FIXED-ANALYSIS ADAPTIVE-SYNTHESIS FILTER BANKS

A Thesis
Presented to
The Academic Faculty

by

Clyde Alphonso Lettsome

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2009

FIXED-ANALYSIS ADAPTIVE-SYNTHESIS FILTER BANKS

Approved by:

Dr. David Anderson,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark J.T. Smith, Advisor
School of Electrical and Computer Engineering
Purdue University

Dr. Russell M. Mersereau
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Yorai Wardi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gail Rosen
Electrical and Computer Engineering
Drexel University

Date Approved: 31 March 2009

To my parents, Whitmore and Elterah Lettsome.

ACKNOWLEDGEMENTS

First and foremost, let me start by thanking my advisors, Dr. Mark J. T. Smith and Dr. Russell M. Mersereau. Dr. Smith has been the best teacher I could have ever dream of having. I hope that I am half as successful as Dr. Smith, in sharing knowledge and wisdom with future generations. Dr. Mersereau took time out to help review my dissertation, sign documents, as well as guide me through the university rules and regulations. Thank you both for your guidance.

With regards to my reading committee, thank you Dr. David Anderson and Dr. Aaron Lanterman for challenging me to find solutions to your complex questions. Special thanks as well to other defense committee members, Dr. Yorai Wardi and especially Dr. Gail Rosen. Dr. Rosen has been both a friend and sincere committee member providing me with great encouragement. Many thanks goes to Dr. David Hertling and Marilou Mycko for leading me through the admission process, helping me find funding for most of the years at the Georgia Institute of Technology, and tolerating the many emailed questions. Special thanks also to Dr. David Veazie for enlightening me by telling me what my advisor expects of me and what I should expect of my advisor.

It would not be fair, if I fail to make mention of my friends and fellow students. Dr. Cherita Corbett, Dr. Raheem Beyah, and Dr. Dalong Li. Thank you all for helping me navigate the politics. A special thanks to all the students of GCATT's third floor, later the CSIP section of Centergy, BGSA, and Caribsa. There are too many worth mentioning. Thus, I will not name them individually. You all have made my stay at CSIP and Georgia Institute of Technology enjoyable. Special thanks goes to Freda "Mika" Johnson and Tamara Clegg. Mika, my grammar editor, reading

about filter bank was not your preferred reading subject. At times, you might have fallen asleep. Nevertheless, you were such a great help. Tamara, I can always count on you with your warm hospitality. Thank you for preparing the food for my proposal and dissertation defense.

Thank you to my business partner Antoine B. Rolle, our small but yet diligent staff, and subcontractors at Calabrix Corporation. Thank you for patience, throughout this journey. You've taken up the slack at work when I was doing research and writing. Our clients never once noticed any delay in fulfilling their needs or meeting their expectations.

This would not have been possible without my parents, Whitmore and Elterah Lettsome. Your support both financially and emotionally, allowed me to dream big. I could never really express how much I appreciate everything. Neither of you attended college. In fact, neither you completed high school initially, as you were forced to leave school to help nuture your younger siblings. Yet despite this, you still understood the importance of an education and found your own ways to encourage and motivate me. Later on, while juggling work and family, mom you received your high school diploma. I can recall the countless nights you spent doing homework, even after going to work and school. Mom you remained discipline in your duties and punctual for work the following day. Somehow, my father inspired me through his pride. Dad, I can recall when you bought a custom made shelf to place all the awards and trophies we earned, after there was no more space left on the old shelf.

To my sister Glenda Lettsome, thanks for being like a mom when mom was studying. I owe a special thanks to my brother, Kevin Lettsome who passed away when I was just twelve. You were a real go-getter and genius amongst us kids. You encouraged me to never stop trying. Rest in peace. Also many thanks to my sister Justine Lettsome, for being like a best friend when we were growing up. Additionally, I would like to thank my cousin Tameka Winchester, for her many invites to dinner, when she knew I could not find time to cook for myself.

Most importantly, thank God, my Lord and Savior Jesus Christ. Through whom, you have blessed me with your patience, guidance and kept me sane through this venture. I would also like to thank God for allowing my parents overcome their health issues over the last couple of years. They've always been encouraging all through these years. Now they are even more proud, to experience this pinnacle of success in my academic career. It was with great joy to place that phone call and tell my parents, I am now Dr. Clyde Alphonso Lettsome.

Lastly, I have a special message to ”my future wife and future kids”, that will one day read this acknowledgement, I hope. A wise man once said, ”People do not plan to fail, they fail to plan.” Because of this, I postponed having a family until I completed my doctorate degree. My greatest fear was failing as a husband, as a father, as a provider, as a protector and as a leader. I waited this long, so that I might be able to give all of you the attention you deserve. Only time will tell if this was a good decision. I trust that it was the best decision for us.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Subband/Wavelet filter analysis-synthesis filters are a major component in many compression algorithms. Such compression algorithms have been applied to images, voice, and video. These algorithms have achieved high performance. Typically, the configuration for such compression algorithms involves a bank of analysis filters whose coefficients have been designed in advance to enable high quality reconstruction. The analysis system is then followed by subband quantization and decoding on the synthesis side. Decoding is performed using a corresponding set of synthesis filters and the subbands are merged together. For many years, there has been interest in improving the analysis-synthesis filters in order to achieve better coding quality. Adaptive filter banks have been explored by a number of authors where by the analysis filters and synthesis filters coefficients are changed dynamically in response to the input. A degree of performance improvement has been reported but this approach does require that the analysis system dynamically maintain synchronization with the synthesis system in order to perform reconstruction.

In this thesis, we explore a variant of the adaptive filter bank idea. We will refer to this approach as fixed-analysis adaptive-synthesis filter banks. Unlike the adaptive filter banks proposed previously, there is no analysis synthesis synchronization issue involved. This implies less coder complexity and more coder flexibility. Such an approach can be compatible with existing subband wavelet encoders. The design methodology and a performance analysis are presented.

CHAPTER I

INTRODUCTION

Filter banks have a rich history of investigation. Much of the attention has been motivated by speech, image, and video compression applications. However, subband filter banks have also been employed in many other important areas such as telemedicine, object detection and classification, denoising and enhancement, image size conversion and sampling rate alteration, and secure signal transmission. In this application context, filter banks are typically viewed as having the form shown in Figure 1. The filter bank (or more precisely, the analysis-synthesis filter bank) contains two primary sections: the analysis section and the synthesis section. The analysis section decomposes the signal into bands or subbands. This can be seen clearly by considering a signal $x[n]$ as the input to the system. First $x[n]$ is filtered by

$$h_0[n], h_1[n], \ldots, h_{M-1}[n],$$

which are typically bandpass filters. In the z-domain, these filters are given by

$$H_0(z), H_1(z), \ldots, H_{M-1}(z).$$

The analysis section implements decimated convolution described by the equations

$$v_k[n] = h_k[n] * x[n] = \sum_{m=0}^{L-1} x[n-m]h_k[m], \qquad k = 0, 1, \ldots, M-1,$$
$$y_k[n] = v_k[Rn]. \tag{1}$$

The channel parameter M and decimation parameter R help define the specific type of filter bank. Depending on the application, filter banks may have as few as two bands (i.e. $M = 2$) or a large number of bands such as 64, 128, or more. The decimation factor R determines the redundancy inherent in the representation. When $R = 1$, the filter bank embodies its highest degree of oversampling. In most applications, the decimation factor is set equal to the number of bands (that is $R = M$) resulting in a memory efficient critically sampled representation.

In the synthesis section, the outputs $y_k[n]$ are recombined to reconstruct the original. In this process, the subbands, $y_k[n]$, are upsampled by a factor of R and then filtered by the synthesis filters $G_k(z)$, after which the channels are summed together.

Depending on the application, some type of signal operation is often performed on the subbands prior to reconstruction. For compression applications, this involves quantizing and coding the subbands. This allows the quantization noise associated with quantization to be preferentially distributed in frequency, which allows one to exploit the spectral roll off that typically accompanies input signals of interest as well as exploit properties of human perception.

There are a number of issues associated with analysis-synthesis filter banks at several levels. First at the filter level, one is typically concerned with the filter characteristics. Often it is desired to have filters that are bandpass in nature with flat passband characteristics, high attenuation stopband characteristics and a narrow transition band. This allows the spectral content of the input to be isolated for independent processing. The designer also has choices with respect to the phase of the filters. Filters may be designed with linear phase, non-linear phase, or approximately linear phase. In many compression applications, linear phase is thought to be preferable. In addition, there is sometimes interest in having constraints placed on the time-domain characteristics of the analysis and synthesis filters. For example, it is sometimes desirable to keep the overshoot and undershoot in the lowpass filter step response small, say below 10 percent. This condition is in contradiction to having sharp frequency domain characteristics. Thus, there is a design tradeoff in play.

At the individual *section* level, one typically seeks to have the filter passbands

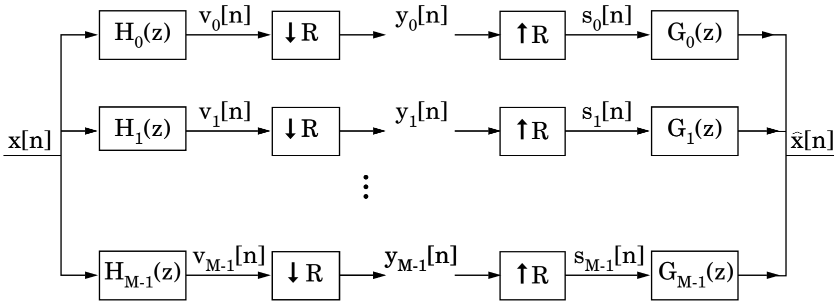


Figure 1: Block diagram of a 1-D uniform band analysis-synthesis filter bank.

collectively cover the entire spectrum without gaps or significant dips in the spectral magnitude response. This desire, when present, comes from the recognition that gaps in coverage represent information loss associated with the spectral region corresponding to the gap. Significant variations in the flatness of the passbands can also be a concern when subbands are quantized. Variations in magnitude can cause quantization noise to be disproportionately amplified across the spectrum.

At the analysis-synthesis system level, one typically considers reconstruction properties. The prevailing view is that exact reconstruction is the ideal objective. However, on occasion, systems with almost exact reconstruction are employed to gain added computational efficiency. These kinds of distortions come in several flavors, such as magnitude distortion, phase distortion, and a mix of both. Again, when these distortions are limited, the impact on performance can be negligible. Another system level issue is delay. It was shown by Nayebi et al. [27] that the overall group delay of the system could be controlled in the filter design process. This is an interesting result for audio signal processing because it allows one to address the pre-echo effect that is audible in conventional low bit rate subband coding algorithms.

At the implementation level, there are other sets of issues to consider. A number of implementation methods have been introduced that provide tradeoffs between arithmetic complexity and decomposition quality. Included among the many implementation structures are tree-structure systems, uniform band systems, lattice structures, ladder structures, cosine modulated systems, lapped transforms, and parallel and cascade filter implementations.

An issue that we have not addressed in this work at all is the use of infinite-duration impulse response (IIR) filters. In this work, we only consider finite impulse response (FIR) filters, as they are the most widely used. But, IIR filter banks have been shown to have advantages and have their uses.

An interesting variation of the analysis-synthesis filter bank was introduced by Nayebi et al. [26]-the so-called time-varying filter bank. This is a novel class of filter banks where the constituent filters are allowed to be adaptive. The authors showed that exact reconstruction can be achieved in a time-varying filter environment by having the analysis filters and synthesis filters change in accordance with a set of conditions that assure exact reconstruction. For every instance of change in the analysis filters, a new set of synthesis filters is employed adaptively that ensures that the input can be reconstructed exactly.

The formulation of time-varying systems was later refined by Sodagar et al. [42, 43] leading to a significantly easier implementation and design process. More specifically, Sodagar et al. introduced the use of a post filter with time-varying filter properties. Now instead of having to invoke a set of adaptive filters for every switching occurrence in the analysis section, the analysis and synthesis filters can be matched one for one. That is, assuming you had two different exactly reconstructing analysis-synthesis filter banks (say A and B), when you switch from A to B in the analysis, one simply switches from A to B in the synthesis section. Without postfiltering, distortion occurs in reconstruction in spite of the fact that both A and B in isolation are exact reconstruction systems. The distortion, however, is transitory. The adaptive postfilter functions as an inverse filter, restoring the input exactly during the periods of transition. When the filters are operating without any switching, the postfilter functions as an identity system.

One of the key issues associated with this type of adaptive filter bank is that one must maintain the synchrony between the analysis and synthesis filters in order to ensure exact reconstruction. This issue was investigated extensively by Arrowood et al. [1, 2]. Both forward and backward adaptive methods of synchronous adaptive analysis-synthesis were reported. It is also noteworthy that a number of authors recognized that introducing adaptivity into the analysis-synthesis system was possible by working in a lattice or ladder structure where exact reconstruction is constrained structurally. In such cases synchronous switching of the analysis and synthesis filters guarantees exact reconstruction. What is lost when doing this is that the analysis and synthesis filter characteristics degrade completely during the interval surrounding the switching. So, again there is a tradeoff.

This class of adaptive filter banks will be reviewed more extensively in Chapter 3 for context. But, in short, the use of this class of filter banks has been limited, owing primarily to the overhead associated with having to track the analysis filter switching pattern in order to maintain synchrony.

In this thesis, we introduce a new class of filter bank that heretofore has not been considered previously. As stated earlier, the current and conventional filter bank employs fixed analysis and fixed synthesis filters in the analysis-synthesis system. The time-varying filter bank introduced by Nayebi et al. and further expanded and explored by others involves adaptive analysis filters that operate in synchrony with adaptive synthesis filters. The former system we call FAFS (fixed analysis, fixed synthesis) and note that it remains the overwhelmingly dominate case. We call the latter case just mentioned AAAS (adaptive analysis, adaptive synthsis) for analogous reasons. It might be apparent at this point that there are two other filter bank classes implied by the nomenclature. The first is Fixed Analysis Adaptive Synthesis (FAAS) and Adaptive Analysis Fixed Synthesis (AAFS). Initial intuition might suggest that both FAAS and AAFS are not particularly interesting classes because exact reconstruction in an adaptive steady state can be shown to be impossible. However, FAAS and AAFS systems are quite interesting when you shift the paradigm. Instead of assuming that exact reconstruction filter banks provide the best outcome, consider that it may be possible to enhance the original or enable more robust performance in the face of subband signal modification.

The image compression example is a good one in which to consider this new paradigm. After an image is decomposed in a traditional subband/wavelet coder, the subbands are quantized. As soon as this happens, information is lost and it becomes impossible to reconstruct the original input exactly. So the issue is no longer about quality in the absence of modification, but quality in the presense of modification. What if FAAS and AAFS systems can be shown to improve reconstruction quality in the presence of quantization relative to FAFS systems. Then such systems become very interesting.

This thesis is devoted to defining, designing, and exploring FAAS systems, and evaluating their utility in compression and interpolation/enlargement applications. The exploration of AAFS systems, although identified, is not explored in this work but is suggested as a topic for a follow-on thesis.

In the chapters that follow, we will provide some background and review for context, and then discuss the new FAAS in detail. More specifically, a brief historical overview of the evolution of filter banks is presented in Chapter 2. This is followed by a discussion in Chapter 3 of the previous work relevant to this thesis that was developed for AAAS systems. In particular, we discuss structure and design issues associated with AAAS systems. The new work begins in Chapter 4 with the development of the time-domain design methodology for odd-length analysis-synthesis filters, the design methodology that is employed subsequently for FAAS systems. FAAS systems are fully developed in this thesis and are applied in compression and enlargement applications. Performance results are presented that demonstrate the advantages of FAAS systems over the traditional FAFS systems.

CHAPTER II

BACKGROUND

This chapter provides a brief overview of the development of filter banks over the years to help make clear how the new class of fixed analysis adaptive synthesis filter banks (FAAS) advances theory and design in this area. It is easy to argue that modern day filter banks date back at least to the early speech spectrum analysis work that was done in the 1930's. During that time, the "analog" sound spectrograph was used to generate time-frequency representations of speech. In the spectrograph, a variable frequency oscillator is used to modulate the input signal. Then, bandpass filters are employed to localize the frequency spectrum. The output of the device is a plot of the average short-term energy as a function of time. Thus, the sound spectrograph behaves as an analysis filter bank.

