Octave-band Directional Decompositions

A Dissertation
Presented to
The Academic Faculty

by

Paul S. Hong

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2005

Octave-band Directional Decompositions

Approved by:

Mark J. T. Smith, Co-advisor
School of ECE
Purdue University

Faramarz Fekri
School of ECE
Georgia Institute of Technology

Russell M. Mersereau, Co-advisor
School of ECE
Georgia Institute of Technology

Paul J. Benkeser
School of ECE
Georgia Institute of Technology

Joel Jackson
School of ECE
Georgia Institute of Technology

Christopher E. Heil
School of Mathematics
Georgia Institute of Technology

Date Approved: 13 July 2005

To my parents,

Dr. Doopyo Alexius Hong and Younsoon Caritas Lee Hong,

for their unending support,

and to my sisters,

the best family anyone could hope to have.

ACKNOWLEDGEMENTS

I'd like to begin by thanking Dr. Mark J. T. Smith who has helped me from the beginning. I realize that it was not easy for him to take on another student as he was in the President's office when I started, but I am inordinately pleased he did as I have found his guidance, advice, and patience indispensable through the years. I feel genuinely lucky to have had the opportunity to work with him.

Next, I want to thank Dr. Russell M. Mersereau who agreed to co-advise me after Dr. Smith left for Purdue University. It was necessary for me to have a local advisor in Atlanta, and Dr. Mersereau really went above and beyond the call of duty whenever he helped me with his insight and straightforwardness.

I would also like to thank my thesis committee. Dr. Joel Jackson and Dr. Paul Benkeser were particularly helpful with my ultrasound questions. And I'd like to extend my thanks to Dr. Faramarz Fekri and Dr. Christopher Heil as well. Additionally, Dr. Lance M. Kaplan (who was on my proposal committee) helped with our regular meetings and had many welcome suggestions with respect to the hyperspectral data in particular.

There are also several people I need to thank because without them, I literally wouldn't have graduated. Christy Ellis was constantly looking out for me, and for that, I am extremely grateful; Kay Gilstrap made sure I didn't starve (on more than one occasion) by keeping me paid; and Charlotte Doughty got me the best cubicle on the floor. Beyond all those things, everyone has been very supporting and friendly, and I appreciate everyone that much more for it.

CSIP is full of helpful and friendly people, but there are a few who really stood out. I'd like to thank Matt Lee for his friendship through my entire time at GCATT and for making lab the funnest place on earth (except when we went to lunch, in which case, lunch was the funnest place on earth). I'd also like to thank Kevin Chan, a fellow Michigander to the bone, who always provided a needed break from research just down the hall. Robert Morris kept me sane at work and home; Tami Randolph is just too cool for words; and Jungwon Kang was always kind enough to laugh at

my jokes. Jose, Clyde, and Dalong were all great labmates as well. There are a ton more CSIPers who have helped me out, thanks to you all.

Outside of CSIP, my family has been extremely supportive. My father Doopyo Hong never pressured me and always encouraged me. Cathy, Benita, Elsie, and Mary all helped me along with a fun conversation or some sympathy, whichever I needed more. They always gave me strength, and I never had to ask for it once.

Thanks also to Elizabeth Brock, the rock on whom I trusted. If I ever needed consolation or just someone to calm me down, Liz was there and was invaluable. Max and Leigh Sprauer were always so supportive all the way from California, and the Guiness Stouts always had an e-mail to brighten my day (especially Kevin Constantino who has a rapier wit!). Tom Drewes and Anne McLaughlin introduced me to rock climbing and were always ready to give me a needed break from the research grind. Thanks also to Hae Eun Min who helped me from going crazy, especially the week before. I have been so lucky to have such great friends, I wish I could thank everyone by name!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis introduces the octave-band directional filter bank (OBDFB) and discusses its use in several applications. The OBDFB is a maximally decimated, perfectly reconstructing filter bank that is able to discriminate with respect to both radial and angular frequency. Previous filter banks have been able to capture either radial or angular frequency content, but not both in a maximally decimated context. Other properties of the OBDFB are its use of separable polyphase filtering and its ability to use either FIR or IIR filters.

Initially, a brief review of multi-dimensional, multi-rate systems is presented. In particular, the derivation of directional filter bank (DFB) is presented as background for the OBDFB. The decimated subbands of the DFB still contain radially low frequency information, but that information is scrambled in frequency space due to modulation and decimation. By applying another filter bank to the subbands of the DFB, radially high and low frequency content can be separated.

With respect to applications, multi-channel texture segmentation is discussed first. Different filter bank configurations are tested and compared to the OBDFB with respect to segmentation accuracy including Gabor filters in the undecimated domain. Next, a novel texture augmentation algorithm for remote sensing is introduced to increase classification accuracy of hyperpixels in hyperspectral data. For the final application, ultrasound despeckling results are provided and compared between the OBDFB, a traditional wavelet decomposition, and adaptive weighted median filtering.

Finally, a new, non-uniform filter bank is proposed and designed to address many issues pertaining to the OBDFB. Although it does not have all of the properties of the OBDFB, it does solve the problem of the DC component being distributed among directional subbands unevenly.

CHAPTER I

INTRODUCTION

For many years, there has been interest in 2-D signal decompositions with refinable angular resolution, stimulated by established and emerging applications such as object and character analysis, texture classification, denoising, segmentation, enhancement, and velocity filtering [30, 60, 86, 91]. Such decompositions are also of interest in applications that attempt to exploit directional properties of the human visual system (HVS) [92, 93].

In parallel, there has been sustained interest in traditional decompositions that separate the spectrum into rectangular short and medium bands from 0 to π such as the discrete wavelet transform (DWT). These decompositions typically result from applying 1-D techniques separably in higher dimensions. There are numerous applications that take advantage of low frequency, mid-band frequency, and high frequency partitions, the most notable of which are image compression and video coding.

1.1 Motivation

More recently, there has been interest in decompositions with both radial frequency partitioning as in the conventional tree-structured filter banks and angular partitioning, each ideally with arbitrary resolution [4, 21]. Examples of both types of partitioning may be seen in Figure 1.1.

These two objectives can be met simultaneously rather easily using undecimated filter banks, but the outcome is a highly redundant and data inefficient representation.

A number of efforts have been made to achieve high angular and radial resolution with minimal redundancy, but virtually all thusfar have been limited with respect to one or more of these objectives [12, 21, 29]. That is, they will have either limited angular resolution, limited radial resolution, and/or various degrees of redundancy. For some applications, the additional condition of near shift invariance is considered, which further constrains the design problem.

In this thesis, a family of fully non-redundant octave-band directional filter banks is introduced

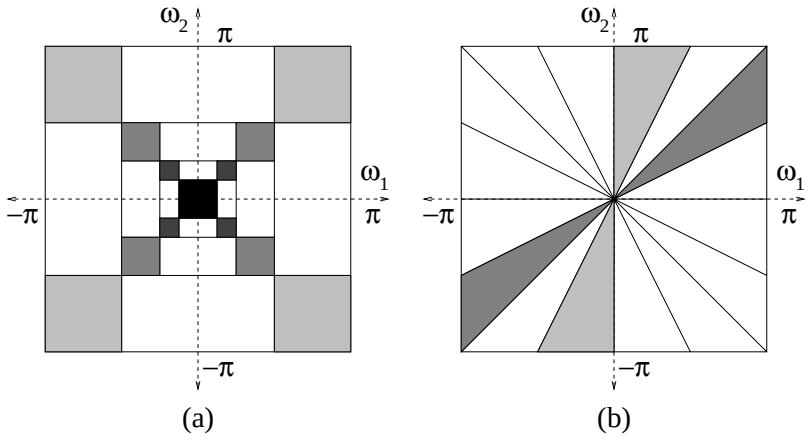


Figure 1.1: Examples of (a) a 3-stage, octave-band decomposition and (b) a uniform, eight-band directional decomposition.

that provides the flexibility to obtain good resolution both angularly and radially. This family of filter banks contains a number of useful properties which include maximal decimation; perfect reconstruction; an efficient, separable implementation; the option for linear-phase or non-linear phase; an arbitrary number of directional and octave-band partitions; and the option of visualizable directional subbands [65]. Consequently, these filter banks prove to be attractive alternatives to some of the decompositions currently employed to extract and manipulate directional information for a variety of applications.

1.2 Organization of Thesis

Background material including discussions of 1-D decompositions and their importance to multidimensional transforms and the directional filter bank are presented in Chapter 2. A thorough understanding of the directional filter bank is necessary for the development of the octave-band directional filter bank.

The octave-band directional filter bank and many of its implementation issues are discussed in Chapter 3. It is derived from the directional filter bank and is able to decompose 2-D signals with respect to both radial and angular frequency. Similar decompositions from the literature are also presented for comparison and completeness.

After the introduction of the OBDFB, three chapters are presented that discuss the results of

applying the OBDFB to three different applications: texture segmentation, hyperspectral data analysis, and ultrasound despeckling (Chapters 4, 5, and 6, respectively). These chapters discuss the individual applications and provide comparisons between the OBDFB and leading algorithms for each application.

Additional work pertaining to filter bank theory is presented in Chapter 7. In particular, a non-uniform rationally sampled multidimensional filter bank is proposed, and the design methodology is presented. The new filter bank addresses many issues with respect to the OBDFB and DFB as well as other decompositions.

Concluding remarks and a summary of the contributions of this thesis are presented in Chapter 8 along with suggestions for future work.

CHAPTER II

BACKGROUND

Filter banks can be designed to have a wide variety of properties. Desirable properties and characteristics of filter banks include:

- Maximal Decimation - the number of coefficients produced by a decomposition is the same as the number of input samples; this is also known as *critically sampled*. This property is associated with computational efficiency as it keeps the number of samples to be processed to a minimum.

- Separable Filtering - the filtering operations can be performed as one-dimensional filtering operations on multidimensional data; for images, the filtering typically is performed row-wise and column-wise. This property is also associated with computational efficiency as separable filtering is much more efficient than the equivalent non-separable filtering.

- Polyphase Form - an efficient implementation of a decimated filter bank where the filtering is performed *after* the downsampling thereby reducing the number of computations.

- Perfect Reconstruction - the coefficients produced by the forward transform can be sent through the inverse transform to reproduce the original signal without any error.

- Tree Structure - the decomposition is implemented in stages where the output of each stage is further subdivided. This allows for an arbitrary number of subbands.

Not all of these characteristics are important in every application, but it is this versatility that makes filter banks so appealing for so many different applications. General discussions on filter banks may be found in [89].

Tree-structured decompositions that subdivide radially low frequency regions are currently very popular. Such decompositions are particularly well-suited to processing natural images which tend to have energy concentrated in radially low frequency regions. One of the limitations of many of

the tree-structured filter banks is their current use for radial frequency discrimination only; it is not trivial to provide discrimination with respect to directional information.

The goal of this thesis is to produce a filter bank that can discriminate with respect to both radial and directional frequencies while maintaining several, if not all, of the previously described desirable filter bank characteristics. The derivation of such a filter bank resides in the analysis and understanding of the directional filter bank (DFB) [9]. Although many valuable additions have been made to the DFB [66, 72, 79], its fundamental features have remained mostly unchanged. The DFB provides a maximally decimated means of analyzing 2-D signals based on their directional content, and has other attractive features such as perfect reconstruction. It also can be implemented efficiently using separable filtering and IIR filters. In [66], Park described a method of reducing the effects of *frequency scrambling* (discussed at length in Section 2.3.4) that also provides a method of rectangularization of the output subband coefficients. However, for all of the DFB's advantages, it is still unable to discriminate between radially high and low frequency information in a maximally decimated fashion. In order to overcome this obstacle, a strong understanding of the inner workings of the DFB is necessary.

2.1 Traditional Octave-band Filter Banks

Typical wavelet decompositions can bee implemented with separable filter banks. The filter bank is constructed in a tree structure where only the subbands corresponding to low frequencies are subdivided. Sample filter magnitudes may be seen in Figure 2.1.

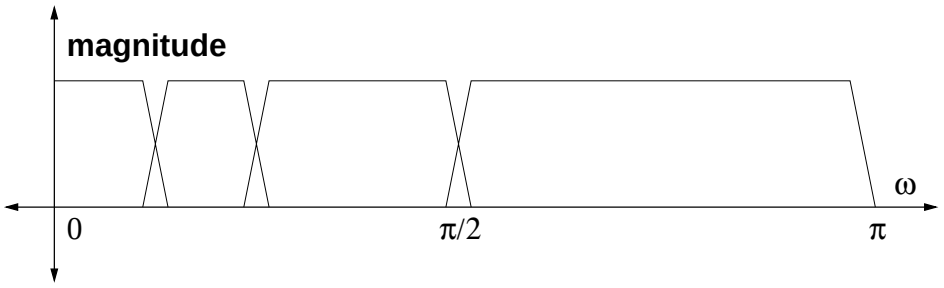


Figure 2.1: Example of a 1-D octave-band decomposition.

These filters have several attractive properties including computationally efficient polyphase forms, maximal decimation, separable implementations, and filter flexibility where a variety of both

FIR and IIR filters can be used.

When these filters are applied row-wise and then column-wise to 2-D signals, they result in decompositions such as the one shown in Figure 1.1(a). These decompositions are particularly useful for image processing because of the tendency of natural images to have a large amount of radially low frequency content. These octave-band decompositions have been very successful in a number of image processing applications, but they are limited by the fact that they are unable to discriminate angularly.

2.2 Current 2-D Decompositions with Radial and Angular Frequency Selectivity

Several 2-D transforms exist that divide the frequency space into directional and octave-band subbands. None of them have all of the properties of the octave-band directional filter bank, but many of them warrant mention for completeness. We shall limit the discussion here to undecimated transforms as the majority of maximally or partially decimated transforms are more intuitively explained after discussing the DFB and OBDFB. In any event, the desired passband geometry of all of these transforms is straightforward. The cortex transform and some representative 2-D Gabor filters are shown in Figure 2.2 as popular examples. As can be seen in the figure, the passband areas are situated both radially and angularly.

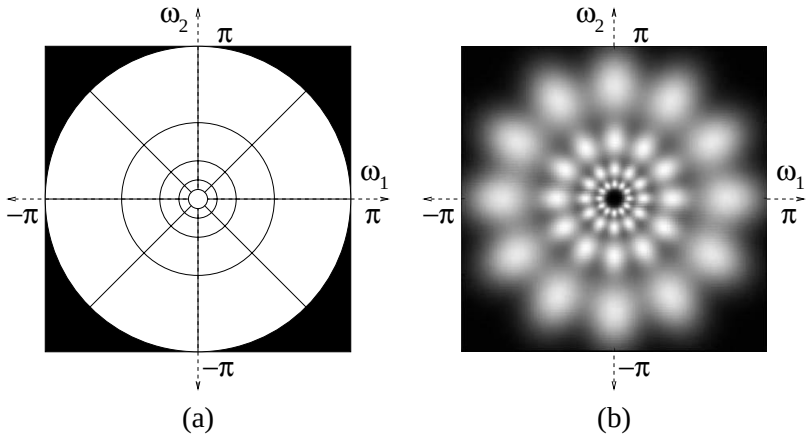


Figure 2.2: Illustration of the 2-D spectrum associated with two popular directional transforms: (a) the cortex transform and (b) 2-D Gabor filters.

Gabor filters, for example, are relatively easy to implement and conceptually simple; they can

be characterized by the following Fourier transform pair:

$$\begin{aligned} h(x,y) &= \frac{1}{2\pi\sigma_x\sigma_y}\exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right]\right\}\cos(2\pi u_0 x), \\ H(u,v) &= A\left(\exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\}+\exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\}\right), \end{aligned}$$

where u_0 is the frequency of a sinusoidal plane wave along the x-axis; σ_x and σ_y are the space constants of the Gaussian envelope along the x and y axes, respectively; $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$; and $A = 2\pi\sigma_x\sigma_y$ [41]. Filters h with arbitrary orientations are obtained by a rotation of the $x-y$ coordinate system. The Fourier representation H is a bit more obvious in the sense that it is clearly a pair of Gaussians oriented in a particular direction with some radial center frequency. The directional partitioning in addition to the octave-band structure is apparent in the figure as well even though the filters themselves are far from ideal with respect to well defined passbands and passband overlap.

Although these decompositions are straightforward to implement, they have several disadvantages. They are data expansive as each subband is simply a filtered version of the input and there is no downsampling. Therefore the expansion is by a factor of the number of channels in the decomposition. Additionally, because the passband geometries necessary to capture directional information are nonrectangular, a separable implementation is not possible. Therefore computational complexity is relatively high.

2.3 The Directional Filter Bank

The DFB originally was proposed by Bamberger and Smith in [9]. The decomposition uses a tree-structure of two-band splits to divide the frequency space into wedge-shaped regions that correspond to spatial directional information. The tree-structure allows for an arbitrary number of directional subbands to be generated (although the number is typically limited to a power of 2). An eight-band decomposition can be seen in Figure 2.3. The following sections discuss the inner workings of the DFB with emphasis on the concepts essential to the derivation of the octave-band directional filter bank (OBDFB). General information on multidimensional signal processing including sampling and filtering may be found in **??**.

An example of two different inputs being decomposed in one branch of the DFB may be seen

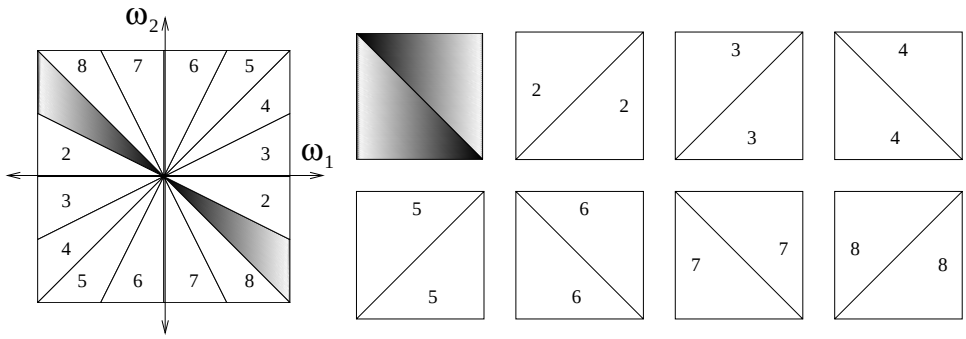


Figure 2.3: An eight-band DFB illustrating the full-rate subbands (left) and their equivalent maximally decimated counterparts (right).

in Figure 2.4. In the figure, the Bamberger polyphase structure is depicted. The first two stages of the DFB use modulators to isolate the desired frequency regions, and after that, resampling matrices are used to shape the different branches accordingly.

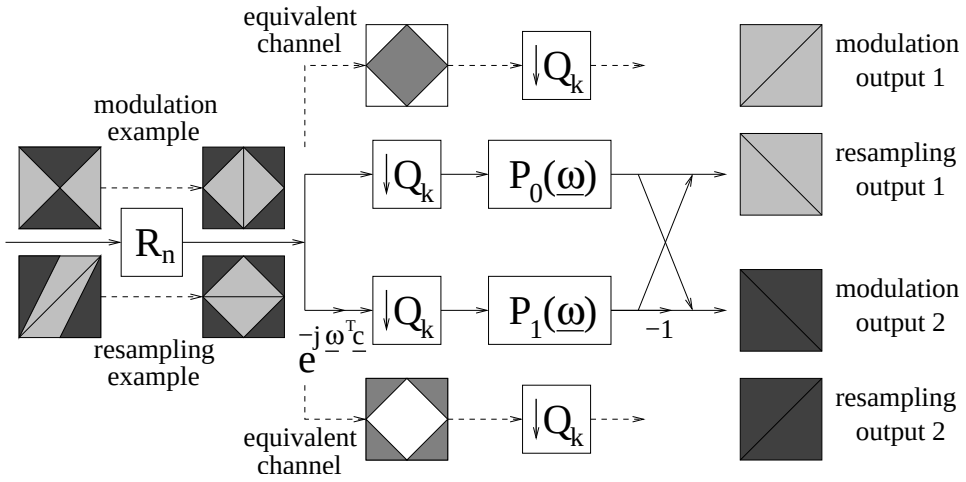


Figure 2.4: The two-band split used in the tree-structured DFB. The traditional polyphase form is depicted as are the equivalent channels.

2.3.1 Quincunx Downsampling

One of the first considerations for the DFB is the downsampling matrices used in the two-band filter banks. For now, let us assume that the diamond-shaped passbands shown in Figure 2.5 represent the regions of interest in the frequency space and that we need a way to downsample them efficiently. Since we are using two-band splits, one of the important properties of this matrix is that it downsamples by a factor of 2 and consequently has a determinant whose absolute value is

equal to 2. Similarly, matrices with a determinant whose absolute value is equal to 1 are unitary resampling matrices that simply rearrange sample points, not throwing any away. There are several quincunx downsampling matrices that generate the sampling lattice shown in Figure 2.6 (the shaded samples) [66]; however, the two most commonly used quincunx matrices create minimal rotation of the data. A sample input signal is shown in Figure 2.6. The two quincunx downsampling matrices used are

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{Q}_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{2.1}$$

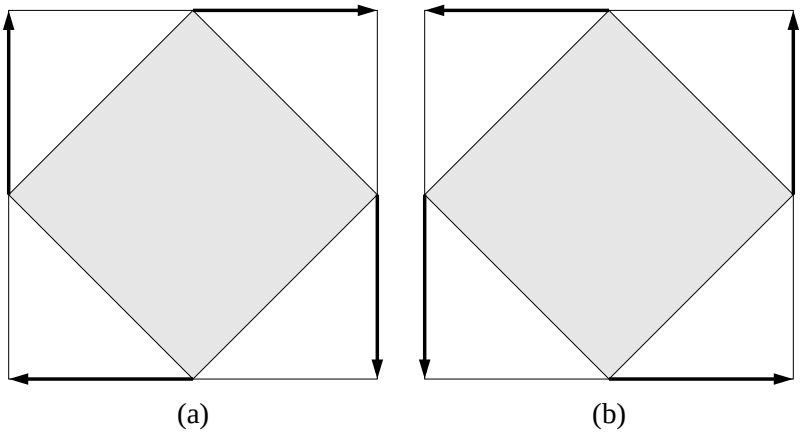


Figure 2.5: The frequency effects from downsampling by (a) $\mathbf{Q}_1$ and (b) $\mathbf{Q}_2$.

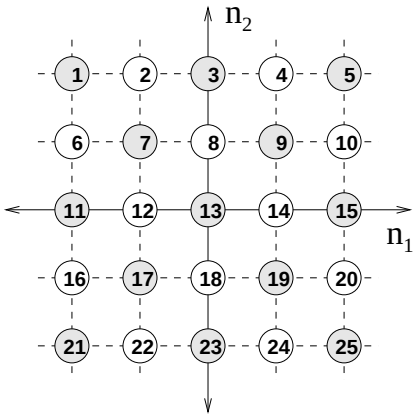
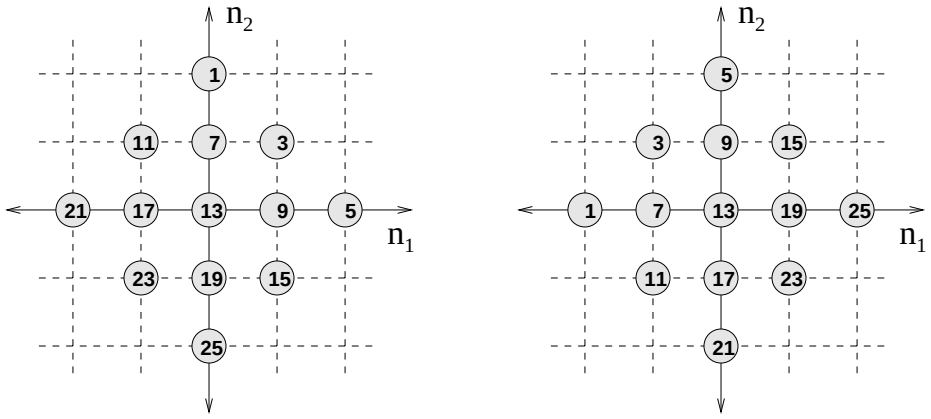


Figure 2.6: An example of labeled input data where the shaded samples indicate a quincunx lattice.

The spatial effect that these matrices have when downsampling may be seen in Figure 2.7. In the figure, the input is assumed to be the signal $x[\mathbf{n}]$ shown in Figure 2.6 where $\mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$. Then the result from downsampling by $\mathbf{Q}_1$, $y_1[\mathbf{n}] = x[\mathbf{Q}_1\mathbf{n}]$, may be seen in Figure 2.7(a) and the result from $\mathbf{Q}_2$, $y_2[\mathbf{n}] = x[\mathbf{Q}_2\mathbf{n}]$, in Figure 2.7(b). Additionally, assuming that $x[\mathbf{n}]$ has a diamond-shaped region of support in the frequency domain, then the effects of spatially downsampling by $\mathbf{Q}_1$ and $\mathbf{Q}_2$ may be seen in Figure 2.5. The expansion in frequency space is analogous to 1-D downsampling, and the counter-rotation is expected because of the inverse relationship between space and spatial frequency.



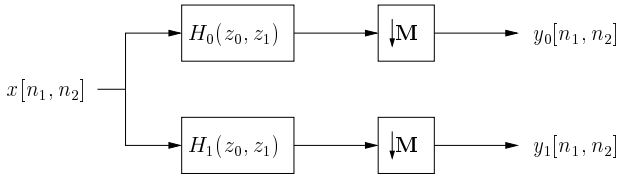
(a) (b)

Figure 2.7: The spatial effects from downsampling by (a) $\mathbf{Q}_1$ and (b) $\mathbf{Q}_2$.

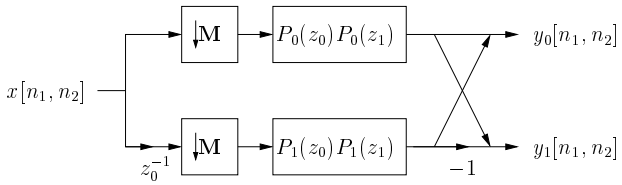
2.3.2 Diamond Filters

In order to obtain diamond-shaped regions of support in the frequency domain, filtering is necessary. The actual design of diamond filters has already been discussed in [16], for example, but the structures used to implement them in a DFB context will now be discussed.

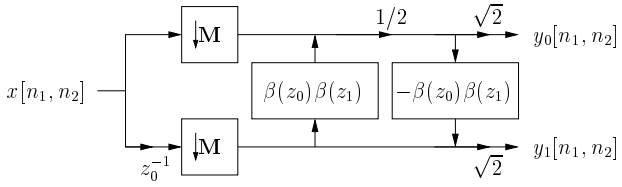
The basic building block of the original DFB is the two-band filter bank using filters with diamond-shaped passbands and stopbands. Different structures for implementing such a filter bank may be seen in Figure 2.8. Conceptually, the first example is the simplest because $H_0(z_0, z_1)$ is simply a diamond filter, and $H_1(z_0, z_1)$ is its complement. Additionally, $\mathbf{M}$ is a quincunx downsampling matrix (not necessarily confined to $\mathbf{Q}_1$ and $\mathbf{Q}_2$), as described earlier. Each of these structures can provide perfect reconstruction, and each has its own set of properties.



(a) Typical Two-band Split



(b) Traditional Polyphase Structure



(c) Ladder Filter Implementation

Figure 2.8: Different two-band structures for implementation: (a) the basic structure, (b) the traditional polyphase structure, and (c) a ladder filter implementation.

The original DFB used modulators, unitary resamplers, diamond filters, and quincunx downsamplers in its two-band splits. The input to the two-band filter bank is modulated in the first two stages or resampled by a unitary resampling matrix in all subsequent stages such that the diamond filters and their complements would capture the desired frequency regions. Different implementation structures of the two-band filter bank may be seen in Figure 2.8. The first filter bank is a staightforward implementation whereas the next two are polyphase implementations. The polyphase structure proposed in [9] is shown in Figure 2.8(b) and its analysis filters are described by the following set of equations:

$$\begin{aligned} H_0(\omega) &= P_0(\mathbf{M}^T\omega) + \exp(j\mathbf{M}^T\omega)P_1(\mathbf{M}^T\omega), \\ H_1(\omega) &= P_0(\mathbf{M}^T\omega) - \exp(j\mathbf{M}^T\omega)P_1(\mathbf{M}^T\omega), \end{aligned}$$

where $\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$, P_0 and P_1 are the polyphase filters, and $\mathbf{M}$ is the quincunx downsampling matrix. This polyphase form allows for perfect reconstruction, separable filtering, and the use of IIR filters that can have guaranteed stability (though not causality). The ladder filter (a biorthogonal filter bank) implementation proposed in [68] and discussed in [83] is shown in Figure 2.8(c), and its analysis filters are described by the following set of equations:

$$\begin{aligned} H_0(z_0, z_1) &= \frac{1}{2}(z_0^{-2N} + z_0^{-1}\beta(z_0 z_1^{-1})\beta(z_0 z_1)), \\ H_1(z_0, z_1) &= -\beta(z_0 z_1^{-1})\beta(z_0 z_1)H_0(z_0, z_1) + z_0^{-4N+1}, \end{aligned}$$

where $\beta(z)$ is a linear phase filter with even length N. To obtain fan filters, we let $z_0 \to -z_0$ in the previous equation.

The ladder structure was used in a DFB implementation in [79] and was shown to provide advantages similar to those of Bamberger's polyphase form. However, the ladder structure has the addition of structurally enforced perfect reconstruction. Both Bamberger's polyphase form and the ladder structure allow the diamond filtering to be performed separably in the decimated domain. However, because of design constraints, the traditional polyphase filters themselves cannot have linear phase although their equivalent non-decimated filters can, whereas the ladder filters can [3, 68]. This allows for more flexibility when dealing with signal boundaries such as the edges of images. The linear phase filters in the ladder implementation also provide higher computational

efficiency as symmetric (and anti-symmetric) filters can halve the number of multiplies of non-linear phase filter of the same length. After informally comparing the traditional polyphase form and the ladder structure, the latter was chosen.

2.3.3 Resampling Matrices

We now discuss the use of diamond filters to generate directional passbands. Successive applications of diamond filters in an octave-band structure appears to yield passbands of lower and lower radial frequency content, not angular frequency content. In order to allow the two-band diamond filter bank to provide the frequency partitions desired, modulators and unitary sampling matrices are used to adjust the input of each two-band filter bank such that overall angular frequency information can be isolated.

Modulating the input by $(-1)^{n_1}$ is simply a shift with respect to ω_1 by π in frequency space. For the first stage, this appears to generate approximately directional bands, with regions 1 and 2 captured in one subband and regions 3 and 4 cpatured in the other in Figure 2.9. After the first stage, modulators serve a similar purpose by allowing the use of diamond filters again in the second stage shown in Figure 2.10.

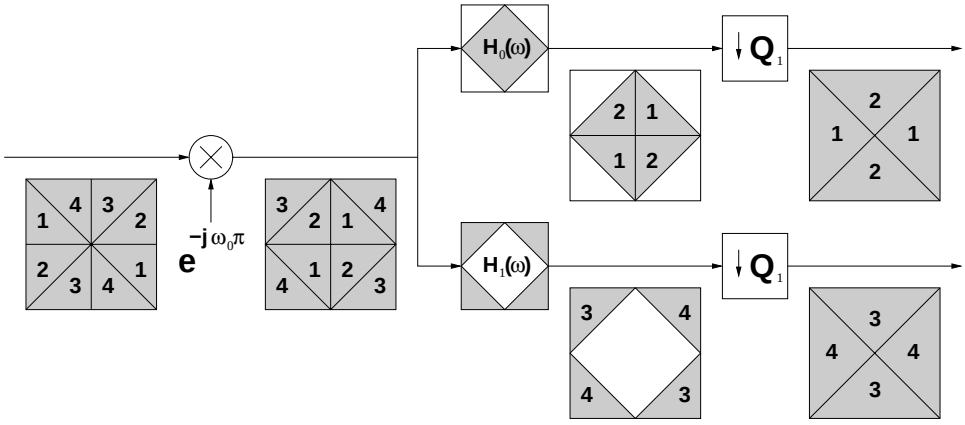


Figure 2.9: The first stage of Bamberger's DFB.

