\right)$, where $A_{k,m}$ is an adjustment constant. In the WSQ standard, $A_{k,m}$ are chosen to be 1 for most of subbands, except for the finest scale ($k = L-1$), in which $A_{k,m}$ are set to be $1.08 \sim 1.42$. In the experiments of this thesis, the adjustment parameters $A_{k,m}$ are chosen to 1 for all subbands, hence they are ignored in Eq.(5.5).

5.2 Numerical Experiments

This section focuses on the comparison of performance of fingerprint image compression techniques based on different transforms. As mentioned before, the FBI WSQ [25] is a wavelet-based standard. Moreover, the semi-NRCT (i.e. the NRCT) is a progressive and critically sampled version of the traditional contourlet transform. Therefore, the semi-NRCT-based fingerprint compression is compared with compressions based on the wavelet transform and the contourlet transform, in all case with the decomposition of the original image into six scales. Moreover, the contourlet transform is calculated with the frequency partition as the semi-NRCT, with 32 directional subbands at the second and the third finest scales, respectively.

In the following experiments, for best results, the source image $x(m,n)$ with a dynamic range of $[0,255]$ is preprocessed to ensure that the statistical distribution of the quantized coarsest subband has a mid-point around zero, as follows

$$\tilde{x}(m,n) = x(m,n) - 128, \quad \text{for } 0 \leq m,n \leq 511. \tag{5.7}$$

At the transform-based decoder section, 128 is added to the reconstructed image to output an image with the dynamic range of about $[0,255]$.

5.2.1 Test Set

Fig.5.4 shows four test images used in the following experiments to examine the performance of the proposed semi-NRCT-based fingerprint image compression. `fingerprint` is a standard test image and is also used in the nonlinear approximation experiments (see Fig.3.22(e) in Chapter 3). `f09`, `f14` and `f23` are three example images from the NIST-4 fingerprint database [47], which contains 8-bit gray-scale fingerprint images of size 512×512. The NIST-4 images are examples of fingerprint images by scanning and digitizing FBI inked cards at a resolution of 500 pixels/inch. Each fingerprint image for NIST-4 has a white region of 32 rows at the bottom of the image. Note that fingerprint identification requires clear ridge endings and ridge bifurcations in fingerprint images [42].

(a) `Fingerprint` (b) `f09`

(c) `f14` (d) `f23`

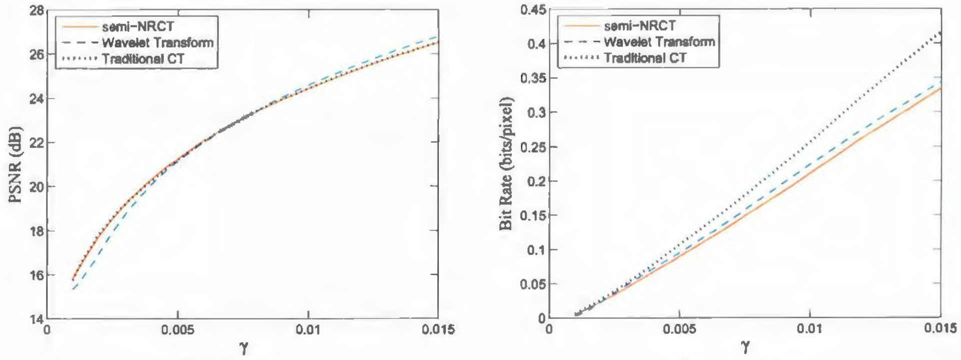
Figure 5.4: Test gray-scale test images of size 512 × 512. (a) `Fingerprint`. (b)-(d) Fingerprint image samples from NIST-4 database, each of which has 32 rows of white space at the bottom of the image.

5.2.2 Test Results

PSNR and Bit Rate as Functions of γ

As mentioned in Section 5.1.3 (see Eq.(5.5) and Eq.(5.6)), the parameter γ controls the quality of the reconstructed image and the bit rate H (bits/pixel) of the compressed data. The image quality can be numerically evaluated by its peak signal-to-noise ratio (PSNR), which is defined in Eq.(3.7) and Eq.(3.8). Therefore, the PSNR and bit rate performances of the semi-NRCT-based compression scheme can be represented as two functions of γ.