Perhaps the oldest type of digital filter bank is the channel vocoder introduced by Dudley [11] during this same time period. The channel vocoder contains a cosine-modulated filter bank of the form

$$h_k[n] = h[n]\cos[\frac{\pi}{M}(n+\alpha)(k+\beta)]. \tag{2}$$

The α and β constants provide variations to the filter bank that can allow for efficient implementations. Such implementations were extensively developed by Malvar [21, 20] much later in the 1990s. In the channel vocoder, the absolute value of each channel sample is computed (which is equivalent to full-wave rectification), followed by lowpass filtering. Dudley used these outputs along with voicing and pitch information to represent speech at a low bit rate.

Another relative of the filter bank is the discrete short-time Fourier transform (STFT), shown in Figure 2, which is a kind of uniform DFT filter bank. It was popular in the area of digital speech processing in the 1970s and early 80s. The STFT uses complex exponential modulators to shift high frequency regions of the spectrum to baseband, after which lowpass filters are used to isolate the shifted band. Because the modulators resemble the kernel of the DFT, the outputs are complex valued just like the outputs of the DFT. Portnoff [32] is credited with much of the pioneering work in developing the relationship between the filter bank and transform interpretations of the discrete STFT as well as deriving the analysis/synthesis reconstruction conditions. The general formulation allows for variable redundancy in the representation. When $R = 1$, the representation is fully redundant, i.e. highly over-sampled. At the other extreme is the case when $R = M$, which is a minimally sampled representation. The minimally sampled representation may be implemented efficiently using a polyphase structure concatenated with an FFT.

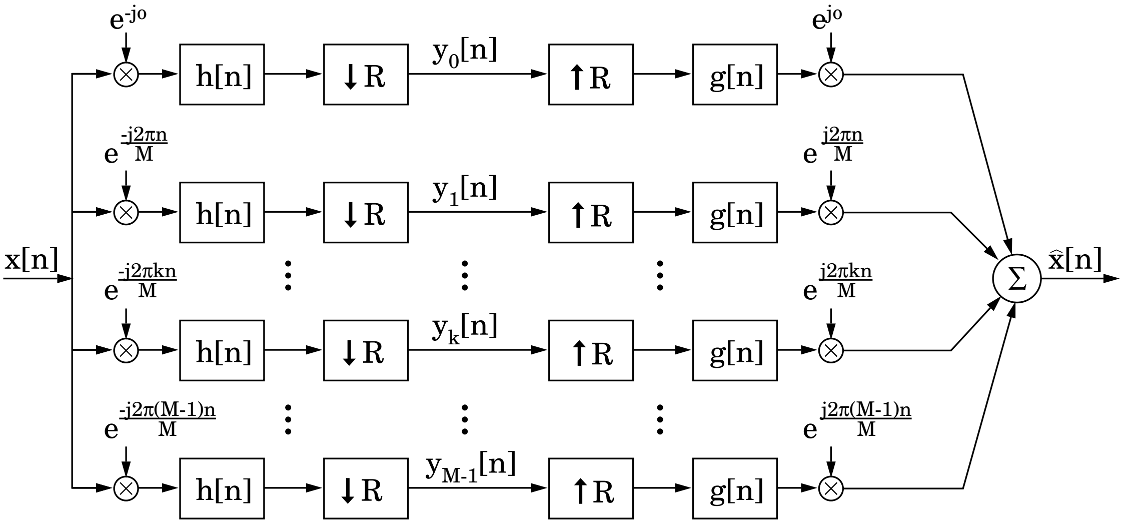


Figure 2: M-channel analysis-synthesis sections of the discrete short-time Fourier transform.

2.1 Quadrature Mirror Filters

A major milestone occurred in 1976 and 1977 with the introduction of Quadrature Mirror Filters (QMFs) and the notion of aliasing cancellation [8]. In this approach, the decomposition is based on a two-band filter bank where $M = 2$ and $R = 2$, as shown in Figure 3. The goal of the analysis filter bank is to decompose the input into

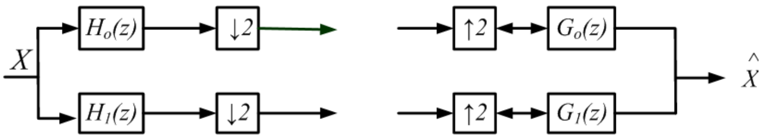


Figure 3: Two-band analysis-synthesis filter bank

critically sampled low frequency and high frequency subbands. In reconstruction, the subbands are interpolated and merged so that the input is reconstructed with high accuracy. Because decimation is involved in the filter bank, the reconstruction issue is not trivial.

The analysis-synthesis equations can written in the z-transform domain, leading to the reconstruction equation

$$\hat{X} = \frac{1}{2}X(-z)[H_0(-z)G_0(z)+H_1(-z)G_1(z)]+\frac{1}{2}X(z)[H_0(z)G_0(z)+H_1(z)G_1(z)], \quad (3)$$

where $x[n]$ is the input and $\hat{x}[n]$ is the reconstructed output. Two components are associated with this equation: an aliasing component

$$\frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)] \quad (4)$$

and a transfer function term

$$\frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)]. \quad (5)$$

Ideally, the filter bank should have the property that the aliasing term reduces to zero and the transfer function term is unity (or an approximation there of).

Croisier, Estaban and Galand [8] proposed a solution to the two-band filter bank reconstruction equation where the analysis and synthesis filters are related as follows:

$$H_1(z) = H_0(-z), \quad (6)$$

$$G_0(z) = H_0(z), \quad (7)$$

and

$$G_1(z) = -H_0(-z). \tag{8}$$

The filter $H_0(z)$ is typically a lowpass filter, which implies that $H_1(z)$ is a highpass filter. By examining the equations, it is evident that the QMF solution results in a complete cancellation of the aliasing term. In order to achieve a unity transfer function, $H_0(z)$ is designed so that the transfer function approximates z^ℓ, where ℓ is the system delay. Two-band QMFs cannot yield a perfect solution, but can be designed to provide very good approximations. Johnston designed a full set of optimized QMFs in 1980 that are still widely used today [15].

2.2 Conjugate Quadrature Filters

Another milestone occurred in 1984 with the introduction of conjugate quadrature filters (CQFs). The CQF solution allowed two-band filter banks to achieve exact reconstruction. In fact, it was shown how to design a broad class of analysis-synthesis filters that satisfy the CQF conditions, including optimal equiripple filters [38]. The CQF solution is defined as

$$H_1(z) = H_0(-z^{-1}), \tag{9}$$

$$G_0(z) = H_0(z^{-1}), \tag{10}$$

and

$$G_1(z) = -H_0(-z). \tag{11}$$

Unlike QMFs, the CQF analysis and synthesis filters are frequency shifted and time reversed versions of $H_0(z)$. It is easily seen after substitution that the aliasing term cancels, i.e.

$$[H_0(-z)H_0(z^{-1}) - H_0(z^{-1})H_0(-z)] = 0 \tag{12}$$

and that the transfer function can be designed so that

$$[H_0(z)H_0(z^{-1}) - H_0(-z^{-1})H_0(-z)] = 2z^{-\ell}. \tag{13}$$

2.3 Tree-Structured Filter Banks

Two-band decompositions are rather uninteresting from an application perspective, since two channel frequency resolution is generally too coarse for most applications. However, two-band filter banks can be cascaded in trees to form an infinite variety of multi-band filter banks with varying frequency resolution. A simple octave-band tree structure is shown in Figure 4. In this simple tree structure, the original signal is split into two subbands. Each proceeding level of decomposition is split into two more subbands from the prior stage's low frequency channel. The ideal bandwidths of the analysis side are illustrated in the frequency response graph in Figure 5.

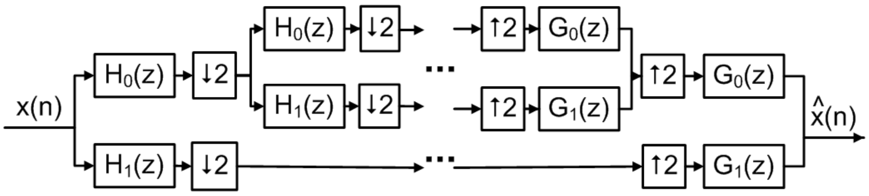


Figure 4: Octave-band tree structure (also known as a dyadic wavelet tree structure).

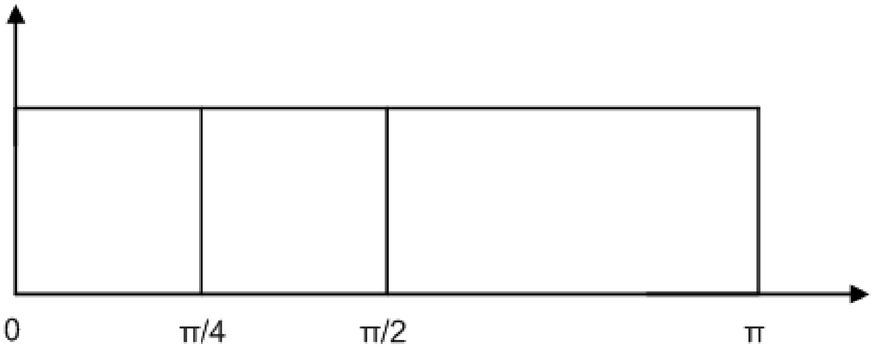


Figure 5: Octave band (also known as wavelet or tree structure) frequency partitioning.

This process of cascading two-band filter banks can be repeated as appropriate for a particular application. The reconstruction process involves a complementary tree structure of two-band synthesis filter banks as shown in Figure 4.

2.4 Efficient Implementations

There have been many implementation structures used to realize filter banks. These structures include direct form implementation, polyphase [3], lattice [45], and ladder (lifting) structures [23].

Polyphase Structure:

The polyphase filter bank is often a more efficient way to implement QMFs than by using a direct form implementation. The polyphase structure is shown in Figure 6. For QMFs the number of multiplications required is reduced by a factor of two.

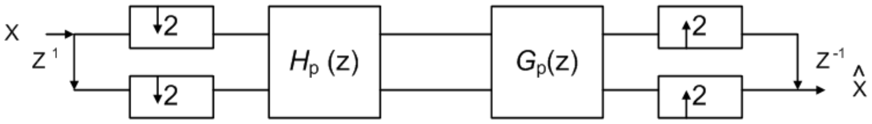


Figure 6: Polyphase filter bank structure.

Lattice Structure:

The use of lattice structures for implementing filter banks was discussed extensively by Vaidyanathan and Hoang [45] and can be used to implement CQFs efficiently. Lattice structures are formed by cascading a series of rotation and delay elements. Each rotation is represented by a matrix of the form

$$\mathbf{R}_i = \begin{bmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{bmatrix}. \tag{14}$$

$\mathbf{R}_i$ can be factored to

$$\mathbf{R}_i = cos\theta_i \begin{bmatrix} 1 & \alpha \\ -\alpha & 1 \end{bmatrix} \tag{15}$$

where α is $\tan\theta$. $\mathbf{R}_i$ is graphically represented in Figure 7 . Each delay is represented by

$$\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}. \tag{16}$$

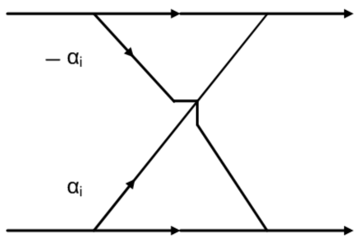


Figure 7: Basic lattice filter section.

Both rotation and delay matrices are paraunitary. As such, cascaded matrices result in an equivalent paraunitary matrix. A block diagram of a lattice structure is shown in Figure 8. On the synthesis side of Figure 8

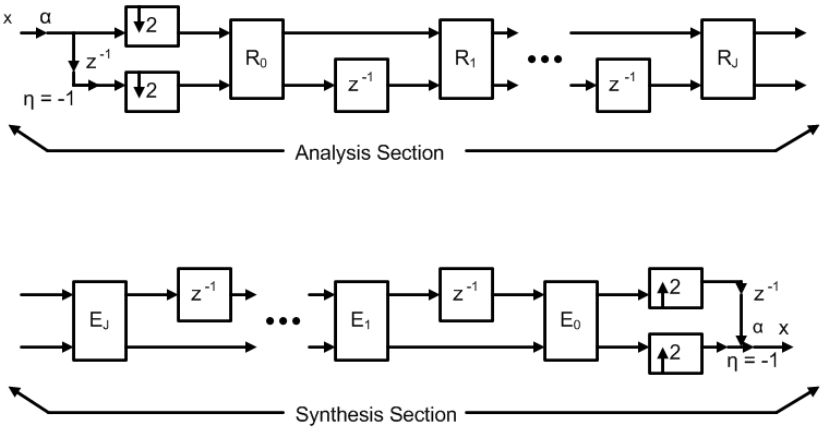


Figure 8: Lattice filter bank structure.

$$\mathbf{E}_i = \mathbf{R}_i^T. \tag{17}$$

The delay element in the synthesis is

$$\Gamma = \begin{bmatrix} z^{-1} & 0 \\ 0 & 1 \end{bmatrix}. \tag{18}$$

Much like the polyphase structure outlined previously, the lattice implementation is efficient and yields exact reconstruction. In addition, lattice structures are robust to coefficient quantization, unlike the direct form implementation.

Ladder Structure:

An even more attractive implementation is the ladder filter implementation. Digital ladder filters were introduced in 1972 by Mitra and Sherwood [23] and are more efficient than polyphase and lattice form structures. It is noteworthy that many authors publishing in the literature appear to be unaware of this original work. Several authors presented identical work in the 1990s under the name ”lifting filters.”

Similar to the lattice structure, ladder structures employ butterflies but with only one wing. A basic two-band ladder structure is shown in Figure 9. Each step in the analysis performs one of two functions:

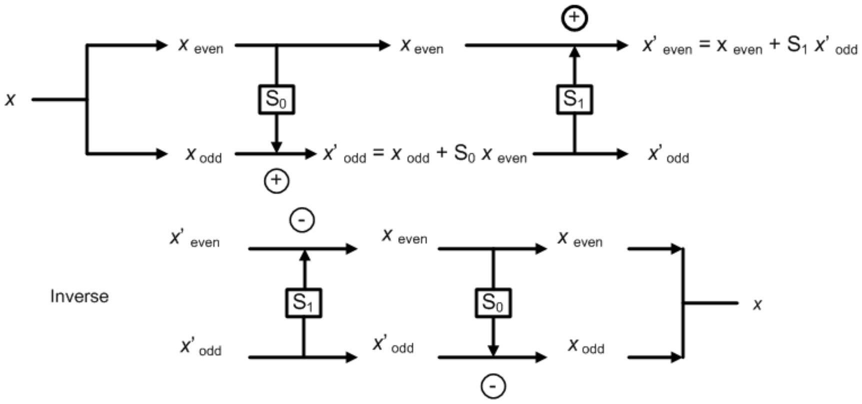


Figure 9: Ladder filter bank structure.

1. The highpass signal is filtered and added to the lowpass channel.
2. The lowpass signal is filtered and added to the highpass channel.

It can be shown that the ladder structure guarantees exact reconstruction structurally and that it can accommodate both linear and nonlinear operators as part of the ladder step operations. An expanded treatment may be found in [10].

2.5 M-Band and Modulated Filter Banks

Shortly after the introduction of exact reconstruction filters in 1984, much attention turned to multiband uniform systems and their reconstruction and implementation issues. Solutions for the M-band case were proposed by a number of authors [44, 30]. Modulated filter banks of the form of Equation (2) were also being explored during this time period. The first to appear in the open literature was presented by Rothweiler at ICASSP83 [35]. Rothweiler's modulated filter banks are like those formerly employed to implement efficient transmultiplexers [24, 13], but in addition address the issue of analysis-synthesis reconstruction. Although this class of modulated filter banks cannot attain exact reconstruction, it's reconstruction is of reasonably high quality [5, 7].