However, now the subbands are oriented such that the wedges need to be split down the long way (see Figure 2.11). In the figure, a single directional subband is shown being split in this fashion. The way to achieve this is not obvious, but it can be done with the use of diamond filters and unitary downsampling matrices. For practical application, unitary resampling means that none of

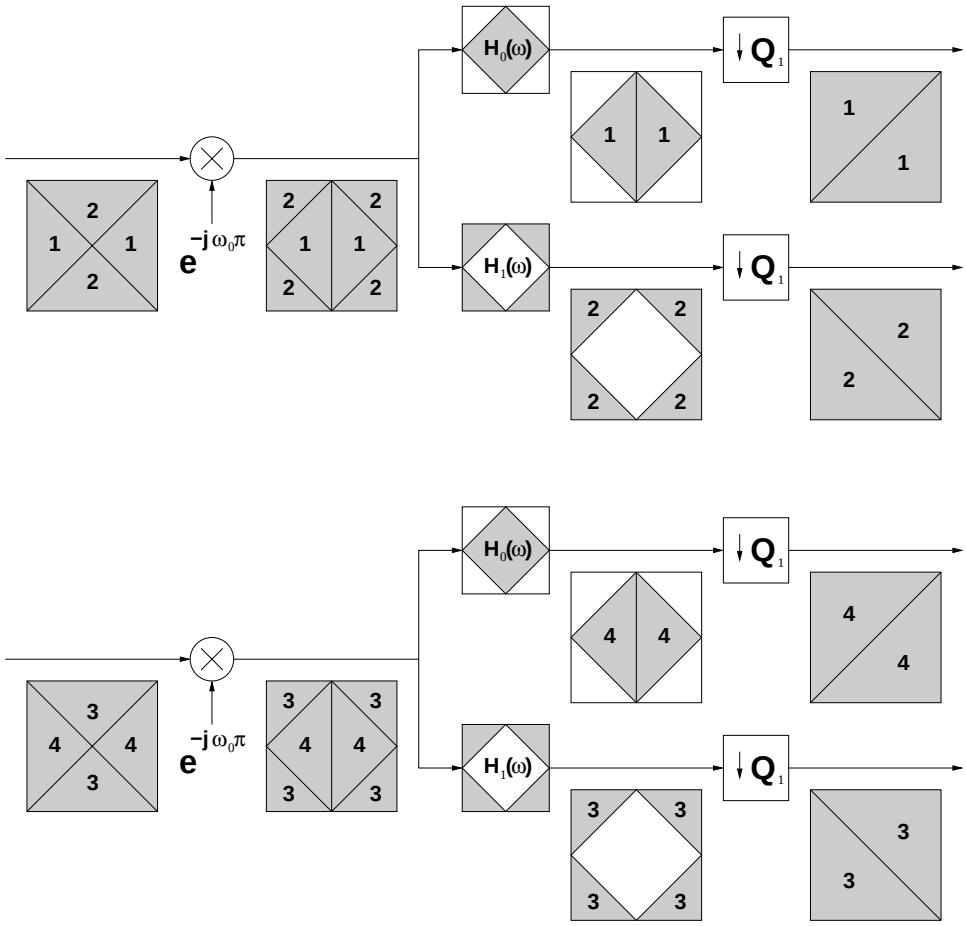


Figure 2.10: The second stage of Bamberger's DFB.

the samples are thrown away, simply rearranged. The following matrices are explicitly defined:

$$\mathbf{R}_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{R}_4 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

The effect that they have on downsampling the signal from Figure 2.6 may be seen in Figure 2.12, and the effect they have in the frequency domain is depicted in Figure 2.13. Once the unitary downsampling is performed, diamond filters are used to separate the appropriate frequency regions.

Because these resampling matrices reshape the geometry of the input signal in the frequency domain, the diamond filters are able to isolated desired regions. Also, because the orientations of the third-stage (shown in Figure 2.14) output subbands are similar to those of the second stage, more

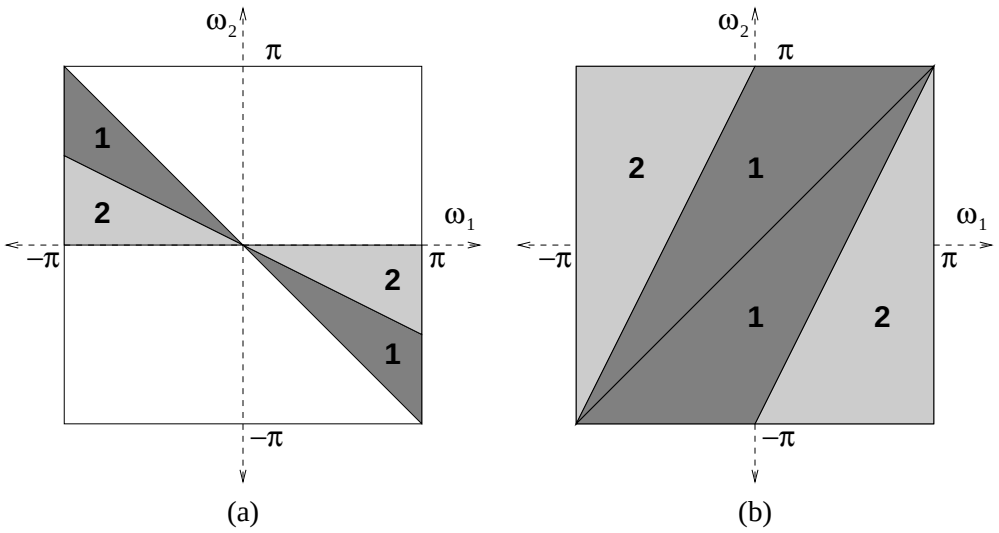


Figure 2.11: The directional split necessary after the second stage is shown above: (a) the two directional bands resulting from splitting one of the original 4 directional bands and (b) the equivalent "long" split necessary to perform the splitting operation on the same subband in the decimated domain.

two-band splits preceded by appropriate unitary matrices may be cascaded to each output to obtain as high an angular frequency resolution as desired.

This tree structure not only allows for an arbitrary number of splits, it also allows for separable filtering at each step which is still more efficient than non-separable multi-dimensional filters. For example, it would still be more efficient to decompose an image using the DFB, and then reconstruct a single directional subband than to filter an image initially using a non-separable directional filter of equivalent size. More specifically, to do a 2^D-band directional decomposition, using length N 1-D filters, the DFB would require approximately $(4 \times D \times N)$ multiplies per sample. If non-separable filters were used, even if they had a similar order filter size of $N \times N$, the number of computations would be $(2^D \times N \times N)$ per output sample.

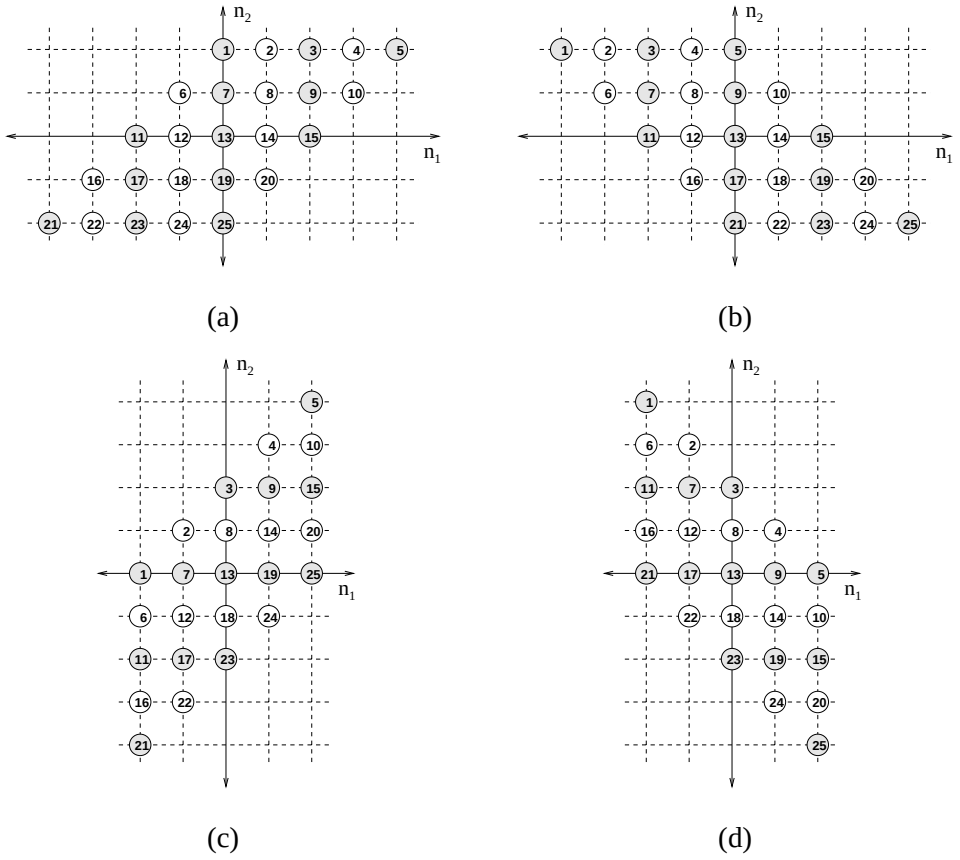


Figure 2.12: Examples of different downsamplings by unitary matrices (a) $\mathbf{R}_1$, (b) $\mathbf{R}_2$, (c) $\mathbf{R}_3$, and (d) $\mathbf{R}_4$.

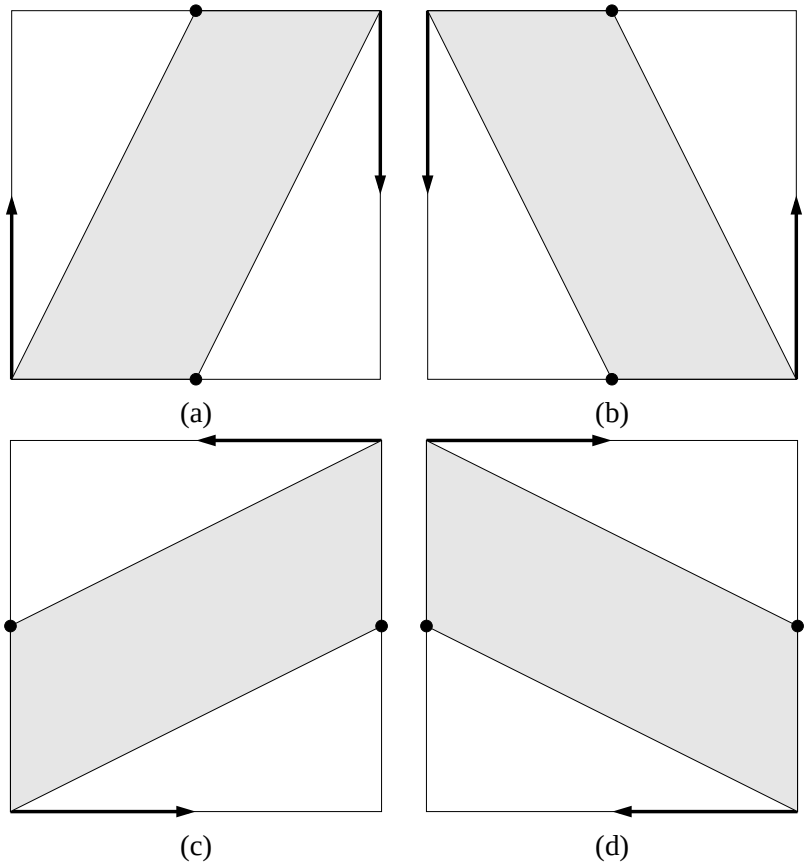


Figure 2.13: Examples of different downsamplings by unitary matrices (a) $\mathbf{R}_1$, (b) $\mathbf{R}_2$, (c) $\mathbf{R}_3$, and (d) $\mathbf{R}_4$.

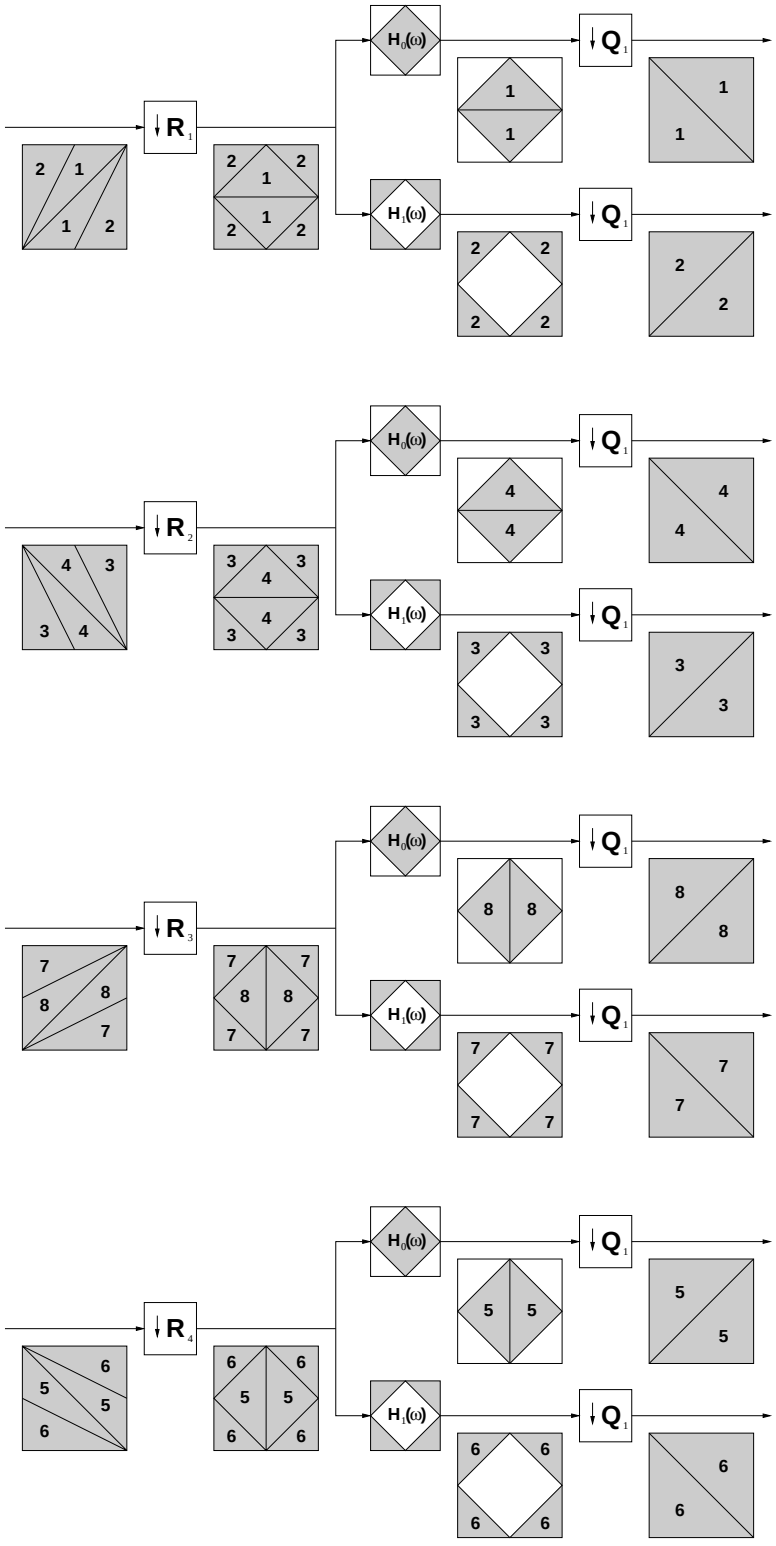
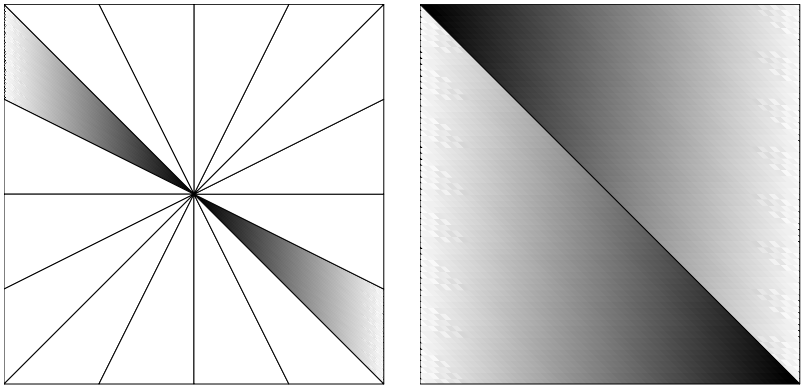


Figure 2.14: The third stage of Bamberger's DFB.

2.3.4 Frequency Scrambling

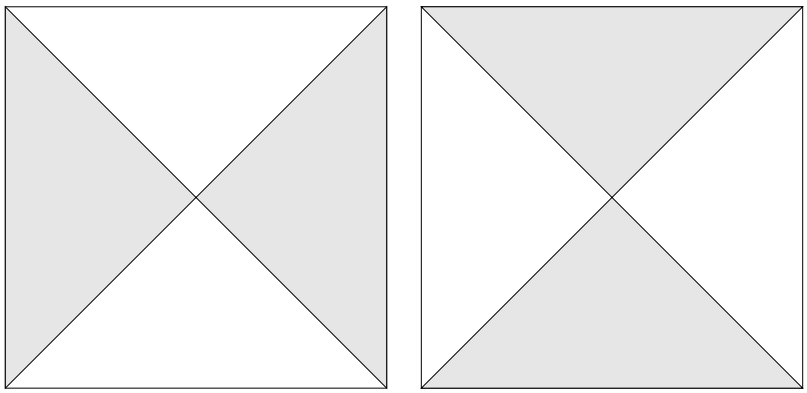
Because of the modulators used in the DFB, radially low frequency information is displaced to radially high frequency areas as seen in Figure 2.15. This phenomenon is referred to as *frequency scrambling* from [65].



(a) (b)

Figure 2.15: Frequency scrambling from (a) the non-decimated equivalent passband and (b) the maximally decimated subband.

Moving the modulators to the end of the filter bank as described in [7] can mitigate some of the major modulation effects, however, this requires modulation of the filters themselves instead of modulation of the signal. So instead of diamond filters, fan filters are used as seen in Figure 2.16. A different design procedure with the same end-result was introduced in [69].



(a) (b)

Figure 2.16: The fan filters associated with (a) $\mathbf{Q}_1$ and (b) $\mathbf{Q}_2$ where the shaded regions denote the passbands.

To compensate for the remaining displacements from aliasing, Park introduced the concept of *post-sampling* (originally referred to as *backsampling*) in [66]. Post-sampling conceptually uses the multirate identities to push all of the downsampling operations from each stage of the DFB into a single downsampling operation by an overall downsampling matrix at the end of the final stage. This concept is illustrated in Figure 2.17. If this downsampling matrix is followed by a unitary matrix that downsampled such that the overall effect was a diagonal downsampling matrix, then the subbands would be rectangular, and the orientation of the directional information would be as similar as possible to the original information with the constraint that it be maximally decimated.

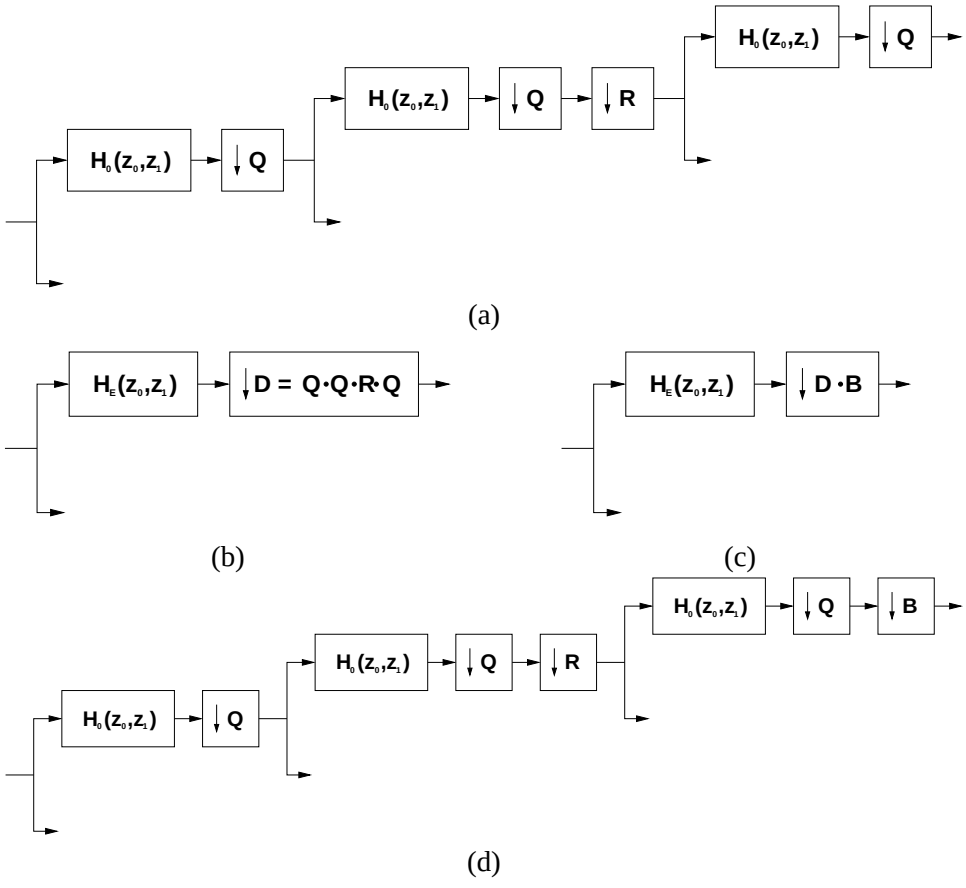


Figure 2.17: Post-sampling illustration (figure from [66]): (a) two paths in an eight-band DFB. (b) simplified structure, (c) simplified structure with post-samplers, and (d) the original two paths with the post-samplers.

In order to remove the rotation that occurs after downsampling by $\mathbf{Q}_1$ in the first two stages, $\mathbf{Q}_1$ is used for the first stage and $\mathbf{Q}_2$ for the second. Then, for the third stage, the post-sampling

matrices are defined as

$$\mathbf{B}_1 = \mathbf{R}_4, \quad \mathbf{B}_2 = \mathbf{R}_3,$$

$$\mathbf{B}_3 = \mathbf{R}_2, \quad \mathbf{B}_4 = \mathbf{R}_1.$$

For successive stages, the defining equation is

$$\mathbf{B}_i^n = \left[\left[\mathbf{B}^{(n-1)} \right]^{-1} \mathbf{R}_i^n \mathbf{Q}_i^n \right]^{-1} \mathbf{T}_i, \tag{2.2}$$

where

$$\mathbf{T}_i = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, & \text{if } i = 1, 2, \\ \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, & \text{if } i = 3, 4. \end{cases} \tag{2.3}$$

Finally, a complete set of DFB coefficients of the input image in Figure 2.18 may be seen in Figure 2.19. The input image is a 2-D chirp function that effectively provides frequency space information in the spatial domain. Consequently, frequency passband geometries are observable directly from the output subbands.

We have discussed the DFB and many of its enhancements including the ladder filter implementation, frequency scrambling, and post-sampling. These improvements will only help us in the next chapter where we discuss the DFB and how to add the ability to discriminate with respect to radial frequency.

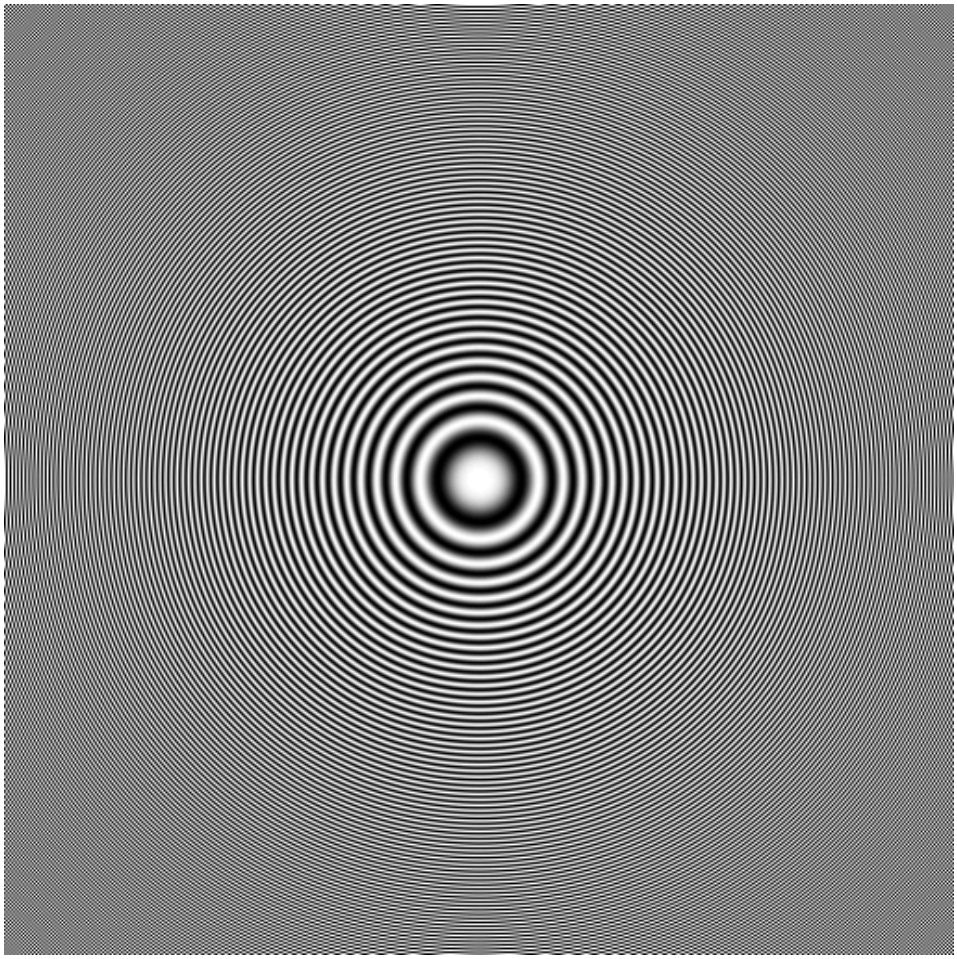


Figure 2.18: A sample input image whose rectangularized DFB coefficients are shown in Figure 2.19.

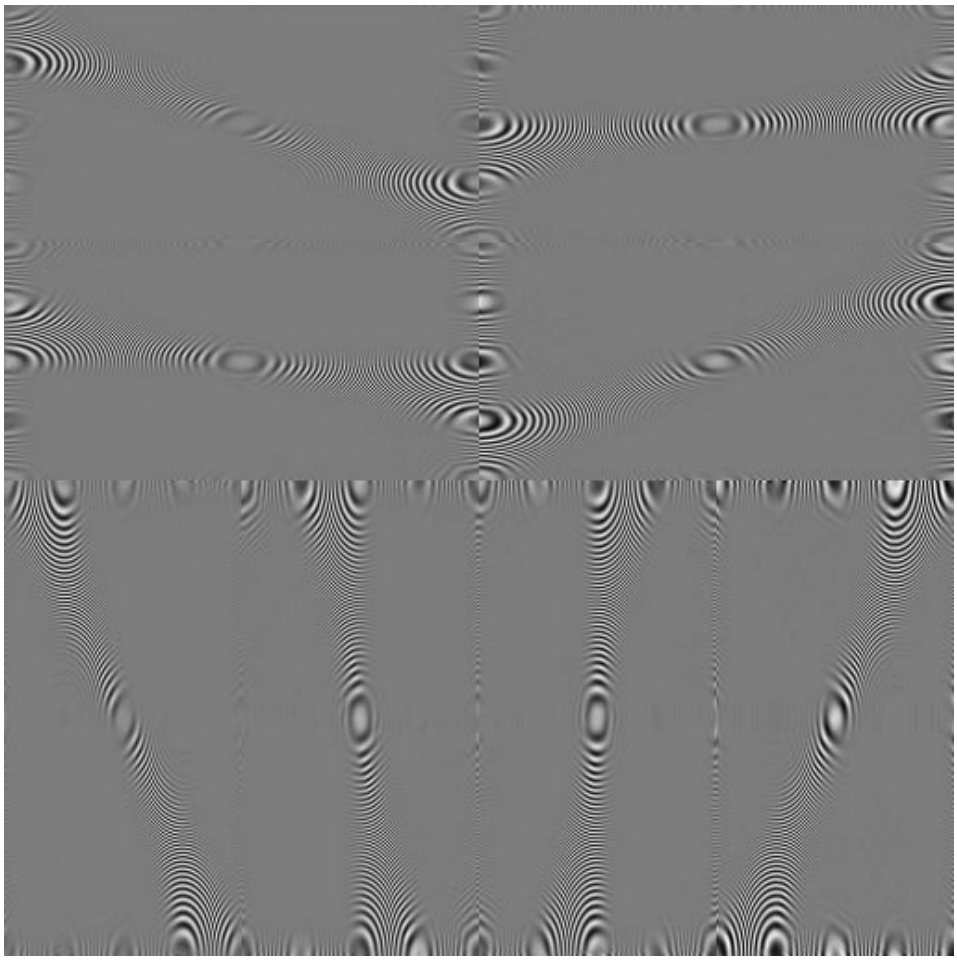


Figure 2.19: An example of the rectangular form of the coefficients of the DFB of the input image shown in Figure 2.18.

CHAPTER III

PROPOSED RESEARCH

Although there have been several enhancements proposed and implemented for the DFB that have given it advantages over other directional decompositions, the DFB is unable to discriminate between radially low and high frequency content in a maximally decimated framework. Consequently, it is not suitable for use in many applications that take advantage of the low frequency concentration of energy in images, for example.

In this chapter, we propose the octave-band directional filter bank (OBDFB) which is an extension to the DFB that allows for both angular and radial frequency selectivity. By first applying the DFB, the OBDFB is then able to isolate radial frequency content within each directional band in a maximally decimated framework.

3.1 The Octave-band Directional Filter Bank

The underlying goal of the OBDFB is to generate a decomposition that can separate radially low and radially high frequency content in addition to directional frequency content and simultaneously retain all of the advantages of the DFB, such as maximal decimation and a separable filter implementation. An example of an OBDFB may be seen in Figure 3.1.