Fig.5.5(a) shows the comparison of the PSNR performance based on γ for the test image `fingerprint`. In Fig.5.5(a), the semi-NRCT has the same PSNR vs. γ performance as the traditional contourlet transform, which implies that the same quantization characteristic results in the same quantization errors for both transforms. Moreover, for lower γ, the wavelet transform has lower PSNR than the semi-NRCT and the contourlet transform in the range of $\gamma < 0.005$. Recall that γ is the reciprocal of the step size of the coarsest-scale quantizer, hence lower values of γ is equivalent



(a) PSNR vs. γ (b) Bit rate vs. γ

Figure 5.5: The curves of (a) the PSNR and (b) bit rate performances based on the parameter γ for the test image `fingerprint`.

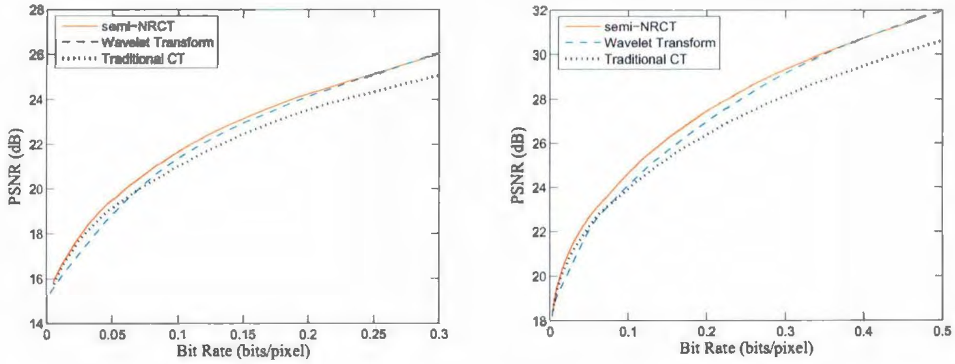
to larger bin widths of the subband quantizers.

The comparison of the bit rate performance according to γ is shown in Fig.5.5(b). With the 4/3 redundancy in its oversampling ratio, the contourlet transform significantly increases the bit rate, compared with the wavelet transform and the semi-NRCT. Moreover, the semi-NRCT can more efficiently represent the abundant delicate curves of the ridges of fingerprint images than can the wavelet transform, due to the directional selectivity of the semi-NRCT. Therefore, the semi-NRCT has the lowest bit rate among the three transforms. The patterns of PSNR and bit rate curves related to γ for the other test images are similar to those for fingerprint images, as shown in Fig.5.5, and are not shown in Fig.5.5.

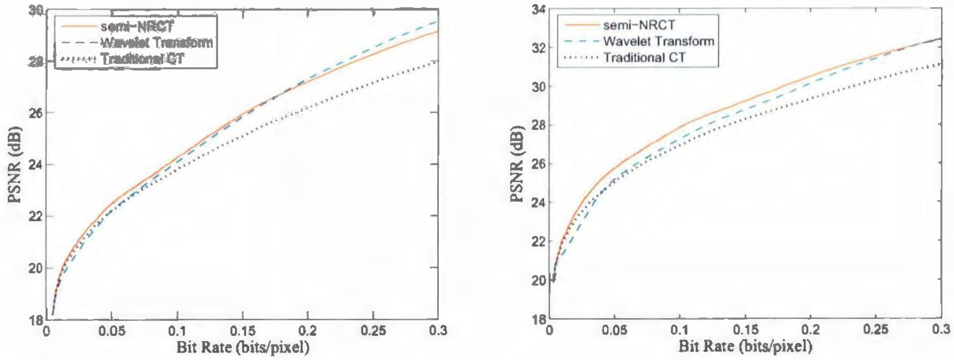
Curves of PSNR vs. Bit Rate

The direct relationship between PSNR and bit rate is more important in practice in evaluating a compression system, where PSNR indicates the quality of the reconstructed images and bit rate represents the efficiency of compression. Fig.5.6 shows the PSNR curves based on bit rate for the four test images and for the three transform-based systems. The semi-NRCT always has higher PSNR performance than the contourlet transform for a given bit rate, since as shown in Fig.5.5, the semi-NRCT requires a lower bit rate than the contourlet transform to achieve the same PSNR for a given γ. The PSNR improvement of the semi-NRCT-based compression scheme over the contourlet-based one derives from the fundamental fact that NRCT is a progressive and critical sampled version of the contourlet transform without the redundancy of oversampling.