The confluence of modulated filter banks and exact reconstruction occurred with the introduction of the time-domain aliasing cancellation (TDAC) method of Princen and Bradley [33]. Interestingly, Malvar, Staelin, and Cassereau arrived at the same solution slightly later but approached the problem from a transform perspective. Although Princen and Bradley were first to publish, their version did not catch on for lack of an efficient implementation. The work of Malvar et al. (called Lapped Transforms) included an efficient FFT-class implementation structure.Because of their efficiency, lapped transforms received widespread attention. Additional improvements to the theory and design of cosine modulated filter banks followed over the subsequent years, primarily adding flexibility with respect to ease of design and the relaxation of previous constraints on length and number of channels [19, 34, 25, 17].

2.6 Time-Domain Design

Another landmark development was the introduction of the time-domain framework by Nayebi et al. Through this framework, low delay filter banks were discovered. In addition time-varying filter banks, non-uniform filter banks, and block-decimation filter banks were also investigated. The time-domain framework is based on describing the analysis filters in matrix form as a multi-channel decimated convolution matrix multiplied by a synthesis filter matrix. All matrix elements represent time-domain filter coefficients. This formulation is used extensively in the chapters that follow in order to develop the FAAS filter banks.

2.7 Applications

As mentioned earlier, the study of analysis-synthesis filter banks was initially motivated by medium rate speech coding. After 1986 when Woods and O'Neil [47] published their paper on subband image coding, attention shifted. Now image compression and denoising are arguably the most popular applications of analysis-synthesis filter banks. For image coding, quantization and coding are performed between the analysis and synthesis sections. For denoising applications, a nonlinear operator is often applied between the analysis and synthesis. Thus, in both cases because the analysis outputs are modified, exact reconstruction is no longer the expectation. Rather one hopes that the appearance of the output image looks good subjectively.

The popular fixed analysis fixed synthesis filter banks presume that the best design is one in which exact reconstruction is preserved in the absence of modification—the thought being that analysis-synthesis distortion can only add to the quantization error and thus reducing or eliminating such distortion improves overall quality. The introduction of time-varying filter banks represented a slight paradigm shift in the sense that it was recognized that time varying filter banks could be used to reduce the effects of quantization noise.

The new FAAS system we will introduce in this thesis builds directly on the AAAS work proposed by Nayebi et al. and developed by Arrowood et al. In the next chapter we will discuss in more detail the operations of AAAS systems.

CHAPTER III

ADAPTIVE-ANALYSIS ADAPTIVE-SYNTHESIS FILTER BANKS

In 1992, it was shown for the first time that exact reconstruction filter banks could be designed with adaptive analysis-synthesis filters [26]. This was an interesting result from a theoretical perspective, but also from an application perspective. In this approach, the analysis filter banks are selected dynamically from among a set of candidate filters according to the properties of the input. In the synthesis process, the corresponding exact reconstruction (ER) filter is employed adaptively in a way that allows the input to be reconstructed without error. In this chapter, we will discuss the history of time-varying filter banks (i.e. adaptive-analysis adaptive-synthesis [AAAS] filter banks) and some of the key milestones to set the stage for the new class of FAAS systems we will introduce in the next chapter.

3.1 Nayebi's Method

Time-varying filter banks were originally introduced by Nayebi et al. [26]. In this formulation, the analysis section contained a set of analysis filters where the decomposition involved switching from one filter to another within the set. Changing from one analysis filter pair to another introduces transitional discontinuities at the switching points, which in turn result in reconstruction errors. In order to reconstruct exactly, a large set of synthesis filter pairs are designed and applied sequentially. Nayebi formulated the reconstruction equations and showed that filters could be designed to satisfy the exact reconstruction condition.

The system proposed by Nayebi is best illustrated by considering a simple two-band system. This system has only two analysis filter sets as shown in Figure 12. The number of synthesis filters needed for reconstruction is dependent on the length of the analysis filters ”L” and the intervals between switching. For a single switch, 2^L synthesis filters are needed per filter set. To illustrate the inner workings of this FAAS filter bank, consider the example from [26] shown in Figure 11 where the signal shown is the input to the synthesis filters.

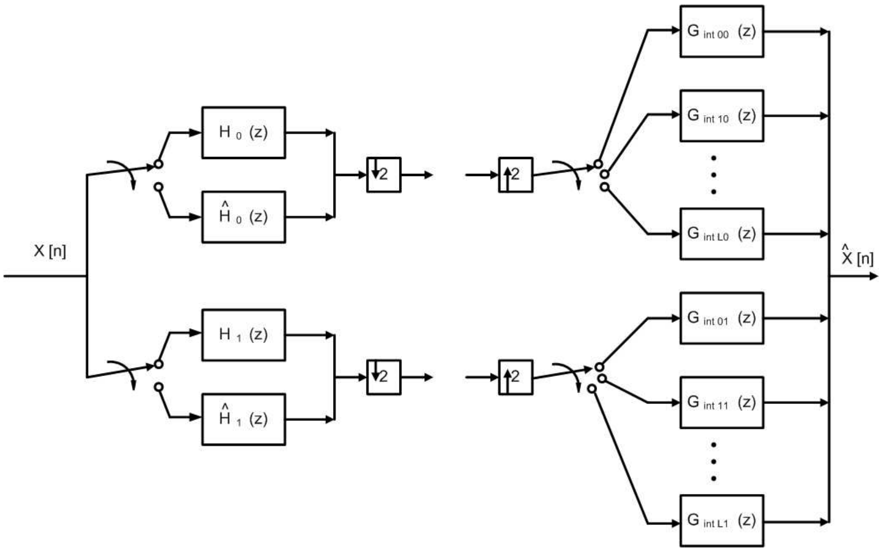


Figure 10: Two-band time-varying filter bank proposed by Nayebi et al.

The sample at n_0 is the transition sample. For this example, all points prior to n_0 were generated by the first set of analysis filters which are embodied in the convolution matrix H. The other points are generated by the second set of analysis filters, whose coefficients are contained in $\hat{H}$. To reconstruct exactly, 2^L different synthesis filter pairs are needed, one for each point in the transition. Equation (19) shows the interpretation where the transition occurs at n_0

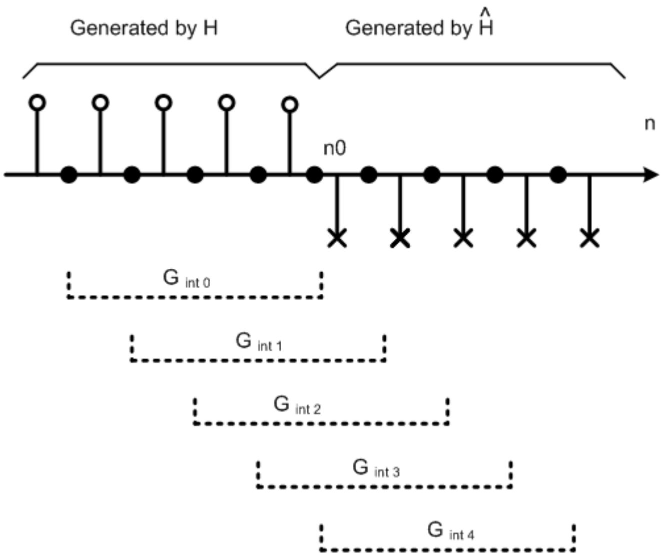


Figure 11: A diagram of analysis-synthesis combinations.

$$A = \begin{bmatrix} h[0] & \mathbf{0} & \dots & 0 \\ h[1] & h[0] & & 0 \\ \vdots & h[1] & & 0 \\ h[n_0-2] & \vdots & & 0 \\ h[n_0-1] & h[n_0-2] & \dots & 0 \\ \hat{h}[n_0] & \hat{h}[n_0-1] & \dots & 0 \\ \hat{h}[n_0+1] & \hat{h}[n_0] & \dots & 0 \\ \hat{h}[n_0+2] & \hat{h}[n_0+1] & \dots & 0 \\ \vdots & h[n_0+2] & & 0 \\ \hat{h}[N-1] & \vdots & & \hat{h}[0] \\ \mathbf{0} & \hat{h}[N-1] & & \hat{h}[1] \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \hat{h}[N-1] \end{bmatrix}. \qquad (19)$$

To find the optimal synthesis filter set $\mathbf{S}[n_0]$ using Equation (19), we obtain

$$\mathbf{S}[n_0] = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}\ . \tag{20}$$

In general, the reconstruction error can be minimized for the system by using an optimization algorithm based on the equation

$$\epsilon_r = \|\mathbf{A}\mathbf{S}[n_0] - \mathbf{B}\|_F^2. \tag{21}$$

The other synthesis filters , such as $\mathbf{S}[n_0 + 1]$, may be determined by relocating the switching point in Equation (19)

3.2 Postfiltering Method

One of the major drawbacks of the method proposed by Nayebi is that a prohibitively large number of synthesis filters had to be designed in order to accommodate switching. Sodagar et al.[43, 42, 22] proposed an important improvement that addressed the filter-bank structure and filter management complexities associated with Nayebi's approach. Specifically, it employed a dynamic FIR postfilter at the end of the analysis-synthesis filter bank. The dynamic postfilter compensates for the transitional switching distortion that results when analysis and synthesis filter pairs are switched in synchrony. Employing this dynamic postfilter, one enjoys the simplicity of being able to switch the analysis filters based on the input characteristics and then directly switch the synthesis filters in lockstep. Sodagar shows how to design these dynamic postfilters [43, 42]. An example of a two-band time-varying FIR filter bank with postfiltering is illustrated in Figure 12.

Sodagar et al. modeled the FIR analysis-synthesis filter bank as a time-varying system with a time-varying postfilter transfer function $\mathbf{T}^{-1}$. The system transfer function is given by

$$\mathbf{T} = \mathbf{S}^T\Lambda\mathbf{A} \tag{22}$$

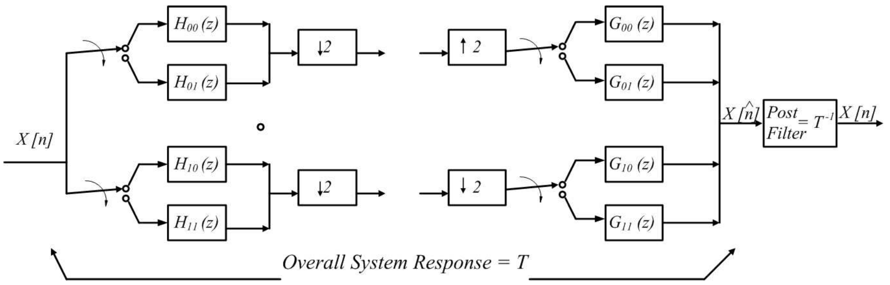


Figure 12: Time-Varying Filter Bank with Postfiltering.

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} \tag{23}$$

and

$$\Lambda = \begin{bmatrix} \lambda \\ \lambda \end{bmatrix} \tag{24}$$

for a two-band system. The submatrices shown in the equations above are defined as

$$\mathbf{H}_0 = \begin{bmatrix} h_0[0] & 0 & 0 & 0 & 0 & \dots & & 0 \\ h_0[1] & h_0[0] & 0 & 0 & 0 & \dots & & 0 \\ h_0[2] & h_0[1] & h_0[0] & & & & & 0 \\ \vdots & \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & h_0[N-2] & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & h_0[N-1] & h_0[N-2] \end{bmatrix}, \tag{25}$$

$$\mathbf{H}_1 = \begin{bmatrix} h_1[0] & 0 & 0 & 0 & 0 & \dots & & 0 \\ h_1[1] & h_1[0] & 0 & 0 & 0 & \dots & & 0 \\ h_1[2] & h_1[1] & h_1[0] & & & & & 0 \\ \vdots & \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & h_1[N-2] & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & h_1[N-1] & h_1[N-2] \end{bmatrix}, \tag{26}$$

and

$$\lambda = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{27}$$

Similar to the **A** matrix, **S** is defined as

$$\mathbf{S} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 \end{bmatrix} \tag{28}$$

where

$$\mathbf{G}_0 = \begin{bmatrix} g_0[0] & 0 & 0 & 0 & 0 & \dots & & 0 \\ g_0[1] & g_0[0] & 0 & 0 & 0 & \dots & & 0 \\ g_0[2] & g_0[1] & g_0[0] & & & & & 0 \\ \vdots & \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & g_0[N-2] & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & g_0[N-1] & g_0[N-2] \end{bmatrix} \tag{29}$$

and

$$\mathbf{G}_1 = \begin{bmatrix} g_1[0] & 0 & 0 & 0 & 0 & \dots & & 0 \\ g_1[1] & g_1[0] & 0 & 0 & 0 & \dots & & 0 \\ g_1[2] & g_1[1] & g_1[0] & & & & & 0 \\ \vdots & \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & g_1[N-2] & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & g_1[N-1] & g_1[N-2], \end{bmatrix}. \tag{30}$$

As mentioned earlier, in the absence of postfiltering, when the analysis and synthesis filters are switched in synchrony, the input **x** and output $\hat{\mathbf{x}}$ are not equal. That is,

$$\hat{\mathbf{x}} \neq \mathbf{Tx}. \tag{31}$$

Sodagar designed the postfilter to be the inverse $\mathbf{T}^{-1}$ of the time-varying system such that

$$\tilde{\mathbf{x}} = \mathbf{T}^{-1}\mathbf{T}\hat{\mathbf{x}} = \mathbf{x}. \tag{32}$$

The inverse filter, in effect, contains rows that are each time-varying postfilters. It should be noted that

$$\mathbf{T} = \mathbf{I} \tag{33}$$

when the filter bank runs in steady state (i.e. when no filter switching occurs). This implies that

$$\hat{\mathbf{x}} = \mathbf{x} \tag{34}$$

under these conditions. Only during the transition period, when the analysis and synthesis filters are switched, is $\mathbf{T} \neq \mathbf{I}$. During transition, the size of $\mathbf{T}$ is proportional to the filter lengths involved. Thus, the only requirement for perfect reconstruction is that $\mathbf{T}$ be invertible.

3.3 Comments on AAAS Filter Bank Applications

Arrowood et al.[1] explored the use of time-varying filter banks for image coding. It should be noted that during this same time, subband image coders using adaptive IIR filters were also being developed by Chung et al. in [6].

Arrowood et al.[1] examined the use of AAAS filter banks to reduce the perceived distortions encountered in low bit rate image coding. One of the most pronounced distortions visible when coding at low bit rates is *ringing*. This type of distortion is prevalent at sharp edge boundaries. Switching between different types of filters with asymmetric impulse responses and varying group delays were shown to have favorable reconstruction properties at edges. Such AAAS systems require analysis of the input image prior to decoding to determine the location of major edges.

The notion of using AAAS filter banks to address this distortion is compelling, since ringing distortion is mitigated by the filters themselves rather than by an investment in additional bits for the quantizers. The downside, however, is that synchronization information must be communicated in some way from the analysis system to the synthesis system in order to keep the filter pairs in synchrony. Synchronization can be done in either a backward adaptive or forward adaptive mode. But either way, it adds a layer of computational overhead to the encoder, which in many situations is unattractive. In addition, the synchronization requirement may result in having to transmit additional bits (as in the case of forward adaptation) or may result in the corruption of the synchronization information after coding (as in the case of backward adaptive systems). Furthermore, the postfilter becomes more complex as the interval between switching decreases and the number of switches increases. As a result, AAAS filter banks have not been widely adopted.

In the next chapter, we introduce FAAS filter banks and highlight the implementation advantages that make them attractive.

CHAPTER IV

INTRODUCTION OF FIXED-ANALYSIS ADAPTIVE-SYNTHESIS FILTER BANKS

In this chapter we introduce fixed-analysis adaptive-synthesis filter banks, the topic of this dissertation. As mentioned previously, there are four classes of filters evident from the nomenclature associated with time-varying filter banks:

- Fixed-Analysis Fixed-Synthesis Filter Banks (FAFS) - the conventional filter banks used in the majority of subband-based systems.
- Adaptive-Analysis Adaptive-Synthesis Filter Banks (AAAS) - the class of time-varying filter banks introduced by Nayebi et al. and further developed by several other authors.
- Adaptive-Analysis Fixed-Synthesis Filter Banks (AAFS) - an unexplored class of filter banks
- Fixed-Analysis Adaptive-Synthesis Filter Banks (FAAS) - an unexplored class of filter banks and topic of this thesis.