Conceptually, the OBDFB extracts radial frequency information from the maximally decimated subbands of the DFB. The application of the DFB accomplishes the goal of providing angular frequency resolution. Because the subbands are maximally decimated, the frequency content of each directional band is aliased such that it fills the frequency space and warps the frequency slices that characterize the subbands of the DFB in the non-decimated domain. The filter passband geometries necessary to achieve the radial frequency partitioning of the decimated directional subbands is motivated through the illustration shown in Figure 3.2. The shaded region in Figure 3.2(a) shows the first directional DFB passband (4^{th} quadrant) next to its downsampled representation (Figures 3.2(b) and 3.2(c)). As is evident from Figure 3.2(a), the dark shaded signal spectrum corresponds to the

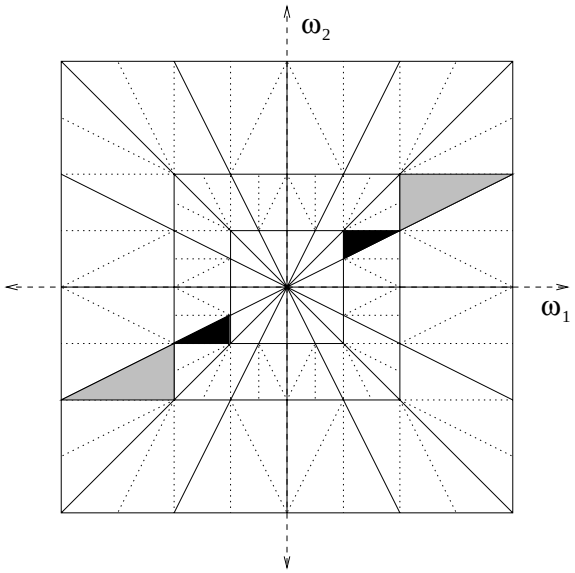


Figure 3.1: A sample decomposition of the OBDFB; the equivalent subbands are shown in the nondecimated domain, and the shaded subbands represent corresponding octave-band subbands within a single directional band.

low frequencies while the light shaded regions correspond to high frequencies. This shading is helpful in keeping track of the original frequency content of the different subbands in the face of aliasing. Figure 3.2(b) (as well as Figure 3.2(c)) illustrates how the low and high frequency components are aliased in the decimated subband domain, and illustrate the crux of the OBDFB.

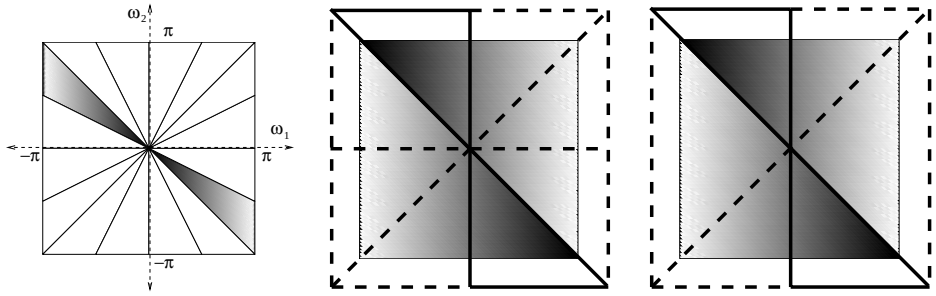


Figure 3.2: Example of (a) a single directional band with a gradient to reflect the low to high frequencies; (b) the same subband in the decimated domain with a four-band DFB overlaid and a (c) three-band (unbalanced) DFB.

The frequency scrambling discussed earlier is integral to the derivation of the OBDFB. The

frequency scrambling in the maximally decimated subband is evident in Figure 3.2(b) (and Figure 3.2(c)); however, if a four-band DFB were applied to the subband in Figure 3.2(b), one of the subbands of the four-band DFB would contain the original, radially low-frequency information denoted by the bold black partition whereas the passbands associated with the radially high frequency region are indicated with dashed line partitions. If this same procedure is applied to each of the directional subbands, one obtains a decomposition with both angular and radial frequency domain separation—an example of which is shown in Figure 3.3. Also, because the DFB generates subbands with similar types of geometries, the radially low-frequency subband would also be oriented such that another four-band DFB could be used to isolate even lower radial frequency information. Consequently, this process can be applied indefinitely to the lowest radial frequency subbands to capture lower and lower frequency content in much the same way the DWT is currently applied to images with the clear distinction that this procedure is performed on subbands that have already performed the directional portion of the decomposition.

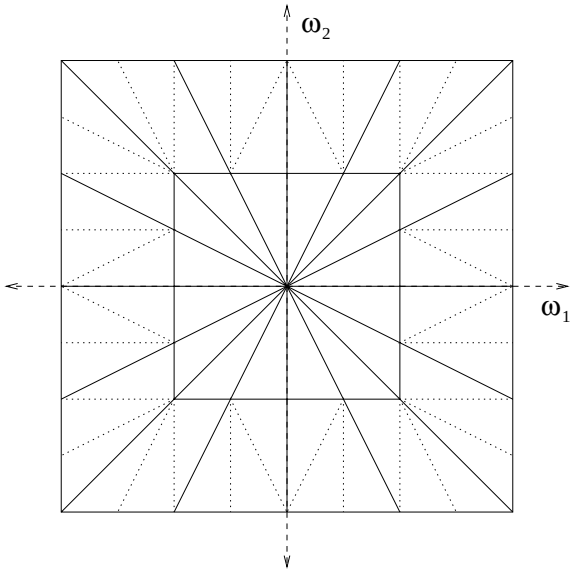


Figure 3.3: A depiction of the OBDFB in the non-decimated, frequency domain. The dotted lines indicate boundaries between subbands where further decomposition is typically unnecessary.

It should be noted that in order to isolate radially low-frequency content, either a four- or three-band DFB may be used (shown in Figures 3.2(b) and 3.2(c), respectively). However, the subbands

of a four-band DFB have an equivalent downsampling described by

$$\mathbf{Q}_1\mathbf{Q}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ were defined in (2.1), whereas the three-band DFB would only have two bands with the equivalent downsampling as (3.1) and one band downsampled only by $\mathbf{Q}_1$. Consequently, the arrangement of coefficients and visualization is simpler for the four-band case despite the extra computation.

3.1.1 Type Definition

The OBDFB is based on a tree structure, where each stage involves different resampling matrices and filters. It is useful to define "type" classifications for the various subbands in order to discuss each instance succinctly. The geometric descriptions for each class follow directly from the relationship between subband number and directional band number as seen in Figure 3.4. The type definition associated with the subband orientation is also in the figure.

For the first stage ($n = 1$), $\mathbf{Q}_1$ is used ($\mathbf{Q}^{[1]} = \mathbf{Q}_1$), and for the second stage ($n = 2$), $\mathbf{Q}_2$ is used ($\mathbf{Q}^{[2]} = \mathbf{Q}_2$). After these two stages, the overall downsampling matrix is described by (3.1), where it is important to realize that the matrix is diagonal; i.e., the downsampling is equivalent to downsampling separably by two with respect to both n_1 and n_2. Additionally, both downsampling matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are used to avoid the inherent rotation associated with using either matrix for both stages. After these stages ($n > 2$), $\mathbf{Q}^{[n]}$ is determined for input subband of type $t^{[n-1]}$ as follows:

$$\mathbf{Q}^{[n]} = \begin{cases} \mathbf{Q}_1, & \text{if } t^{[n-1]} = 2, 3, \\ \mathbf{Q}_2, & \text{if } t^{[n-1]} = 1, 4. \end{cases} \tag{3.2}$$

Subsequently, the following resampling matrices are used to provide the proper filtering in the remaining stages of the tree:

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix},$$

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R}_4 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}.$$

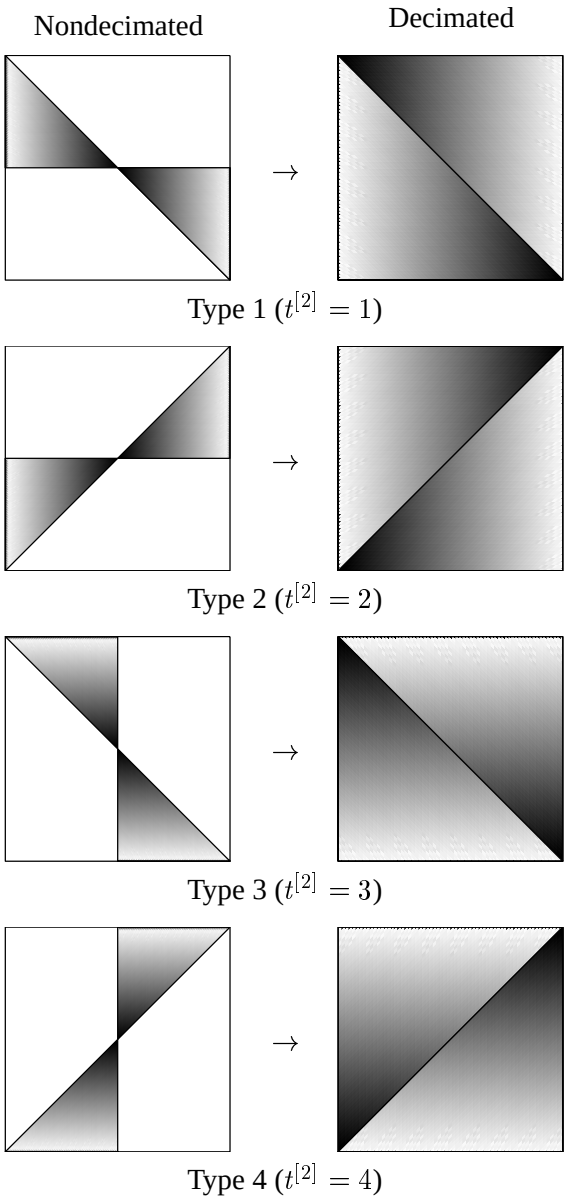


Figure 3.4: Type definitions from the output of a 4-band DFB in 2-D frequency space where the decimated subbands are shown modulated by $(-1)^{n_1+n_2}$ for clarity.

These type definitions are helpful to keep subband descriptions brief. Also, they will be even more convenient after the discussion of subband orientation in Section 3.1.2.

3.1.2 Subband Orientation

An important consideration with respect to the further decomposition of directional subbands (either for increased angular resolution or separation of radial frequency content) is the orientation of the subbands. By "orientation," we mean where the radially low frequency content is and whether or not the subband has been skewed from its original geometry (beyond was is necessary to keep it maximally decimated). When doing a purely directional decomposition, this is not as important a concern because the directional decomposition can proceed perfectly well even if the subbands are rotated or skewed, but because the octave-band, four-band DFB is added, the orientation becomes extremely important for consistency.

An example of the problem of orientation is illustrated in Figure 3.5. If the sample shown in the figure were the final directional stage, the application of a four-band DFB to the original subbands would yield different geometries for the radially higher frequency subbands which is undesirable because we would want all of the directional bands to have similar partitions. In order to alleviate this inconsistency, post-samplers (different from those in [65]) are used to retain consistent subband geometries.

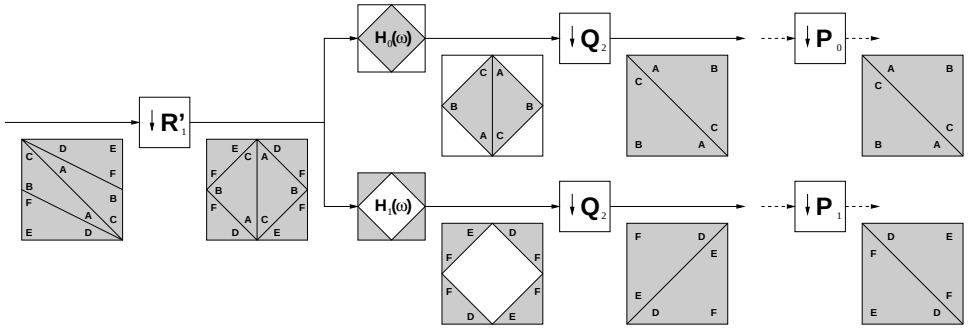


Figure 3.5: Post-sampling example for a subband with $t = 1$ where the subbands have all been modulated for visual simplicity.

The following are the definitions for the post-sampling matrices $\mathbf{P}_{1,t}$ and $\mathbf{P}_{2,t}$ (dependent on type t):

$$\mathbf{P}_{1,1} = \mathbf{I}_2, \quad \mathbf{P}_{2,1} = \mathbf{R}_4; \qquad \mathbf{P}_{1,2} = \mathbf{R}_3, \quad \mathbf{P}_{2,2} = \mathbf{I}_2;$$

$$\mathbf{P}_{1,3} = \mathbf{I}_2, \quad \mathbf{P}_{2,3} = \mathbf{R}_2; \qquad \mathbf{P}_{1,4} = \mathbf{R}_1, \quad \mathbf{P}_{2,4} = \mathbf{I}_2;$$

where $\mathbf{I}_2$ is the 2×2 identity matrix. For computational efficiency, the post-samplers can be combined (or removed in the case of $\mathbf{I}_2$) with the resampling matrices that follow in successive directional decomposition stages. It should be noted that these post-sampling matrices $\mathbf{P}$ are different from Park's post-sampling matrices $\mathbf{B}$ from (2.2) although they can be combined in the same way that the resampling matrices can be combined with the $\mathbf{P}$ after each stage.

It should be noted that the type definitions from Figure 3.4 are preserved after the octave-band portion of the OBDFB for the radially low frequency components. When performing type enforcement, for $n \geq 2$, $t^{[n+1]} = t^{[n]}$. This is somewhat intuitive after examination of the triangular geometry of the low-frequency portion of a directional band. Because the overall downsampling of a four-band DFB is simply a downsampling by two in each direction, the type enforcement is guaranteed. This is also useful conceptually, that subbands and their radially low frequency portions have the same type for successive capturing of low-frequency content.

Finally, Park's post-sampling is responsible for the rectangularization of subband coefficients. It is also what allows an intuitive subband arrangement for convenient processing by algorithms such as zero-tree coding as seen in Figure 3.6.

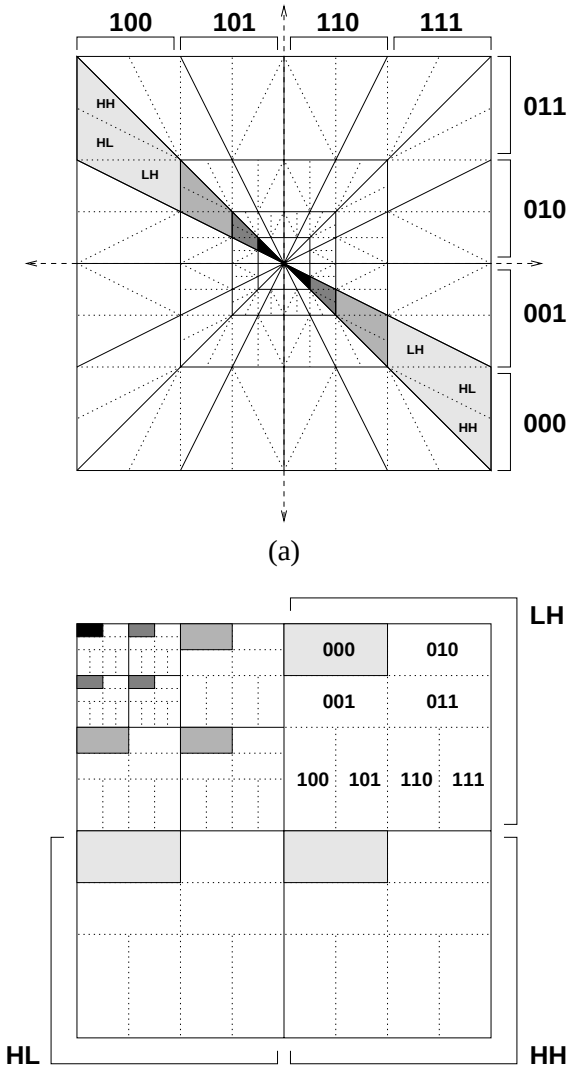


Figure 3.6: A sample OBDFB with 8 directions and 3 octave-band splits: (a) the decomposition in the frequency domain and (b) the subband arrangement. All of the shaded regions correspond to the same directional band, while the typical octave-band subband arrangement is essentially preserved.

3.1.3 Computational Efficiency

The OBDFB has many advantages over other image decompositions including the use of linear-phase filters, perfect reconstruction, and control of both angular and radial frequency resolution. Gabor filters are representative of many non-decimated decompositions and will be used in the following example. The greatest advantage of the OBDFB over Gabor filters comes from the computational efficiency and to a lesser extent, reduced memory requirements. Because the OBDFB is maximally decimated, computations are reduced at each stage in addition to saving memory. Furthermore, its tree structure allows for the use of two-band filter banks that have polyphase forms that allow efficient separable filtering.

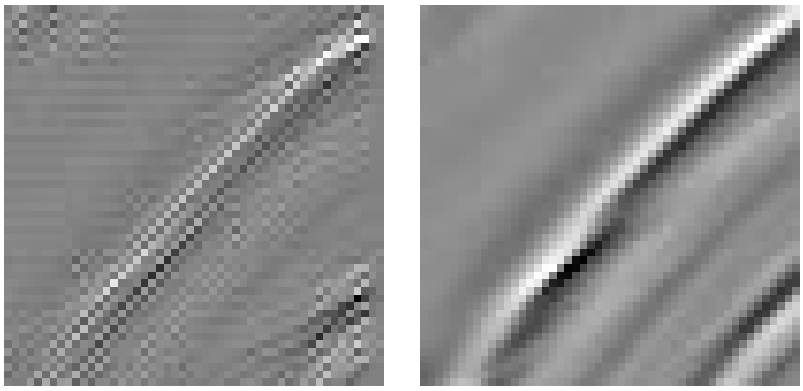
As a simple experiment, it is enlightening to compare the arithmetic complexity of the OBDFB to a direct form implementation of an equivalent Gabor decomposition. Consider as an example a decomposition with N_d directions and N_o octave bands. If the filters are of size $N \times N$, then the Gabor decomposition would require $N_d \times N_o \times N \times N$ multiplies per pixel. For a comparable OBDFB, the number of multiplies for the directional decomposition is $2 \times N \times \log_2 N_d$ per pixel, and the number for the octave-decomposition is $2 \times 2 \times N \times \frac{1-(\frac{1}{4})^{No-1}}{1-\frac{1}{4}}$ multiplies per pixel. Thus, a 256×256 image decomposed using 4 directions and 3 octave bands with filters of length $N = 12$ would require 113 million multiplies for the Gabor filter bank but only 7.1 million multiplies for the OBDFB. These savings increase dramatically with filter length N.

3.1.4 Image Border Processing

Because image borders can introduce discontinuities to processing methods, we address them explicitly. For a finite-length signal, nothing is assumed about it outside of its range of indices $0 \leq n < N$. In order to filter near the boundaries, values need to be used from beyond the range of indices. Typical solutions for border processing include zero padding, boundary value replication, periodic extension, and symmetric extension [64]. Zero padding assumes signal values of zero outside of the range; boundary value replication is similar except that it assumes values equal to the value of the closest boundary sample; periodic extension simple treats the signal as an infinite-length periodic signal with length N where the original signal is simply repeated; symmetric extension first mirrors the original signal at the boundary and then periodically extends this new signal. Because it removes discontinuities at signal boudaries [8, 87] and can also be extended to images, symmetric extension is highly desirable and is used in this thesis when applicable.

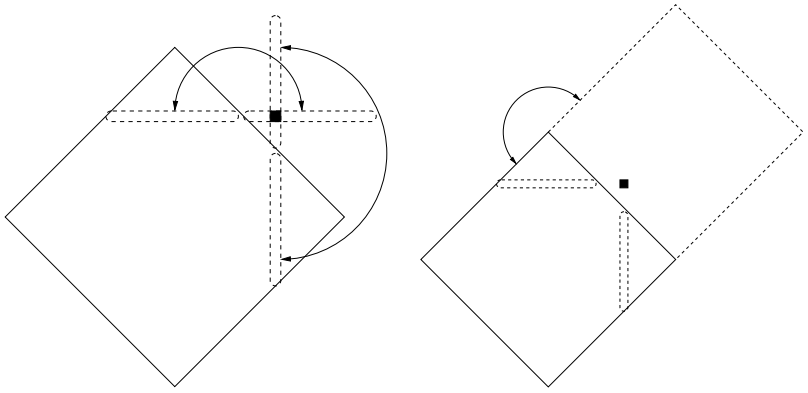
Currently, because of the separable processing, image borders are being handled in an "inconsistent" fashion. An example of this effect is shown in Figure 3.7 where the upper, right corner of the *Lenna* image is shown after reconstructing single directional bands using the DFB and an equivalent non-separable FIR directional filter (generated using the DFB). The underlying reason for this effect is illustrated in Figure 3.8. The inconsistency of the extended point makes the processing of this image quite different than if a non-separable directional filter were used instead of the DFB. However, because all of the downsamplings are known, it should be possible to symmetrically extend in a given direction as shown in Figure 3.8(b). Because the ladder filter implementation is used (i.e. perfect reconstruction is structurally enforced), this method of symmetric extension is feasible and provides more control over image border characteristics.

One solution to this discrepancy is to conceptually perform post-sampling first (for subband rectangularization) and then do the appropriate 1-D filtering now at angles dictated by the post-sampling matrices. This is presented visually in Figure 3.9. This solution is attractive for several reasons. The first is that we no longer have to keep track of the "zero position" that is typically very important for each subband to make sure that the coset vector delays are accurate. Although the coset vectors will change based on their interaction with the post-sampling matrix, it will be



(a) (b)

Figure 3.7: (a) This corner section was taken from a *Lenna* image that was reconstructed from a single subband of the DFB; (b) this is from the same location of a *Lenna* image that was filtered by the reconstructed subband of a DFB of a delta function (as an FIR filter).



(a) (b)

Figure 3.8: The reason behind the image border discrepancy. (a) The type of symmetric extension being used in the DFB (1-D symmetric extension). (b) The proposed type of extension for consistency with respect to the original image.

trivial to shift one subband with respect to its polyphase pair. Additionally, symmetric extension at this point is much simpler because it is the same as the rectangular case. The only change is to the filtering, but that remains essentially 1-D. It should be noted that this solution only works in the case of the biorthogonal polyphase form because of its structurally enforced perfect reconstruction.

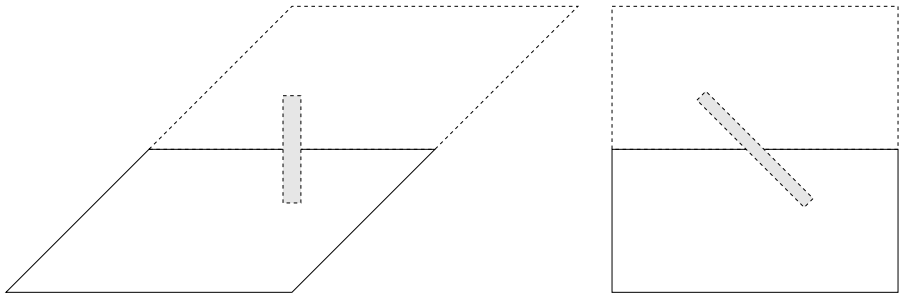


Figure 3.9: One solution is to conceptually perform post-sampling, filter in 1-D at angles corresponding to the post-sampling matrices and then invert the post-sampling.

3.1.5 Notation

In the first stage of the DFB, the subband that includes the ω_0-axis is arbitrarily chosen to start as the 0 subband, and the subband that includes the ω_1-axis is chosen as 1. Naming proceeds from the lower right to the upper right of the 0 subband and then from the upper left to the upper right of the 1 subband. This notation is demonstrated in Figure 3.10.

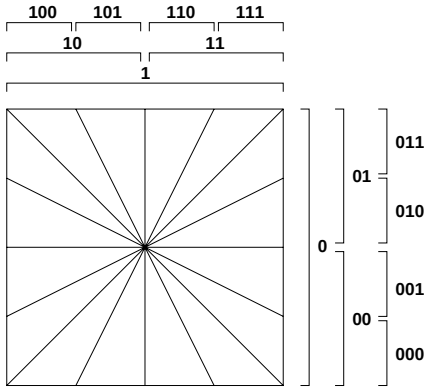


Figure 3.10: The labeling procedure of subbands of an eight-band DFB.

An instantiation of an OBDFB is signified by $\Delta_{S_O}^{N_D}$, where N_D denotes the number of directional bands in the directional decomposition, and S_O denotes the number of octave-band stages.

Furthermore, individual subbands may be referenced by their directional index first and then their octave-band position. For example, in Figure 3.1, the black subband pair would be referred to as $\Delta_2^8(011,\mathrm{LL},\mathrm{LH})$, and the gray subbands $\Delta_2^8(011,\mathrm{LH})$.

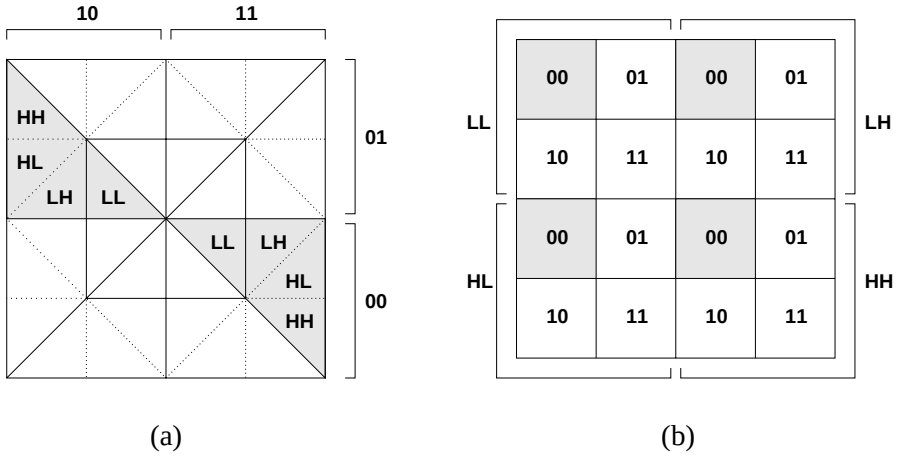


Figure 3.11: (a) A Δ_1^4 OBDFB and (b) the mapping of its coefficients; the highlighted directional subband in (a) has corresponding coefficients that are highlighted in (b).

3.1.6 Non-uniform Directional Resolutions

Another advantage of the OBDFB is its ability to change the angular resolution of the radially low frequency subbands in each direction. Because each low freqeuncy region has the same type t as the directional band it is in, showing that it is possible to change the angular resolution for one means it is possible for all of them.

The ability to increase angular resolution in lower frequency subbands is accomplished by adding two-band DFB analysis sections to the original OBDFB to subdivide the radially low frequency subbands directionally. The overall structure obviously retains the advantages of the OBDFB and DFB because it still uses the same two-band filter banks from the original DFB. Type assignment is useful here because the analysis block parameters are determined by the type of the low-freqeuncy subband which, in turn, was determined to be the same as the type of its corresponding directional band, a type that is retained even after the octave-band portion of the OBDFB (for low-frequency regions only). The block diagram may be seen in Figure 3.12, and the equivalent frequency partitioning may be seen in Figure 3.13. The frequency partitioning may also be compared to the one depicted in Figure 3.3 (the OBDFB decomposition before the increase in angular resolution).

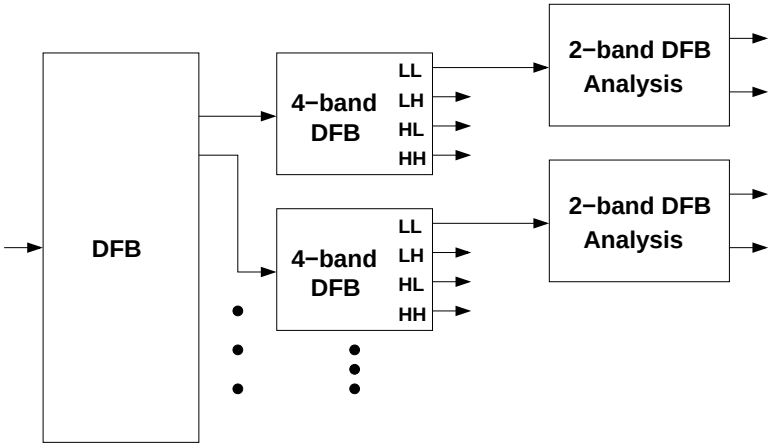


Figure 3.12: A block diagram for the increase of angular resolution in the low frequency region.

Sometimes it is desirable to reduce the angular resolution for radially lower frequency subbands, rather than increase the resolution (i.e., have fewer directional subbands for radially lower frequency regions). This is true for decompositions that attempt to emulate the cortex transform [92], for example. In this situation, the addition of a two-band DFB synthesis section effectively combines two

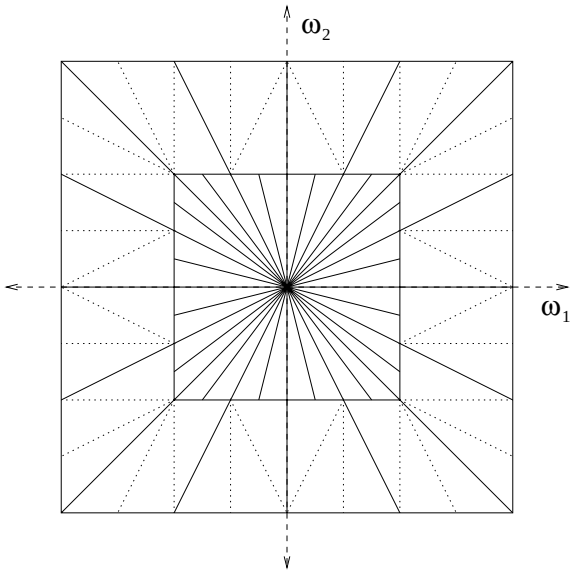


Figure 3.13: An example of an OBDFB with higher angular resolution in the low frequency region.

directional bands into its "original" lower angular frequency equivalent. The effect of performing this alteration on the OBDFB from Figure 3.3 may be seen in block diagram form in Figure 3.14 and in the frequency domain in Figure 3.15.

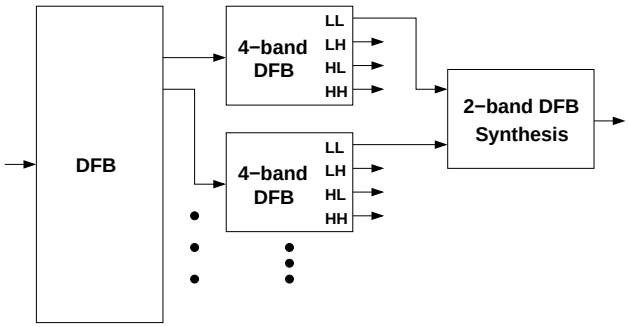


Figure 3.14: A block diagram for the reduction of angular resolution in the low frequency region.

This concept can be taken further by applying it successively to other low frequency subbands in combination with other octave-band structures or to completely reconstruct the lowest frequency region, for example. In the latter case, a system like the one shown in Figure 3.14 might be followed by another octave-band structure (thereby isolating low-frequency content from the newly merged

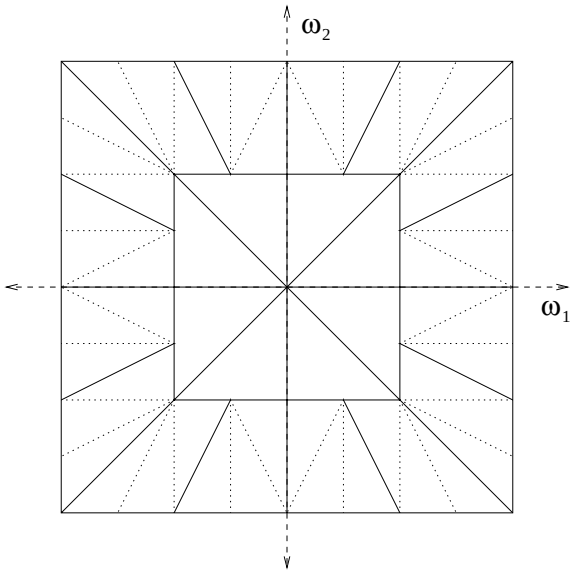


Figure 3.15: An example of an OBDFB with lower angular resolution in the low frequency region.

directional bands) and then reconstruct the lowest frequency subbands entirely. The equivalent decomposition is shown in Figure 3.16 which depicts such a transform.