In addition, as shown in Fig.5.6, the semi-NRCT-based fingerprint image compression outperforms the wavelet-based one for lower bit rates approximately in the range of (0, 0.2) bits/pixel. In fact, all the three transform-based compression schemes can obtain reconstructed images close to the original image for higher bit rates. Therefore, compression at lower bit rates attracts more research interests, and at such rates some compression schemes can cause significant corruption in reconstructed images.



(a) `Fingerprint` (b) `f09`

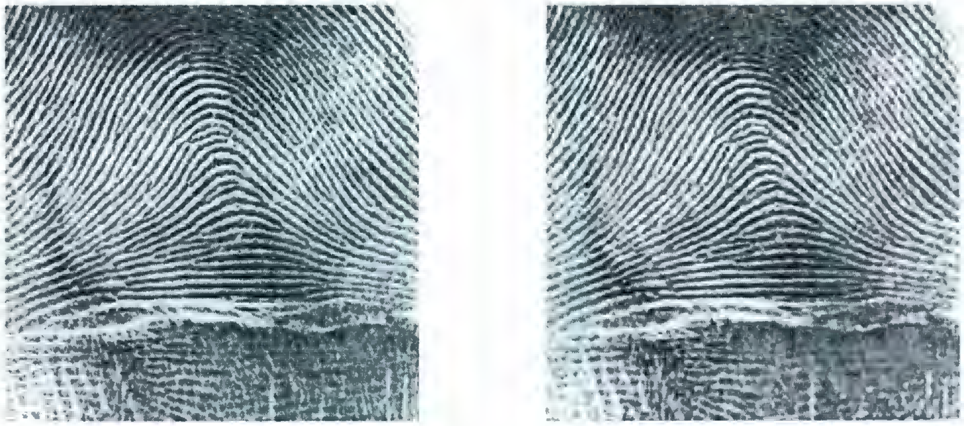


(c) `f14` (d) `f23`

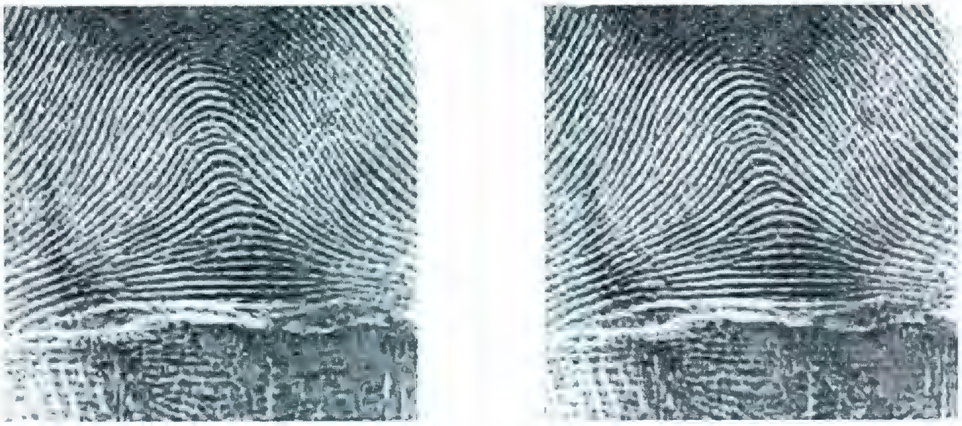
Figure 5.6: The comparison of PSNR vs. bit rate curves for the four test images and for the three transform-based compression systems.