The last two classes, in contrast to the first two, are asymmetric. That is, both FAFS and AAAS are adaptive or non-adaptive in both analysis and synthesis, while AAFS and FAAS filters are only adaptive in one of the two section, and hence are asymmetric. Both of these asymmetric formulations have some appealing aspects. Both have relatively low complexity. And both can be made compatible with existing standard subband/wavelet image coders. But how do the AAFS and FAAS system compare?

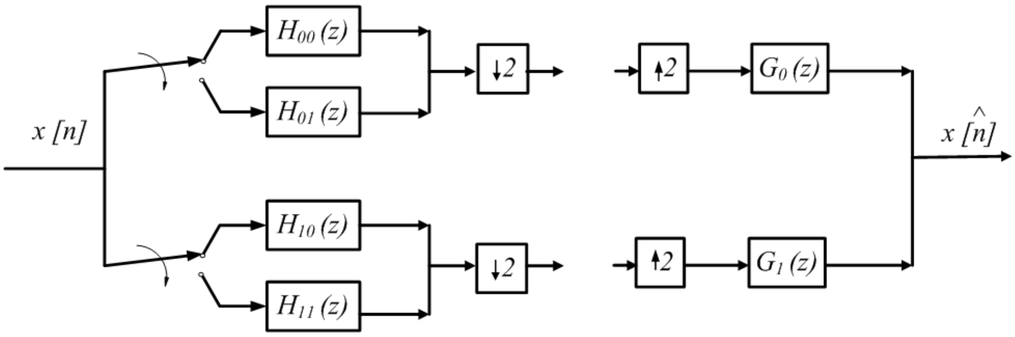


Figure 13: Basic block diagram of an adaptive analysis fixed synthesis Filter Bank.

A block diagram of the AAFS filter bank is shown in Figure 13. Because all of the adaptation occurs on the analysis side in the AAFS system, freedom to remove artifacts and compensate distortion are limited, largely because quantization occurs after the adaptive filtering. Employing the adaptive filters to correct distortions is much more intuitive if done in the synthesis section, and thus is the motivation for exploring the FAAS system.

A block diagram of the FAAS system is shown in Figure 14. What makes FAAS filter banks particularly interesting is their potential to exploit phase diversity in coding and enlargement applications. It is well known that at high bit rates the output of quantization can be modeled as the signal plus the quantization noise associated with the signal at that spatial location. The design of FAAS filters involves a multiplicity of synthesis filter pairs. To preserve continuity of the magnitude response, we design the FAAS filters to all have approximately the same magnitude response but different phases. Each of these filter pairs will generate a unique reconstruction. The differences among them are in their phase shifts. If reconstructions are performed on the same quantized signal based on a diversity of synthesis filters with different phases, the resulting reconstructions will each contain the signal plus the associated noise spatially displaced. Since spatial regions with high amplitude changes generate proportionately higher quantization noise and since this noise is now spatially shifted

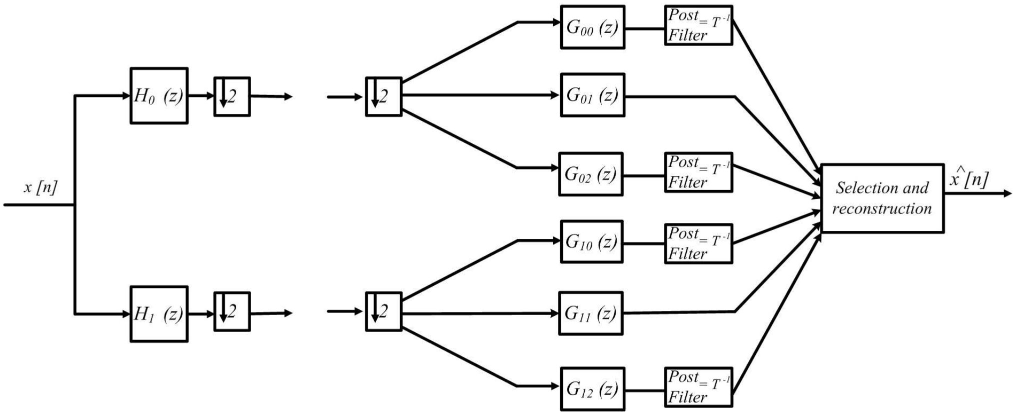


Figure 14: Basic block diagram of a fixed analysis adaptive synthesis filter bank.

across the diverse reconstructions, that noise in theory can be suppressed as part of the process of merging the images together.

For simplicity, we have considered three-phase reconstruction, where we employ a low delay, a linear phase, and a high delay filter set, which corresponds to the block diagram shown in Figure 14. Once the three reconstructions are computed, we can synthesize the final output adaptively by choosing the best pixels (on a pixel-by-pixel basis) from among the three reconstructions, thereby exploiting the phase diversity of the system.

To illustrate the exploitation of phase diversity in a fixed-analysis adaptive-synthesis system, consider the step responses for a low delay and high delay filters as shown in Figure 15. The phase dispersion results in relatively large oscillation on the back end of the step for the low delay case with no oscillation at the onset of the edge.

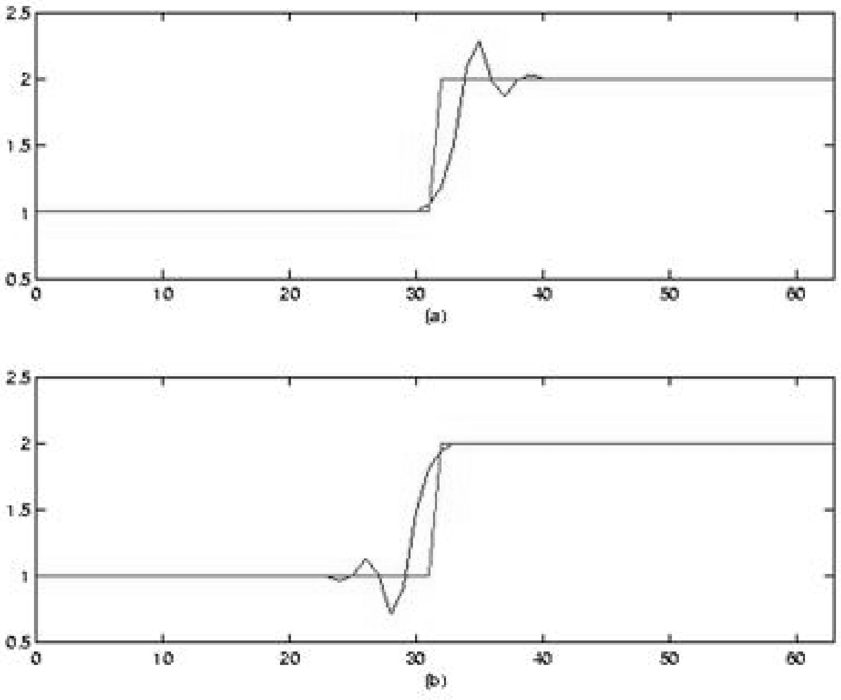


Figure 15: (a)low delay lowpass filter step response. (b)high delay lowpass filter step response.

The reverse characteristics are apparent for the high delay filter case. For the linear phase case (not shown in the figure), the oscillations are smaller in amplitude but are distributed evenly on both sides of the edge. Each of these outputs is shifted spatially by different amounts dictated by the characteristics of the filters.

One can quickly see that going pixel by pixel we can reconstruct the original step function. The three reconstructed signals will result in three different displacements and three dispersions for the noise (relative to the image), which potentially can be exploited.

The fixed-analysis adaptive-synthesis filter bank reported here is based on designing a set of synthesis filters, where each synthesis filter set is optimized to have minimum reconstruction distortion and a different overall system group delay. Of these filters, only one will achieve ER with the fixed analysis filter.

There are potentially many methods for synthesizing the final image. The method we pursued primarily is based on an empirically derived rule based on reducing the filtering overshoots and undershoots that appear in the image when coded at low bit rates. In practice, it is not necessary to reconstruct all of the images in the synthesis section, but only these relevant sub-regions of the image. Thus computation complexity in the synthesis section can be minimized. In the next chapter we consider the design of the filter sets, which is critical to obtaining high quality performance in compression and enlargement applications.

CHAPTER V

FAAS FILTER BANKS FILTER DESIGN

In this chapter, we introduce a design method for the FAAS filter bank, which is based on time-domain reconstruction equations. The formulation provides control over the critical filter characteristics and system properties such as analysis filter ripple, transition bandwidth, reconstruction fidelity, system group delay, and more. The method is based on a matrix equation representation where analysis and synthesis filter coefficients are optimized iteratively to reduce a weighted cost function. There are two classes of FIR analysis-synthesis filters typically employed for subband decompositions: even-length filters and odd-length filters. Nayebi et al. developed the design method for the even-length case, which we present next for completeness. The odd-length case, which is markedly different is new and is introduced in the subsequent subsection. A design example is included to illustrate the effectiveness of the new method.

5.1 Even-Length Filters

Nayebi proposed the time-domain formulation for ER given by $AS = B$ where

$$\underbrace{\begin{bmatrix} \mathbf{P}_0^T & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}_1^T & \mathbf{P}_0^T & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{P}_2^T & \mathbf{P}_1^T & \mathbf{P}_0^T & & \vdots \\ \mathbf{P}_3^T & \mathbf{P}_2^T & \mathbf{P}_1^T & \ddots & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{P}_{L-1}^T & \mathbf{P}_{L-2}^T & \mathbf{P}_{L-3}^T & \cdots & \mathbf{P}_0^T \\ \mathbf{0} & \mathbf{P}_{L-1}^T & \mathbf{P}_{L-2}^T & \ddots & \mathbf{P}_1^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{L-1}^T \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{Q}_0 \\ \\ \mathbf{Q}_1 \\ \\ \vdots \\ \\ \mathbf{Q}_{L-2} \\ \\ \mathbf{Q}_{L-1} \end{bmatrix}}_{\mathbf{S}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \\ \mathbf{J}_R \\ \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}}. \tag{35}$$

In these equations **A** is a block Toeplitz matrix of analysis filter coefficients, **S** is a matrix of synthesis filter coefficients, and matrix **B** is the reconstruction matrix containing the exchange matrix $\mathbf{J}_R$. $\mathbf{J}_R$ controls the group delay characteristics of the system being designed. Matrix **A** is expressed in terms of submatrix **P**, where

$$\mathbf{P} = [\mathbf{P}_0|\mathbf{P}_1|\ldots|\mathbf{P}_{L-1}]$$

and

$$\mathbf{P}_i^T = [h_0(i)h_1(i)].$$

Matrix **S** is expressed in terms of submatrix **Q**, where

$$\mathbf{Q} = [\mathbf{Q}_0|\mathbf{Q}_1|\ldots|\mathbf{Q}_{L-1}].$$

To be more specific,

$$\mathbf{Q}_0 = [g_0(0)g_0(1)],$$

$$\mathbf{Q}_1 = [g_1(0)g_1(1)],$$

$$\mathbf{Q}_2 = [g_0(2)g_0(3)],$$

$$\mathbf{Q}_3 = [g_1(2)g_1(3)],$$

and so on. Finally, the exchange matrix J_r is defined as

$$\mathbf{J}_R = \begin{bmatrix} 0 & 0 & \ldots & 1 \\ 0 & & \cdot & 0 \\ \vdots & \cdot & & 0 \\ 1 & 0 & \ldots & 0 \end{bmatrix}. \tag{36}$$

As stated before, the position of the submatrix $\mathbf{J}_R$ in $\mathbf{B}$ controls the system delay. If matrix $\mathbf{J}_R$ is moved to the top of $\mathbf{B}$, the analysis-synthesis filter bank realizes a minimum delay system. Similarly, if $\mathbf{J}_R$ is moved to the bottom, the result is a maximum delay system.

To illustrate the design method, an example is presented next. Suppose we wish to design a two-band maximally decimated filter bank with even-length 6-tap analysis-synthesis filters with a group delay of 6 samples. The analysis lowpass filter is

$$h_0 = [h_0[0], h_0[1], h_0[2], h_0[3], h_0[4], h_0[5]] \tag{37}$$

and the analysis highpass filter is

$$h_1 = [h_1[0], h_1[1], h_1[2], h_1[3], h_1[4], h_1[5]]. \tag{38}$$

Similarly, the synthesis filter coefficients are

$$g_0 = [g_0[0], g_0[1], g_0[2], g_0[3], g_0[4], g_0[5]] \tag{39}$$

and

$$g_1 = [g_1[0], g_1[1], g_1[2], g_1[3], g_1[4], g_1[5]]. \tag{40}$$

The corresponding analysis submatrices are given by

$$\mathbf{P}_0^T = [h_0(0)h_1(0)],$$

$$\mathbf{P}_1^T = [h_0(1)h_1(1)],$$

and so on. Similarly, the synthesis submatrices are given by

$$\mathbf{Q}_0 = [g_0(0)g_0(1)],$$

$$\mathbf{Q}_1 = [g_1(0)g_1(1)],$$

$$\mathbf{Q}_2 = [g_0(2)g_0(3)],$$

$$\mathbf{Q}_3 = [g_1(2)g_1(3)],$$

and so on until we reach to $\mathbf{Q}_5$. Expressed strictly in terms of the analysis and synthesis coefficients, the reconstruction equation becomes

$$\begin{bmatrix} h_0[0] & h_1[0] & 0 & 0 & 0 & 0 \\ h_0[1] & h_1[1] & 0 & 0 & 0 & 0 \\ h_0[2] & h_1[2] & h_0[0] & h_1[0] & 0 & 0 \\ h_0[3] & h_1[3] & h_0[1] & h_1[1] & 0 & 0 \\ h_0[4] & h_1[4] & h_0[2] & h_1[2] & h_0[0] & h_1[0] \\ h_0[5] & h_1[5] & h_0[3] & h_1[3] & h_0[1] & h_1[1] \\ 0 & 0 & h_0[4] & h_1[4] & h_0[2] & h_1[2] \\ 0 & 0 & h_0[5] & h_1[5] & h_0[3] & h_1[3] \\ 0 & 0 & 0 & 0 & h_0[4] & h_1[4] \\ 0 & 0 & 0 & 0 & h_0[5] & h_1[5] \end{bmatrix} \begin{bmatrix} g_0[0] & g_0[1] \\ g_1[0] & g_1[1] \\ g_0[2] & g_0[3] \\ g_1[2] & g_1[3] \\ g_0[4] & g_0[5] \\ g_1[4] & g_1[5] \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{41}$$

Starting with a reasonable best guess for the analysis coefficients, the synthesis coefficients are obtained from the equation

$$\mathbf{S} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}. \tag{42}$$

The reconstruction error (ϵ_r) can be computed at this point using the equation

$$\epsilon_r = \|\mathbf{AS} - \mathbf{B}\|_F^2 \tag{43}$$

where $||.||_F^2$ is the Frobenius norm defined by the equation

$$\mathbf{X} = \sum_i \sum_j |x_{ij}|^2. \tag{44}$$

Through the use of iterative optimization of the analysis filter coefficients to minimize ϵ_r, coefficients for the filter bank can be obtained. The **A** matrix has a dimension of 10×6. The **S** matrix has a dimension of 6×2. This leads to a **B** matrix that is 10×2. Considering this, the **A** matrix has $2L - R$ rows and LM/R columns, and the S matrix has LM/R rows and L columns. This leads to a **B** matrix of $2L - R$ rows and R columns with a possible minimum delay of $R - 1$ samples and a maximum delay of $(2L - 1)R - 1$ samples.