A benefit of this subband combination method, besides its ability to allow for different angular resolutions, is that it is not restricted to the OBDFB. Transforms such as the CRISP–contourlet [56] (described in more detail later) may take advantage of this by combining directional subbands that are divided along the frequency axes, for example.

Finally, although the OBDFB provides a large amount of flexibility in dividing up the frequency space, there are situations where the type of split may not be defined until after the signal has been analyzed. For example, some decision criterion whose output determines whether a directional or octave-band split is used, may be applied during the image decomposition. This family of filter banks may be used for that purpose as well. The output of such a hybrid structure may be seen in Figure 3.17. It is accompanied by a block diagram of the filter bank components illustrated in Figure 3.18.

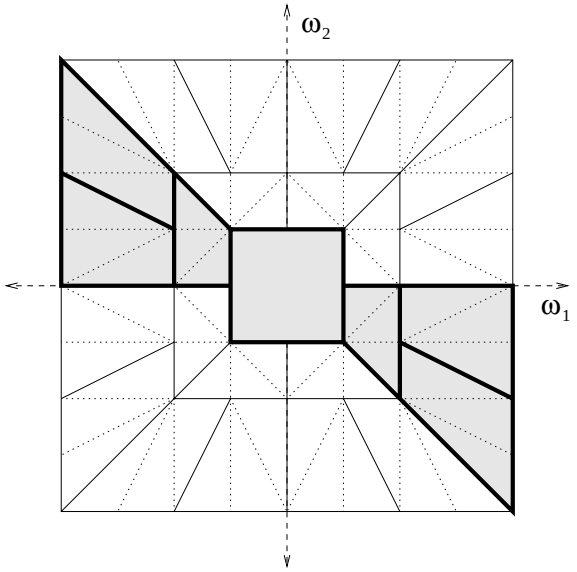


Figure 3.16: An OBDFB where the low frequency regions were first combined once to reduce the angular resolution, then fully reconstructed.

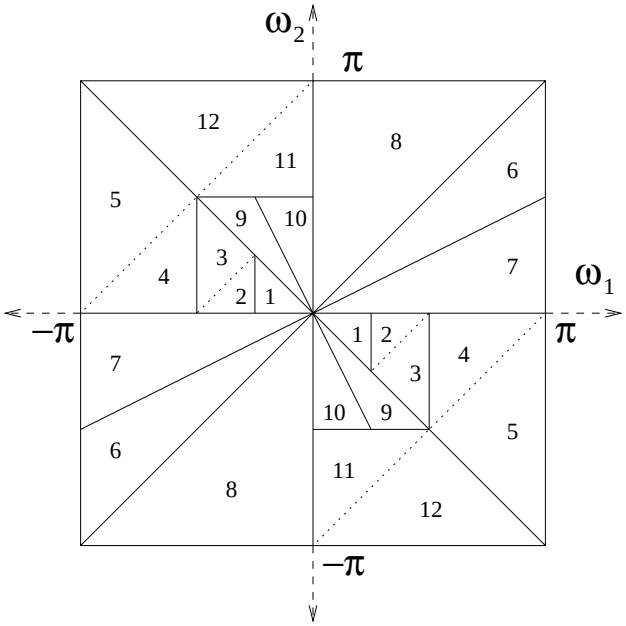


Figure 3.17: An example of an octave-band DFB decomposition whose structure appears in Figure 3.18.

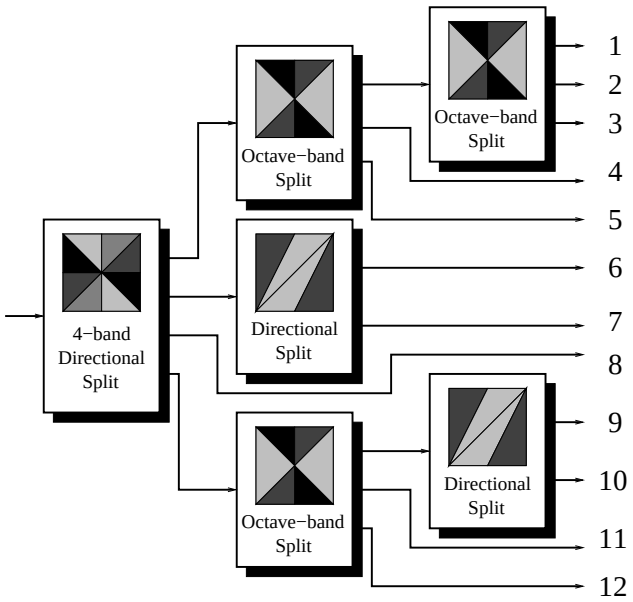


Figure 3.18: The associated diagram with the example shown in Figure 3.17.

3.1.7 Implementation

Actual implementation of the octave-band DFB successfully produced subbands with the desired passbands. For future reference, unless otherwise noted, the filter coefficients for $\beta(z)$ from Figure 2.8(c) were generated using the Parks-McClellan algorithm in Matlab:

```
beta = remez(11, [0, 0.8, 1, 1], [1, 1, 0, 0]);
```

resulting in the following coefficients:

$$\beta(z) = [-0.0144, 0.0272, -0.0526, 0.0972, -0.1929, 0.6300, \tag{3.3}$$

$$0.6300, -0.1929, 0.0972, -0.0526, 0.0272, -0.0144], \tag{3.4}$$

which were then modulated by $(-1)^n$ in order to produce hour-glass filters (instead of diamond filters) whose orientations were dependent on the quincunx matrix (and coset vector) used for downsampling.

A sample octave-band DFB subband displayed in the non-decimated frequency domain may be seen in Figure 3.19. The radially low frequency region was isolated from the rest of the directional band as described in Section 3.1.

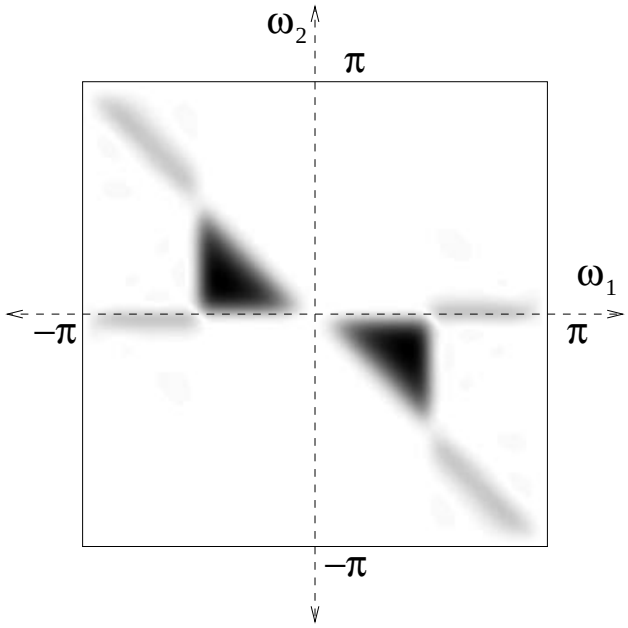


Figure 3.19: A sample frequency response of the analysis filters of a passband of the OBDFB Δ_1^4.

Residual shadows can exist in radially higher frequency passbands, along the boundaries of the directional band in Figure 3.19, because of the passband orientations relative to one another and the nonidealities of the filters used. The orientation of an arbitrary directional subband may be seen in Figure 3.20 in the decimated domain. If the four-band DFB were applied to perform an octave-band split, the radially low frequency area would be captured in one of its subbands. However, that particular subband would have boundaries adjacent to the radially high frequency areas as indicated in the figure. Those boundary effects cause the shadows in the non-decimated domain. The impact of these shadows can be minimized, however, simply by applying filters with tighter transition bands.

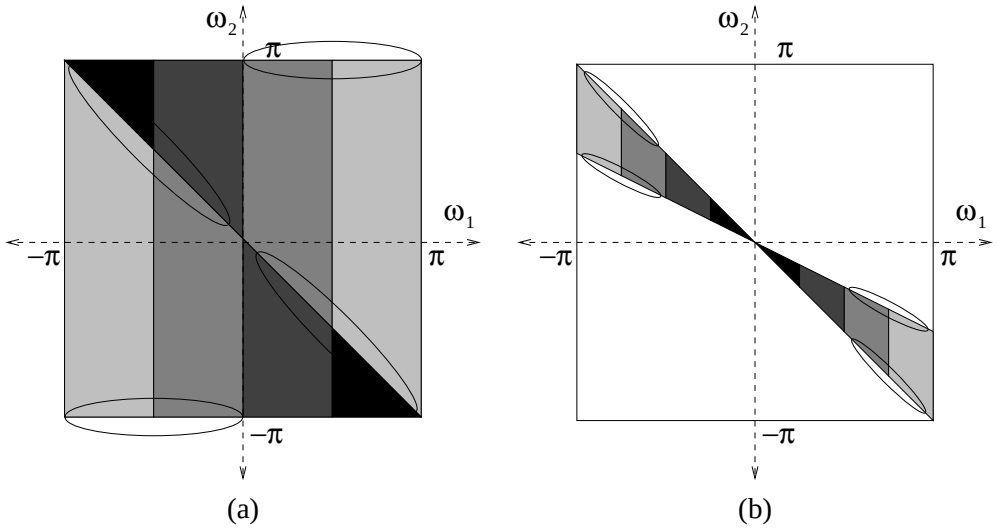


Figure 3.20: Shadow mapping of the OBDFB.

The DFB, when applied to an image, is effective in isolating directional information, as illustrated in Figure 3.21. The original *Cameraman* image is shown in (a) next to the subband depicted in Figure 3.19, shown in (b). As noted earlier, because the image in (b) is a subband in the decimated domain, frequency scrambling is present. The image was modulated, however, to restore DC to the center of the frequency cell. Strong directional information may be seen at the edges of the lower jacket as well as the left tripod leg. Additionally, the equivalent non-octave split directional subband is provided in (c), as well as the non-decimated version of (b) in (d) which is visually accurate because there is no aliasing present.

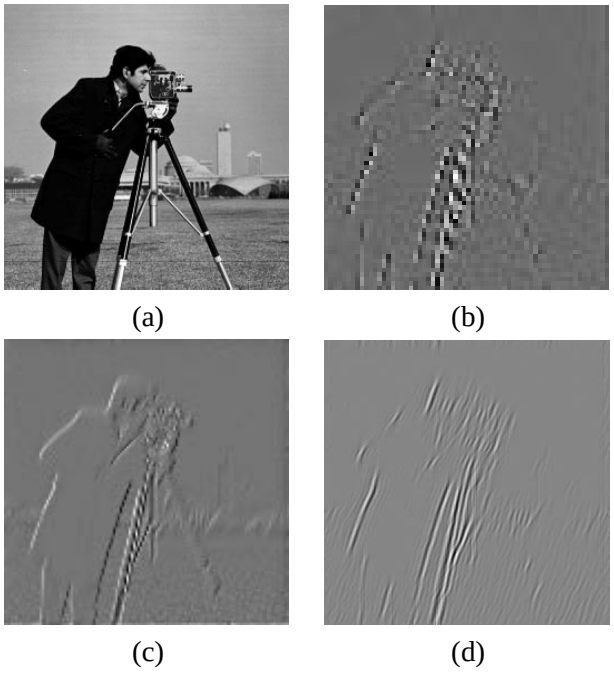
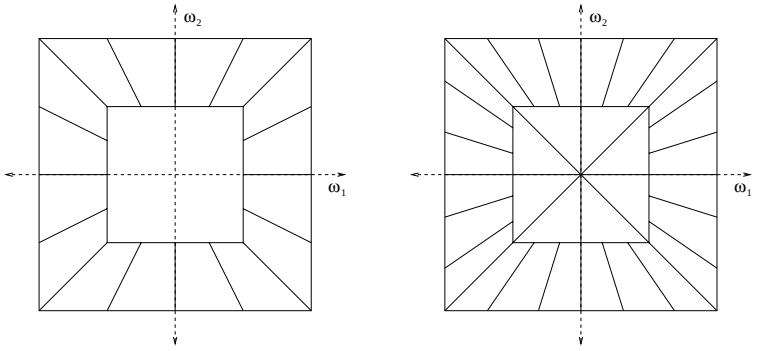


Figure 3.21: (a) The original *Cameraman* image (256 pixels $\times$ 256 pixels), (b) the subband (64 pixels $\times$ 64 pixels) associated with the passband shown in Figure 3.19, (c) the purely directional subband (128 pixels $\times$ 128 pixels), and (d) the non-decimated version of (b).

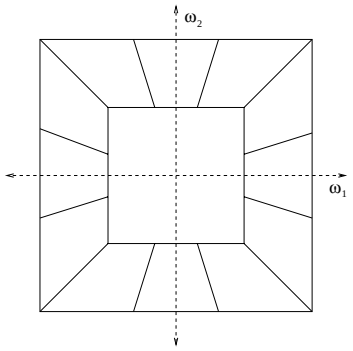
3.2 Other Decompositions

There are several other decompositions that divide the frequency space both angularly and radially. The transforms shown in Figure 3.22 were chosen because of their similarities to the OBDFB.



(a) Curvelets

(b) CRISP–contourlets



(c) Non-uniform DFB

Figure 3.22: Other decompositions with both angular and radial frequency discrimination.

Bamberger Pyramids were originally introduced in **??** and then later introduced in [21] as curvelets using pyramidal directional filter banks (PDFB) by Do and Vetterli. They are discussed at length in [22], and a sample decomposition is shown in Figure 3.22(a). Using a Laplacian pyramid, the DFB is applied to the radially high frequency subbands at the desired angular frequency resolution. This decomposition is flexible in that regard, and it is relatively simple to implement as both the Laplacian pyramid and DFB are computationally efficient. However, because this decomposition uses the Laplacian pyramid, it can increase the number of coefficients by up to 33%; i.e., it is

data expansive. For applications such as compression, this is extremely counterproductive.

More recently, Lu and Do introduced CRISP–contourlets in [56] (Figure 3.22(b)). Like the OBDFB, it is maximally decimated and can discriminate angularly and radially. And unlike the OBDFB, it is able to subdivide the radially high frequency subbands into directional subbands (not the three triangle-shaped regions produced by the OBDFB). Conceptually, this is achieved by decomposing the output subbands of a DFB by a non-uniform filter bank. The main problem with CRISP–contourlets is the implementation. Because the OBDFB was based on diamond (hourglass) filters, it has an extremely efficient implementation. CRISP–contourlets, however, rely on non-separable filters that do not (currently) have a useful polyphase form. Additionally, this decomposition does not solve the propagation of the DC problem inherent to the DFB.

A non-uniform DFB (nuDFB) was presented by Nguyen and Oraintara in [63] (Figure 3.22(c)). Unlike the other decompositions, the nuDFB has non-uniform subband geometries as well as non-uniform downsampling (the radially low frequency subband is downsampled by a factor of 2 whereas the radially high frequency directional subbands are downsampled by a factor of 8). This decomposition appears to have similar properties to the PDFB; however, it does not divide subbands along the major frequency axes and it is not data expansive. Unfortunately, in its current form, it does not provide perfect reconstruction and it does not have an efficient implementation. Another problem is its lack of flexibility. Because the other transforms are based primarily on the DFB, arbitrary numbers of directional subband (powers of two) are archievable. The nuDFB requires a minimum of six directional bands that require specialized filter banks if greater angular resolution is desired.

CHAPTER IV

TEXTURE SEGMENTATION

The ability to classify the pixels of an image based on their surrounding pixels and textural cues plays a significant role in a number of image processing applications such as medical image processing, remote sensing, and computer vision. Typically, texture classification is divided into two subproblems: feature extraction and classification. This chapter focuses on the feature extraction aspect of the texture classification problem. More explicitly, filter banks and wavelets are able to generate features from images by decomposing them based on different frequency regions. By processing each subband and stacking the outputs, feature vectors can be formed. But with so many different decompositions, criteria must be defined to narrow the possibilities.

The discrete wavelet transform (DWT) has enjoyed success in the field of texture classification. Randen and Husøy [71] did an extensive study of filter bank and wavelet decompositions used in texture classification. Their experiments primarily tested different filter types while varying subband geometries over only four patterns.

Another common approach to texture segmentation and analysis is to use a polar-logarithmic Gabor filter bank where each Gabor filter represents a separate frequency channel thereby mimicking the human visual system (HVS). This approach is well-established for the texture segmentation problem [41] and has been used in a wide variety of texture analysis systems and has been analyzed with respect to a number of aspects already [19, 35]. One of the major reasons to use Gabor filters is because of its emulation of the cortex transform [92]. A side-by-side comparison is shown in Figure 2.2 where the analogous subband structures are evident. Additionally, Gabor filters are relatively easy to implement and conceptually simple. For the purposes of this chapter, the Gabor filter system will be considered the gold standard.

Other decompositions have been used for the texture segmentation problem such as the directional filter bank (DFB) used by Rosiles in [81]. Although the DFB is unable to provide an

octave-band decomposition, its ability to discriminate angularly still allowed it to extract meaningful features in a maximally decimated and computationally efficient manner. The success of both the DFB and Gabor filters (along with a myriad of wavelet decompositions) provides a strong case for multichannel approaches to texture classification, and in particular, results obtained using the DFB indicate that there is not necessarily a large trade-off between computational efficiency and high classification accuracy.

As seen in [38], the octave-band directional filter bank (OBDFB) is able to capture similar directional and octave-band information as Gabor filters in a maximally decimated fashion in addition to having a computationally efficient implementation also requiring less memory than overcomplete decompositions. The purpose of this chapter is to provide a comparison of the features extracted by the OBDFB and other decompositions and to analyze actual texture segmentation results.

4.1 System Overview

There are many different configurations used for texture classification using filtering techniques; however, fundamental aspects of such systems are shared. System complexity was a factor in the choice of components, but the underlying units are typically the same from system to system. After the input image is decomposed, local energy estimates from each subband are used to create feature vectors which are then classified. Because the purpose of this research is to compare the OBDFB with other decompositions, the system was made as modular as possible such that switching between the decompositions was easily accomplished. As for Gabor filter bank parameters, the σ values were chosen to obtain even coverage of the spectrum given the number of directions and octave bands. Ultimately, the overall texture classification system from [71] was used with minor adjustments.

For energy estimation, the square-magnitude of the subband coefficients is smoothed in both the horizontal and vertical directions by using a Gaussian smoothing filter of the form

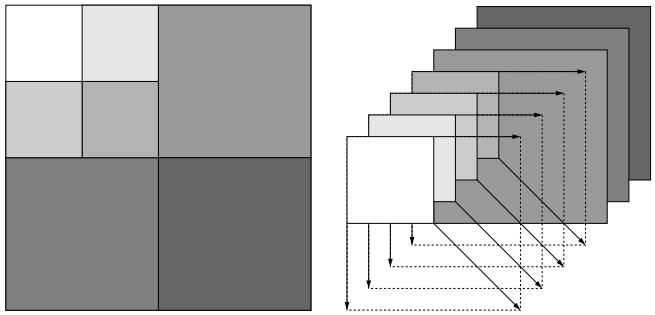
$$h_G[n] = \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left(-\frac{1}{2}\frac{n^2}{\sigma_s^2}\right), \tag{4.5}$$

where

$$\sigma_s = \frac{1}{2\sqrt{2}f_0}, \tag{4.6}$$

where f_0 is the normalized radial center frequency of the subband being processed [71]. Although there are different methods for choosing f_0 for each subband based on its frequency-domain geometry, the best empirical results came from choosing f_0 such that $\sigma_s = 8$, and that value was used for the experiments presented in this chapter. The $\log$ is then taken of the smoothed data (although different nonlinearities can be used). A filter providing a higher amount of smoothing was used with the OBDFB because of the manner of subband aliasing that occurs from decimation.

The decimated decompositions used typically had subbands that were of different sizes (i.e., subbands at different scales had different sizes). In order to generate features corresponding to every pixel of the original image, the decimated subbands were interpolated using zero-order hold as shown in Figure 4.23. To remain consistent between the decimated and undecimated transforms, smoothing was performed after the resizing to keep the computational comparison purely an aspect of the decompositions and not any post processing.



(a) (b)

Figure 4.23: (a) The subbands of a typical filter bank decomposition. (b) The reshaped and rearranged subbands creating feature vectors.

For the OBDFB and Gabor filter bank decompositions, the low frequency region was removed separately with a normalized cut-off frequency of $\frac{\pi}{16}$ and processed as its own dimension within the feature vector. This was necessary for both transforms as the Gabor filter bank implementation used excludes the low frequency region and the OBDFB tends to channel the DC energy into a single directional band thereby skewing the energy distribution among the directional bands. The other decompositions did not require this preprocessing step.

For classification, type 1 learning vector quantization (LVQ) used by Randen [71] was employed. The basic principle of LVQ is to move codebook vectors closer to training data points that

correspond to the same class and further away from training points from different classes during codebook generation/training. If m_i represents the codebook vectors, then some vector x is declared to be in the same class C as the nearest m_i which will be denoted as m_n. The training process is described for input $x[k]$ (where k denotes time) as

$$m_n[k+1] = \begin{cases} m_n[k] + \alpha[k](x[k] - m_n[k]), & \text{if } x, m_n \in C, \\ m_n[k] - \alpha[k](x[k] - m_n[k]), & \text{otherwise}, \end{cases}$$

where $0 < \alpha[k] < 1$ and $\alpha[k]$ may be constant or decrease monotonically with time, and the m_i for $i \neq n$ do not change. The number of codebook vectors used was 800, and the training data was kept strictly separate from the test data.

After codebook generation, each of the test vectors was assigned to a class according to comparison with the codebook vectors, and texture maps were generated.

4.2 Experiments

The following experiments were used to compare the performance of the OBDFB to that of other decompositions. Because of the modularity of the texture segmentation system, the only modification between tests was usually the change in decomposition.

4.2.1 Test Images

All of the test images are shown in Figure 4.24. They consist of five five-texture images, two 16-texture images, two 10-texture images, and three two-texture images (shown respectively in Figures 24(a)–24(l)). These are the same textures used in [71] and [75]. Because the overall systems are very similar, direct comparison with their results is possible.

4.2.2 Alternate Decompositions

The different subband decompositions used in addition to Gabor filters for comparison are shown in Figure 4.25. The overriding intent of using these decompositions was to compare subband geometries, not filter design techniques. Consequently, two-band filter banks similar to those used in the OBDFB were used to generate these decompositions with the same 1-D biorthogonal filter bank coefficients from Section 3.4 (12-tap, linear-phase, etc.). The main differences involve the

effective downsampling matrices as these filter banks act separably on the rows and columns and downsample accordingly.

As previously mentioned, the σ values for the Gabor filter banks were chosen to obtain even coverage of the spectrum given the number of directions and octave bands. Additionally, the Gabor filter passbands were chosen such that they correlated as closely as possible to the equivalent OBDFB passbands. In the case of the radially high frequency areas, a single Gabor filter was used to approximate the corresponding three OBDFB subbands. The exception to this was the implementation of a 6-direction Gabor filter bank as this is a common configuration for texture segmentation and was included for completeness.

4.2.3 Results

Two scoring methods were adopted for direct comparisons with [71] and [75]. The first score is percent inaccuracy which is simply the total number of incorrectly classified pixels divided by the total number of pixels. The second score was presented in [75] and is the average ranking with respect to each method (where a rank of 1 corresponds to the highest performance).

The average decimated transform results are shown in Tables 4.1 and 4.2 (the full results are shown in Tables 4.5 and 4.6). Although the OBDFB has the highest overall performance with 4 directional bands and 2 octave-band splits (followed closely by the 8 directional, 2 octave-band decomposition), the largest margin of improvement is in the 10-texture case. Although there are only two test cases here, the OBDFB more than halves the percent inaccuracy of the best performing ladder implementation.

Another interesting trend in the decimated case is the fact that inaccuracy tends to increase after 2 octave-band divisions. This could be due to a number of factors. First, the number of bands is increased with this extra split, and the classifier may not perform well because of this. The fact that the number of bands doubles when we move from 4 to 8 directional bands would seem to nullify this argument, but it could still be that the trade-off of useful information in this frequency range does not offset the increase in number of features. Second, the increase in number of subbands indirectly affects the size of the subbands where an extra octave-band split will reduce the smallest subband size by a factor of 4 (2 in each direction). Reducing the effective subband resolution only

Table 4.1: Average texture segmentation results (% inaccurate) using critically sampled transforms. The decompositions are labeled first by type, then by number of directions and number of octave bands, respectively.

Decomp.	Number of Textures				all
	5	16	10	2	
Filter Bank A	21.5	43.1	63.8	6.8	28.5
Filter Bank B	23.6	45.6	63.6	6.1	29.6
Filter Bank C	28.4	52.5	68.9	9.2	34.4
Filter Bank D	20.5	37.3	63.3	5.5	26.7
OBDFB 4,0	37.8	61.6	50.4	6.0	35.9
OBDFB 4,1	25.4	42.3	30.1	3.3	23.5
OBDFB 4,2	23.0	38.9	29.7	3.6	21.9
OBDFB 4,3	25.3	43.9	35.3	3.8	24.7
OBDFB 8,0	32.4	48.5	40.0	5.3	29.6
OBDFB 8,1	25.3	40.2	31.5	3.5	23.4
OBDFB 8,2	24.2	38.2	31.0	3.4	22.4
OBDFB 8,3	31.1	43.2	38.2	5.6	27.9

increases the percent inaccuracy. Nonetheless, a happy medium seems to exist at 2 octave-band splits for either 4 or 8 directional bands.

The average nondecimated transform results are shown in Tables 4.3 and 4.4 (the full results are shown in Tables 4.7 and 4.8). Technically, the OBDFB performs best in the nondecimated domain, but the difference between it and the Gabor filter bank is negligible. Because we are treating the Gabor filter bank as the gold standard, this confirms that the OBDFB can perform well in the undecimated domain and seems to indicate that the radial/angular frequency decompositions perform better than either alone.

With respect to the number of octave-band splits, the move from 2 to 3 seems to make little difference for both the OBDFB and the Gabor filter bank thereby suggesting that there is little information in that last frequency band. This also suggests that the degredation in accuracy in the decimated domain is due to the decimation of the subbands.

On average, the loss in accuracy from the best OBDFB scores in the full-rate domain to the decimated domain is 5%. The trade-off between accuracy and computational savings favors the computational savings in this case.

Table 4.2: Average texture segmentation ranks using critically sampled transforms. The decompositions are labeled first by type, then by number of directions and number of octave bands, respectively.

Decomp.	Number of Textures				all
	5	16	10	2	
Filter Bank A	3.0	5.5	10.5	5.3	5.3
Filter Bank B	5.6	7.0	10.5	4.7	6.4
Filter Bank C	9.0	10.5	11.0	9.0	9.6
Filter Bank D	3.4	3.5	10.0	5.3	5.0
OBDFB 4,0	11.6	12.0	8.0	9.0	10.4
OBDFB 4,1	6.0	6.5	2.5	6.3	5.6
OBDFB 4,2	4.0	3.5	2.0	6.0	4.1
OBDFB 4,3	6.0	7.5	4.5	6.3	6.1
OBDFB 8,0	8.4	9.0	6.0	8.0	8.0
OBDFB 8,1	6.6	4.5	3.5	4.7	5.3
OBDFB 8,2	4.6	2.0	3.0	4.0	3.8
OBDFB 8,3	9.8	6.5	6.5	9.3	8.6

Table 4.3: Average texture segmentation results (% inaccurate) using full-rate transforms. The decompositions are labeled first by type, then by number of directions and number of octave bands, respectively.

Decomp.	Number of Textures				all
	5	16	10	2	
Filter Bank A	21.4	37.8	30.1	4.5	21.4
Filter Bank B	20.5	38.9	27.1	4.5	20.7
Filter Bank C	22.9	41.6	28.8	3.6	22.2
Filter Bank D	19.4	34.9	26.1	2.5	18.9
Gabor 4,1	29.2	52.6	38.4	3.1	28.1
Gabor 4,2	19.3	38.7	30.9	2.1	20.2
Gabor 4,3	17.9	36.8	25.3	2.3	18.4
Gabor 6,1	26.6	45.2	33.9	2.6	24.9
Gabor 6,2	18.4	35.2	23.8	1.9	18.0
Gabor 6,3	17.6	34.6	23.8	2.4	17.7
Gabor 8,1	26.5	43.4	32.3	2.5	24.3
Gabor 8,2	18.1	33.8	24.1	2.1	17.7
Gabor 8,3	17.2	34.6	24.2	2.2	17.5
OBDFB 4,0	36.8	62.9	47.0	6.0	35.2
OBDFB 4,1	22.9	34.0	23.6	1.7	19.6
OBDFB 4,2	20.0	33.6	22.5	1.8	18.1
OBDFB 4,3	18.9	35.2	25.9	1.6	18.5
OBDFB 8,0	30.0	50.2	38.8	4.3	28.4
OBDFB 8,1	19.4	33.0	22.7	1.7	17.8
OBDFB 8,2	18.4	32.7	21.4	1.5	17.1
OBDFB 8,3	20.8	34.6	24.8	1.7	19.0

Table 4.4: Average texture segmentation ranks using full-rate transforms. The decompositions are labeled first by type, then by number of directions and number of octave bands, respectively.

Decomp.	Number of Textures				all
	5	16	10	2	
Filter Bank A	12.4	13.0	11.5	10.7	11.9
Filter Bank B	11.4	12.0	9.5	10.3	10.9
Filter Bank C	14.4	17.0	11.5	12.7	13.9
Filter Bank D	8.2	6.5	8.5	11.3	8.8
Gabor 4,1	19.2	19.5	19.0	14.0	17.9
Gabor 4,2	8.2	14.5	15.5	12.0	11.4
Gabor 4,3	6.6	11.5	11.0	13.3	9.8
Gabor 6,1	16.4	17.0	16.5	12.0	15.4
Gabor 6,2	7.6	8.5	8.0	9.7	8.3
Gabor 6,3	6.2	8.0	8.0	11.0	8.0
Gabor 8,1	16.0	16.0	14.5	11.3	14.6
Gabor 8,2	7.2	5.0	8.0	10.3	7.8
Gabor 8,3	5.6	7.0	7.5	10.7	7.4
OBDFB 4,0	21.0	21.0	21.0	19.3	20.6
OBDFB 4,1	11.8	6.0	8.0	10.0	9.8
OBDFB 4,2	8.8	6.5	6.0	10.7	8.4
OBDFB 4,3	8.2	6.5	10.5	8.3	8.3
OBDFB 8,0	17.0	19.0	16.5	11.3	15.8
OBDFB 8,1	6.6	6.0	6.5	7.7	6.8
OBDFB 8,2	7.0	4.0	4.0	6.0	5.8
OBDFB 8,3	11.2	6.5	9.5	8.3	9.4

4.3 Summary

The OBDFB provides a computationally efficient alternative to using traditional (and non-traditional) octave-band decompositions as well as Gabor filters for texture segmentation. Although the resulting features are not exactly the same, the most important structures are retained and its classification results in the decimated domain are comparable even to those of the full-rate Gabor filters, considering the significant reduction in computations.

Table 4.5: Texture segmentation results (% inaccurate) using critically sampled transforms. The decompositions are labeled first by type, then by number of directions and number of octave band splits, respectively.