Reconstructed Images

Detailed comparisons of the transform-based fingerprint image compressions on the test image `fingerprint` are shown in Fig.5.7. The original image `fingerprint` is illustrated in Fig.5.7(a). The wavelet transform, the traditional contourlet trans-



(a) Original image `fingerprint` (b) Semi-NRCT: PSNR = 22.32 dB



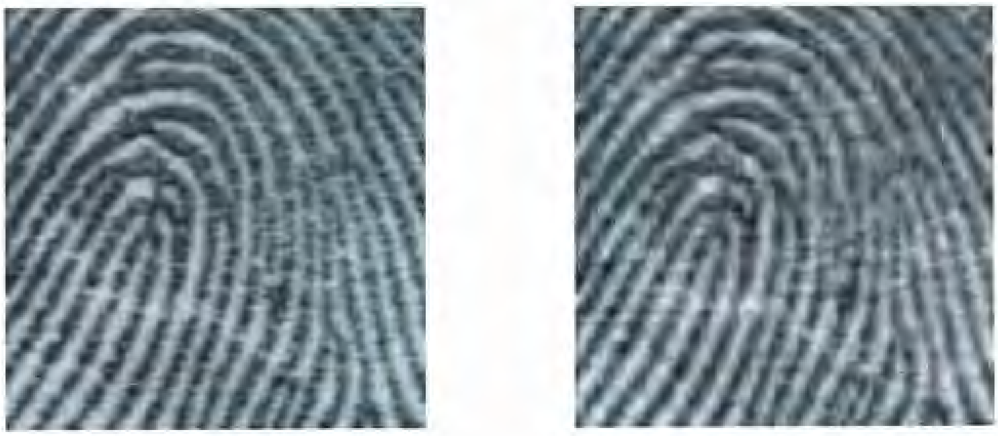
(c) DWT: PSNR = 22.04 dB (d) CT: PSNR = 21.62 dB

Figure 5.7: Comparison of reconstructed images generated from transform-based fingerprint image compressions. (a) The original image `fingerprint`, which is to be compressed at the bit rate of 0.12 bits/pixel. Reconstructed images by (b) the semi-NRCT-based, (c) the wavelet-based, and (d) the contourlet-based fingerprint image compressions, each with its resultant PSNR.

form and the semi-nonredundant contourlet transform are computed for this image, then the coefficient images are respectively quantized to achieve the data rate of 0.12 bits/pixel. This specific bit rate is chosen to obtain good NRCT results with acceptable quality for image identification. Note that the bit rates for the following experiments in Fig.5.8-Fig.5.10 are chosen differently to ensure their corresponding NRCT results with lowest acceptable qualities for fingerprint idenfitication.

The reconstructed images based on the three different transforms are shown in Fig.5.7(b)-(d). The semi-NRCT-based compressed image in Fig.5.7(b) has the highest PSNR among the three transforms, and it provides acceptable image quality with little degradation compared with the original image. With the same data rate of 0.12 bits/pixel, the wavelet-based compressed image in Fig.5.7(c) has significant corruption in some areas, such as the area at right-bottom corner. The contourlet-based compressed image in Fig.5.7(d) has a lower PSNR than the other two transforms; however, it presents more oriented details than the wavelet transform due to its directional selectivity. However, due to its improvement in sampling over the contourlet transform, the semi-NRCT-based compression shows a slight enhancement in the reconstructed image over that for the contourlet-based compression (see the lower right corners in Fig.5.7(b) and (d)).

A zoom of the center of the test image `f09` is shown in Fig.5.8(a). Fig.5.8 illustrates a detailed comparison of the zoomed portion of "f09" at the bit rate of 0.2 bits/pixel. The semi-NRCT-based reconstructed image in Fig.5.8(b) preserves more oriented details when compared with the wavelet-based and the contourlet-based reconstructed images in Fig.5.8(c) and (d). The wavelet-based compressed image in Fig.5.8(c) contains some blocking artifacts and is significantly blurred in the middle-right area. Moreover, the contourlet-based compressed image includes some artificial dots around the right-top corner of Fig.5.8(d). Due to its oversampling, the contourlet transform