Although the reconstruction error is minimized, the filter's frequency domain characteristics may not be of high quality. Consequently, a cost function is included in the error term to assure that the frequency response characteristics are optimized. Specifically, we use a frequency error component s^2 where

$$s_t^2 = 1/2 * s_{low}^2 + 1/2 * s_{high}^2. \tag{45}$$

For a two-band system, the frequency error component can be decomposed into two frequency error components. One component, s_{low}^2, associated with the lowpass frequency error where

$$s_{low}^2 = \underbrace{\frac{1}{N/2}\sum_{k=1}^{\frac{N}{2}}(|h_x(k)| - \sqrt{2})^2}_{passband\ energy} + \underbrace{\frac{1}{N/2}\sum_{k=\frac{N}{2}+1}^{N}(|h_x(k)| - 0)^2}_{stopband\ ripple} \tag{46}$$

and one component, s_{high}^2,where

$$s_{high}^2 = \underbrace{\frac{1}{N/2}\sum_{k=1}^{\frac{N}{2}}(|h_x(k)| - 0)^2}_{stopband\ energy} + \underbrace{\frac{1}{N/2}\sum_{k=\frac{N}{2}+1}^{N}(|h_x(k)| - \sqrt{2})^2}_{passband\ ripple} \tag{47}$$

associated with the highpass frequency error. Each component has a term that controls the passband deviation and stopband ripple. For optimization we use a weighted error of the form

$$\epsilon_t = \alpha * \epsilon_r + (1 - \alpha) * s_t^2. \tag{48}$$

More details associated with the method may be found in [30, 27, 29, 28, 31]. The flexibility inherent in the method allow users to balance the many tradeoffs faced when designing filters.

5.2 Designing Odd-Length Filters

In this section, we develop the design method for odd-length filters, which heretofore has not been done. Many of the best known filters used in subband image coding are odd-length filters. The most popular of these filters are the Daubechies 9/7 filters and the LeGall 5/3 filters. The popularity is due to their superior performance over their even-length counterpart. In addition, odd-length linear phase filters do not exhibit fractional delay in the subband images, which makes them attractive.

Much like the previously developed even-length time-domain approach, this new approach employs a matrix equation to ensure reconstruction conditions, but embodies some notable differences. For simplicity and without loss of generality, this new method is presented for a maximally decimated filter bank.

Unlike the even-length case, the constituent filters in an odd-length filter bank have different lengths. This is a consequence of reconstruction conditions and perhaps evident through the product filter. Recall from Section 5.1, that the product filter must be a halfband filter where all even coefficients are zero except the mid-point coefficient. This implies that product filters are constrained to have lengths of $L_p = 3 + 4(i)$, where $i = 0, 1, 2, ...$ Thus the product filter length L_p can be 3,7,11,15, and so on. To help visualize this, consider Figure 16. These are the coefficients of the product filter of the Daubichies 9/7 filter (15-tap lowpass product filter). Through quick inspection we can see that all even coefficients are equal to zero except at $n = 0$. This means we can never have a linear phase product filter that is of length 1,5,9,13,

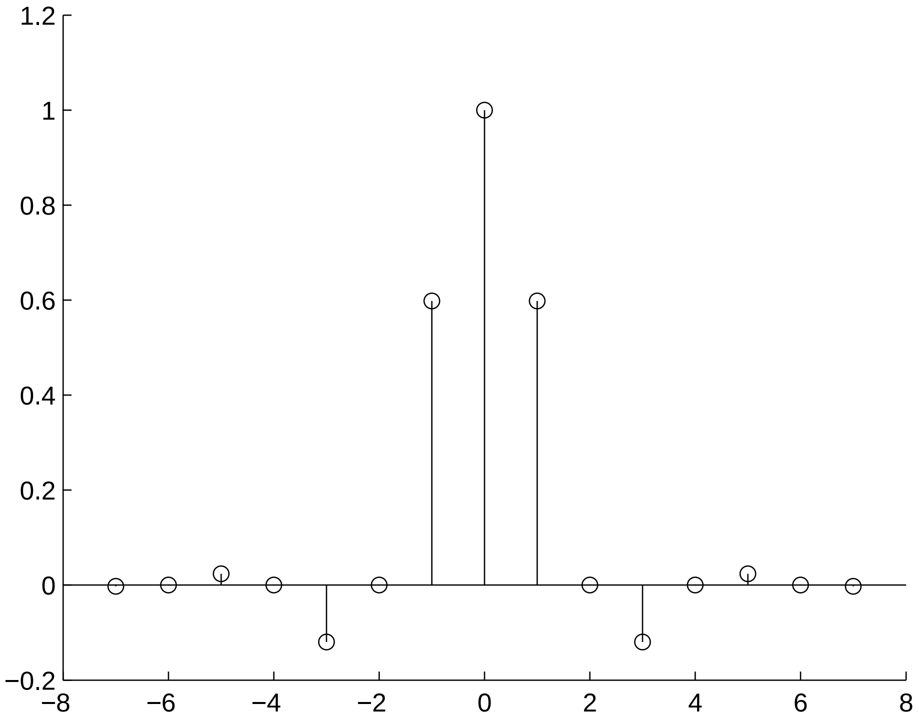


Figure 16: Daubichies 9/7 lowpass product filter.

and so on because the end coefficients would be zero. This would also be true for low delay and high delay product filters. Decomposing a product filter of length $Lp = 3 + 4(i)$ into two odd-length filters clearly cannot be done unless the lengths are different, which dictates a different set of design equations.

For odd-length filters we define the reconstruction equation as follows:

$$\underbrace{\begin{bmatrix} \mathbf{P}_0^T & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_1^T & \mathbf{P}_0^T & \mathbf{0} & \vdots & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_2^T & \mathbf{P}_1^T & \mathbf{P}_0^T & \vdots & \vdots & \\ \mathbf{P}_3^T & \mathbf{P}_2^T & \mathbf{P}_1^T & \ddots & \ddots & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ \mathbf{P}_{K-1}^T & \mathbf{P}_{K-2}^T & \mathbf{P}_{K-3}^T & \dots & \mathbf{P}_0^T & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{K-1}^T & \mathbf{P}_{K-2}^T & \ddots & \mathbf{P}_1^T & h_0(0) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{K-1}^T & h_0(L-1) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{Q}_0 \\ \\ \mathbf{Q}_1 \\ \\ \vdots \\ \\ \mathbf{Q}_{K-2} \\ \\ \mathbf{Q}_{K-1} \end{bmatrix}}_{\mathbf{S}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \\ \mathbf{J}_R \\ \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}}, \tag{49}$$

where $\mathbf{A}$ is a block Toeplitz matrix of analysis filter coefficients, $\mathbf{S}$ is a matrix of synthesis filter coefficients, and matrix $\mathbf{B}$ is the reconstruction matrix containing the exchange matrix $\mathbf{J}_R$. $\mathbf{J}_R$ controls the group delay characteristics of the system being designed. Matrix $\mathbf{A}$ is expressed in terms of $\mathbf{P}$, where

$$\mathbf{P} = [\mathbf{P}_0 | \mathbf{P}_1 | \dots | \mathbf{P}_{K-1}]$$

Matrix $\mathbf{S}$ is expressed in terms of $\mathbf{Q}$, where

$$\mathbf{Q} = [\mathbf{Q}_0 | \mathbf{Q}_1 | \dots | \mathbf{Q}_{K-1}].$$

The length parameter in Equation (49) is set to the length of the larger filter and is denoted as K. To adjust the shorter filter to length K, the shorter filter is zero-padded at the back end.

To create a more lucid picture of equation structure consider the following. Suppose we wish to design a two-band filter bank with 7-tap and 5-tap analysis-synthesis filters with a 6 sample group delay. The analysis lowpass filter is

$$h_0 = [h_0[0], h_0[1], h_0[2], h_0[3], h_0[4]] \tag{50}$$

and the analysis highpass filter

$$h_1 = [h_1[0], h_1[1], h_1[2], h_1[3], h_1[4], h_1[5], h_1[6]]. \tag{51}$$

It is our goal to maintain alias cancelation for this system and thus,

$$G_0(z) = H_1(-z) \tag{52}$$

and

$$G_1(z) = -H_0(-z). \tag{53}$$

Given Equations (52) and (53) the synthesis filter coefficients are of length 7 and 5, where the synthesis lowpass filter is

$$g_0 = [g_0[0], g_0[1], g_0[2], g_0[3], g_0[4], g_0[5], g_0[6]] \tag{54}$$

and the synthesis highpass filter is

$$g_1 = [g_1[0], g_1[1], g_1[2], g_1[3], g_1[4]] \tag{55}$$

In terms of the submatrices,

$$\mathbf{P}_1^T = [h_0(1)h_1(1)],$$

$$\mathbf{P}_2^T = [h_0(2)h_1(2)],$$

and so on, and

$$\mathbf{Q}_0 = [g_0(0)g_0(1)],$$

$$\mathbf{Q}_1 = [g_1(0)g_1(1)],$$

$$\mathbf{Q}_2 = [g_0(2)g_0(3)],$$

$$\mathbf{Q}_3 = [g_1(2)g_1(3)],$$

$$\vdots$$

In terms of the coefficients, Equation (49) becomes

$$\begin{bmatrix} h_0[0] & h_1[0] & 0 & 0 & 0 & 0 & 0 \\ h_0[1] & h_1[1] & 0 & 0 & 0 & 0 & 0 \\ h_0[2] & h_1[2] & h_0[0] & h_1[0] & 0 & 0 & 0 \\ h_0[3] & h_1[3] & h_0[1] & h_1[1] & 0 & 0 & 0 \\ h_0[4] & h_1[4] & h_0[2] & h_1[2] & h_0[0] & h_1[0] & 0 \\ 0 & h_1[5] & h_0[3] & h_1[3] & h_0[1] & h_1[1] & 0 \\ 0 & h_1[6] & h_0[4] & h_1[4] & h_0[2] & h_1[2] & h_0[0] \\ 0 & 0 & 0 & h_1[5] & h_0[3] & h_1[3] & h_0[1] \\ 0 & 0 & 0 & h_1[6] & h_0[4] & h_1[4] & h_0[2] \\ 0 & 0 & 0 & 0 & 0 & h_1[5] & h_0[3] \\ 0 & 0 & 0 & 0 & 0 & h_1[6] & h_0[4] \end{bmatrix} \begin{bmatrix} g_0[0] & g_0[1] \\ g_1[0] & g_1[1] \\ g_0[2] & g_0[3] \\ g_1[2] & g_1[3] \\ g_0[4] & g_0[5] \\ g_1[4] & 0 \\ g_0[6] & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{56}$$

For any given set of analysis filters, the synthesis filter coefficients can be calculated using (42). The reconstruction error (ϵ_r) can also be calculated using Equation (44).

For this example, the **A** matrix has a dimension of 11 × 7. The **S** matrix has a dimension of 7 × 2. This results in a **B** matrix that is 11 × 2. For the general odd-length case, the **A** matrix has $2K - R - (K \ \ mod \ \ R)$ rows and $rnd(M\frac{K}{R})$ columns. The **S** matrix has $rnd(M\frac{K}{R})$ and R columns. This means that **B** has dimension $2K - R - (K \ \ mod \ \ R)$ by R. In order to specify the system group delay, $\mathbf{J}_R$ is placed in the $\ell - 1$ position where ℓ is the desired system delay. Thus, for a system delay of 6 samples, $\mathbf{J}_R$ is placed in the 5 position from the top of **B** as shown in Equation (56).

The frequency domain characteristics of the filter can be handled as before using a cost function

$$s_t^2 = 1/2 * s_{low}^2 + 1/2 * s_{high}^2, \tag{57}$$

where s_{low}^2, is lowpass frequency error given by Equation (46) and, s_{high}^2, given by

Equation (47) for the highpass component. The analysis filters are then optimized iteratively to minimize the weighted cost function given by Equation (48).

5.2.1 Odd-Length Filter Design Example

In this subsection we present an example. We begin with the Daubechies 9/7 product filter whose pole-zero plot is shown in Figure 17. Our objective is to develop a new set of odd-length near-linear phase filters. We know from [40] that near-linear phase filters can be obtained by factoring the product filter such that the analysis filter contains selectively distributed zeros from the product filter. The synthesis filter contains the remaining zeros from the product filter. Consider the lowpass Daubechies 9/7 pole-zero plot in Figure 17. It is possible to factor the filter into two near-linear phase odd-length filters whose pole-zero plots are shown in Figure 20(a) and Figure 20(b).

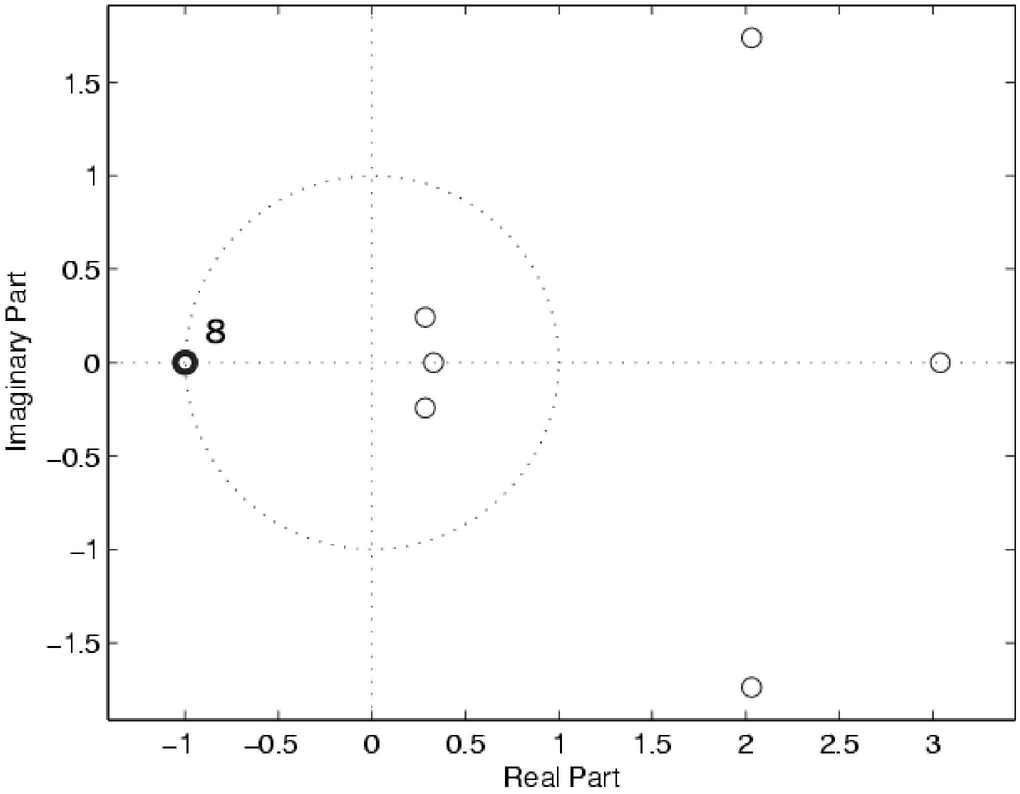
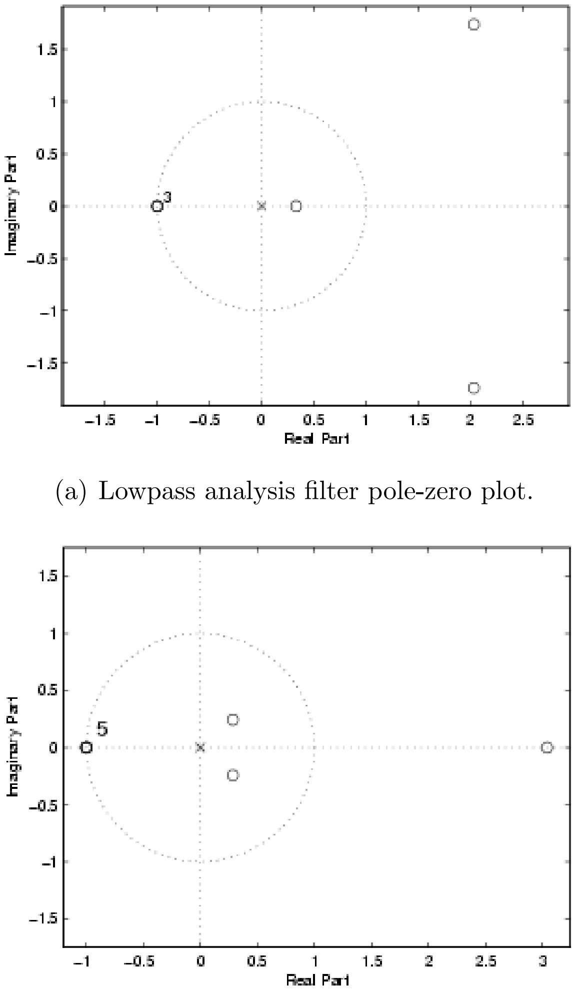


Figure 17: Pole-zero plot for Daubechies 9/7 product filter.