Decomp.	Test Set												mean
	24(a)	24(b)	24(c)	24(d)	24(e)	24(f)	24(g)	24(h)	24(i)	24(j)	24(k)	24(l)	
Filter Bank A	13.5	22.7	29.5	24.2	17.6	42.7	43.4	61.5	66.0	14.4	1.9	4.1	28.5
Filter Bank B	14.1	26.7	34.0	21.0	22.5	47.1	44.2	62.8	64.4	12.6	1.3	4.6	29.6
Filter Bank C	17.3	33.4	38.4	23.2	29.7	50.8	54.1	74.9	62.9	18.2	2.2	7.2	34.4
Filter Bank D	13.9	19.1	32.0	16.5	20.8	36.2	38.3	67.1	59.6	9.6	1.5	5.5	26.7
OBDFB 4,0	16.1	35.9	55.4	49.0	32.8	56.7	66.6	46.9	53.8	2.1	5.3	10.8	35.9
OBDFB 4,1	11.3	31.3	33.6	28.3	22.5	36.2	48.4	20.9	39.3	2.1	3.3	4.6	23.5
OBDFB 4,2	10.3	28.1	29.6	28.3	18.9	33.1	44.8	26.4	33.1	1.9	3.0	6.0	21.9
OBDFB 4,3	9.4	32.2	36.3	25.6	23.0	40.2	47.6	31.6	39.1	1.9	2.5	7.1	24.7
OBDFB 8,0	12.8	24.5	47.0	44.5	33.1	36.5	60.4	29.7	50.4	2.3	2.1	11.5	29.6
OBDFB 8,1	13.2	27.5	32.3	30.3	23.2	33.3	47.1	23.2	39.9	1.4	1.9	7.1	23.4
OBDFB 8,2	10.7	26.4	31.4	31.0	21.4	32.3	44.0	27.0	34.9	1.4	1.4	7.4	22.4
OBDFB 8,3	15.3	32.7	42.3	35.9	29.2	35.7	50.7	35.1	41.4	2.4	2.5	11.9	27.9

Table 4.6: Texture segmentation ranks using critically sampled transforms. The decompositions are labeled first by type, then by number of directions and number of octave bands, respectively.

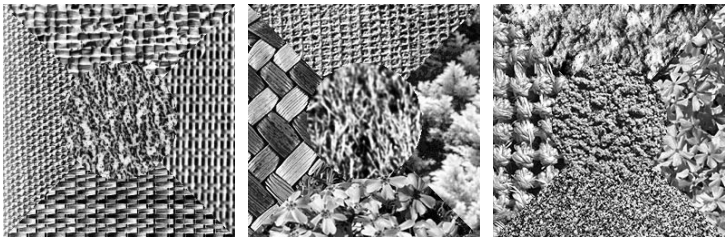
Decomp.	Test Set												mean
	24(a)	24(b)	24(c)	24(d)	24(e)	24(f)	24(g)	24(h)	24(i)	24(j)	24(k)	24(l)	
Filter Bank A	7.0	2.0	1.0	4.0	1.0	9.0	2.0	9.0	12.0	11.0	4.0	1.0	5.3
Filter Bank B	9.0	5.0	7.0	2.0	5.0	10.0	4.0	10.0	11.0	10.0	1.0	3.0	6.4
Filter Bank C	12.0	11.0	9.0	3.0	10.0	11.0	10.0	12.0	10.0	12.0	7.0	8.0	9.6
Filter Bank D	8.0	1.0	4.0	1.0	3.0	6.0	1.0	11.0	9.0	9.0	3.0	4.0	5.0
OBDFB 4,0	11.0	12.0	12.0	12.0	11.0	12.0	12.0	8.0	8.0	5.0	12.0	10.0	10.4
OBDFB 4,1	4.0	8.0	6.0	6.0	6.0	5.0	8.0	1.0	4.0	6.0	11.0	2.0	5.6
OBDFB 4,2	2.0	7.0	2.0	7.0	2.0	2.0	5.0	3.0	1.0	3.0	10.0	5.0	4.1
OBDFB 4,3	1.0	9.0	8.0	5.0	7.0	8.0	7.0	6.0	3.0	4.0	9.0	6.0	6.1
OBDFB 8,0	5.0	3.0	11.0	11.0	12.0	7.0	11.0	5.0	7.0	7.0	6.0	11.0	8.0
OBDFB 8,1	6.0	6.0	5.0	8.0	8.0	3.0	6.0	2.0	5.0	2.0	5.0	7.0	5.3
OBDFB 8,2	3.0	4.0	3.0	9.0	4.0	1.0	3.0	4.0	2.0	1.0	2.0	9.0	3.8
OBDFB 8,3	10.0	10.0	10.0	10.0	9.0	4.0	9.0	7.0	6.0	8.0	8.0	12.0	8.6

Table 4.7: Texture segmentation results (% inaccurate) using full-rate transforms. The decompositions are labeled first by type, then by number of directions and number of octave band splits, respectively.

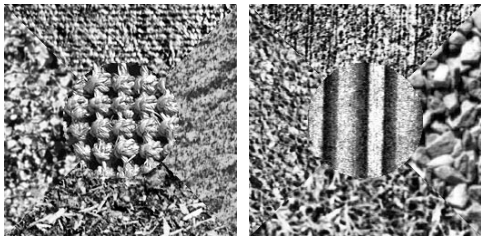
Decomp.	Test Set												mean
	24(a)	24(b)	24(c)	24(d)	24(e)	24(f)	24(g)	24(h)	24(i)	24(j)	24(k)	24(l)	
Filter Bank A	12.3	23.3	30.7	22.0	18.8	35.7	40.0	35.4	24.8	9.9	0.8	2.9	21.4
Filter Bank B	12.3	22.0	30.3	19.0	19.0	39.5	38.3	31.5	22.7	9.8	0.7	2.9	20.7
Filter Bank C	13.7	25.1	32.2	20.5	23.2	42.1	41.1	32.6	25.0	6.6	1.0	3.3	22.2
Filter Bank D	11.2	15.9	30.0	18.5	21.6	34.5	35.4	31.3	20.9	3.4	0.9	3.2	18.9
Gabor 4,1	13.8	31.9	31.1	39.2	30.3	47.8	57.4	37.8	39.1	1.4	5.2	2.5	28.1
Gabor 4,2	10.1	24.1	27.1	24.1	11.1	35.3	42.1	28.5	33.3	0.8	2.4	3.0	20.2
Gabor 4,3	6.2	21.3	27.8	24.2	10.0	33.5	40.1	18.9	31.7	0.6	2.5	3.7	18.4
Gabor 6,1	11.4	27.1	30.0	34.4	29.9	37.7	52.6	28.1	39.6	1.0	4.7	2.0	24.9
Gabor 6,2	9.6	19.5	27.4	24.8	10.7	30.4	39.9	17.7	30.0	0.5	2.0	3.2	18.0
Gabor 6,3	6.3	17.6	27.7	25.7	10.4	29.2	40.1	17.5	30.1	0.5	1.7	4.8	17.7
Gabor 8,1	10.5	26.9	30.2	34.5	30.5	36.1	50.7	25.7	38.8	0.9	4.5	2.2	24.3
Gabor 8,2	7.8	19.0	27.4	25.3	11.0	28.6	39.1	17.3	30.9	0.5	2.4	3.5	17.7
Gabor 8,3	6.6	17.4	27.8	23.7	10.5	29.3	40.0	16.9	31.5	0.5	1.8	4.4	17.5
OBDFB 4,0	17.7	38.5	52.4	44.4	31.2	59.2	66.7	42.8	51.3	1.8	4.7	11.5	35.2
OBDFB 4,1	9.9	27.4	27.3	24.4	25.4	29.4	38.5	19.4	27.8	0.9	2.6	1.7	19.6
OBDFB 4,2	10.5	26.0	26.1	23.4	14.0	30.9	36.3	20.4	24.7	0.9	2.0	2.5	18.1
OBDFB 4,3	11.3	20.8	27.4	20.7	14.2	35.3	35.2	23.2	28.6	0.7	1.2	3.0	18.5
OBDFB 8,0	11.3	25.0	41.4	42.5	29.8	41.7	58.7	27.8	49.7	0.5	1.8	10.5	28.4
OBDFB 8,1	9.2	20.2	27.2	20.5	19.6	29.6	36.3	18.5	27.0	0.7	1.5	2.8	17.8
OBDFB 8,2	9.9	18.4	28.0	19.4	16.3	29.5	35.9	17.1	25.7	0.6	1.5	2.4	17.1
OBDFB 8,3	12.8	18.7	29.8	22.4	20.2	33.3	36.0	21.7	28.0	0.9	1.4	2.8	19.0

Table 4.8: Texture segmentation ranks using full-rate transforms. The decompositions are labeled first by type, then by number of directions and number of octave bands, respectively.

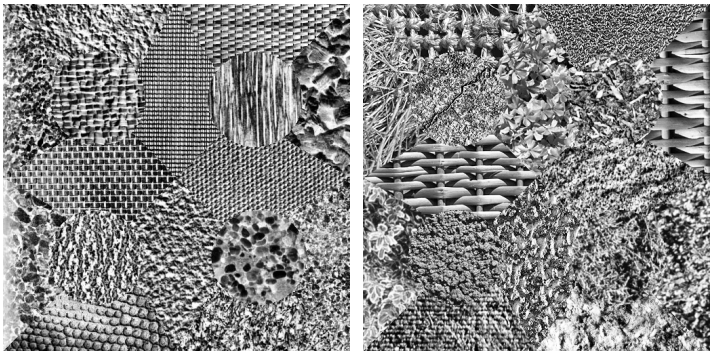
Decomp.	Test Set												mean
	24(a)	24(b)	24(c)	24(d)	24(e)	24(f)	24(g)	24(h)	24(i)	24(j)	24(k)	24(l)	
Filter Bank A	16.0	12.0	17.0	7.0	10.0	14.0	12.0	19.0	4.0	21.0	2.0	9.0	11.9
Filter Bank B	17.0	11.0	16.0	2.0	11.0	17.0	7.0	17.0	2.0	20.0	1.0	10.0	10.9
Filter Bank C	19.0	15.0	19.0	4.0	15.0	19.0	15.0	18.0	5.0	19.0	4.0	15.0	13.9
Filter Bank D	12.0	1.0	13.0	1.0	14.0	11.0	2.0	16.0	1.0	18.0	3.0	13.0	8.8
Gabor 4,1	20.0	20.0	18.0	19.0	19.0	20.0	19.0	20.0	18.0	16.0	21.0	5.0	17.9
Gabor 4,2	9.0	13.0	2.0	11.0	6.0	13.0	16.0	15.0	16.0	10.0	15.0	11.0	11.4
Gabor 4,3	1.0	10.0	9.0	12.0	1.0	10.0	13.0	7.0	15.0	7.0	16.0	17.0	9.8
Gabor 6,1	15.0	18.0	14.0	17.0	18.0	16.0	18.0	14.0	19.0	15.0	19.0	2.0	15.4
Gabor 6,2	6.0	7.0	7.0	14.0	4.0	7.0	10.0	5.0	11.0	2.0	13.0	14.0	8.3
Gabor 6,3	2.0	3.0	8.0	16.0	2.0	2.0	14.0	4.0	12.0	5.0	9.0	19.0	8.0
Gabor 8,1	10.0	17.0	15.0	18.0	20.0	15.0	17.0	12.0	17.0	13.0	18.0	3.0	14.6
Gabor 8,2	4.0	6.0	6.0	15.0	5.0	1.0	9.0	3.0	13.0	1.0	14.0	16.0	7.8
Gabor 8,3	3.0	2.0	10.0	10.0	3.0	3.0	11.0	1.0	14.0	3.0	11.0	18.0	7.4
OBDFB 4,0	21.0	21.0	21.0	21.0	21.0	21.0	21.0	21.0	21.0	17.0	20.0	21.0	20.6
OBDFB 4,1	7.0	19.0	4.0	13.0	16.0	4.0	8.0	8.0	8.0	12.0	17.0	1.0	9.8
OBDFB 4,2	11.0	16.0	1.0	9.0	7.0	8.0	5.0	9.0	3.0	14.0	12.0	6.0	8.4
OBDFB 4,3	13.0	9.0	5.0	6.0	8.0	12.0	1.0	11.0	10.0	8.0	5.0	12.0	8.3
OBDFB 8,0	14.0	14.0	20.0	20.0	17.0	18.0	20.0	13.0	20.0	4.0	10.0	20.0	15.8
OBDFB 8,1	5.0	8.0	3.0	5.0	12.0	6.0	6.0	6.0	7.0	9.0	7.0	7.0	6.8
OBDFB 8,2	8.0	4.0	11.0	3.0	9.0	5.0	3.0	2.0	6.0	6.0	8.0	4.0	5.8
OBDFB 8,3	18.0	5.0	12.0	8.0	13.0	9.0	4.0	10.0	9.0	11.0	6.0	8.0	9.4



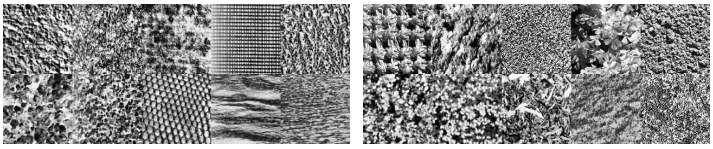
(a) (b) (c)



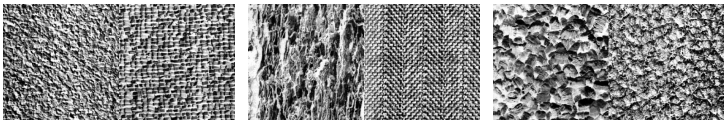
(d) (e)



(f) (g)

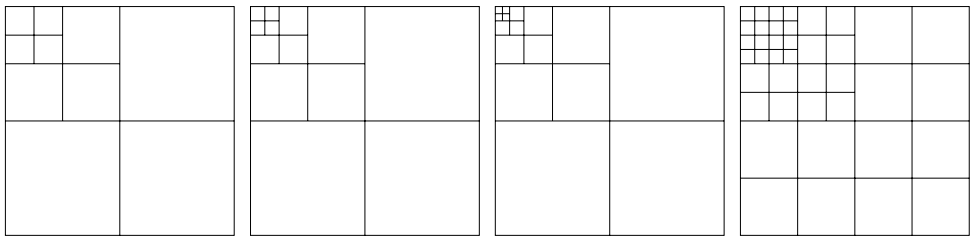


(h) (i)



(j) (k) (l)

Figure 4.24: Texture test images.



(a) Filter Bank A (b) Filter Bank B (c) Filter Bank C (d) Filter Bank D

Figure 4.25: Different subband decompositions.

CHAPTER V

HYPERSPECTRAL DATA

Hyperspectral images are gathered by sensors calibrated to different electromagnetic (EM) frequency bands to generate a set of images at different wavelengths. This collective set of images is referred to as a *hyperspectral data cube* made up of *hyperpixels* that are vectors formed by all of the pixels with the same spatial coordinates varying across the EM bands. Although image segmentation has been treated extensively in signal processing literature, segmentation of hyperspectral images has received less attention, and because of the high dimensionality, has its own unique challenges [51].

Texture information is used on a regular basis for segmentation and classification of a hyperspectral scene, i.e., dividing up the image into recognizable areas based on physical structures or surrounding materials. Texture classification methods typically fall into one of two categories: statistical approaches and filter-based approaches. Focusing on popular filter-based approaches, these tend to provide good results, but implementation issues may hinder their use with hyperspectral data. A computationally efficient method would help alleviate some of the already constraining computational complexity inherent in hyperspectral data.

In this section, we exploit texture information in order to enhance segmentation performance. We introduce a novel algorithm that incorporates texture features based on different configurations of the octave-band directional filter bank (OBDFB) originally discussed in [37]. Texture information and wavelets have been used in the past [52,59,74,84] for hyperspectral data; however, in this paper, it is used as a supplement to the spectral information in a simple, computationally efficient system.

A popular hyperspectral data set is the "AVIRIS" data set, short for "airborne visible/infrared imaging spectrometer." AVIRIS imagery uses 224 contiguous spectral channels with wavelengths ranging from 400 to 2500 nanometers. Data sets are typically very large because of the high dimensionality of the hyperpixels, and consequently, computational efficiency is a major factor in most applications regarding hyperspectral data.

5.1 System Overview

An overall system diagram for the texture augmentation is shown in Figure 5.26. First, texture features are extracted from a composite image of the input hyperspectral data. Then the original hyperpixels (or dimension-reduced versions) and texture feature vectors are run through maximum likelihood (ML) classifiers, independently, where the outputs are the ML "scores" normalized as probabilities. These scores are then combined and post-processed, and the final classification is performed. The baseline system follows the top branch in the figure and deals only with the spectral features of the hyperspectral data. There is the option to perform principal component analysis for dimension reduction as described in [31]; results are provided for both cases.

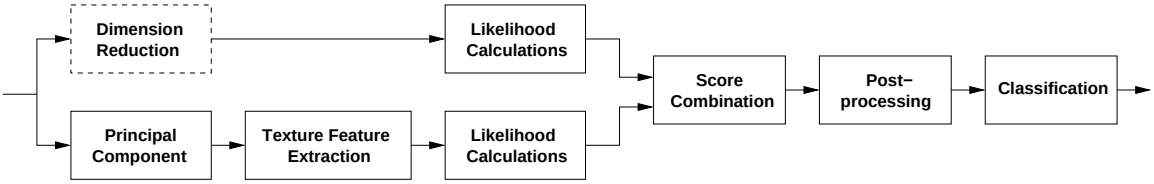


Figure 5.26: A block diagram outlining the texture augmentation system.

5.1.1 Dimension Reduction

In order to make the hyperspectral problem more tractable, dimension reduction is often implemented on the hyperpixels. This is done not only for computational savings but also to allow classifiers to generalize and be able to identify feature vectors outside of those provided in the training data. For high dimensional systems, more training data is necessary to prevent classifiers from simply "memorizing" training data for each class. Consequently, dimension-reduced data is a practical alternative. Although this dimension reduction can take many different forms, one of the most commonly used algorithms is principal component analysis (PCA). Using this technique, an optimal transform (in terms of decorrelating the input) is calculated and applied to the data to be reduced. First, the covariance matrix $\mathbf{C}$ is calculated from the training data. Then the eigenvalues are computed and ordered from largest to smallest as $\lambda_1, \ldots, \lambda_n$. In order to perform dimension reduction, only the first k largest values ($k < n$) are used. The corresponding eigenvectors are then used as the column vectors of a transform matrix $\mathbf{A}$, and $\mathbf{A}$ is applied to the hyperpixels $\mathbf{x}_{i,j}$ as

follows:

$$\mathbf{y}_{i,j} = \mathbf{A}^T \mathbf{x}_{i,j}, \tag{5.7}$$

where $\mathbf{y}_{i,j}$ are the dimension-reduced hyperpixels.

It should also be noted that this is the same procedure used to extract the principal component ($k = 1$) used in the texture feature extraction.

5.1.2 Texture Feature Extraction

The composite image used for texture feature extraction is constructed by projecting the hyperpixels onto the principal eigenvector associated with the overall covariance matrix of the training data as described in Section 5.1.1. This image contains much of the high frequency information associated with scene topography and the majority of the scene variance [88]. Because the texture features extracted can themselves be of high dimensionality for single 2-D images, the composite image is used instead of extracting features for each spectral frame which would cause excessive data expansion.

The extraction of texture features is accomplished using the method described in [71]. Initially, a subband decomposition is performed on the purely 2-D composite image. A local energy estimation is based on the log-magnitudes of the subband coefficients, and then smoothing is performed on each output subband using a Gaussian smoothing filter whose cut-off frequency is determined by the frequency characteristics of each subband. Finally, the processed output subbands are used to construct feature vectors by "stretching" and "stacking" the processed subbands as shown in Figure 4.23. The reshaping of the subbands is particularly necessary for outputs of the DFB and OBDFB as they can have unusual geometries due to the various downsampling matrices used. However, they ultimately have rectangular shapes (not necessarily square) when used in conjunction with the *post-sampling* from [65].

In particular for the directional decompositions, because multivariable Gaussian pdf's are assumed, it was necessary to generate texture feature vectors that were rotationally invariant. To this end, a post-processing technique is used after the original texture feature vector formation. In each texture feature, the energy associated with the subbands affiliated with a given direction is calculated for each directional wedge, and then the direction with the maximum energy is placed "first" in the

vector. Then, depending on which neighboring directional wedge has more energy, the directional wedges are ordered either clockwise or counterclockwise as shown in Figure 5.27. In this way, consistency with respect to mean and variance calculations is achieved, again, assuming single-modal Gaussian pdf's.

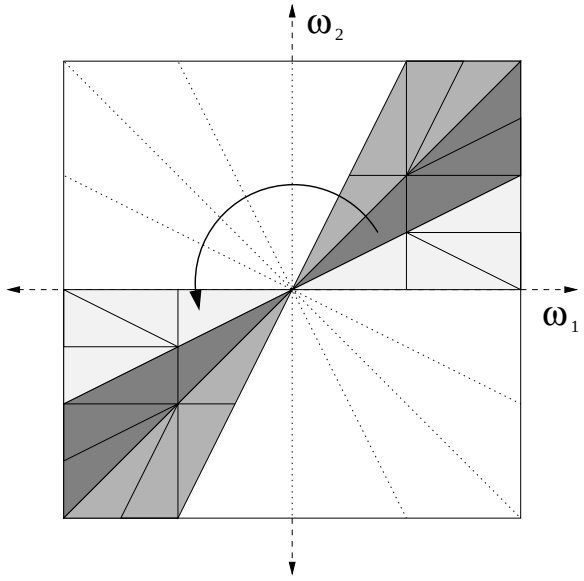


Figure 5.27: Directional feature construction where the darkest subband represents the one with the greatest energy. In this case, because the directional wedge in the counter-clockwise direction has more energy of the two neighbors, the coefficients will be taken in that direction.

5.1.3 Likelihood Calculation and Score Combination

The likelihood calculation blocks from Figure 5.26 refer to the likelihood calculations of each feature with respect to the different classes for both the raw hyperspectral data and the texture feature vectors. If the classes ω_n are assumed to have pdf's $p(x|\omega_n)$, then the likelihoods for each class are calculated by evaluating that classes's pdf by the vector in question. These likelihoods are retained for the N classes. It should be noted that the pdf's of each class are assumed to be single-modal Gaussians whose means and covariances are estimated from training data (discussed more in Section 5.2.3).

In order to use both sets of likelihoods, they need to be combined in a useful way. To that end, the likelihood measures are first converted to *a posteriori* probabilities via Bayes' theorem where the *a priori* probabilities $P(\omega_n)$ are assumed to be equal for all classes since no spatial information

is being used at this time. Then the product rule is used for the combination as described in [47]. Because the *a priori* probabilities are assumed to be equal for all classes (there is no reason to prefer one class to another at this point), the product rule is simplified; the decision rule is to assign the hyperpixel (and equivalent texture feature vector) to class ω_j for

$$j = \arg\max_n \prod_{i=1}^{M} P(\omega_n|x_i),$$

where $M = 2$ in this case because there are only two systems being integrated together.

5.1.4 Post-processing

The post-processing technique used on the combined scores is a multiplication of the scores (probabilities) in a particular window surrounding a given pixel. This has the effect of adding in spatial information to the decision criterion because now a pixel is affected by its neighbors. The window size was varied, and results for different sizes are given in Section 5.2. Formally, the technique is described by the following equation:

$$\mathbf{p}'_{i,j} = \prod_{(n,m)\in W_{i,j}} \mathbf{p}_{n,m}, \tag{5.8}$$

where $\mathbf{p}_{n,m}$ is the vector of probabilities associated with each class for a particular pixel to be classified at spatial location (n, m), $W_{i,j}$ is a window of size $M \times M$ at (i, j), and $\mathbf{p}'$ is the new output. This system is a means to include prior information that class labels between pixels should be consistent; i.e., a pixel is more likely to belong to the same class as its neighbors. The larger M is, the more neighboring pixels are "considered"; however, if M is too large, the windowing process will "blur" the scores too much and impair the ability of the classifier to detect different classes in spatially small regions.

5.2 Experimental Set-up and Results

The experiments used a hyperspectral data set consisting mostly of land features such as bog and fen as well as different types of vegetation such as jack pine and black spruce. A full description of the data set, algorithm and parameters, as well as classification results with discussion follow.

5.2.1 The Data Set

The AVIRIS data set used for training and testing was taken from the Boreal forest in Canada as part of the Boreas project and has a spatial resolution of 30 m. Three scenes were provided each taken at three different times of year: 04/94, 07/94, and 09/94. Of the original 224 AVIRIS spectral bands, 179 (168 in one case) were retained and used for both the baseline system (classification using just the spectral features) and to provide the composite image for texture classification.

Mean vectors and standard deviations of the hyperspectral data set *ojp* in 07/94 for both the raw data and for the dimension-reduced data may be seen in Figures 5.28 and 5.29, respectively. While some of the spectral signatures are relatively unique, the majority of them are quite similar. This similarity is expected for like classes such as white and black spruce, but unlike classes such as aspen and fen also exhibit similarities, making the classification problem more difficult. This, in turn, prompts the use of more features such as texture features to aid in the classification process.

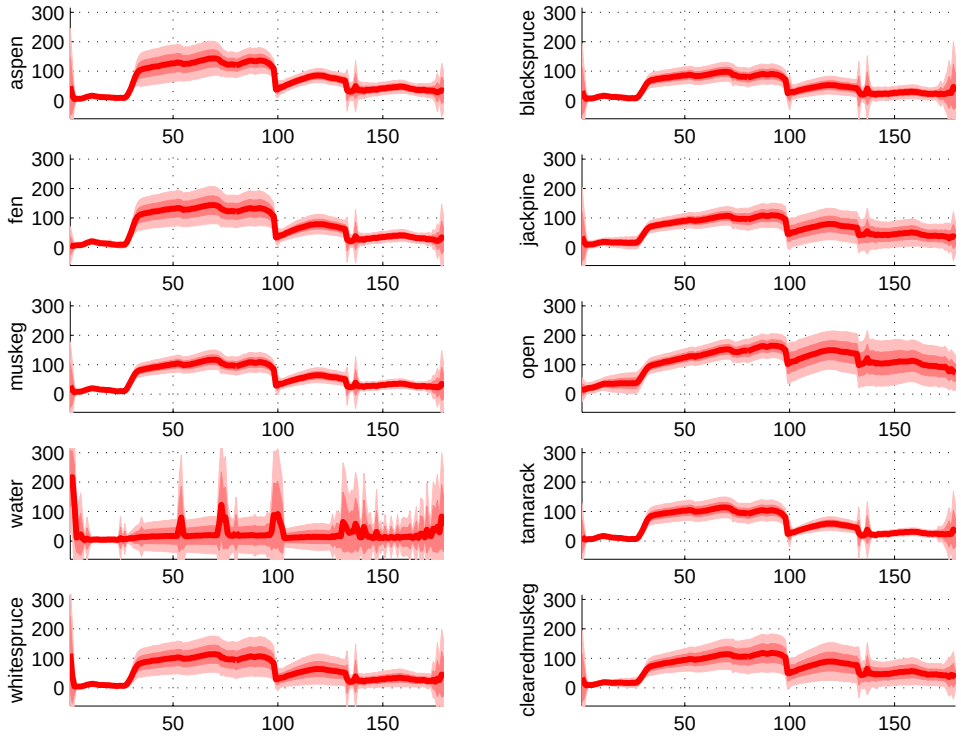


Figure 5.28: Mean vectors of different classes of a hyperspectral data set where each different shade of red denotes a standard deviation.

The ground truths of the two different scenes displayed in the results are shown in Figures 5.30

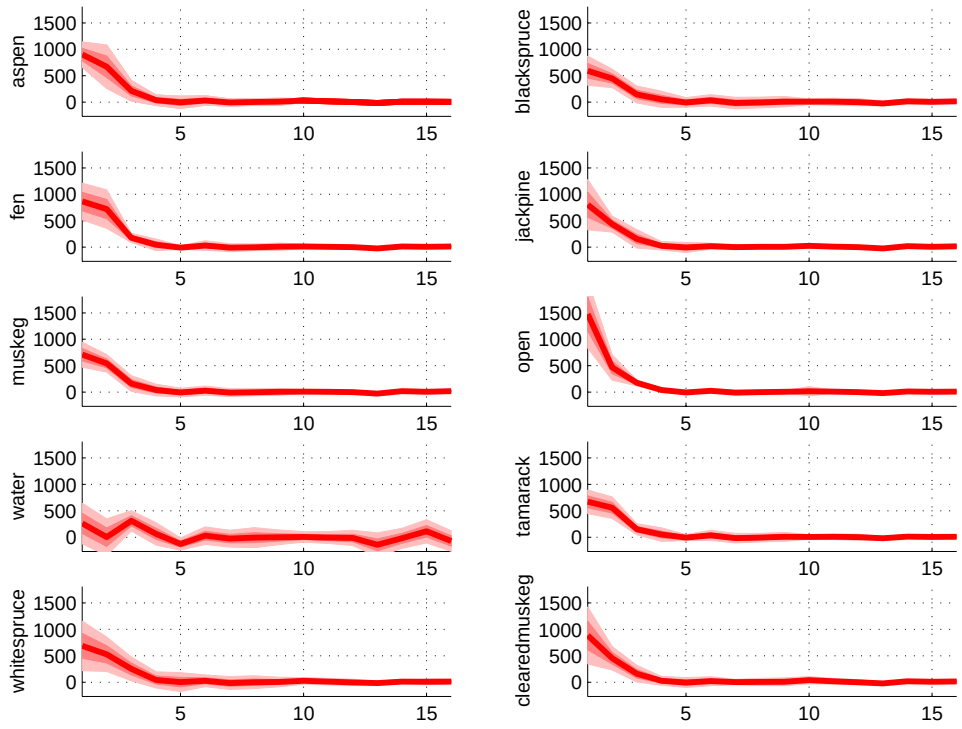


Figure 5.29: Mean vectors of different classes after dimension reduction where each different shade of red denotes a standard deviation.

and 5.31. Along with the ground truth, individual maps of each of the classes present in each scene are also presented. As seen, each class is not represented equally. In Figure 5.30, the classes present (and their abundance in the scene) are aspen (2.0%), black spruce (13.4%), fen (3.5%), jack pine (53.5%), muskeg (11.4%), open (12.0%), water (0.1%), tamarack (3.7%), white spruce (0.1%), and cleared muskeg (0.3%). In Figure 5.31, the classes present (and their abundance in the scene) are aspen (66.0%), black spruce (14.8%), fen (3.7%), jack pine (3.4%), water (2.7%), tamarack (2.1%), and white spruce (7.4%).

5.2.2 OBDFB Parameters

Different configurations of the OBDFB may be seen in Figure 5.32. Results using the first three are presented. The final two configurations produced subbands with very poor spatial resolutions, and consequently, they were not used during experimentation.

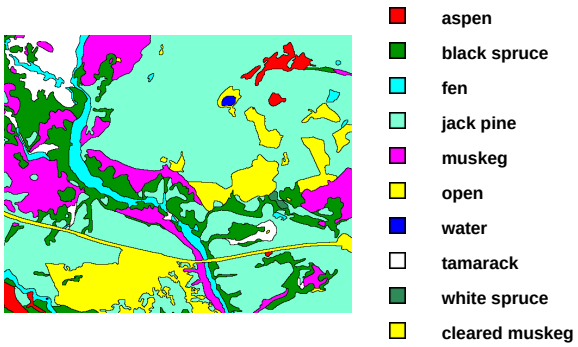


Figure 5.30: The ground truth for scene *ojp*.