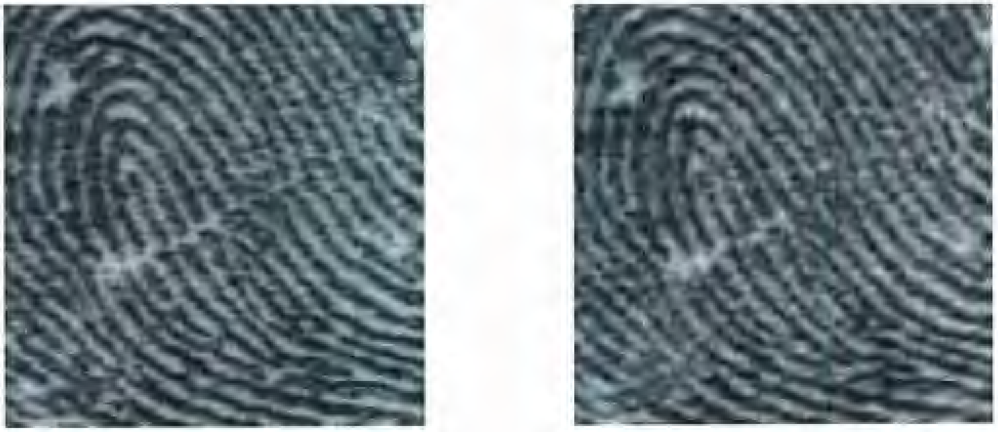
(a) Original image `f09`

(b) Semi-NRCT: PSNR = 27.45 dB

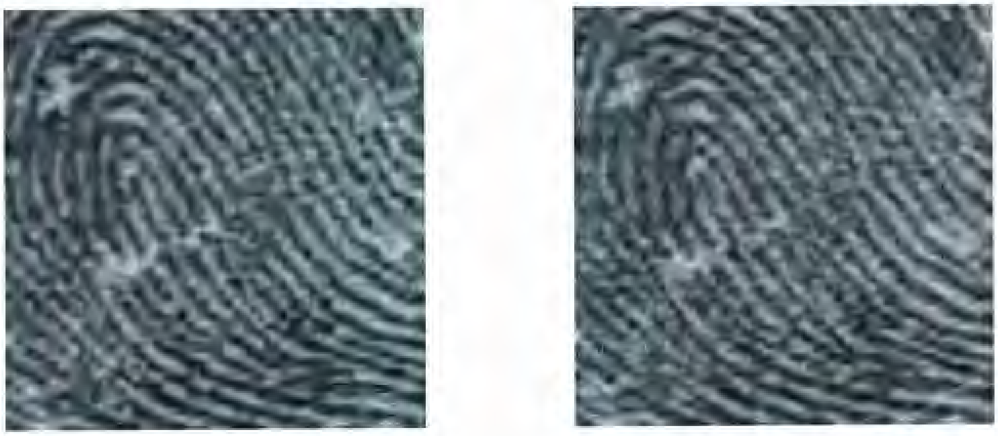
(c) DWT: PSNR = 26.96 dB

(d) CT: PSNR = 26.39 dB

Figure 5.8: Detailed comparison of the transform-based fingerprint image compressions at the bit rate of 0.2 bits/pixel. (a) A zoom-in area of the original image "f09". The reconstructed images by (b) the semi-NRCT-based compression preserves more detailed features of fingerprint ridges than that by (c) the wavelet transform, and (d) the traditional contourlet transform.



(a) Original image `f14` (b) Semi-NRCT: PSNR = 28.26 dB



(c) DWT: PSNR = 28.51 dB (d) CT: PSNR = 27.17 dB

Figure 5.9: Image compression of the test image `f14` at the data rate of 0.25 bits/pixel. (a) A zoomed area of the original image `f14`. (b) The semi-NRCT-based compressed image is close to the original one. (c) The wavelet-based compressed image has noticeable blurring, although it has the highest PSNR among the three transforms. (d) The contourlet-based compressed image has little degradation compared with the semi-NRCT-based image.

has more coefficients than the semi-NRCT and the wavelet transform. This implies that the number of bits allocated to each contourlet coefficient is lower than that of the other two transforms, if the total bit rate is fixed. Therefore, the contourlet transform has the lowest PSNR among the three transforms in Fig.5.8. However, the contourlet transform has the advantage of directional selectivity compared with the wavelet transform. Therefore, the image compressed by the contourlet transform retains more directional details than does the wavelet compressed image (compare the middle-right areas of Fig.5.8(c) and (d)).