(a) Lowpass analysis filter pole-zero plot.

(b) Lowpass synthesis filter pole-zero plot.

Figure 18: Pole-zero plots for 9/7 lowpass near-linear phase filters.

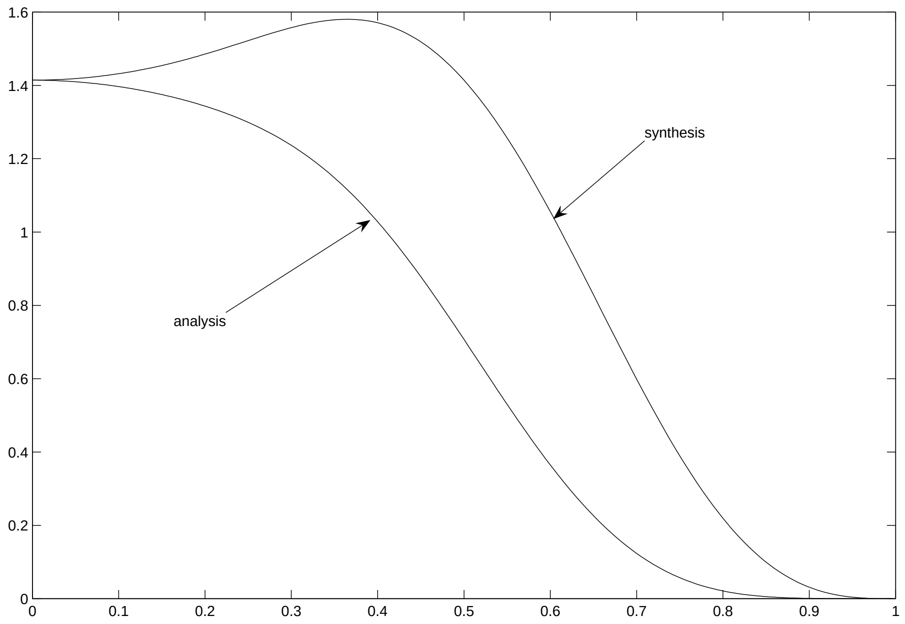
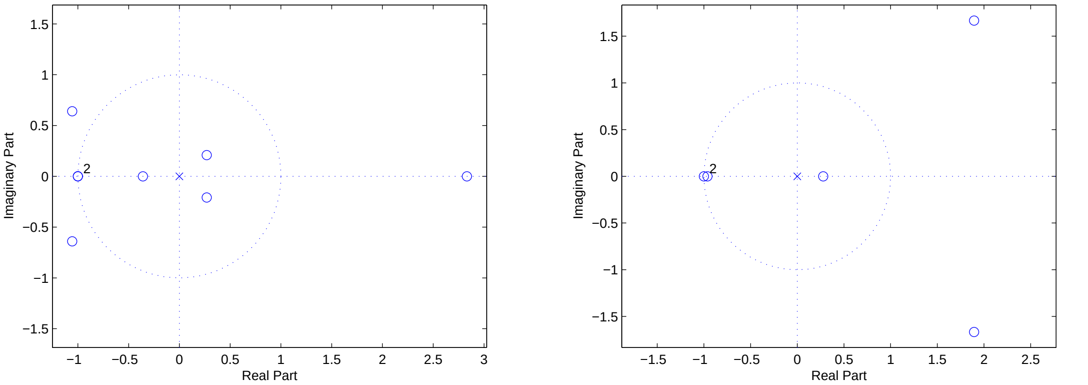


Figure 19: Frequency plot for 9/7 lowpass near-linear phase filters.

Although these filters were designed from the Daubichies 9/7 and combine to create a system having a reconstruction error on the order of 10^{-14}, the frequency response characteristics are degraded as shown in Figure 19. The degradation is not huge compared to the original but it is noticeable. There is a larger overshoot in the lowpass synthesis filter and a larger undershoot in the analysis lowpass filter.

Employing the new time-domain design method presented in Section 5.2, we were able to develop the filters whose pole-zero plots are shown in Figure 20 and whose frequency response plots are shown in Figure 21. The passband characteristics are noticably improved with only a small increase in reconstruction error. The resulting reconstruction error is $1.366875369975381 \times 10^{-12}$ which is negligible. The coefficients for the new filters are shown in Table 1.

In summary, we have introduced a design method for odd-length filters that provides control over filter frequency characteristics and system reconstruction error. Similar to Nayebi's approach, this method uses a time-domain approach coupled with



(a) Lowpass analysis filter's pole-zero plot. (b) Lowpass synthesis filter's pole-zero plot.

Figure 20: Pole-zero plot for 9/7 ER near-linear phase filters.

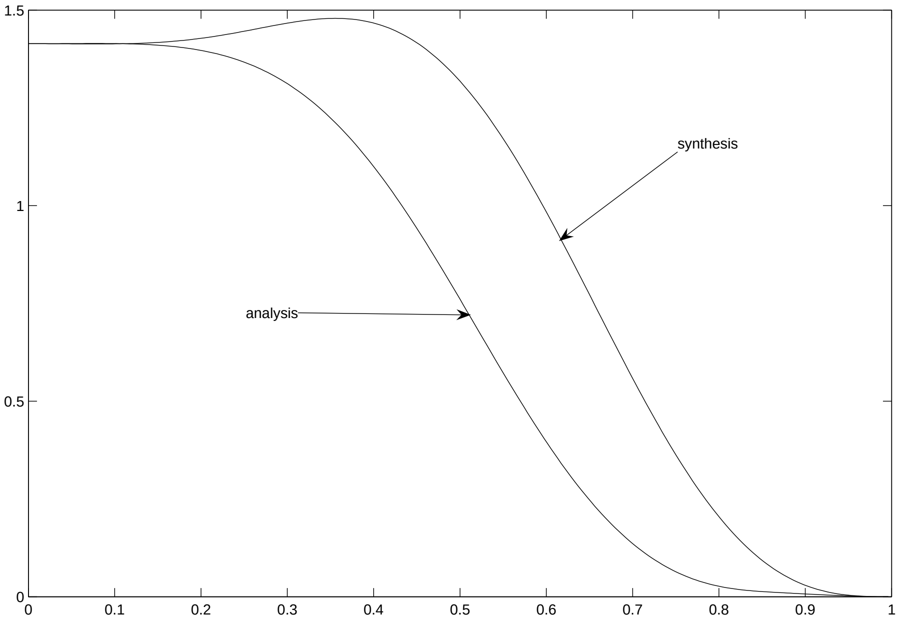


Figure 21: Frequency plot for 9/7 ER near-linear phase filters.

H_0	H_1
0.05291311404179	-0.06971017838208
0.05838675082778	-0.07691802049972
-0.27517187079671	0.11840024237207
-0.69542616558931	0.64682815800842
-0.53910281122032	-0.87421545720016
-0.05908080620355	0.13719664367785
0.06374250317152	0.11841861202362
-0.01098656022147	0
-0.00948771638282	0

Table 1: Coefficients for the near-linear phase odd-length 9/7 filters.

a frequency error component and also provides explicit control over the system delay. The most important fact about this new method is that it allows for the design of odd-length filters in the time-domain that may be able to compete against commonly used filters such as the Daubechies 9/7 and LeGall 5/3 filters.

CHAPTER VI

FAAS SYSTEM DESIGN FOR ADAPTIVE SYNTHESIS

In the previous chapter, we discussed time-domain filter design and introduced a design method to handle analysis-synthesis filters of uneven length. Such flexibility is necessary when considering odd-length filters. In this chapter, we address the design of FAAS filters optimized for high quality adaptive synthesis. Recall from Chapter 4, that for convenience we are focusing on FAAS systems with three synthesis filter sets that work in conjunction with one analysis filter set to reconstruct images and reduce distortion. Thus, we need to design a system with a fixed analysis filter set and with three filter sets that can be used to reconstruct images and reduce distortion that is visible in low bit-rate subband/wavelet coders. In the remainder of this chapter, we introduce the method to design a complete filter set with even-length and odd-length filters and show a few examples.

The FAAS system is shown again in Figure 22 for convenience. In this particular configuration of the FAAS filter bank, which is the one we explore in detail as part of this thesis, the analysis filter set is fixed and reconstruction is based on three synthesis filter sets. The identity property of the analysis-synthesis reconstruction equations indicates that for a given analysis filter pair, only one synthesis filter pair can reconstruct the input exactly in the absence of subband modification. Given this constraint, the system design task is to design

$$(G_{00}, G_{10}), (G_{01}, G_{11}), (G_{02}, G_{12})$$

with respect to (H_0, H_1) to

(a) minimize the error over the course of operation, and

(b) improve the subjective quality when subbands are quantized.

Objective (b) does not lend itself to a numerical formulation and typically requires observation and experimentation. Objective (a) however, can be addressed mathematically as discussed next.

The three synthesis filter sets can be designed collectively with a common analysis filter.

$$\mathbf{S}_0 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}_0 \tag{58}$$

$$\mathbf{S}_1 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}_1 \tag{59}$$

$$\mathbf{S}_2 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}_2 \tag{60}$$

Equation (58) is used to obtain the low delay synthesis filters. Equation (59) is used to obtain the linear phase synthesis filters. Similarly, Equation (60) is used to obtain the high delay synthesis filters. Using Equation (43), a reconstruction error is computed for each analysis-synthesis combination given by

$$\epsilon_{r0} = \|\mathbf{AS}_0 - \mathbf{B}_0\|_F^2 \tag{61}$$

$$\epsilon_{r1} = \|\mathbf{AS}_1 - \mathbf{B}_1\|_F^2 \tag{62}$$

and

$$\epsilon_{r2} = \|\mathbf{AS}_2 - \mathbf{B}_2\|_F^2 \tag{63}$$

where ϵ_{r0}, ϵ_{r1}, and ϵ_{r2} are the low delay, linear phase, and high delay reconstruction errors, respectively. The filter frequency domain characteristics should also be optimized for each adaptive synthesis set. These errors are given by

$$s_{00} = \frac{1}{N}\sum_{k=1}^{\frac{N}{2}}(|G_{00}(k)| - \sqrt{2})^2 + \frac{1}{N}\sum_{k=\frac{N}{2}+1}^{N}|G_{00}(k)|^2, \tag{64}$$

$$s_{01} = \frac{1}{N}\sum_{k=1}^{\frac{N}{2}}(|G_{01}(k)| - \sqrt{2})^2 + \frac{1}{N}\sum_{k=\frac{N}{2}+1}^{N}|G_{01}(k)|^2, \tag{65}$$

$$s_{02} = \frac{1}{N}\sum_{k=1}^{\frac{N}{2}}(|G_{02}(k)| - \sqrt{2})^2 + \frac{1}{N}\sum_{k=\frac{N}{2}+1}^{N}|G_{02}(k)|^2 \tag{66}$$

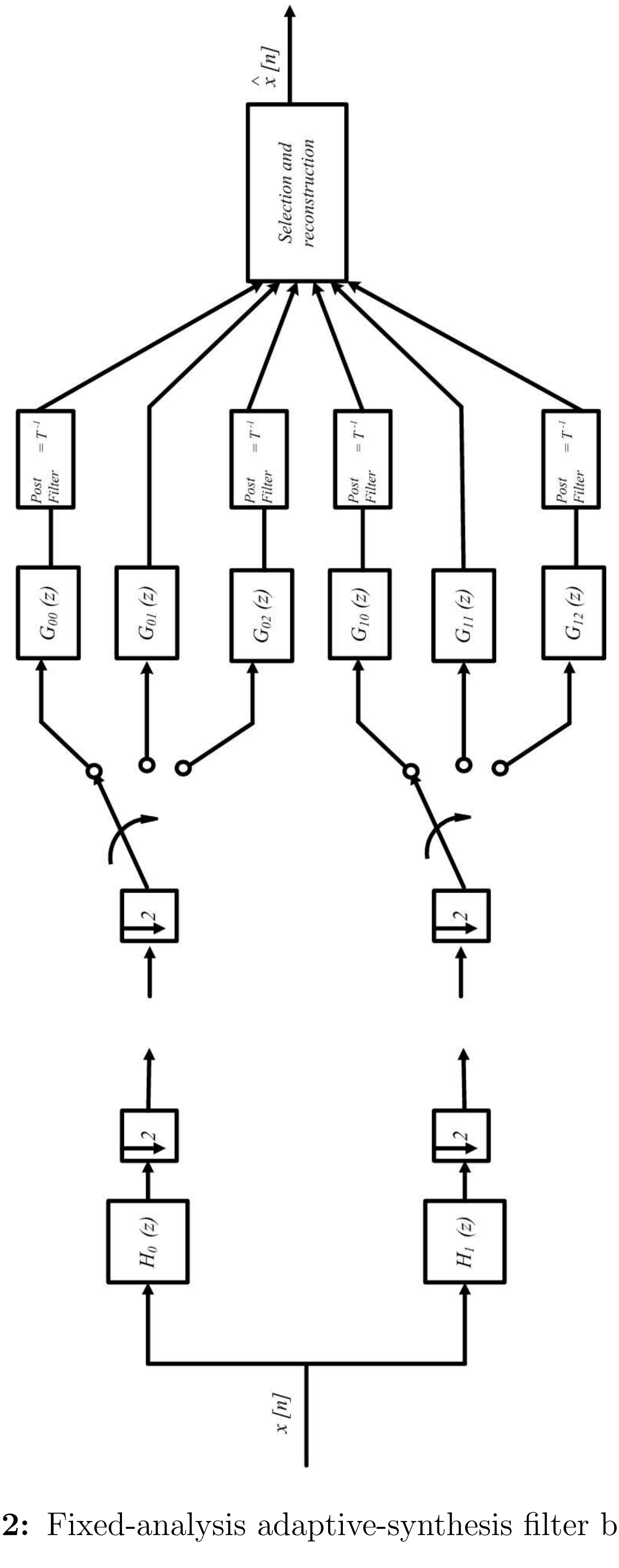


Figure 22: Fixed-analysis adaptive-synthesis filter bank structure.

for the lowpass filters and

$$s_{10} = \frac{1}{N}\sum_{k=1}^{\frac{N}{2}} |G_{10}(k)|^2 + \frac{1}{N}\sum_{k=\frac{N}{2}+1}^{N} (|G_{10}(k)| - \sqrt{2})^2, \tag{67}$$

$$s_{11} = \frac{1}{N}\sum_{k=1}^{\frac{N}{2}} |G_{11}(k)|^2 + \frac{1}{N}\sum_{k=\frac{N}{2}+1}^{N} (|G_{11}(k)| - \sqrt{2})^2, \tag{68}$$

$$s_{12} = \frac{1}{N}\sum_{k=1}^{\frac{N}{2}} |G_{12}(k)|^2 + \frac{1}{N}\sum_{k=\frac{N}{2}+1}^{N} (|G_{12}(k)| - \sqrt{2})^2 \tag{69}$$

for the highpass filters.

With the reconstruction errors and frequency domain errors isolated as error components a total weighted error can be calculated. We can specify the tradeoff between reconstruction error and frequency domain error via the equation

$$\epsilon_t = \alpha * \epsilon_r + (1 - \alpha) * s, \quad 0 < \alpha < 1 \tag{70}$$

where α is the weighting factor and

$$\mathrm{s} = \sum_i \sum_j |s_{ij}|. \tag{71}$$

To specify the reconstruction error tradeoff among the three synthesis filter pairs we can again employ the use of weighting factors,

$$\beta_0 + \beta_1 + \beta_2 = \beta, \quad \beta = 1 \tag{72}$$

where

$$0 \leq \beta_0, \beta_1, \mathrm{and} \beta_2 \leq 1. \tag{73}$$

The composite weighted reconstruction error then becomes

$$\epsilon_r = \beta_0 * \epsilon_{r0} + \beta_1 * \epsilon_{r1} + \beta_2 * \epsilon_{r2} \tag{74}$$

To specify the tradeoff in frequency-domain error among the synthesis filters, we can employ a similar set of constrained weights γ_0, γ_1, γ_2 resulting in the composite weighted error term

$$s_i = \gamma_0 * s_{i0} + \gamma_1 * s_{i1} + \gamma_2 * s_{i2}. \tag{75}$$

One could ascribe preferential weightings to the lowpass and highpass filters within each synthesis filter set, but for this investigation we constrained them to be equally weighted. The degrees of freedom present in the implied design process are many - too many in fact. How then does one specify the weights

$$\alpha, \beta_0, \beta_1, \beta_2, \gamma_0, \gamma_1, \gamma_2?$$

For the gamma values that govern the relative emphasis of frequency domain error among the synthesis filters, uniform weighting was used, since there is no rationale for emphasizing one over another.