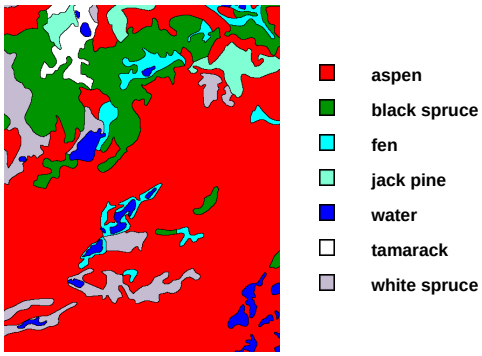


Figure 5.31: The ground truth for scene *oa*.

5.2.3 Training and Testing

The hyperspectral data sets were divided into 20 blocks of size 128×128 hyperpixels, and each scene was tested and trained on individually; i.e., one hyperspectral data set provided one set of results independent of the other hyperspectral sets. For $j, k \in [1, \ldots, 20]$, each block b_k was used for testing while training was performed on blocks b_j where $j \neq k$. In this way, we were able to maximize the use of the provided data. Additionally, the OBDFB was performed on a block-by-block basis. This also was to maximize the amount of training and testing data available as the region of support for the filters in the OBDFB would cause potential overlap of testing and training data.

The baseline system was implemented in two ways. The first way uses the raw hyperpixels, and the second way uses principal component analysis to perform dimension reduction from 179 (the

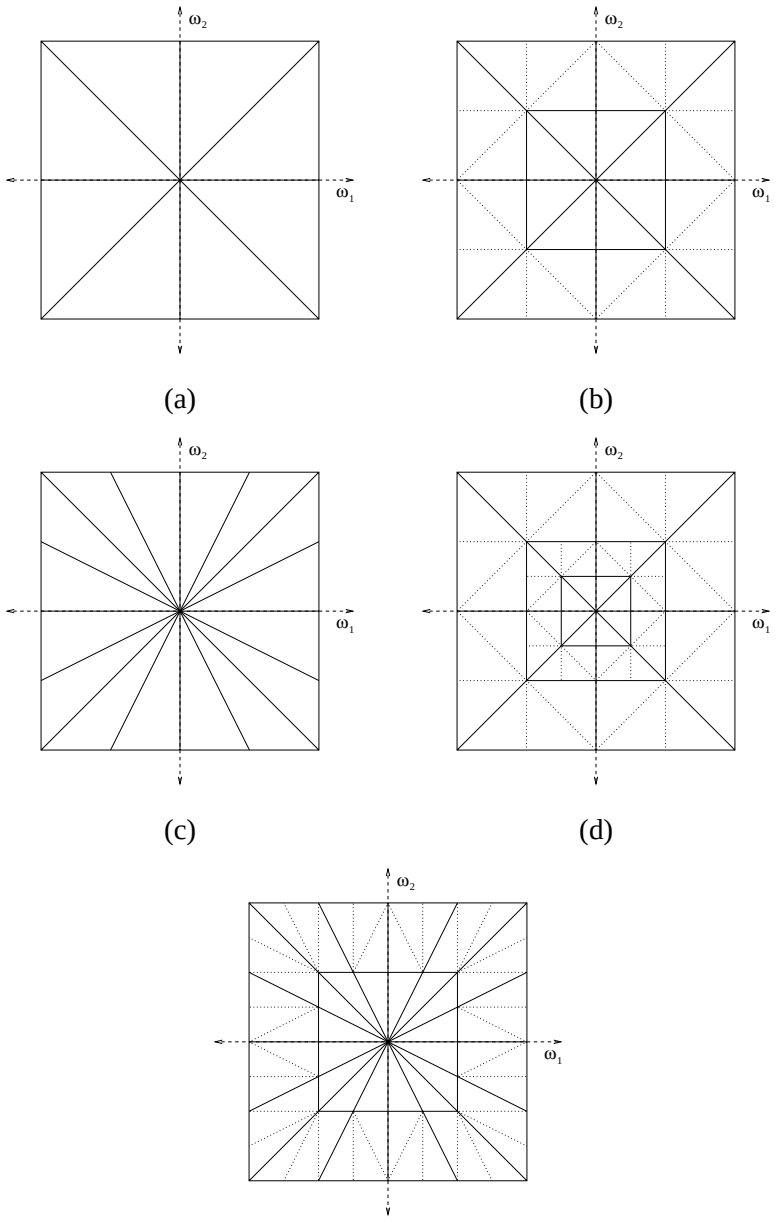


Figure 5.32: The different configurations of the OBDFB used in the experiments; (a) OBDFB 4,0, (b) OBDFB 4,1, (c) OBDFB 8,0, (d) OBDFB 4,2, and (e) OBDFB 8,1 (where the results from (d) and (e) were not presented because of the small spatial size of the subbands).

corrected raw hyperpixel length) to 16. In both cases, maximum likelihood scores are generated after approximating the parameters of the pdf's of the classes. Both sets of results are presented in the following section.

5.2.4 Results

By supplementing the classification of hyperpixels with the texture feature information, accuracy was generally increased. The results without using dimension reduction and using dimension reduction may be seen for two hyperspectral scenes in Tables 5.9 and 5.10, and sample output maps may be seen in Figures 5.33 and 5.34. The experiment was performed over 3 different sets of parameters for the OBDFB that were shown in Figure 5.32. The number of directions were either 4 or 8; in the case of 4 directional bands, 0 and 1 octave-band divisions were used; and in the case of 8 directional bands, no octave-band divisions were used. More octave-band divisions had been tested, but because of the maximal decimation, more divisions correspond to spatially smaller subbands that are sometimes too small to be usable. Additionally, overall results for all of the data sets are presented in Table 5.11, and confusion matrices for the *ojp* scene in 07/94 are presented in Tables 5.12–5.15 for the raw data, dimension-reduced data, and the equivalent texture-augmented systems.

By reviewing Tables 5.9 and 5.10, approximate increases in accuracy of 15% and 3%, respectively, is obtained for two different hyerspectral scenes. The texture augmentation seems to factor in at least some spatial information as it tends to have a smoothing effect. However, it is acting as more than a smoothing filter as seen in the maps shown in Figures 5.33 and 5.34 (i.e., the maps associated with the two scenes associated with the tables). For the *ojp* scene, the increase in classification accuracy comes from the correct classification of classes that are more abundant, such as jack pine, for example. In the center of the image, for example, we expect the OBDFB to add texture information for that large area of jack pine. For the *ojp* scene, only a slight increase in classification accuracy is shown in Table 5.10. However, the algorithm is still outperforming typical processing techniques that act solely on the scores generated by the baseline system. A simple processing technique after the baseline sytem would have misclassified some of the aspen pixels around the bodies of water. However, the texture augmentation system was able to correctly identify these regions as aspen.

The overall results for all the scenes over all the times are shown in Table 5.11. In general, the texture augmentation algorithms performed best for 07/94 and 09/94. Although the best results were shown for the *ojp* scene, there are slight improvements in the other scenes, and many of them are similar to the ones demostrated in Figure 5.34 where the improvements were not necessarily

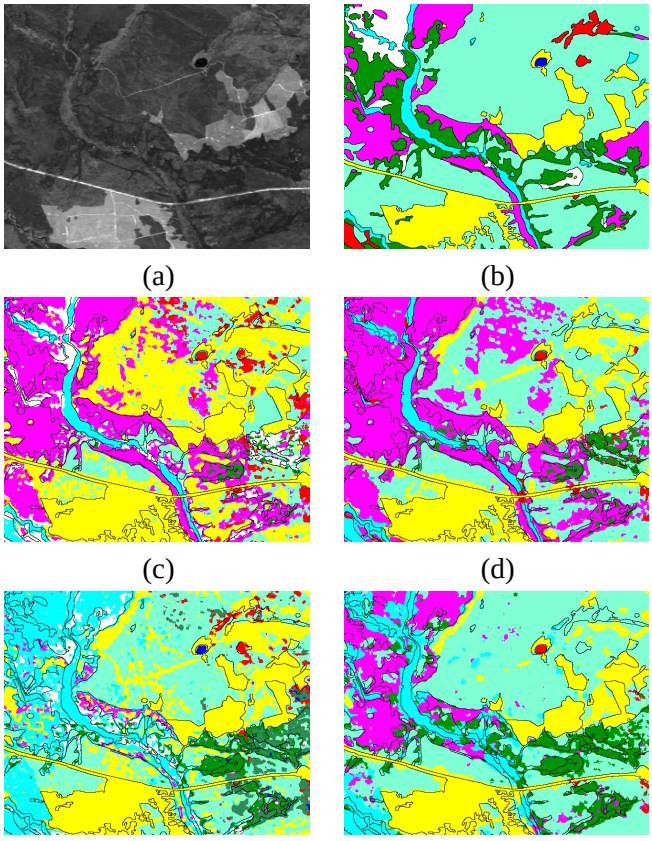


Figure 5.33: Maps for the *ojp* scene in 07/94 associated with (a) the principal component, (b) the ground truth, (c) the baseline system, (d) the OBDFB with 4 directions and 1 octave-band division, (e) the baseline system using dimension reduction, and (f) the OBDFB, all with a post-processing window size of 7.

shown in percent accuracy. Additionally, a similar conclusion to the one found in [71] is that the decompositions with more subbands (in the high frequency regions, in particular) appear to perform the best overall as the best scores were generally achieved by the 4 direction, 1 octave-band division case. This obviously was balanced with the subband spatial resolution issue discussed earlier.

As for the confusion matrices in Tables 5.12–5.15 for the raw data, dimension-reduced data, and the equivalent texture-augmented systems, the greatest performance boosts involved the most abundant classes. For example, comparing the classification accuracy of jack pine in Tables 5.12 and 5.14 (i.e., the baseline system without dimension reduction), we see a big increase since the baseline system had misclassified 48.8% of the jack pine as open. This is also evident in Figure 5.33 in the center of images (c) and (d). These large areas are excellent candidates for texture augmentation

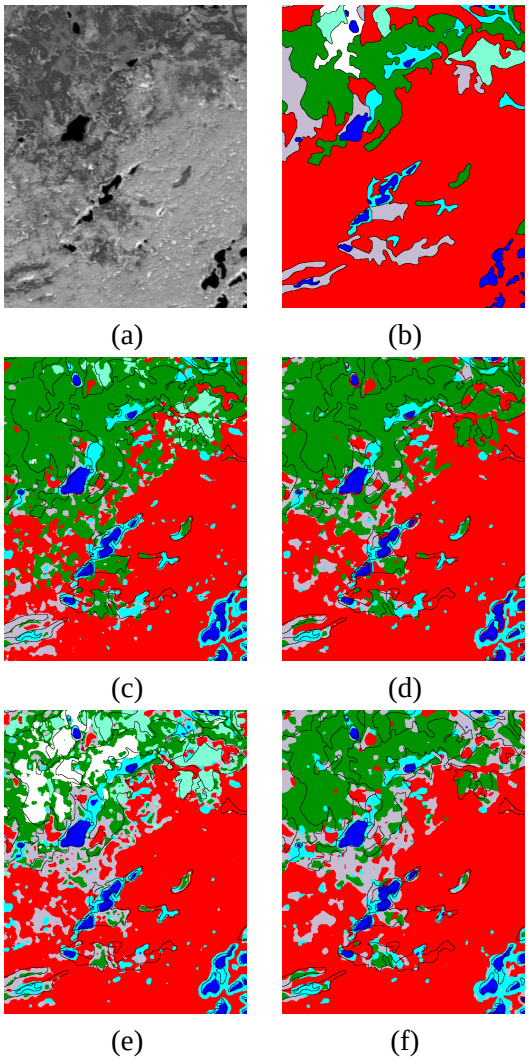
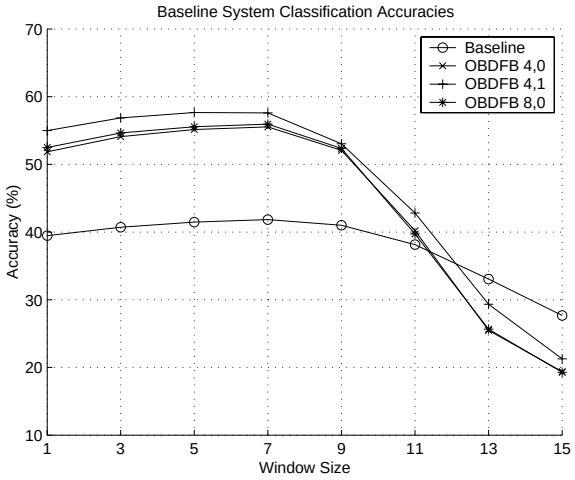


Figure 5.34: Maps for the *oa* scene in 07/94 associated with (a) the principal component, (b) the ground truth, (c) the baseline system, (d) the OBDFB with 4 directions and 1 octave-band division, (e) the baseline system using dimension reduction, and (f) the OBDFB, all with a post-processing window size of 9.

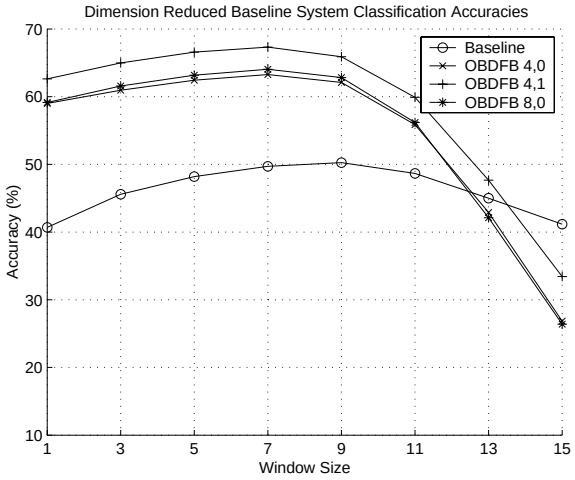
because they provide enough area for texture extraction to train and test well. At the same time, classes that are not abundant or that are spread thin spatially suffer in the sense that few pixels were identified as those classes regardless of accuracy. This is evident in Tables 5.14 and 5.15 where very few (not necessarily zero because of the precision) pixels were classified as water, tamarack, white spruce, or cleared muskeg. In any case, for this scene, the texture augmentation system was able to increase accuracy for the majority of classes.

In addition to performing the experiments over different OBDFB parameters, the window size

in the post-processing block was varied, and one set of results are shown in Figure 5.35. These were plotted for the *ojp* scene in 07/94 for all the configurations of the OBDFB tested.



(a)



(b)

Figure 5.35: (a) The accuracies with respect to the post-processing window size for the baseline system and the three OBDFB configurations used for texture augmentation. (b) The accuracies with respect to the post-processing window size for the baseline system using dimension reduction and the three OBDFB configurations used for texture augmentation.

It was also noticed that the texture features, when classified independently, had more trouble in the upper regions outside of the aspen area. This is somewhat intuitive as there are many transitions in that area and the region of support of the equivalent filters is relatively large. This addresses a potential hindrance as there were not particularly good instances of large sections of each class

to train over. There certainly are some areas that are entirely one class, but the fact that a maximally decimated filter bank is being used means that the resolution is getting lower for the smaller subbands.

5.3 Summary

In this paper, the OBDFB was used to extract texture information that, in turn, was used to increase classification accuracy of land features in hyperspectral scenes in a computationally efficient manner. The various scores for each class for both the baseline system and for the texture feature vectors were combined using the product rule in a maximum likelihood context. Future work may involve the decomposition of more composite bands with a more complex dimension reduction scheme for the resulting large feature vectors.

Table 5.9: Texture augmentation results (% accurate); the baseline results and the results using texture augmentation are given first, and then the corresponding results using dimension reduction. This is the *ojp* scene in 07/94 with window size 7. The classes present (and their abundance in the scene) are aspen (2.0%), black spruce (13.4%), fen (3.5%), jack pine (53.5%), muskeg (11.4%), open (12.0%), water (0.1%), tamarack (3.7%), white spruce (0.1%), and cleared muskeg (0.3%).

Decomp.	Classes										Overall
	aspen	black spruce	fen	jack pine	muskeg	open	water	tamarack	white spruce	cleared muskeg	
Baseline	24.5	5.4	72.7	27.9	85.4	98.3	0.0	32.2	0.0	0.0	41.8
OBDFB 4,0	29.6	8.2	74.9	52.6	85.8	98.3	0.0	32.1	0.0	0.0	55.5
OBDFB 4,1	4.0	13.0	80.4	57.9	87.2	98.3	0.0	0.0	0.0	0.0	57.6
OBDFB 8,0	21.4	8.6	75.1	53.7	85.9	98.3	0.0	28.2	0.0	0.0	55.9
Dim. Red.	23.9	21.4	86.2	55.9	10.4	97.1	75.4	9.2	26.1	4.9	49.7
OBDFB 4,0	33.4	32.2	84.7	70.8	44.8	97.1	0.0	17.1	0.0	1.8	63.3
OBDFB 4,1	1.2	42.5	85.7	75.7	56.2	97.0	0.0	0.0	0.0	0.0	67.3
OBDFB 8,0	22.9	34.0	83.4	72.9	42.6	97.0	0.0	14.5	0.0	0.8	64.0

Table 5.10: Texture augmentation results (% accurate); the baseline results and the results using texture augmentation are given first, and then the corresponding results using dimension reduction. This is the *oa* scene in 07/94 with window size 9. The classes present (and their abundance in the scene) are aspen (66.0%), black spruce (14.8%), fen (3.7%), jack pine (3.4%), water (2.7%), tamarack (2.1%), and white spruce (7.4%).

Decomp.	Classes							Overall
	aspen	black spruce	fen	jack pine	water	tamarack	white spruce	
Baseline	67.3	93.3	55.6	43.2	79.0	1.0	17.0	65.1
OBDFB 4,0	72.2	90.6	54.5	41.6	78.7	3.4	18.6	68.0
OBDFB 4,1	74.4	92.0	46.6	0.0	74.5	0.0	16.5	67.7
OBDFB 8,0	72.5	90.0	54.6	31.7	78.1	1.6	20.1	67.9
Dim. Red.	73.7	48.4	52.3	61.9	73.9	67.6	27.7	65.3
OBDFB 4,0	73.9	52.5	54.1	55.6	77.4	65.7	27.7	65.9
OBDFB 4,1	75.6	77.3	39.4	0.0	74.5	0.0	20.6	66.3
OBDFB 8,0	73.5	54.4	52.6	42.5	77.0	51.0	31.4	65.4

Table 5.11: Overall texture augmentation results (% accurate) for all three hyperspectral scenes at three different times using a window size of 7.

Decomp.	*oa*			*obs*			*ojp*		
	04/94	07/94	09/94	04/94	07/94	09/94	04/94	07/94	09/94
Baseline	76.2	64.6	64.0	68.9	71.0	68.2	28.2	41.8	29.7
OBDFB 4,0	74.0	67.8	65.8	67.9	70.9	68.1	30.3	55.5	34.7
OBDFB 4,1	72.3	67.3	66.4	65.5	69.1	66.1	34.3	57.6	36.0
OBDFB 8,0	73.5	67.6	65.9	67.3	70.1	67.3	30.6	55.9	35.3
Dim. Red.	65.9	64.8	61.9	61.3	63.7	60.4	37.2	49.7	28.9
OBDFB 4,0	64.4	65.6	64.3	60.8	64.5	61.2	40.2	63.3	39.9
OBDFB 4,1	66.0	66.3	67.1	60.5	64.5	61.3	48.9	67.3	43.9
OBDFB 8,0	64.4	65.1	63.8	60.8	63.6	60.3	41.0	64.0	41.2

Table 5.12: Confusion matrix of the 07/94 *ojp* data set using the baseline system. Each row of the table shows what percentage of that class was labeled as each of the different classes.

Class	aspen	black spruce	fen	jack pine	muskeg	open	water	tamarack	white spruce	cleared muskeg
aspen	24.5	0.0	22.2	12.5	0.1	30.1	0.0	10.6	0.0	0.0
black spruce	0.3	5.4	2.8	5.5	62.0	3.2	0.0	20.8	0.0	0.0
fen	0.0	0.0	72.7	1.6	7.8	3.4	0.0	13.9	0.0	0.6
jack pine	4.9	0.3	0.3	27.9	16.7	48.8	0.0	1.1	0.0	0.0
muskeg	1.3	1.9	0.3	3.1	85.4	1.7	0.0	6.2	0.0	0.0
open	0.6	0.0	0.0	0.8	0.2	98.3	0.0	0.1	0.0	0.0
water	84.7	0.0	8.5	3.6	0.0	3.1	0.0	0.0	0.0	0.0
tamarack	0.0	6.2	4.7	0.3	55.3	1.3	0.0	32.2	0.0	0.0
white spruce	54.1	3.0	0.0	1.5	20.7	11.3	0.0	9.4	0.0	0.0
cleared muskeg	3.2	2.6	11.0	2.3	16.1	61.6	0.0	3.2	0.0	0.0

Table 5.13: Confusion matrix of the 07/94 *ojp* data set using the baseline system and dimension reduction. Each row of the table shows what percentage of that class was labeled as each of the different classes.

Class	aspen	black spruce	fen	jack pine	muskeg	open	water	tamarack	white spruce	cleared muskeg
aspen	23.9	0.0	31.7	19.6	0.0	21.1	0.0	1.3	2.4	0.0
black spruce	0.0	21.4	38.7	7.7	3.9	2.2	0.4	18.0	7.5	0.1
fen	0.0	1.5	86.2	2.1	0.5	3.0	0.0	5.9	0.3	0.4
jack pine	2.1	1.7	8.5	55.9	0.2	26.2	0.1	0.2	4.8	0.3
muskeg	0.0	4.9	46.8	3.8	10.4	16.3	0.1	9.5	8.1	0.0
open	0.5	0.1	0.6	1.1	0.0	97.1	0.0	0.0	0.3	0.2
water	3.4	1.3	1.8	4.4	0.0	3.6	75.4	0.0	0.0	10.1
tamarack	0.0	10.8	77.0	0.5	1.1	0.6	0.2	9.2	0.5	0.0
white spruce	31.2	28.4	1.3	1.1	0.0	8.5	0.0	0.0	26.1	3.4
cleared muskeg	0.3	5.2	11.0	7.9	0.0	59.3	1.7	1.0	8.7	4.9

Table 5.14: Confusion matrix of the 07/94 *ojp* data set using the texture augmentation system. Each row of the table shows what percentage of that class was labeled as each of the different classes.

Class	aspen	black spruce	fen	jack pine	muskeg	open	water	tamarack	white spruce	cleared muskeg
aspen	4.0	0.0	30.0	45.7	0.5	19.8	0.0	0.0	0.0	0.0
black spruce	2.4	13.0	4.4	7.9	70.6	1.8	0.0	0.0	0.0	0.0
fen	3.9	0.2	80.4	1.8	10.8	2.8	0.0	0.0	0.0	0.0
jack pine	0.7	0.5	0.7	57.9	21.4	18.8	0.0	0.0	0.0	0.0
muskeg	1.2	4.5	0.4	5.3	87.2	1.4	0.0	0.0	0.0	0.0
open	0.5	0.0	0.0	1.0	0.1	98.3	0.0	0.0	0.0	0.0
water	91.7	0.0	7.0	0.8	0.0	0.5	0.0	0.0	0.0	0.0
tamarack	2.4	7.5	11.9	1.6	76.5	0.2	0.0	0.0	0.0	0.0
white spruce	22.6	18.2	0.0	21.4	25.4	12.4	0.0	0.0	0.0	0.0
cleared muskeg	4.8	5.5	11.4	7.5	18.4	52.3	0.0	0.0	0.0	0.0

Table 5.15: Confusion matrix of the 07/94 *ojp* data set using the texture augmentation system with dimension reduction. Each row of the table shows what percentage of that class was labeled as each of the different classes.

Class	aspen	black spruce	fen	jack pine	muskeg	open	water	tamarack	white spruce	cleared muskeg
aspen	1.2	0.4	35.1	59.4	0.0	4.0	0.0	0.0	0.0	0.0
black spruce	0.2	42.5	15.5	15.1	25.3	1.4	0.0	0.0	0.0	0.0
fen	0.1	4.0	85.7	4.5	3.0	2.7	0.0	0.0	0.0	0.0
jack pine	0.5	3.9	5.3	75.7	1.8	12.8	0.0	0.0	0.0	0.0
muskeg	0.2	12.7	14.0	13.8	56.2	3.1	0.0	0.0	0.0	0.0
open	0.2	0.2	0.3	2.4	0.0	97.0	0.0	0.0	0.0	0.0
water	88.9	4.1	1.3	3.9	0.0	1.8	0.0	0.0	0.0	0.0
tamarack	0.0	17.6	43.3	6.4	32.3	0.4	0.0	0.0	0.0	0.0
white spruce	1.7	64.3	3.2	20.7	0.0	10.0	0.0	0.0	0.0	0.0
cleared muskeg	1.8	13.2	4.2	22.9	6.0	51.9	0.0	0.0	0.0	0.0

CHAPTER VI

ULTRASOUND DESPECKLING

Ultrasonic imaging is noninvasive, low-cost, portable, and real-time. Because of these characteristics, it is becoming more and more popular as a medical diagnostic tool. Image quality is therefore extremely important to maximize its effectiveness for diagnoses. The speckle that is inherent to ultrasound imagery reduces image quality as well as the ability of a human observer to discriminate fine details [2]. Additionally, it is arguable whether or not speckle presents any relevant information at all.

Many different speckle reduction algorithms have been proposed; however, we focus our attention on real-time systems. Adaptive weighted median filtering, for example, has received much attention in the literature because of its good results and simple implementation [44, 55]. Wavelet thresholding [13, 14, 36, 82] has also received attention because of its computational efficiency and success in natural image denoising.

However, we propose a new despeckling system based on the octave-band directional filter bank (OBDFB). Advantages of using the OBDFB is superior performance over existing systems, real-time computational efficiency, and the ability to be appended to already existing systems as a post-processing step. The purpose of this chapter is to compare despeckling results generated by the OBDFB and other well-established algorithms.

6.1 Background

The term “speckle” refers to the granular pattern that appears on B-scans in ultrasound imagery. It is considered to be multiplicative noise (versus additive) and is caused by the scattering of the ultrasonic beam from microscopic tissue inhomogeneities. Speckle noise is a major limitation of image quality in ultrasound images. This can be a significant hindrance to medical diagnoses both by doctors as well as automated computer algorithms because it can obscure fine image details.

If $f[\mathbf{n}]$ represents the speckled ultrasound image, it can be modeled by

$$f[\mathbf{n}] = g[\mathbf{n}] \cdot e[\mathbf{n}], \tag{6.9}$$

where $g[\mathbf{n}]$ is the true image, and $e[\mathbf{n}]$ is the speckle noise. By taking the logarithm of both sides, the equation becomes

$$\log(f[\mathbf{n}]) = \log(g[\mathbf{n}]) + \log(e[\mathbf{n}]), \tag{6.10}$$

where the noise component is now additive, not multiplicative. This leads to the more common image denoising problem where the typical assumption is that the noisy image can be modeled as

$$y[\mathbf{n}] = x[\mathbf{n}] + \eta[\mathbf{n}], \tag{6.11}$$

where $x[\mathbf{n}]$ is the original image, and $\eta[\mathbf{n}]$ is additive white noise.

The purpose of ultrasound despeckling is to isolate the true image $g[\mathbf{n}]$ (or equivalently, $x[\mathbf{n}]$). This can be accomplished in a number of ways including wavelet thresholding [13, 14, 36], filter banks [77], nonlinear diffusion [1], and adapted weighted median filtering [44, 55].

6.2 Wavelet Thresholding

Wavelet thresholding involves the thresholding (typically soft-thresholding versus hard-thresholding) of computed wavelet coefficients. A separate threshold is computed for each subband and is applied separately. After thresholding, the inverse wavelet transform is applied.

There are two types of thresholding: hard-thresholding and soft-thresholding. Hard-thresholding is computed by

$$x_t = \begin{cases} x, & \text{for } |x| \geq T, \\ 0, & \text{for } |x| < T, \end{cases} \tag{6.12}$$

where x is the value to be compared to some threshold T, and x_t is the resulting value. Soft-thresholding is similarly computed by

$$x_t = \begin{cases} \operatorname{sgn}(x)(|x| - T), & \text{for } |x| \geq T, \\ 0, & \text{for } |x| < T. \end{cases} \tag{6.13}$$

It should be noted that hard-thresholding introduces a discontinuity at $x = T$ whereas soft-thresholding is continuous over all values of x. Soft-thresholding is usually chosen over hard-thresholding because of better visual quality. Thus, that is the method we use in our experiments.

The optimal threshold in denoising applications is defined to be the threshold which minimizes the expected squared error. An acceptable approximation to the optimal threshold is

$$T_B = \frac{\sigma_n^2}{\sigma_x}, \tag{6.14}$$

where σ_n is the standard deviation of the noise, and σ_x is the standard deviation of $x[\mathbf{n}]$. The standard deviation of the noise $\hat{\sigma}_n$ can be approximated by

$$\hat{\sigma}_n = \frac{\text{median}(|Y[i,j]|)}{0.6745}, \tag{6.15}$$

where $Y[i,j]$ are the coefficients of the HH_1 subband. Similarly, σ_x can be approximated by

$$\hat{\sigma}_x = \max(\sigma_y - \sigma_n, 0). \tag{6.16}$$

The value of σ_y can be approximated from the observed data.

6.3 Spatially Adaptive Wavelet Thresholding

Although wavelet thresholding as described above performs well, the introduction of a spatially varying threshold value for each subband has produced some of the best image denoising results to date [13, 14]. The spatially adaptive wavelet thresholding (SAWT) algorithm is based on the idea of computing spatially local statistics on coefficients with similar context.

Context values $Z[i,j]$ are computed for each coefficient in a subband by taking the mean of the absolute values of its eight neighbor coefficients and its parent coefficient. These context values are then used to determine the most similar coefficients and use those to approximate local statistics such as mean and variance. More formally, for a given subband, $Z[i,j]$ are computed for each coefficient coordinate (i,j) as previously described. Then these context values are ordered. Finally, for each context value, L values before and after it in the ordered list are used to approximate its variance $\sigma_n[i,j]$. Mathematically, this is described as follows:

$$\hat{\sigma}_x^2[i,j] = \max\left(\frac{1}{2L+1}\sum_{[k,l]\in B_{ij}} Y[k,l]^2 - \sigma_n^2, 0\right), \tag{6.17}$$

where the B_{ij} is the set of all indices whose coefficients fall within L positions of the coefficient at (i, j) in the ordered list of $Z[i, j]$. The purpose is to compute $\sigma_x[i, j]$ using data that is as statistically similar as possible thereby producing the most reliable threshold value.

Finally, the new threshold calculation is

$$T_B[i, j] = \frac{\sigma_n^2}{\sigma_x[i, j]}, \tag{6.18}$$

where it should be noted $T_B[i, j]$ varies with (i, j); i.e., the thresholding is performed on a coefficient-by-coefficient basis.

Originally, the SAWT algorithm also took advantage of the fact that shift-invariant transforms typically outperformed shift-varying transform for image denoising by using shift-invariant transforms and altering the context and local statistics calculations. For the purposes of performing real-time image denoising, however, we omit the use of shift-invariant transforms because they inherently require more system memory (because they are non-decimated) and more computation.

6.4 Experiments

The outputs using the SAWT algorithm and the OBDFB were compared to the original ultrasound images in addition to the outputs using several competing methods. In order to make fair comparisons, alternate algorithms were limited to systems that could be expected to perform realistically in real time. The different images were compared in subjective AB tests.