A comparison of these transform-based compressions on a zoom of the test image `f14` is illustrated in Fig.5.9, where images are compressed at the bit rate of 0.25 bits/pixel. A zoomed portion of `f14` is shown in Fig.5.9(a). By being compared with the semi-NRCT-based compressed image in Fig.5.9(b), the wavelet-based compressed image has a slight blurring in the middle-left area in Fig.5.9(c). Note that the semi-NRCT-based reconstructed image has lower PSNR, but provides finer ridge details, than the wavelet-based one. The contourlet-based compressed image in Fig.5.9(d) has little degradation with lower PSNR compared with the semi-NRCT-based compressed image.

Fig.5.10 illustrates the comparison of the transform-based fingerprint image compressions on the test image `f23` at the data rate of 0.23 bits/pixels. A zoom-in area of `f23` is shown in Fig.5.10(a). The semi-NRCT-based reconstructed image in Fig.5.10(b) is very close to the original image, while the wavelet-based and the contourlet-based compressed images have slight blurring around the middle-left areas in Fig.5.10(c) and (d).

The experimental results in Fig.5.7 through Fig.5.10 demonstrate visually that the semi-NRCT-based image compression can preserve more directional details in reconstructed fingerprint images when compared with other two transform-based ones

(a) Original image `f23` (b) Semi-NRCT: PSNR = 31.15 dB

(c) DWT: PSNR = 30.92 dB (d) CT: PSNR = 29.92 dB

Figure 5.10: The detailed comparison of a zoom of the test image `f23` at the data rate of 0.23 bits/pixel. (a) A zoomed area of the original image `f23`. Reconstructed images by the fingerprint image compression schemes based on (b) the semi-NRCT, (c) the wavelet transform, and (d) the traditional contourlet transform.

for a given bit rate.

5.3 Summary

This chapter proposes a new fingerprint image compression scheme based on the semi-NRCT, which is a combination of the nonredundant contourlet transform and the wavelet transform. The performance of the proposed fingerprint image compression system is evaluated by the peak signal-to-noise ratio of the reconstructed image and the bit rate of the compressed data. Both PSNR curve comparisons and detailed visual examination of the reconstructed images demonstrate that the proposed semi-NRCT-based algorithm outperforms both wavelet-based and contourlet-based compression, especially at lower bit rates in the range around $(0, 0.2)$ bits/pixel.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis has developed a new multiresolution and multidirectional image representation, namely the nonredundant contourlet transform (NRCT), and has applied it to fingerprint image compression with favorable results.

In Chapter 2, various image representation techniques, including the wavelet transform and the contourlet transform, were reviewed from the literature. Due to its directional selectivity, the contourlet transform can efficiently track oriented structures in images. Unfortunately, the contourlet transform had the drawback of a 4/3 redundancy in its oversampling ratio. In order to eliminate the redundancy, Chapter 3 developed an innovative framework of filter banks to achieve the same frequency partition as the contourlet transform. The new image representation is called the nonredundant contourlet transform. The NRCT starts with the 2-directional wavelet transform which was also developed in this thesis. The NRCT is generated by first implementing the 2-directional wavelet transform, and then directionally decomposing the higher frequency bands. If the basic NRCT algorithm is iterated on the coarse scale, then a multiscale and multidirection image representation is achieved.

The proposed nonredundant contourlet transform possesses various valuable properties, including sparse expression, critical sampling, perfect reconstruction, multiresolution, and multidirection. These properties reveal that the NRCT can efficiently track the oriented structure of images. In the nonlinear approximation experiments in Chapter 3, the NRCT was shown to outperform the wavelet transform when low ratios of coefficients to original pixels are retained. Moreover, the NRCT has better peak signal-to-noise performance than the traditional contourlet transform.

Chapter 4 investigated the computational complexity of the NRCT, and compared it with the wavelet transform and the contourlet transform. The one-level NRCT, with m directional subbands in the highpass scale, has the computational complexity with an upper bound $\mathcal{O}(\log_2 m)$. The examination of running time in Matlab shows that the time complexity of the nonredundant contourlet transform is tolerable for standard images, such as of size 512×512.