For the beta values that govern the relative reconstruction error, values were chosen to minimize overall reconstruction error. In most cases, the selection/reconstruction algorithm will employ the linear phase filters. When image edges are encountered (which are not that often relatively speaking), the minimum and high delay filters are used. Thus, we reason that the reconstruction errors should be weighted in proportion to the frequency of use of the given filter in the reconstruction process. In this way the overall reconstruction error is reduced.

For the α value that governs the tradeoff between reconstruction and frequency domain errors, we relied on settings that worked well in previous AAAS experiments.

A remaining issue to address in the design process is the position of the exchange matrix in **B** of Equations (58), (59), and (60). To take advantage of the greatest spread in phase diversity we use a linear phase analysis filter in conjunction with a low delay, a linear phase, and a high delay synthesis filter set. The choice of a linear phase synthesis filter was motivated by the desire to be compatible with existing subband/wavelet coders. To achieve the low delay, linear phase, and high delay design specifications the exchange matrix in **B** was placed in the first, middle, and last positions respectively in **B**. The resulting system group delays are

$$\ell_1 = n, \tag{76}$$

$$\ell_0 = n - integer(n/2), \tag{77}$$

and

$$\ell_2 = n + integer(n/2), \tag{78}$$

where n is the order of the filter, integer(x) is the integer portion of the argument, ℓ_1 is the group delay that yields linear phase, ℓ_0 is the group delay that yields a combination of a linear phase filter for the analysis subsystem and a low delay synthesis subsystem, and ℓ_2 is the group delay obtained from the combination of a linear phase filter for the analysis subsystem and a high delay synthesis subsystem.

Minimizing the cost function in Equation (70), in combination with the given group delays, leads to a set of FAAS filters. Designing in this manner allows us to meet the design goals and attain a system compatible with current systems. In the following sections, we will take a closer look at the design methodology and its use for designing even-length and odd-length filters for FAAS systems.

6.1 Even-Length Adaptive Synthesis Filter Design

The multi-filter synthesis structure inherent in FAAS filter banks provides the potential to exploit phase diversity as mentioned previously. In Chapter 5, we provided an example to illustrate the design of a single optimized analysis-synthesis filter bank (with even or uneven length). In the next subsections, we walk through the design of an even-length and odd-length FAAS filter bank, as a system of one fixed analysis filter pair and three lowpass/highpass synthesis filter pairs.

Even-length Example

Even length filters, such as QMFs, are still employed in many image compression codecs. Here we consider the design of a compatible FAAS filter bank. For this system, the filters are 7th order (n) filters. Given this, the linear phase analysis and the linear phase synthesis filter combination yield a system group delay of seven samples. Each filter set contributes a group delay of approximately 3.5 samples. Using Equation (77), the linear phase analysis filter in combination with the low delay synthesis filter yields a system group delay of 4 samples. This is obtained from the 3.5 sample delay of the analysis filter and 0.5 sample delay from the synthesis filter. Finally, from Equation (78), the linear phase analysis filter in combination with the high delay synthesis filter results in a system group delay of 10 samples. That is, this combination contains a 3.5 sample delay from the analysis filter and 6.5 samples from the synthesis filter. These all lead to a system capable of exploiting phase diversity to reduce distortion.

With the previous portion of the design addressed, we continue by addressing the determination of the weighting factors, α and the βs, and the initial coefficients for the analysis filters in the **A** matrix of Equation (35). Since our design objective for this system is to make it compatible with current compression coders, consider using a scaled version of the Johnston's eight-tap(A) (scaled for a two-band system) [15] for the initial analysis filter coefficients for the **A** matrix in Equation (35). To achieve a large degree of compatibility with systems that employ the Johnston eight-tap(A) filters, α in Equation (70) is set to approximately 1. Similarly, β_1 in Equations (72) and (74) are set to approximately 1 also. Thus, β_0 and β_2 in each equation are set to approximately 0. Minimizing Equation (70) yields the coefficients in Tables 2, 3, and 4 for this even-length FAAS system.

Table 2: Johnston eight-tap(A) analysis filter coefficients (normalized for a passband gain of $\sqrt{2}$)

H_0	H_1
0.00037436976317	0.00037436976317
-0.01628634319845	0.01628634319845
0.01626542450770	0.01626542450770
0.70668561798825	-0.70668561798825
0.70668561798825	0.70668561798825
0.01626542450770	-0.01626542450770
-0.01628634319845	-0.01628634319845
0.00037436976317	-0.00037436976317

Table 3: Johnston eight-tap(A) synthesis lowpass filter and complementary synthesis lowpass filters

G_{00}	G_{01}	G_{02}
0.70677847266626	0.00037498769053	0.00000860859150
0.70677790627290	-0.01627617029410	0.00000863742685
0.01627985228855	0.01627984112172	0.00000020140233
-0.01627618601797	0.70677818952471	0.00037518634246
0.00037518634246	0.70677818952471	-0.01627618601797
0.00000020140233	0.01627984112172	0.01627985228855
0.00000863742685	-0.01627617029410	0.70677790627290
0.00000860859150	0.00037498769053	0.70677847266626

Table 4: Johnston eight-tap(A) synthesis highpass filter and complementary synthesis highpass filters

G_{10}	G_{11}	G_{12}
0.70677847266626	-0.00037498769053	0.00000860859150
-0.70677790627290	-0.01627617029410	-0.00000863742685
0.01627985228855	-0.01627984112172	0.00000020140233
0.01627618601797	0.70677818952471	-0.00037518634246
0.00037518634246	-0.70677818952471	-0.01627618601797
-0.00000020140233	0.01627984112172	-0.01627985228855
0.00000863742685	0.01627617029410	0.70677790627290
-0.00000860859150	0.00037498769053	-0.70677847266626

Table 2 contains the Johnston 8(A) analysis filter. Table 3 contains the lowpass synthesis filters for the FAAS system. and Table 4 contain the coefficients for the highpass synthesis filters. It is important to note that the filter coefficients G_{01} and G_{11} from Tables 3 and 4 are the Johnston 8(A) (normalized by $\sqrt{2}$).

The reconstruction errors for each filter combination are shown in Table 5. At first glance, the reconstruction errors for the linear phase - low delay and linear phase - high delay combination may appear large. But, these filters are used only for small segments in an image while the linear phase-linear phase combination are used significantly more often to reconstruct a given signal. Thus, the overall reconstruction error for the image is very small.

Table 5: Johnston's eight-tap(A) filters and complementary filters reconstruction errors

Filter Pairs (analysis-synthesis)	Reconstruction Error
linear phase - low delay	0.00106003613975
linear phase - linear phase	$5.957290674789945 * 10^{-10}$
linear phase - high delay	0.00106003613975

6.2 Odd-Length Adaptive Synthesis Filter Design

Odd-length Example

In this subsection, we present the design of an odd-length FAAS filter bank, compatible with codecs that employ the popular Daubechies 9/7 and LeGall 5/3 filters. For this system, we consider a nine-tap filter and a seven-tap filter. The linear phase analysis and linear phase synthesis filter combination yield an overall system group delay of 7 samples. The analysis filter contributes a group delay of 4 samples, while the synthesis filter contributes a group delay of 3 samples. Using Equation (77), the linear phase analysis filter in combination with the low delay synthesis filter yields a system group delay of 5 samples. This is obtained from the 4 sample delay of the analysis filter and 1 sample delay from the synthesis filter. Finally, from Equation (78), the linear phase analysis filter in combination with the high delay synthesis filter results in a system group delay of 11 samples. That is, this combination contains a 4 sample delay from the analysis filter and 7 samples from the synthesis filter. Similar to its even-length counterpart, these all lead to a system capable of exploiting phase diversity to suppress distortion.

We continue by addressing the weighting factors, α and the βs, and the initial coefficients for the analysis filters in the **A** matrix of Equation (35). Since the SPIHT coder uses the Daubechies 9/7 filters, we consider using the Daubechies 9/7 [9] for the initial analysis filter coefficients for the **A** matrix in Equation (35). To achieve a large degree of compatibility with systems that employ the Daubechies 9/7 filters, α in Equation (70) is set to approximately 1. Similarly, β_1 in Equations (72) and (74) is set to approximately 1 also. Thus, β_0 and β_2 in each equation are set to approximately 0. Minimizing Equation (70) yields the coefficients in Tables 6, 7, and 8 for this odd-length FAAS system.

Table 6: Daubechies 9/7 analysis filter coefficients

H_0	H_1
0.03782845550726	-0.06453888262870
-0.02384946501956	0.04068941760916
-0.11062440441844	0.41809227322162
0.37740285561283	-0.78848561640558
0.85269867900889	0.41809227322162
0.37740285561283	0.04068941760916
-0.11062440441844	-0.06453888262870
-0.02384946501956	0
0.03782845550726	0

Table 7: Daubechies 9/7 synthesis lowpass filter and complementary synthesis lowpass filters

G_{00}	G_{01}	G_{02}
0.79222481028799	-0.06453888262892	0
0.38154281168348	-0.04068941760936	-0.00000000000007
-0.04344260785188	0.41809227322190	0.00023522048207
-0.03865456887373	0.78848561640626	-0.06416579134413
0.00023522048207	0.41809227322190	-0.04344260785188
-0.00209828791646	-0.04068941760937	0.41372534362802
0	-0.06453888262892	0.79222481028799
0	-0.00000000000014	0.42402313704934
0	0	0

Table 8: Daubechies 9/7 synthesis highpass filter and complementary synthesis highpass filter

G_{10}	G_{11}	G_{12}
0.37417351498122	-0.03782845550717	0
-0.82245340753992	-0.02384946501974	0
0.38103462269512	0.11062440441872	0.00013787080251
0.07911389212137	0.37740285561309	-0.03760977389242
-0.02497324250280	-0.85269867900966	-0.02497324250280
-0.02702748264951	0.37740285561307	0.10884194235998
0.00013787080251	0.11062440441872	0.38103462269512
-0.00122987860735	-0.02384946501973	-0.84693820851846
0	-0.03782845550717	0.37417351498122

Table 6 contains the Daubechies 9/7 analysis filter, Table 7 contains the lowpass synthesis filters for the FAAS system, and Table 8 contain the coefficients for the highpass synthesis filters. The reconstruction errors for each filter combination are shown in Table 9.

Table 9: Daubechies 9/7 filters and complementary filters reconstruction errors

Filter Pairs (analysis-synthesis)	Reconstruction Error
linear phase - low delay	0.20566879027821
linear phase - linear phase	$3.376026106177021 * 10^{-026}$
linear phase - high delay	0.00807868404910

CHAPTER VII

DISTORTION SUPPRESSION

Subband image coders operating at low bit rates are known to exhibit distortion in the form of ringing at edges. To alleviate this problem, AAAS filter banks were investigated in work reported by Arrowood where a set of ringing removal strategies were applied to effectively exploit phase diversity. Arrowood applied length switching and phase switching techniques to mitigate ringing, but focused mostly on phase switching. He applied the low delay results at the beginning of discontinuities, high delay results at the end of discontinuities and linear phase results elsewhere.

Determining the appropriate place to switch filters in an AAAS filter bank is not a trivial procedure. Arrowood employed a first-order difference equation on the rows and columns of the input to detect major edges. Best results were obtained using a threshold in the range of 45 to 55. To ensure encoder/decoder synchrony, switch point locations were transmitted separately to the decoder as side information, which constitutes a forward adaptive system. Arrowood also considered two methods to do this. These two methods were highpass split information and edge detection.

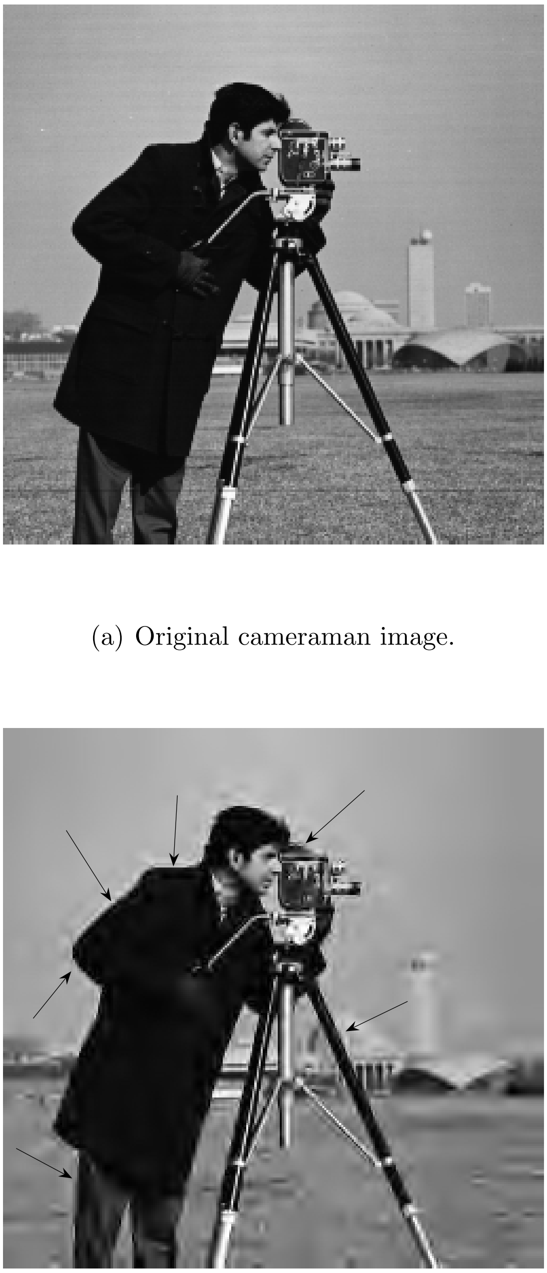
Arrowood also considered using the HL and LH subbands, as they embody directional edge information. Together they can be used as an edge detector to trigger switching. This approach is attractive because the HL and LH subbands are available and thus don't involve additional arithmetic. For forward adaptive systems, he was able to determine the major switch points and suppress distortion. However, additional bits had to be transmitted to specify the switching point locations. For his backward adaptive implementation, the switch points were extracted from HL and LH at the receiver side. Consequently, no additional sync bits were necessary.

The disadvantage here is that at low bit-rate edge information in the HL and LH sometimes does not survive quantization, resulting in sync errors.

In the sections that follow, we consider the FAAS filter bank structure and introduce a new approach to suppressing ringing distortion.

7.1 Distortion Suppression Strategies

A key advantage of FAAS filter banks is the potential to achieve higher performance than conventional filter banks. In particular, these filter banks provide a way to suppress ringing distortion in reconstruction. An example of ringing distortion is shown in Figure 23. The reconstructed 256×256 cameraman image is shown here compressed to 0.25 bpp. The ringing distortion manifests itself as alternating light and dark bands adjacent to object edges. Ringing distortion is caused by a loss of information in the high frequency channel. Such losses occur when images are coded at low bit rates. The cause of the distortion can be seen clearly through a simple 1-D example. Consider a step edge that has been filtered by $H_0(z)$ in the low frequency channel of the filter bank, an illustration of which is shown in Figure 24.



(a) Original cameraman image.

(b) Cameraman image compressed to 0.25 bpp.

Figure 23: Illustration of ringing distortion in cameraman image.

The result is effectively the step response of the lowpass filter. The overshoot and undershoot associated with the step response is the source of the ringing distortion that appears when coding is performed at low bit rates.