6.4.1 Alternate Algorithms

In order to test the performance of the OBDFB for ultrasound speckle reduction, a handful of algorithms was chosen for comparison. The algorithms used were limited to real-time or near real-time systems.

6.4.1.1 SAWT using a Biorthogonal Filter Bank

The first and most obvious algorithm to compare the OBDFB against is the SAWT algorithm using a traditional octave-band decomposition. In this case, a biorthogonal filter bank was used for implementation at four scales of decomposition similar to the configuration in [36]. This amounts to the original SAWT algorithm using a decimated biorthogonal filter bank.

6.4.1.2 *Adaptive Weighted Median Filtering*

Adaptive weighted median filtering (AWMF) [44, 55] is used for comparison because it is the most used real-time model due to its simplicity, computational efficiency, and good performance. As expected of median filtering in general, AWMF tends to do a good job of removing speckle noise and retaining edges in image structures.

The simplest explanation of weighted median filtering involves the weighted median in one dimension. The weighted median of a sequence $\{X_i\}$ is defined as the median of the extended sequence formed by taking each term X_i and repeating it w_i times where $\{w_i\}$ are the corresponding weights. If the weight is negative, the value is negated and then repeated by a factor of $|w_i|$. For example, if $w_1 = 2$, $w_2 = 3$, and $w_3 = -2$, and we wish to take the weighted median of the sequence $\{X_1, X_2, X_3\}$, then the equivalent calculation becomes

$$y_{\mathrm{WM}} = \mathrm{median}(X_1, X_1, X_2, X_2, X_2, -X_3, -X_3). \tag{6.19}$$

Extending this to multiple dimensions is analogous to the extension of one-dimensional filtering to multi-dimensions.

For a window size of $N \times N$, the adaptive weights for pixel position (i, j) of the image is given by the following equation:

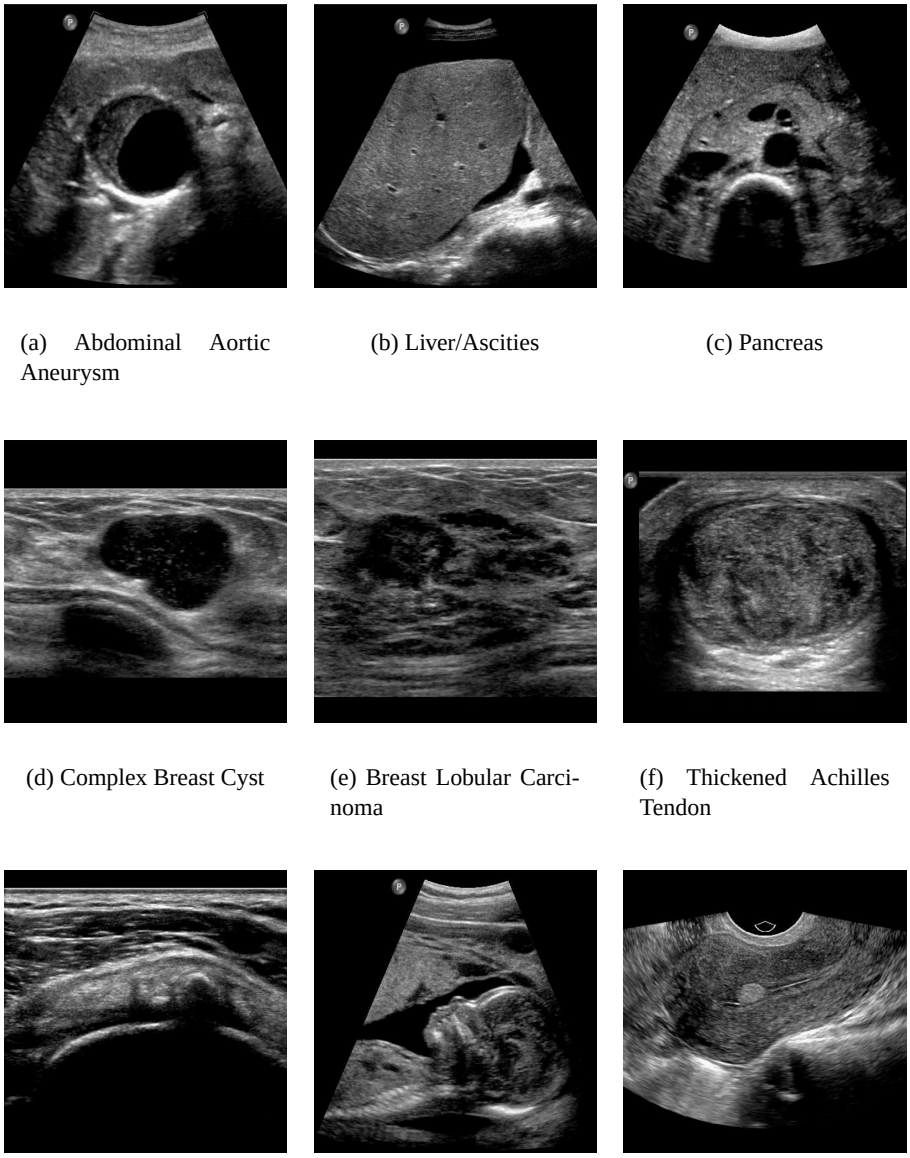
$$w_{m,n} = w_0 - g\sqrt{(i-m)^2 + (j-n)^2}\,\sigma_{i,j}^2/\mu_{i,j} \tag{6.20}$$

where m and n each vary from $-\frac{(N-1)}{2}$ to $\frac{(N-1)}{2}$, w_0 is the central weight, g is a scaling factor, and $\sigma_{i,j}^2$ and $\mu_{i,j}$ are the local variance and mean, respectively, of the window centered around point (i, j).

For the experiments, window size $N = 9$, $w_0 = 99$, and $g = 5$. Additionally, negative coefficients were set to zero in accordance with other algorithms.

6.4.2 Results

The images used for the subjective test are shown in Figures 6.36 and 6.37 taken from [67] with permission. These images were chosen because they are representative of the different types of ultrasound imagery used in medicine.



(a) Abdominal Aortic Aneurysm

(b) Liver/Ascities

(c) Pancreas

(d) Complex Breast Cyst

(e) Breast Lobular Carcinoma

(f) Thickened Achilles Tendon

(g) Calcific Tendinopathy of Rotator Cuff

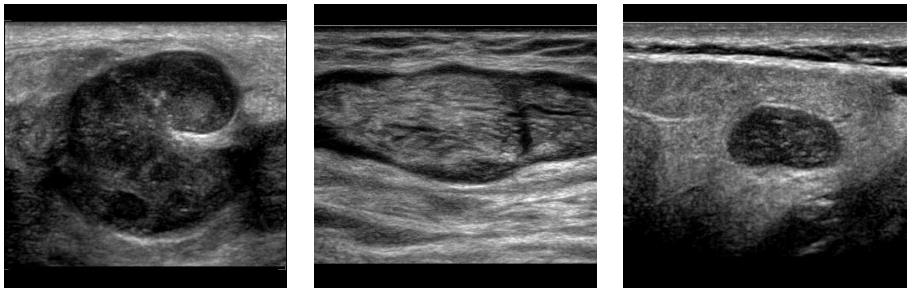
(h) 17-week Fetal Profile

(i) Endometrial Polyp

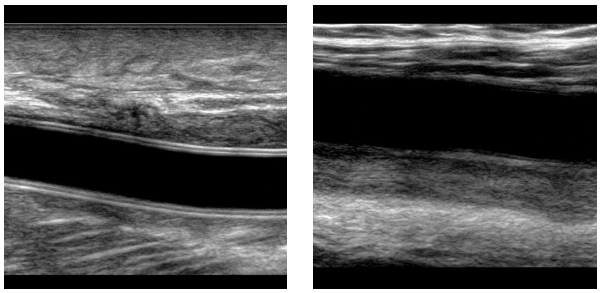
Figure 6.36: First set of ultrasound test images taken from [67] with permission.

Only the best performing set of parameters was used for each method; although the different parameters were varied primarily through published values. For the AWMF, window size $N = 9$, $w_0 = 99$, and $g = 5$ were used and negative weights were set to zero. The decomposition shown in Figure 25(c) was used for the standard SAWT algorithm which is equivalent to four octave-band splits. As for the OBDFB, 8 directions and 3 octave-band splits were used.

(a) Neonatal Head (b) Neonatal Head Bleed (c) Pediatric Liver



(d) Parotid Gland Mass (e) Lymph Node (f) Parotid Gland Adenoma



(g) Gortex®Graft (h) Superficial Femoral Vein Thrombosis/Artery

Figure 6.37: Second set of ultrasound test images taken from [67] with permission.

The subjective tests were set up as AB tests where two images were presented at a time. All combinations of the original and the outputs processed by the AWMF, standard SAWT, and the OBDFB were used on the 17 test images. The test was administered on the Internet for simplicity and had 47 respondents.

The results of the subjective tests are presented in Table 6.16. Sample results of the different algorithms are shown in Figure 6.38 and in Figure 6.39 with zoom. A strong trend in the results was

Table 6.16: Ultrasound subjective test results in percent preferred.

Method	Original	AWMF	SAWT	OBDFB	Overall
Original	n/a	87.6	51.1	81.0	73.2
AWMF	12.4	n/a	14.0	26.9	17.8
SAWT	48.9	85.6	n/a	81.7	72.2
OBDFB	19.0	73.1	18.3	n/a	36.8

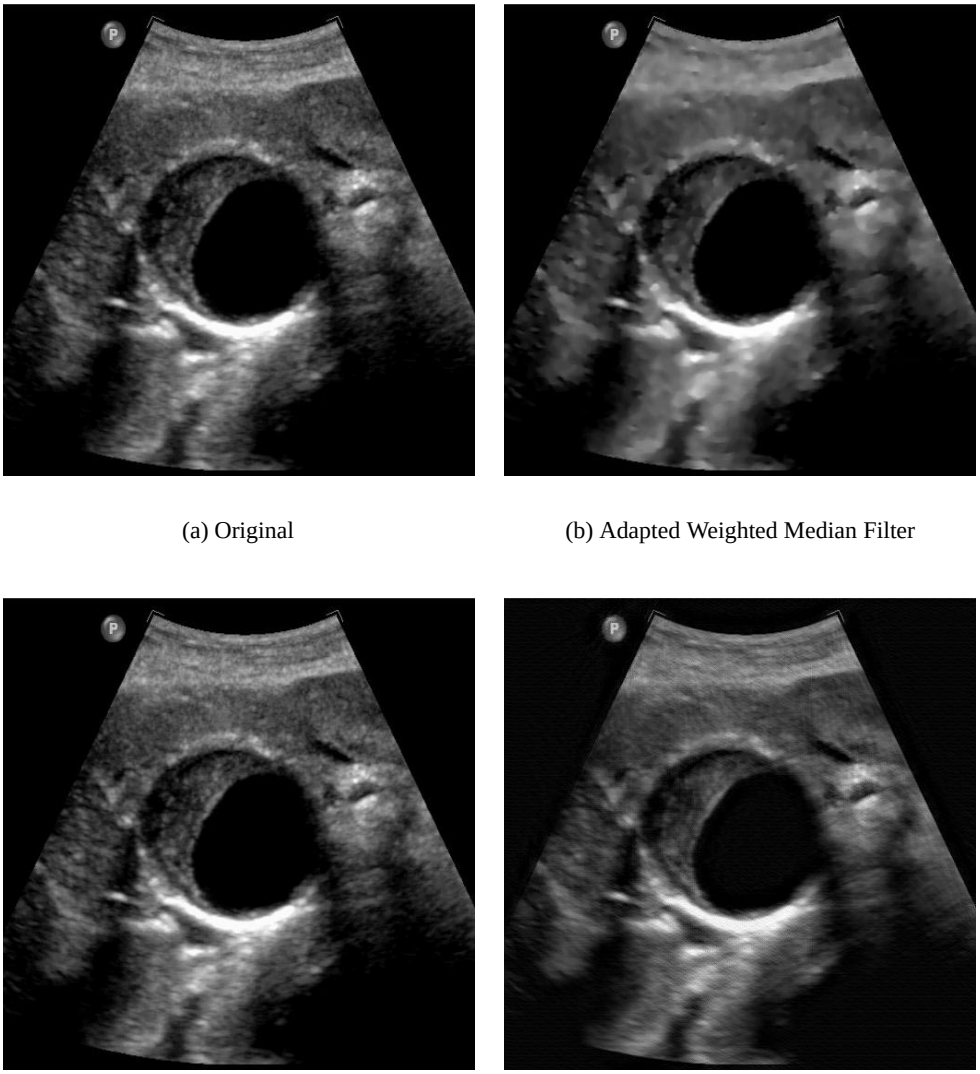
that the standard SAWT algorithm produced output that was very close to the original input. This is confirmed in the table where the original and the SAWT output are nearly 50%. The scaling to the threshold was increased and decreased on a sample set, but ultimately it was left at the original value as reported in [44]. Because of the small change from the original to the standard SAWT output, their similar performance is not surprising.

After the initial examination, the results with respect to the OBDFB were disappointing except against the AWMF; however, similarities between the original image and the standard SAWT algorithm effective translate to the original image outperforming any despeckling whatsoever. This does not appear to be the case in the sample outputs. In the Figure 6.38, the OBDFB clearly smooths homogeneous regions such as that near the top of the image yet retains boundaries between obviously disparate structures such as near the middle of the image whereas the AWMF overcompensates the smoothing action. This is more obvious in Figure 6.39. This holds true over all of the images.

Possible explanations for the disparity between subjective score and observation include the pool of testers and the instructions. Very few participants had any medical background or experience viewing ultrasound imagery. Consequently, their interpretation of the instructions may have led to ambiguity. The instructions themselves were clear, instructing participants to “Please choose the image that appears to be LESS NOISY while retaining features such as organ boundaries.” However, further analysis in a more controlled situation such as a medical doctor comparing an ultrasound image with ground truth before and after processing is necessary for more conclusive results.

6.5 Summary

A new algorithm for real-time ultrasound image despeckling is presented. It is a modification of the SAWT algorithm using the OBDFB. Subjective test results are presented as are a comparison of



(a) Original

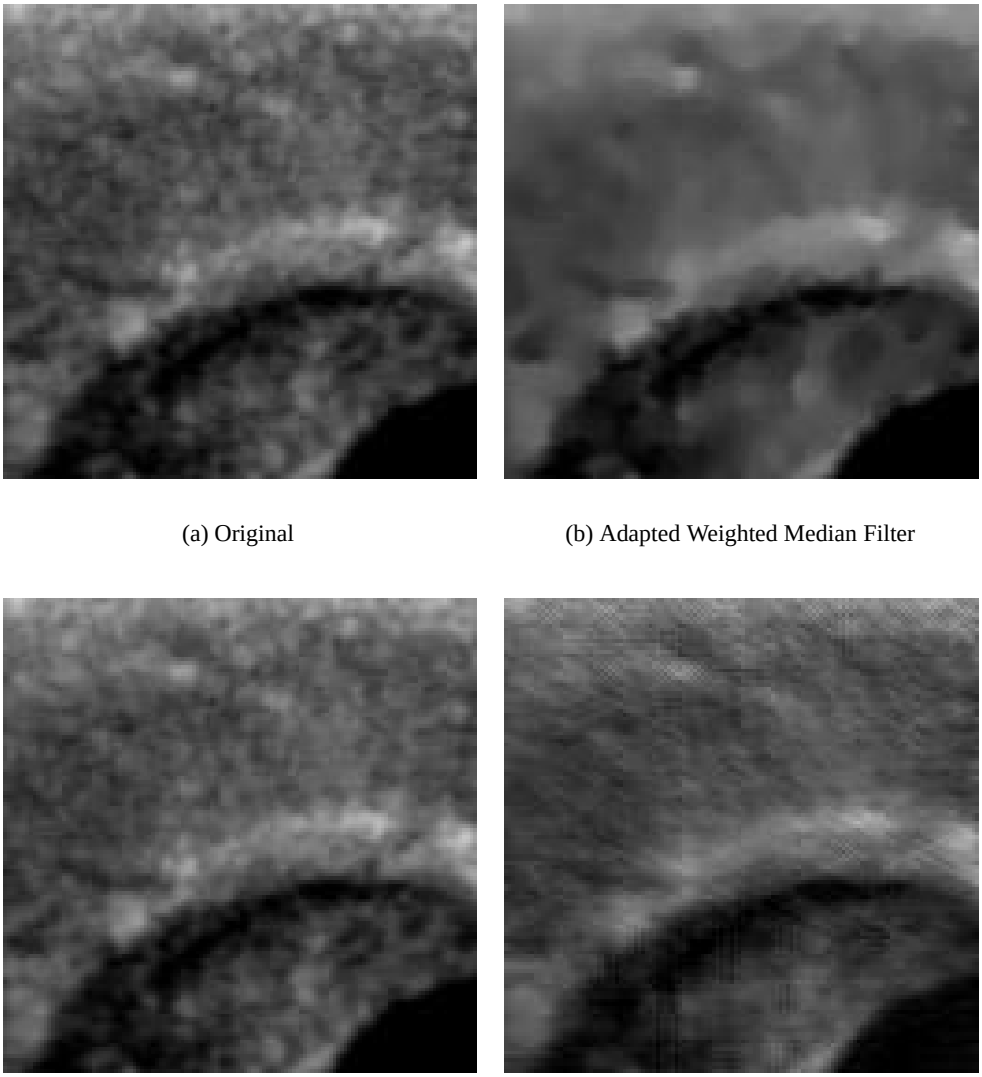
(b) Adapted Weighted Median Filter

(c) SAWT using Biorthogonal Wavelets

(d) SAWT using the OBDFB

Figure 6.38: Sample ultrasound despeckling results.

sample outputs.



(a) Original

(b) Adapted Weighted Median Filter

(c) SAWT using Biorthogonal Wavelets

(d) SAWT using the OBDFB

Figure 6.39: Sample ultrasound despeckling results with zoom.

CHAPTER VII

MODIFIED OBDFB

As part of this investigations, we explored a modification to the OBDFB structure in an attempt to better handle the low frequencies and angular frequencies. We examined the potential of non-uniform and rationally-sampled filter banks [18, 25, 34, 46, 50].

7.1 Non-uniform Rationally-sampled Filter Bank

Although the OBDFB is able to perform angular and radial frequency partitioning in a maximally decimated framework, it still has some drawbacks. First of all, and this is inherent to Bamberger's DFB, energy around DC channels unevenly across the directional bands. Although the OBDFB addresses most of the area around DC and mid-band frequencies particularly well, without ideal filters, this leakage is inevitable in the current framework. Additionally, the current incarnation of the DFB (and OBDFB) generates subband boundaries along the principle frequency axes thereby dividing the energy there between subbands instead of trying to consolidate that energy into a single subband.

To address many of these issues, a filter bank with frequency partitions described in Figure 7.40 is proposed. This filter bank removes the low frequency partition by itself thereby removing the DC energy issue, and it contains directional bands that include the principal frequency axes. These high radial frequency bands can be further decomposed to provide finer angular resolution if so desired.

Although a similar filter bank was introduced in [63], there are some important property differences in the modified OBDFB we consider now. First of all, the cut-off frequency of the low frequency regions are different. Second, and more importnantly, the decomposition in [63] is restricted to a minimum of six directions whereas the proposed decomposition can have as few as two. This is useful for further decomposition of lower frequency partitions when a lower angular resolution is desired.

The proposed structure of the new filter bank is shown in Figure 7.41. The modulators are

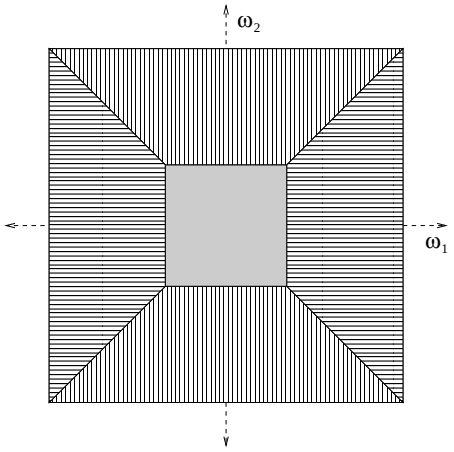


Figure 7.40: The radially low frequency region is denoted H_l and is indicated by the shaded region near DC; the radially high frequency vertical passband is denoted H_v and is indicated by vertical lines; and the radially high frequency horizontal passband is denoted H_h and is indicated by horizontal lines.

included in order to "center" the hexagonal passbands for the vertical and horizontal directional bands and to avoid aliasing. Notice that the downsampling matrix $\mathbf{M}$ is the same for all three bands. Additionally, we are under the assumption that this is a critically sampled system, and this will be shown in its design.

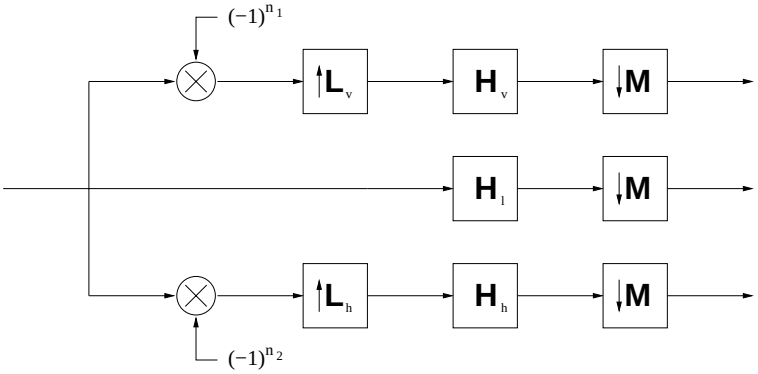


Figure 7.41: The non-uniform rationally-sampled filter bank.

To summarize, advantages of this new system include the following:

- Both angular and radial frequency selectivity is achieved simultaneously.
- Frequency scrambling of radially low frequency information aliasing to radially high frequency areas is avoided.

- Similarly, directional aliasing is avoided.
- Unlike the DFB and OBDFB, this new structure does not propogate DC into a single subband.
- This transform has the potential for further decomposition both angularly and radially (i.e., it can have a tree structure).
- Directional passbands include the fundamental frequency axes (i.e., does not have subband boundaries along those axes).
- The new filter bank uses rational sampling matrices for maximal decimation.

Obstacles to the proposed filter bank include the following:

- The filters appear to be nonseparable in both the nondecimated and decimated domains.
- A polyphase implementation is not immediately obvious.
- The downsampling of the two directional subbands requires rational sampling matices (i.e., upsampling and downsampling by integer matrices).
- Perfect reconstruction is not enforced yet, and its conditions are unclear (at least relatively speaking with respect to a non-rationally sampled filter bank).

7.1.1 Design Procedure

This section describes the different considerations in designing the proposed filter bank. Although the majority of the design process was completed, the actual filter design procedures such as those discussed in [33] were beyond the scope of this thesis. Related materials may be found in [20, 28, 40, 49, 62].

7.1.1.1 Cut-off Frequency

Let us assume that the cut-off frequency of the low frequency region is ω_c. We define b to be ω_c/π for convenience. Consider first the region denoted by H_0. This region can be modulated so it appears to be a contiguous hexagon centered at the origin (and repeated every 2π) in the normalized frequency space. Now the problem becomes finding a rational sampling matrix such that these hexagons can be packed as seen in Figure 7.42

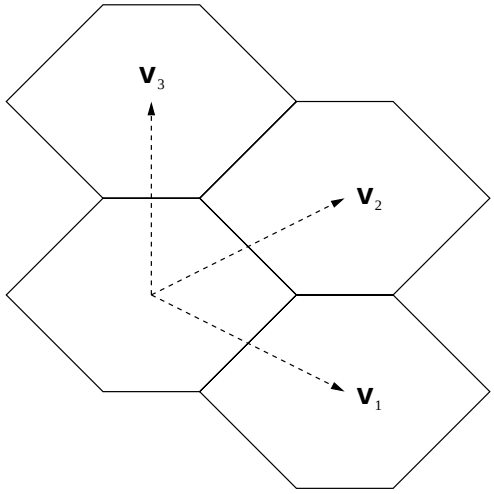


Figure 7.42: Different frequency vectors for Hexagonal regions of support.

In order to accomplish this, we define

$$\mathbf{u}_1 = \begin{bmatrix} b+1 \\ b-1 \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} b+1 \\ 1-b \end{bmatrix}, \text{and } \mathbf{u}_3 = \begin{bmatrix} 0 \\ 2-2b \end{bmatrix}. \tag{7.21}$$

Using

$$\mathbf{U}^T\mathbf{V} = 2\pi\mathbf{I}, \tag{7.22}$$

we can find the decimation matrix $\mathbf{V}$ for each $\mathbf{U}$ associated with three combinations of vectors from above. They become

$$\mathbf{U}_1 = \begin{bmatrix} b+1 & b+1 \\ b-1 & 1-b \end{bmatrix} \pi, \quad \mathbf{V}_1 = \begin{bmatrix} \frac{1}{b+1} & \frac{1}{b+1} \\ \frac{1}{b-1} & \frac{1}{1-b} \end{bmatrix}, \tag{7.23}$$

$$\mathbf{U}_2 = \begin{bmatrix} b+1 & 0 \\ 1-b & 2-2b \end{bmatrix} \pi, \quad \mathbf{V}_2 = \begin{bmatrix} \frac{2}{b+1} & \frac{-1}{b+1} \\ 0 & \frac{1}{1-b} \end{bmatrix}, \tag{7.24}$$

$$\mathbf{U}_3 = \begin{bmatrix} b+1 & 0 \\ b-1 & 2-2b \end{bmatrix} \pi, \quad \mathbf{V}_3 = \begin{bmatrix} \frac{2}{b+1} & \frac{1}{b+1} \\ 0 & \frac{1}{1-b} \end{bmatrix}, \tag{7.25}$$

where each of the decimation matrices $\mathbf{V}_k$ have $\det(\mathbf{V}_k) = \frac{-2}{(b+1)(b-1)}$.

In Figure 7.43, we see a typical branch of a rationally sampled filter bank. For now, assume that branch to be H_v. From [17, 24, 45], if we want to have commutative downsamplers and upsamplers

for a computationally efficient implementation, we must have $\mathbf{LM} = \mathbf{ML}$. If we define $\mathbf{H} = \mathbf{L}^{-1}\mathbf{M}$ and $\mathbf{R} = \mathbf{L}\mathbf{M}^{-1}$, then

$$\begin{aligned} \mathbf{LM} = \mathbf{ML} &\Rightarrow \mathbf{LMM}^{-1} = \mathbf{MLM}^{-1} && (7.26) \\ &\Rightarrow \mathbf{M}^{-1}\mathbf{L} = \mathbf{LM}^{-1}, && (7.27) \end{aligned}$$

and

$$\begin{aligned} \mathbf{H} &= \mathbf{L}^{-1}\mathbf{M}, && (7.28) \\ &= (\mathbf{M}^{-1}\mathbf{L})^{-1}, && (7.29) \\ &= (\mathbf{L}\mathbf{M}^{-1})^{-1}, && (7.30) \\ &= \mathbf{R}^{-1}. && (7.31) \end{aligned}$$

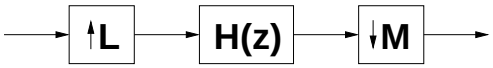


Figure 7.43: A single branch of a rationally-sampled filter bank.

A convenient assumption to make is that all three branches have the same diagonal downsampling matrix $\mathbf{M}$ in the rational portion. Because we need to downsample the radially low frequency portion by $\frac{1}{b}$ in both directions, we can assume that $\mathbf{M} = \begin{bmatrix} \frac{1}{b} & 0 \\ 0 & \frac{1}{b} \end{bmatrix}$ without loss of generalization. If it turns out that b is rational, it is still implementable. Another reason this is useful is because it means that $\mathbf{L}$ and $\mathbf{M}$ commute which is a necessary condition for an efficient polyphase form.

Since $\mathbf{V}_n = \mathbf{L}^{-1}\mathbf{M} \Rightarrow \mathbf{V}_n\mathbf{M} = \mathbf{L}^{-1}$, we obtain

$$\begin{aligned} \mathbf{L}_1 &= (b\mathbf{V}_1)^{-1}, & \mathbf{L}_2 &= (b\mathbf{V}_2)^{-1}, & \mathbf{L}_3 &= (b\mathbf{V}_3)^{-1}, \\ &= \begin{bmatrix} \frac{b+1}{2b} & \frac{b-1}{2b} \\ \frac{b+1}{2b} & \frac{1-b}{2b} \end{bmatrix}, & &= \begin{bmatrix} \frac{b+1}{2b} & \frac{1-b}{2b} \\ 0 & \frac{1-b}{b} \end{bmatrix}, & &= \begin{bmatrix} \frac{b+1}{2b} & \frac{b-1}{2b} \\ 0 & \frac{1-b}{b} \end{bmatrix}. \end{aligned} \quad (7.32)$$

In order for the $\mathbf{L}$ matrices to remain integer, $\frac{b+1}{2b}$ and $\frac{b-1}{2b}$ must be integers. Thus $\frac{1}{2} + \frac{1}{2b}$ and $\frac{1}{2} - \frac{1}{2b}$ must be integers. This implies that $\frac{1}{2b} = k + \frac{1}{2}$ and consequently $\frac{1}{2} = (k + \frac{1}{2})b \Rightarrow b = \frac{1}{2(k+\frac{1}{2})} = \frac{1}{2k+1}$, where k is an integer. For $k = 0, 1, 2, \ldots$, $b = 1, \frac{1}{3}, \frac{1}{5}, \ldots$, respectively. Although b was allowed to be rational, it actually has the form $\frac{1}{k_{\text{odd}}}$ where k_{odd} is an odd integer. To maximize the

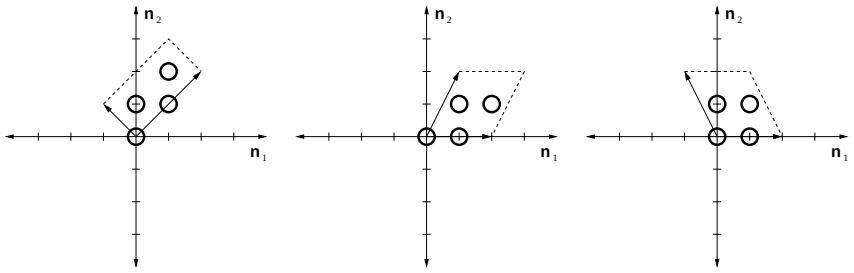
area of the low frequency region, we choose $k = 0$ so that $b = \frac{1}{3}$ (i.e., $\omega_c = \frac{\pi}{3}$). It should be noted that this is a bit smaller than the usual $\frac{\pi}{2}$ cut-off frequency.

7.1.1.2 Upsampling Matrix

Now that b is determined, we have

$$\mathbf{L}_1 = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix}, \mathbf{L}_2 = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}, \text{and } \mathbf{L}_3 = \begin{bmatrix} 2 & -1 \\ 0 & 2 \end{bmatrix}. \tag{7.33}$$

The coset vectors of these upsampling matrices are shown in Figure 7.44. Because $\mathbf{L}_1$ keeps its axes orthogonal and only introduces a small amount of rotation, we select it to use for that branch; i.e., we define $\mathbf{L}_v = \mathbf{L}_1$.



(a) $\mathbf{L}_1$ (b) $\mathbf{L}_2$ (c) $\mathbf{L}_3$

Figure 7.44: Coset vectors of the different upsamplers.