Based on an appropriately designed filter bank structure, the NRCT is an extension of the discrete wavelet transform, hence it is easily compatible with the DWT. Chapter 3 developed a new transform called the semi-NRCT, which is a combination of the NRCT and the wavelet transform. The semi-NRCT has its potential to capture the directional ridge structure of fingerprint images, while saving the computational complexity of the NRCT. Therefore, Chapter 5 applied the semi-NRCT in a transform-based compression scheme. The experiments in Chapter 5 demonstrate that the semi-NRCT-based fingerprint image compression has enhanced performance, especially at lower bit rates, when compared with the wavelet-based and the contourlet-based compressions, with the same transform-based compression scheme.

This thesis contributes a progressive version of the contourlet transform with critical sampling, and hence extends its applications to image compression field.

With an acceptable computational complexity, the nonredundant contourlet transform presents significant potential for image representation and compression.

6.2 Future Work

Due to the binary tree structure of filter banks for the nonredundant contourlet transform, the size of input images is restricted in this thesis to be a power of two, such as 256 × 256, 512 × 512. However, a generic transformation should not restrict the size of input images. In fact, it should be possible to extend the proposed NRCT to allow an arbitrary input image size using a method called nonexpansive symmetric extension [48, 49].

The efficiency of the NRCT system may be improved as well. As mentioned in Chapter 4, most of the computational complexity of the NRCT comes from the sheared-parallelogram (SPR) filter bank. In this thesis, the SPR filter bank is designed based on the analysis/synthesis structure, using the transformations of variables technique. If the SPR filter bank can be implemented in a more efficient structure, then the corresponding NRCT system will be speeded up.

In Chapter 5, the proposed NRCT was applied to fingerprint image compression. This thesis focuses on the comparison of transform-based compressions schemes among different transforms. So the entropy coding following quantization is simplified to the ideal variable-length coding, which assigns a bit rate based on the entropy of quantized image data. In fact, the work could be extended to include a practical coding scheme (e.g. with source coding and channel coding) so as to compare the performance of the extended system with JPEG2000 and FBI WSQ standards.

As mentioned in previous chapters, the nonredundant contourlet transform can be easily compatible with the wavelet transform. This thesis developed a new transform called the semi-NRCT, which is a combination of the wavelet transform and the NRCT. The semi-NRCT provides an approach to partly replace the wavelet transform with the NRCT. Therefore, a combination of the semi-NRCT and the prevalent wavelet-based techniques, such as JPEG2000 and FBI WSQ, is another possibility to be explored.

References

[1] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, April 1983.

[2] P. S. Addison, *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine, and Finance.* Bristol, UK; Philadelphia: Institute of Physics Pub., 2002.

[3] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," *Proceedings of the IEEE*, vol. 84, no. 4, pp. 626–638, April 1996.

[4] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. New York: Academic, 1999.

[5] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer.* Upper Saddle River, N.J.: Prentice Hall, 1998.

[6] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, September 2001.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 2002.

[8] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, December 2005.

[9] E. J. Candès and D. L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," in *Curve and Surfaces*, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Saint-Malo, Nashville, TN: Vanderbilt University Press, 1999, pp. 105–120.

[10] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: Anisotropic multidirectional representation with separable filering," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1916–1933, July 2006.

[11] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, October 2006.

[12] R. Eslami and H. Radha, "Translation-invariant contourlet transform and its application to image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3362–3374, November 2006.

[13] J. Zhou, A. L. Cunha, and M. N. Do, "Nonsubsampled contourlet transform: Construction and application in enhancement," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, September 2005, pp. 469–472.

[14] N. Baaziz, "Adaptive watermarking schemes based on a redundant contourlet transform," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, September 2005, pp. 221–224.

[15] A. Bouzidi and N. Baaziz, "Contourlet domain feature extraction for image contect authentication," in *Proc. IEEE 8th Workshop on Multimedian Signal Processing*, October 2006, pp. 202 206.

[16] Y. Lu and M. N. Do, "CRISP-contourlets: A critically sampled directional multiresolution image representation," in *Proc. SPIE, Wavelets: Applications in Signal and Image Processing X*, M. A. Unser, A. Aldroubi, and A. F. Laine, Eds., vol. 5207, 2003, pp. 655 665.

[17] T. T. Nguyen and S. Oraintara, "A multiresolution directional filter bank for image applications," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, vol. 3, May 2004, pp. 37 40.

[18] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 882 893, April 1992.

[19] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[20] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure and Applied Math*, vol. 45, no. 5, pp. 485 560, June 1992.

[21] S. Park, M. J. T. Smith, and R. M. Mersereau, "Improved structures of maximally decimated directional filter banks for spatial image analysis," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1424–1431, November 2004.

[22] T. T. Nguyen and S. Oraintara, "Multiresolution direction filterbanks: Theory, design, and applications," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3895–3905, October 2005.

[23] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proceedings of the IEEE*, vol. 78, no. 1, pp. 56–93, January 1990.

[24] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, April 1992.

[25] Federal Bureau of Investigation (FBI), "WSQ gray-scale fingerpring image compression specification," *IAFIS-IC-0110v2*, February 1993.

[26] D. Esteban and C. Galand, "Application of quadrature mirror filters to split band voice coding schemes," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, vol. 2, May 1977, pp. 191–195.

[27] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, vol. 5, April 1980, pp. 291–294.

[28] M. J. T. Smith and S. L. Eddins, "Analysis/synthesis techniques for subband image coding," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 38, no. 8, pp. 1446–1456, August 1990.

[29] M. J. T. Smith and T. Barnwell, "A new filter bank theory for time-frequency representations," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 35, no. 3, pp. 314–327, March 1987.

[30] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, U.K.: Cambridge Univ. Press, 1988.

[31] Y. Lin and P. P. Vaidyanathan, "Theory and design of two-dimensional filter banks: A review," *Multidimensional Systems and Signal Processing*, vol. 7, no. 3-4, pp. 263–330, 1996.

[32] D. B. H. Tay and N. G. Kingsbury, "Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables," *IEEE Trans. Image Process.*, vol. 2, no. 40, pp. 466–480, October 1993.

[33] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 1999.

[34] S. Phoong, C. W. Kim, P. P. Vaidyanathan, and R. Ansari, "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Trans. Signal Process.*, vol. 43, no. 3, pp. 649–665, March 1995.

[35] E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 502–522, April 1985.

[36] M. N. Do, "Contourlet toolbox," freely available from Matlab Central (http://www.mathworks.com/matlabcentral).

[37] P. Hoang and P. P. Vaidyanathan, "Non-uniform multirate filter banks: Theory and design," in *Proc. IEEE Symp. Circuits Syst.*, vol. 1, May 1989, pp. 371–374.

[38] J. Kovačević and M. Vetterli, "Perfect reconstruction filter banks with rational sampling factors," *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2047–2066, June 1993.

[39] P. P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitray M, having the perfect-reconstruction property," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 35, no. 4, pp. 476–492, April 1987.

[40] P. P. Vaidyanathan and P. Hoang, "Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction qmf banks," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 36, no. 1, pp. 81–94, January 1988.

[41] S. Zhang and C. Moloney, "A redundancy reduction scheme for the contourlet transform," in *Proc. IEEE Newfoundland Electrical and Computer Engineering Conf. (NECEC)*, St. John's, NL, Canada, November 2006.

[42] C. M. Brislawn, "The FBI fingerprint image compression specification," in *Wavelet Image and Video Compression.* Springer-Netherlands, 2002, pp. 271–288.

[43] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 1996.

[44] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Englewood Cliffs, N.J.: Prentice Hall PTR, 1995.

[45] Y. Oike, M. Ikeda, and K. Asada, “A high-speed XGA 3-D image sensor and its applications,” in *World Automation Congress*, vol. 18, 28 June - 1 July 2004, pp. 209–216.

[46] R. B. Wells, *Applied Coding and Information Theory for Engineers.* Upper Saddle River, N.J.: Prentice Hall, 1999.

[47] National Institute of Standards and Technology (NIST), “Fingerprint Samples for NIST Special Database 4,” http://www.nist.gov/srd/nistsd4.htm.

[48] C. M. Brislawn, “Classification of nonexpansive symmetric extension transforms for multirate filter banks,” *Applied and Computational Harmonic Analysis*, vol. 3, no. 4, pp. 337–357, October 1996.

[49] R. H. Bamberger, S. L. Eddins, and V. Nuri, “Generalized symmetric extension for size-limited multirate filter banks,” *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 82–87, January 1994.