Although downsampling by $\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$ is straightforward, upsampling by $\mathbf{L}_1$ is not as obvious. The modulated vertical radially high frequency passband is shown in Figure 7.45 along with its upsampled version without its replicas. The upsampled passband shown defines the necessary geometry for H_v.

The algorithm from [33] uses objective and constraint formulations to design near-perfect reconstruction filter banks in the spatial domain. It extends the upsamplers in rationally-sampled filter banks from the filter banks with only downsamplers. If this algorithm is being used, then there are a few stipulations on the upsampling matrix of the last branch. All of the branches need to start with the same unitary resampling (i.e., have the same leading matrix in their Smith forms [26, 27]). First, we decompose $\mathbf{L}_v$. Although we would like the diagonal matrix of the Smith form to be

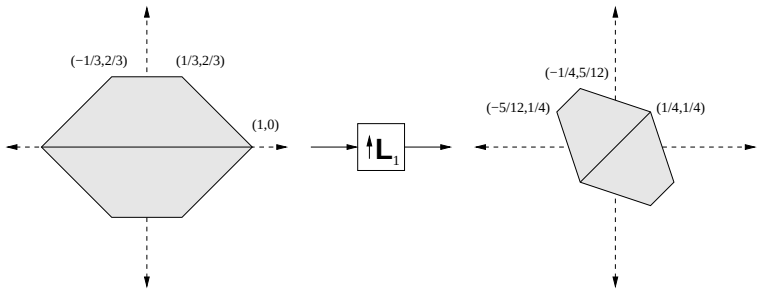


Figure 7.45: Upsampling a hexagon in frequency using $\mathbf{L}_1$.

$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ (for simplicity), we can see that this is not possible from the following:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} 1 & \frac{-1}{2} \\ 1 & \frac{1}{2} \end{bmatrix}$$

$$\Rightarrow \begin{array}{ll} ae + bg = 1 & ce + dg = 1 \\ af + bh = \frac{-1}{2} & cf + dh = \frac{1}{2} \end{array}$$

where we know that all of the entries of the matrices must be integers. Obviously, this cannot be the case with the final set of equations. So instead, after some inspection, we have the following for $\mathbf{L}_0$:

$$\begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \tag{7.34}$$

$$\begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}. \tag{7.35}$$

Since $\begin{bmatrix} 1 & -2 \\ 1 & 2 \end{bmatrix}$ does not have a Smith form with one of the two leading unitary matrices from above, we use instead

$$\begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \tag{7.36}$$

$$\begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}. \tag{7.37}$$

We can arbitrarily choose which form to use in the Gardos algorithm which is equivalent to choosing downsampling by four in either the horizontal or vertical directions. In either case, we choose $\mathbf{L}_h = \begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix}$ which orients the hexagon into the same final orientation as $\mathbf{L}_v$ (i.e., $H_v = H_h$).

7.1.1.3 Polyphase Form

The polyphase form for a rational multi-dimensional filter bank [18] provides a computationally efficient implementation. The following example elucidates the derivation of the polyphase form. We start with the single branch shown in Figure 7.43. If we assume that $|det(\mathbf{L})| = 2$ and $|det(\mathbf{M})| = 3$, we can perform all of the steps shown in Figure 7.46 assuming that all of the coset vectors of $\mathbf{M}$ can be written as $\mathbf{k}_i = \mathbf{M}\mathbf{k}_{i1} + \mathbf{L}\mathbf{k}_{i2}$. Additionally, the $\mathbf{j}_i$'s are the coset vectors of $\mathbf{L}$. We get the structure shown in Figure 7.46(a) by substituting in the polyphase form of just the filter $H(z)$ and the downsampler and by substituting our new form of the coset vectors of $\mathbf{M}$. The transition from Figure 7.46(a) to Figure 7.46(b) is made by moving the upsamplers and downsamplers across their respective delays. Because $\mathbf{ML} = \mathbf{LM}$, we can commute the upsamplers and downsamplers (Figure 7.46(c)) and then finally use the polyphase form of the upsampler followed by a filter to get Figure 7.46(d).

The following vectors were found by doing an exhaustive search where all of the coset vectors of $\mathbf{M}$ were substituted into the $\mathbf{k}_{i2}$'s. Only a single value of $\mathbf{k}_{i2}$ from the coset vectors revealed a

unique integer solution $\mathbf{k}_{i1}$. For $\mathbf{L}_v$, we have

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{7.38}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{7.39}$$

$$\begin{bmatrix} 2 \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ -2 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \tag{7.40}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \tag{7.41}$$

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \tag{7.42}$$

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{7.43}$$

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \tag{7.44}$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \tag{7.45}$$

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mathbf{L}_v \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{7.46}$$

and for $\mathbf{L}_h$, we have

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{7.47}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \tag{7.48}$$

$$\begin{bmatrix} 2 \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \tag{7.49}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{7.50}$$

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \tag{7.51}$$

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \tag{7.52}$$

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} -2 \\ 0 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \tag{7.53}$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{7.54}$$

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mathbf{L}_h \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{7.55}$$

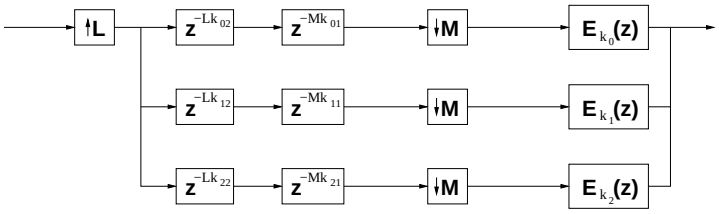
where the equation is of the form $\mathbf{k}_i = \mathbf{M}\mathbf{k}_{i1} + \mathbf{L}\mathbf{k}_{i2}$.

7.1.1.4 Further Decomposition

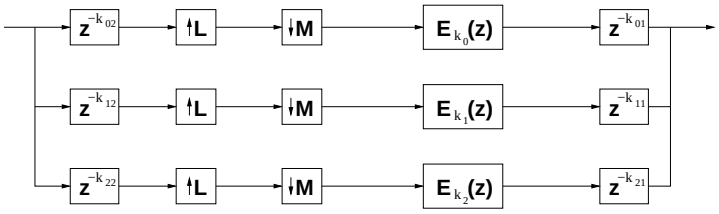
As discussed earlier, further decomposition of the directional bands is possible. In order to keep the subband divisions from falling on the original principal frequency axes, we can subdivide the directional bands into thirds as shown in Figure 7.47. After downsampling, these subbands can have the same geometry as the directional subbands and can be further divided accordingly.

7.2 Summary

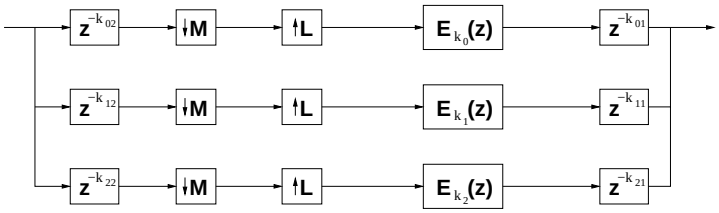
A non-uniform rationally-sampled filter bank was introduced that mitigates many of the disadvantages of the OBDFB. It removes the DC area initially thereby circumventing the propagation of DC energy into directional bands, and it does not create subband boundaries along principal frequency axes.



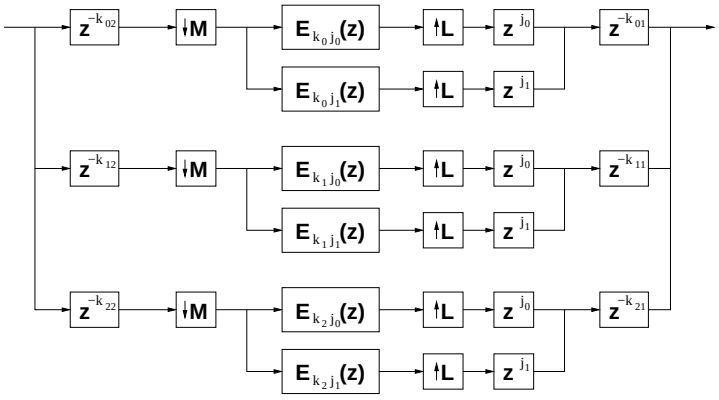
(a)



(b)



(c)



(d)

Figure 7.46: Sample polyphase form derivation from [18]. Assume $|det(\mathbf{L})| = 2$ and $|det(\mathbf{M})| = 3$ for this example.

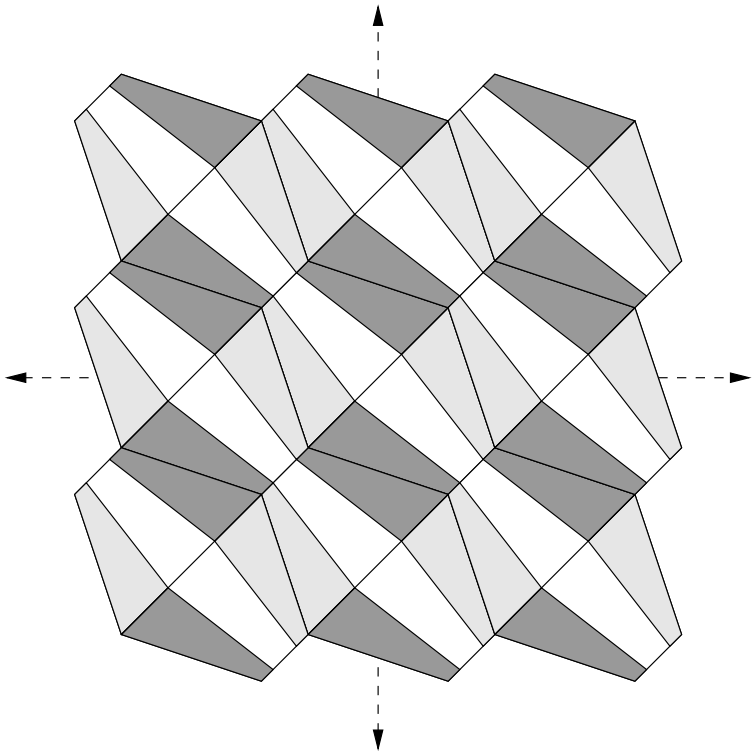


Figure 7.47: Potential further decomposition of the directional bands.

CHAPTER VIII

CONCLUSION AND FUTURE WORK

The octave-band directional filter bank has been proposed and applied to a number of applications. Specific contributions to filter bank theory and each application are discussed below. Additionally, potential future work is presented afterwards.

8.1 Contributions of Thesis

The main contribution of this research was to develop a filter bank that could decompose a 2-dimensional signal with respect to both radial and angular frequency in a maximally decimated and computationally efficient manner. The OBDFB is able to achieve this goal in addition to retaining the separable implementation, perfect reconstruction, and polyphase forms of the DFB because of its successive applications of a DFB used to capture radially low frequency information from decimated directional subbands.

We have applied the OBDFB to several applications, the first of which was texture segmentation. Filter banks and wavelet decompositions have enjoyed a large amount of success when trying to segment an image into different regions based on texture. The OBDFB was compared to traditional octave-band decompositions in the decimated domain as well as Gabor filters in the undecimated domain over a diverse set of test and training data. It was able to outperform the traditional octave-band decompositions in the decimated domain and perform on par with the Gabor filters which were used as the gold standard. The computational savings when using the decimated transforms arguably compensates for the loss in accuracy in that domain.

Similarly, the OBDFB was used to increase classification of hyperpixels in hyperspectral data. Several sets of ground-truthed AVIRIS hyperspectral imagery from the Boreal forest in Canada were used for testing and training of a classification algorithm. In the baseline system, maximum likelihood was used to classify hyperpixels. Using the OBDFB, texture information was added to the baseline system by combining scores produced first by the baseline system and then by the

texture system. For both the original and dimension-reduced data, the OBDFB texture augmentation improved classification accuracy.

Additionally, the OBDFB was used to perform ultrasound despeckling. The speckling that occurs in ultrasound imagery can be treated in a similar fashion as additive noise in natural images. Consequently, a modified version of the SAWT algorithm using the OBDFB was applied to a series of ultrasound images and compared to a traditional wavelet decomposition and an adaptive weighted median filtering algorithm in subjective A-B testing.

Finally, a non-uniform octave-band directional filter bank was introduced that uses rational sampling to accomodate the non-uniform channels. The design procedure was presented from the initial subband geometries to the final resampling matrices. This filter bank avoids many of the disadvantages of the DFB and OBDFB but introduces its own set of issues.

8.2 Future Work

There are several open problems with respect to both filter bank theory and design as well as applications.

- The non-uniform octave-band directional filter bank should be explored in comparison to new decompositions. Additionally, an appropriate filter design procedure needs to be applied to create perfectly reconstructing filters. One of the most promising structures appears to be the ladder filter structure as perfect reconstruction is structurally enforced.

- One of the strengths of the OBDFB is its computational efficiency which is on the order of traditional, separable wavelet decompositions. Therefore, application of the OBDFB to real-time video applications should be explored and implemented, even beyond texture segmentation and ultrasound despeckling.

- For three-dimensional signals, an equivalent velocity-filtering filter bank should be explored. Such a filter bank would generate subbands with conic passband geometries for the higher radial frequency areas and should also discriminate with respect to radial frequency. Unfortunately, the intuitive subband geometries are not able to be made in a maximally decimated way; however, alternative geometries should be explored as video processing continues to

become a staple in signal processing.

REFERENCES

[1] ABD-ELMONIEM, K. Z., YOUSSEF, A.-B. M., and KADAH, Y. M., "Real-time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion," *IEEE Transactions on Biomedical Engineering*, vol. 49, pp. 997–1014, Sept. 2002.

[2] ACHIM, A., BEZERIANOS, A., and TSAKALIDES, P., "Novel bayesian multiscale method for speckle removal in medical ultrasound images," *IEEE Transactions on Medical Imaging*, vol. 20, pp. 772–783, Aug. 2001.

[3] ANSARI, R., KIM, C. W., and DEDOVIC, M., "Structure and design of two-channel filter banks derived from a triplet of halfband filters," *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 46, pp. 1487–1496, Dec. 1999.

[4] ANTOINE, J.-P., MURENZI, R., and VANDERGHEYNST, P., "Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns," *Applied and Computational Harmonic Analysis*, vol. 6, pp. 314–345, 1999. available online at http://www.idealibrary.com.

[5] ARCE, G. R., "A general weighted median filter structure admitting negative weights," *IEEE Transactions on Signal Processing*, vol. 46, pp. 3195–3205, Dec. 1998.

[6] ARCE, G. R. and MCLOUGHLIN, M. P., "Theoretical analysis of the max/median filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, pp. 60–69, Jan. 1987.

[7] BAMBERGER, R. H., "New results on two and three dimensional directional filter banks," in *Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, pp. 1286–1290, 1993.

[8] BAMBERGER, R. H., EDDINS, S. L., and NURI, V., "Generalized symmetric extension for size-limited multirate filter banks," *IEEE Transactions on Image Processing*, vol. 3, pp. 82–87, Jan. 1994.

[9] BAMBERGER, R. H. and SMITH, M. J. T., "A filter bank for the directional decomposition of images: Theory and design," *IEEE Transactions on Signal Processing*, vol. 40, pp. 882–893, Apr. 1992.

[10] BOVIK, A. C., HUANG, T. S., and MUNSON, JR., D. C., "A generalization of median filtering using linear combinations of order statistics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, pp. 1342–1350, Dec. 1983.

[11] BURKHARDT, R., DIMALANTA, A., and PELI, T., "Hyperspectral data compression using adaptive wavelet packets," in *Workshop on Multi/Hyperspectral Technology and Applications*, 2002.

[12] CARRÉ, P., ANDRES, E., and FERNANDEX-MALOIGNE, C., "Discrete rotation for directional orthogonal wavelet packets," in *IEEE Proceedings: ICIP*, pp. 257–260, Oct. 2001.

[13] CHANG, S. G., YU, B., and VETTERLI, M., "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, pp. 1532–1546, Sept. 2000.

[14] CHANG, S. G., YU, B., and VETTERLI, M., "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing*, vol. 9, pp. 1522–1531, Sept. 2000.

[15] CHEN, T., DJOKOVIĆ, I., and VAIDYANATHAN, P. P., "Results on multidimensaionl nonuniform rational maximally-decimated filter banks with orthonormal filters," in *Twenty-sixth Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 145–149, 1992.

[16] CHEN, T. and VAIDYANATHAN, P. P., "Multidimensional multirate filters and filter banks derived from one-dimensional filters," *IEEE Transactions on Signal Processing*, vol. 41, pp. 1749–1765, May 1993.

[17] CHEN, T. and VAIDYANATHAN, P. P., "On the choice of rational decimation systems for multidimensional signals," in *IEEE Proceedings: ICASSP*, vol. 5, pp. 527–530, 1993.

[18] CHEN, T. and VAIDYANATHAN, P. P., "The role of integer matrices in multidimensional multirate systems," *IEEE Transactions on Signal Processing*, vol. 41, pp. 1035–1047, Mar. 1993.

[19] CLAUSI, D. A. and JERNIGAN, M. E., "Designing Gabor filters for optimal texture separability," *Pattern Recognition*, vol. 33, pp. 1835–1849, 2000.

[20] COOKLEV, T., NISHIHARA, A., YOSHIDA, T., and SABLATASH, M., "Regular multidimensional linear phase FIR digital filter banks and wavelet bases," in *IEEE Proc.: ICASSP*, vol. 2, pp. 1464–1467, May 1995.

[21] DO, M. N. and VETTERLI, M., "Pyramidal directional filter banks and curvelets," in *IEEE Proceedings: ICIP*, pp. 158–161, Oct. 2001.

[22] DO, M. N., *Directional Multiresolution Image Representations*. PhD thesis, Swiss Federal Institute of Technology, 2001.

[23] DUDGEON, D. E. and MERSEREAU, R. M., *Multidimensional Digital Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

[24] EVANS, B. L., "Designing commutative cascades of multidimensional upsamplers and downsamplers," *IEEE Signal Processing Letters*, vol. 4, pp. 313–316, Nov. 1997.

[25] EVANS, B. L., BAMBERGER, R. H., and MCCLELLAN, J. H., "Rules for multidimensional multirate structures," *IEEE Transactions on Signal Processing*, vol. 42, pp. 762–771, Apr. 1994.

[26] EVANS, B. L., GARDOS, T., and MCCLELLAN, J. H., "Imposing structure on Smith-form decompositions of rational resampling matrices," *IEEE Transactions on Signal Processing*, vol. 42, pp. 970–973, Apr. 1994.

[27] EVANS, B. L., TEICH, J., and KALKER, T. A., "Families of Smith form decompositions to simplify multidimensional filter bank design," in *Twenty-eighth Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 363–367, 1994.

[28] EVANS, B. L., TEICH, J., and SCHWARZ, C., "Automated design of two-dimensional rational decimation systems," in *Twenty-eighth Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 498–502, 1994.

[29] FERNANDES, F. C. A., VAN SPAENDONCK, R. L., and BURRUS, C. S., "A directional, shift-insensitive, low-redundancy, wavelet transform," in *IEEE Proceedings: ICIP*, pp. 618–621, Oct. 2001.

[30] FUKUDA, S. and HIROSAWA, H., "A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, pp. 2282–2286, Sept. 1999.

[31] FUKUNAGA, K., *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[32] GALLAGHER, JR., N. C. and WISE, G. L., "A theoretical analysis of the properties of median filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 1136–1141, Dec. 1981.

[33] GARDOS, T. R., *Analysis and Design of Multidimensional FIR Filter Banks*. PhD thesis, Georgia Institute of Technology, 1993.

[34] GOPINATH, R. A. and BURRUS, C. S., "On upsampling, downsampling, and rational sampling rate filter banks," *IEEE Trans. on Signal Processing*, vol. 42, pp. 812–824, Apr. 1994.

[35] GRIGORESCU, S. E., PETKOV, N., and KRUIZINGA, P., "Comparison of texture features based on Gabor filters," *IEEE Trans. on Image Processing*, vol. 11, pp. 1160–1167, Oct. 2002.

[36] GUPTA, S., KAUR, L., CHAUHAN, R. C., and SAXENA, S. C., "A wavelet based statistical approach for speckle reduction in medical ultrasound images," in *TENCON: Conference on Convergent Technologies for Asia-Pacific Region*, pp. 534–537, Oct. 2003.

[37] HONG, P. S., KAPLAN, L. M., and SMITH, M. J. T., "Hyperspectral image segmentation using filter banks for texture augmentation," in *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, pp. 254–258, 2003.

[38] HONG, P. S., KAPLAN, L. M., and SMITH, M. J. T., "A comparison of the octave-band directional filter bank and Gabor filters for texture classification," in *IEEE Proceedings: ICIP*, pp. 1541–1544, 2004.

[39] HONG, P. S. and SMITH, M. J. T., "An octave-band family of non-redundant directional filter banks," in *IEEE Proceedings: ICASSP*, pp. 1165–1168, 2002.

[40] IKEHARA, M. and NGUYEN, T. Q., "On 2D perfect reconstruction linear phase filter banks," in *Thirty-First Asilomar Conference on Signals, Systems, and Computers*, pp. 721–725, 1997.

[41] JAIN, A. K. and FARROKHNIA, F., "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[42] JAYANT, N., JOHNSTON, J., and SAFRANEK, R., "Signal compression based on models of human perception," *Proc. of the IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.

[43] JIA, X. and RICHARDS, J. A., "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 37, pp. 538–542, Jan. 1999.

[44] KARAMAN, M., KUTAY, M. A., and BOZDAGI, G., "An adaptive speckle suppression filter for medical ultrasonic imaging," *IEEE Transactions on Medical Imaging*, vol. 14, pp. 283–292, June 1995.

[45] KHANSARI, M. R. K. and CHEN, T., "Finding commutative multidimensional downsamplers and upsamplers," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2002–2004, Aug. 1995.

[46] KHANSARI, M. R. K. and LEON-GARCIA, A., "Multidimensional perfect reconstruction filter banks with generalized subsampling," in *IEEE Proc.: ICASSP*, vol. 3, pp. 185–188, 1993.

[47] KITTLER, J., HATEF, M., DUIN, R., and MATAS, J., "On combining classifiers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, Mar. 1998.

[48] KOVAČEVIĆ, J. and VETTERLI, M., "Perfect reconstruction filter banks with rational sampling rate changes," in *IEEE Proc.: ICASSP*, vol. 3, pp. 1785–1788, 1991.

[49] KOVAČEVIĆ, J. and VETTERLI, M., "Design of multidimensional non-separable regular filter banks and wavelets," in *IEEE Proc.: ICASSP*, vol. 4, pp. 389–392, Mar. 1992.

[50] KOVAČEVIĆ, J. and VETTERLI, M., "Perfect reconstruction filter banks with rational sampling factors," *IEEE Trans. on Signal Processing*, vol. 41, pp. 2047–2066, June 1993.

[51] LANDGREBE, D. A., *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, New Jersey: John Wiley & Sons, 1993.

[52] LENNON, M., MOUCHOT, M. C., MERCIER, G., and HUBERT-MOY, L., "Segmentation of hedges on CASI hyperspectral images by data fusion from texture, spectral and shape analysis," in *IEEE Proc. of the Int. Geoscience and Remote Sensing Symp.*, vol. 2, pp. 825–827, 2000.

[53] LIEW, S. C., CHANG, C. W., and LIM, K. H., "Hyperspectral land cover classification of EO-1 Hyperion data by principal component analysis and pixel unmixing," in *IEEE Proc. of the Int. Geoscience and Remote Sensing Symp.*, vol. 6, pp. 3111–3113, 2002.

[54] LIU, B. and BRUTON, L. T., "The design of N-band nonuniform-band maximally decimated filter banks," in *Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1281–1285, 1993.

[55] LOUPAS, T., MCDICKEN, W. N., and ALLAN, P. L., "An adaptive weighted median filter for speckle suppression in medical ultrasound images," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 129–135, Jan. 1989.

[56] LU, Y. and DO, M. H., "CRISP-contourlets: a critically sampled directional multiresolution image representation," in *Proceedings of the SPIE conference on Wavelet Applications in Signal and Image Processing X*, Aug. 2003.

[57] MACIEJEWSKI, A. A. and ROBERTS, R. G., "An example of principal component analysis applied to correlated images," in *IEEE Proc. of the 33rd Southeastern Symp. on System Theory*, pp. 269–273, 2001.

[58] MAN, H., *On Efficiency and Robustness of Adaptive Quantization for Subband Coding of Images and Video Sequences*. PhD thesis, Georgia Institute of Technology, 1999.

[59] MERCIER, G. and LENNON, M., "On the characterization of hyperspectral texture," in *IEEE Geoscience and Remote Sensing Symposium*, vol. 5, pp. 2584–2586, 2002.

[60] MUJICA, F. A., LEDUC, J.-P., MURENZI, R., and SMITH, M. J. T., "A new motion parameter estimation algorithm based on the continuous wavelet transform," *IEEE Transactions on Image Processing*, vol. 9, pp. 873–888, May 1999.

[61] NAGAI, T., FUTIE, T., and IKEHARA, M., "Direct design of nonuniform filter banks," in *IEEE Proc.: ICASSP*, vol. 3, pp. 2429–2432, 1997.

[62] NAYEBI, K., BARNWELL III, T. P., and SMITH, M. J. T., "The design of perfect reconstruction nonuniform band filter banks," in *IEEE Proc.: ICASSP*, vol. 3, pp. 1781–1784, 1991.

[63] NGUYEN, T. T. and ORAINTARA, S., "A multiresolution directional filter bank for image applications," in *IEEE Proc.: ICASSP*, vol. 3, pp. 37–40, May 2004.

[64] NURI, V. and BAMBERGER, R. H., "A theoretical framework for the analysis and design of size-limited multirate filter banks," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 311–314, 1992.

[65] PARK, S., SMITH, M. J. T., and MERSEREAU, R. M., "A new directional filter bank for image analysis and classification," in *IEEE Proc.: ICASSP*, vol. 3, pp. 1417–1420, Mar. 1999.

[66] PARK, S.-I., *New Directional Filter Banks and Their Applications in Image Processing*. PhD thesis, Georgia Institute of Technology, 1999.

[67] PHILIPS, "http://www.medical.philips.com/main/products/ultrasound/image_ library/."

[68] PHOONG, S.-M., KIM, C. W., VAIDYANATHAN, P. P., and ANSARI, R., "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Transactions on Signal Processing*, vol. 43, pp. 649–665, Mar. 1995.

[69] PUN, K. S. C. and NGUYEN, T. Q., "A novel and efficient design of multidimensional PR two-channel filter banks with hourglass-shaped passband support," *IEEE Signal Processing Letters*, vol. 11, pp. 345–348, Mar. 2004.

[70] RAMASUBRAMANIAN, D. and KANAL, L. N., "Fast integer wavelet transforms and their application to hyperspectral image compression," in *Workshop on Multi/Hyperspectral Technology and Applications*, 2002.

[71] RANDEN, T. and HUSØY, J. H., "Filtering for texture classification: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 291–310, Apr. 1999.

[72] RANDOLPH, T. R., *Image Compression and Classification Using Nonlinear Filter Banks*. PhD thesis, Georgia Institute of Technology, 2001.

[73] REDMILL, D. W. and BULL, D. R., "Nonlinear perfect reconstruction critically decimated filter banks," *IEEE Electronic Letters*, vol. 32, pp. 310–311, Feb. 1996.

[74] RELLIER, G., DESCOMBES, X., ZERUBIA, J., and FALZON, F., "A Gauss-Markov model for hyperspectral texture analysis of urban areas," in *IEEE International Conference on Pattern Recognition*, vol. 1, pp. 692–695, 2002.

[75] RIVAZ, P. D., *Complex Wavelet Based Image Analysis and Synthesis*. PhD thesis, University of Cambridge, 2000.

[76] ROSENTHAL, J., *Filters and Filterbanks for Hexagonally Sampled Signals*. PhD thesis, Georgia Institute of Technology, 2002.

[77] ROSILES, J. G., *Image and Texture Analysis using Biorthogonal Angular Filter Banks*. PhD thesis, Georgia Institute of Technology, 2004.

[78] ROSILES, J. G. and SMITH, M. J. T., "Image denoising using directional filter banks," in *IEEE Proceedings: ICIP*, pp. 292–295, 2000.

[79] ROSILES, J. G. and SMITH, M. J. T., "Texture classification with a biorthogonal directional filter bank," in *IEEE Proceedings: ICASSP*, pp. 1549–1552, 2001.

[80] ROSILES, J. G. G., "Image and texture analysis with angular biorthogonal filter banks." Dissertation Proposal, Georgia Institute of Technology, July 2000.

[81] ROSILES, R.-G. and SMITH, M. J. T., "Texture segmentation with a biorthogonal directional decomposition," in *World Multiconference on Systemics and Informatics*, July 2000.

[82] SELESNICK, I. W., "A new complex-directional wavelet transform and its application to image denoising," in *IEEE Proc.: ICIP*, vol. 3, pp. 573–576, 2002.

[83] SHAH, I. A. and KALKER, T. A. C. M., "On ladder structures and linear phase conditions for bi-orthogonal filter banks," in *IEEE Proc.: ICASSP*, vol. 3, pp. 181–184, Apr. 1994.

[84] SHI, M. and HEALEY, G., "Hyperspectral texture recognition using a multiscale opponent representation," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 41, pp. 1090–1095, May 2003.

[85] SHMULEVICH, I. and ARCE, G. R., "Spectral design of weighted median filters admitting negative weights," *IEEE Signal Processing Letters*, vol. 8, pp. 313–316, Dec. 2001.

[86] SIMONCELLI, E. P. and ADELSON, E. H., "Noise removal via Bayesian wavelet coring," in *IEEE Proc.: ICIP*, pp. 379–382, Sept. 1996.

[87] SMITH, M. J. T. and EDDINS, S. L., "Analysis/synthesis techniques for subband image coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1446–1456, Aug. 1990.

[88] TYO, J. S., "PC-based pseudocolor rendering for hyperspectral imagery," in *Proceedings of the 2003 ARL Advanced Sensors CTA Conference*, pp. 189–193, Apr. 2003.

[89] VAIDYANATHAN, P. P., *Multirate Systems and Filter Banks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

[90] VISCITO, E. and ALLEBACH, J. P., "The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices," *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 29–41, Jan. 1991.

[91] WANG, Y.-P., WU, Q., CASTLEMAN, K., and XIONG, Z., "Image enhancement using multi-scale oriented wavelets," in *IEEE Proc.: ICIP*, pp. 610–613, Oct. 2001.

[92] WATSON, A. B., "The cortex transform: Rapid computation of simulated neural images," *Computer Vision, Graphics, Image Processing*, vol. 39, no. 3, pp. 311–327, 1987.

[93] WATSON, A. B., "Efficiency of a model human image code," *Journal of the Optical Society of America A*, vol. 4, pp. 2401–2417, Dec. 1987.

VITA

Paul S. Hong was born in New Castle, Delaware in 1976. He attended Seaholm High School in Birmingham, Michigan, and graduated in June 1994. He graduated from the University of Michigan, Ann Arbor, Michigan, with a Bacchelor in Science Degree from the Electrical Engineering and Computer Science Department in December 1997. He received a Masters of Science in Electrical and Computer Engineering Degree from the Georgia Institute of Technology, Atlanta, Georgia, in May 2000. His advisor was Dr. Mark J. T. Smith, and he was eventually co-advised by Dr. Russell M. Mersereau. His research interests include multi-rate, multi-dimensional filter banks applied to texture segmentation, hyperspectral imagery, and ultrasonic imagery. He is a member of Eta Kappa Nu, Tau Beta Pi, and IEEE.