There is a tradeoff between aliasing cancelation and ringing distortion that is important to manage. Both of these factors are functions of high frequency channel information loss, which, in turn, is a function of the bit rate. To illustrate this point, consider Figure 25 which shows the various high frequency subband images for different bit rates. By inspecting these subbands, it is evident that as the bit rate is lowered, the amount of information present in the high frequency subbands is reduced, leading to the ringing distortions we see.

Low delay and high delay reconstruction filters should be designed carefully to minimize both imaging distortion and ringing distortion. To mitigate imaging distortion, the synthesis filters should have good stopband rejection characteristics. To suppress ringing distortion, the step response characteristics should have minimal ripples. These two objectives are contradictory, suggesting the need to impose constraints in both the frequency domain and the time domain.

To start, consider the method developed in Chapter 6. In general, this method allows us to obtain a set of complementary low delay and high delay filters to aid in distortion suppression. In Chapter 6, the Daubechies 9/7 filters were considered as a specific case and complementary synthesis filters were designed.Although the frequency domain coefficients have been optimized, an interesting phenomenon occurs with the low delay (G_{00}) and high delay (G_{02}) filters when they are applied in an FAAS filter bank to suppress ringing. Recall that the filters G_{00} and G_{02} are applied after the subband images have been upsampled. When the subband has a high sustained amplitude, the upsampled subband will contain interleaved zeros. Thus, if the sum of the odd filter coefficients is different from the sum of the even coefficients, oscillations will occur in the output. For linear phase filters (like the Johnston QMFs), the sum of

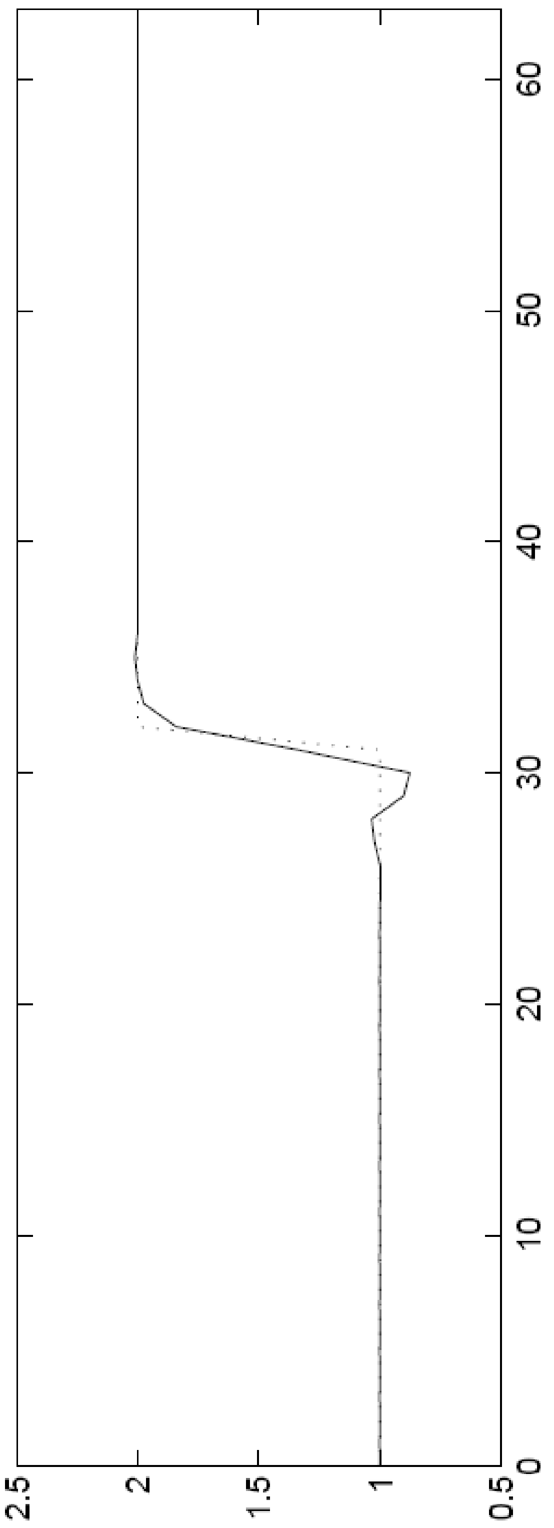
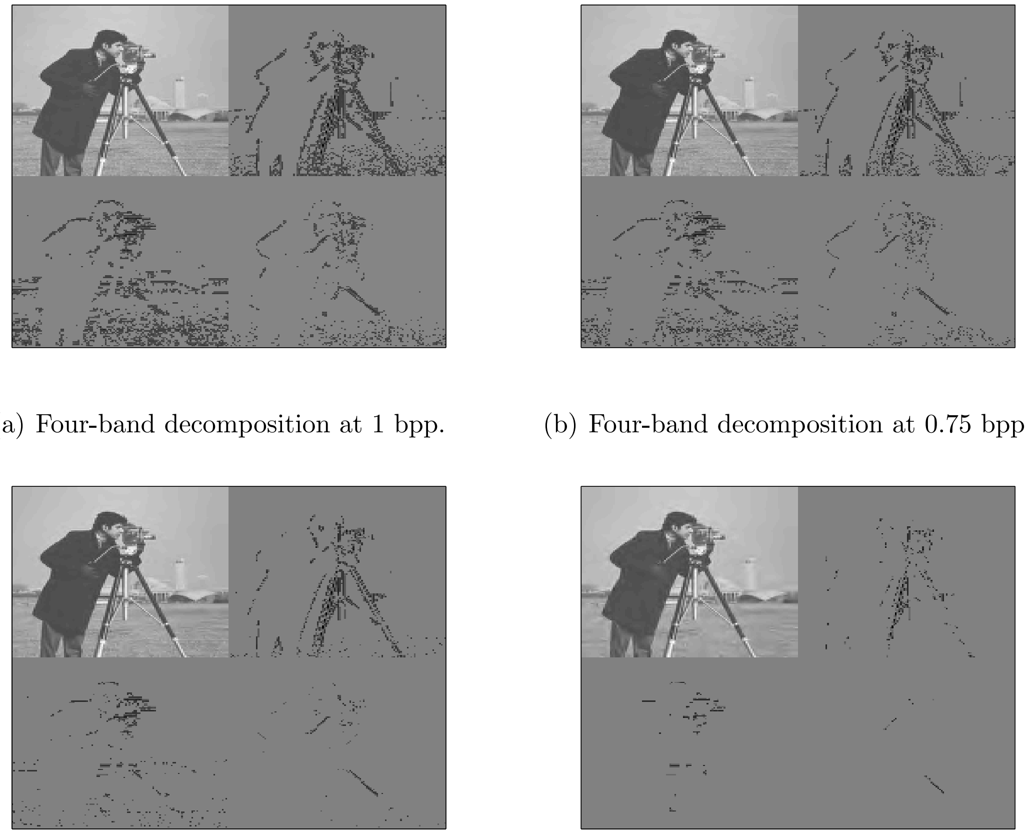


Figure 24: Step function response in the lowpass channel.



(a) Four-band decomposition at 1 bpp.

(b) Four-band decomposition at 0.75 bpp.

(c) Four-band decomposition at 0.5 bpp.

(d) Four-band decomposition at 0.25 bpp.

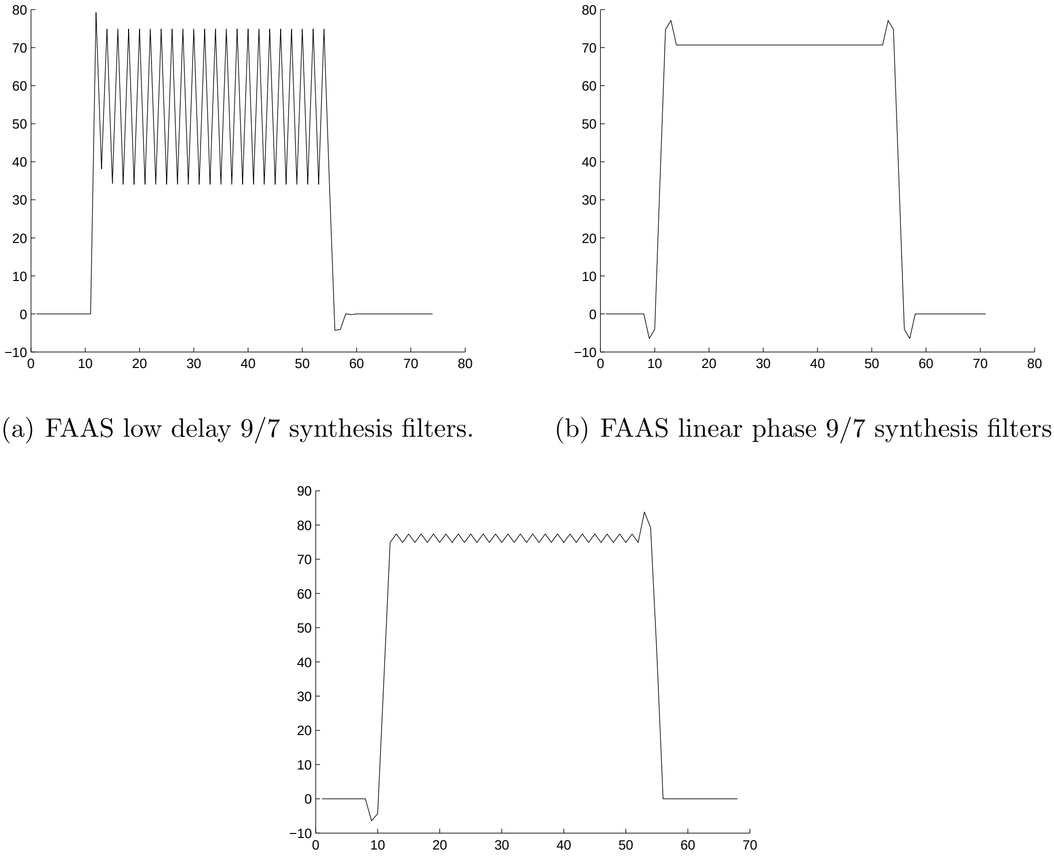
Figure 25: Illustration of image subbands for the cameraman image coded at four different bit rates.

the even and odd coefficients is exactly the same (which is a property of even-length linear phase filters). But such is not naturally true for low delay or high delay filters. Figure 26 illustrates this point graphically. In the figure, three FAAS examples of complementary Daubechies 9/7 synthesis filters are shown: a) low delay, b) linear phase, and c) high delay. For the low delay, the difference in the sum of the odd and even coefficients is high, resulting in the high oscillations seen in Figure 26(a). Small oscillations are visible for the high delay case. It turns out the sum difference in odd and even coefficients is smaller. No oscillations occur for the linear phase case.

To address this issue, an explicit constraint is introduced in the design process to assure that the odd and even coefficient sums differences are negligible. It should be noted that this sum difference issue is simply a component of reconstruction error, which could have been addressed by using a higher weighting for the reconstruction error in the original design procedure. Emphasizing the even/odd sum difference as an explicit component within the reconstruction error has a perceptual advantage in that it can be directly associated with a distortion that is visible.

In an effort to maximize FAAS distortion suppression, we have redesigned the synthesis filters in accordance with the discussion above. In addition, to reduce computation complexity while at the same time reducing ringing in the tails of the low and high delay filters, we constrain the low amplitude trailing and leading coefficients to be zero. Since these amplitudes are so small, their effect on the frequency response characteristics is negligible and hence this represents an opportunity to reduce arithmetic complexity.

Table 10 lists the coefficients of the redesigned FAAS synthesis filters for the Johnston eight-tap(A) filters. Similarly, Table 11 lists the coefficients for the complementary highpass filters. Table 12 lists the reconstruction errors for each of the reconstruction pairs. One can observe that the reconstruction errors have been reduced significantly compared to the filters designed in Chapter 6.



(a) FAAS low delay 9/7 synthesis filters.

(b) FAAS linear phase 9/7 synthesis filters.

(c) FAAS high delay 9/7 synthesis filters.

Figure 26: Example of FAAS synthesis filters applied to an upsampled block.

1. Scaling the filter coefficients so that the sum of the coefficients are scaled to $\frac{2}{\sqrt{2}}$ for the lowpass filter coefficients (0 for the highpass). That means a magnitude of $\frac{1}{\sqrt{2}}$ for the even coefficients and a magnitude of $\frac{1}{\sqrt{2}}$ for the odd coefficients is required (-$\frac{1}{\sqrt{2}}$ and $\frac{1}{\sqrt{2}}$ for the highpass).

2. Minimizing the coefficients in the "tail" of the filter by reducing the magnitude of the filter's tail coefficients or using shorter filters [14].

These steps will increase the reconstruction error and image distortion but they will lead to less ringing distortion.

Table 10: Johnston eight-tap(A) Redesigned Synthesis Lowpass Filter and Complementary Synthesis Lowpass Filter

$G_{00}Redesigned$	$G_{01}Redesigned$	$G_{02}Redesigned$
0.69084041185006	0.00037498769053	0
0.72378694465759	-0.01627617029410	0
0.01590045844083	0.01627984112172	0
-0.01668016347104	0.70677818952471	0.00036591045377
0.00036591089566	0.70677818952471	-0.01668016298951
0	0.01627984112172	0.01590045801345
0	-0.01627617029410	0.72378694417606
0	0.00037498769053	0.69084041271932

Table 11: Johnston eight-tap(A) Redesigned Synthesis Highpass Filter and Complementary Synthesis Highpass Filter

$G_{10}Redesigned$	$G_{11}Redesigned$	$G_{12}Redesigned$
-0.69084041185006	-0.00037498769053	0
0.72378694465759	-0.01627617029410	0
-0.01590045844083	-0.01627984112172	0
-0.01668016347104	0.70677818952471	0.00036591045377
-0.00036591089566	-0.70677818952471	0.01668016298951
0	0.01627984112172	0.01590045801345
0	0.01627617029410	-0.72378694417606
0	0.00037498769053	0.69084041271932

Table 12: Johnston eight-tap(A) Redesigned Filters Reconstruction Errors

Filter Pairs Redesigned	Reconstruction Error
linear phase - low delay	0.00214737453486
linear phase - linear phase	$5.957275160953383 * 10^{-10}$
linear phase - high delay	0.00214737444664

This same procedure was applied to the odd-length filter for redesign. The original (Chapter 6) FAAS Daubechies 9/7 complementary synthesis lowpass filters are listed in Table 13. As before, the odd/even coefficient sum difference was constrained to be negligible. In addition, note that in the complementary Daubechies 9/7 synthesis lowpass filters in Table 13, there is only one virtual zero coefficient per filter. Keeping with the approach to reduce the tail and reduce computational complexity, the filter coefficients that are of low amplitude are replaced with zeros. This yields coefficients listed in Table 14 and Table 15. The reconstruction errors for the filter combinations are shown in Table 16. The effects of shortening and redesigning of the low delay synthesis filter and high delay filters are evident from the example shown in Figure 27.

Table 13: Complementary Daubechies 9/7 synthesis lowpass filters

G_{00}	G_{02}
0.79222481028799	-0.00000000000007
0.38154281168348	0.00023522048207
-0.04344260785188	-0.06416579134413
-0.03865456887373	-0.04344260785188
0.00023522048207	0.41372534362802
-0.00209828791646	0.79222481028799
0	0.42402313704934

Table 14: Redesigned Daubechies 9/7 complementary synthesis lowpass filter

G_{00}	G_{01}	G_{02}
0.74789653543220	-0.06453888262897	0
0.79166508748431	-0.04068941760930	0.00022205891726
-0.04101181316292	0.41809227322225	-0.05865186334346
-0.08020455820952	0.78848561640516	-0.04101181316292
0.00022205891726	0.41809227322225	0.37817288321209
-0.00435374808825	-0.04068941760931	0.74789653543220
0	-0.06453888262897	0.38758576131792

Table 15: Redesigned Daubechies 9/7 complementary synthesis highpass filters

G_{10}	G_{11}	G_{12}
0.36225423798497	-0.03782845550717	0
-0.75371272132055	-0.02384946501974	0
0.36889678548531	0.11062440441872	0.00013347893558
0.07250155009196	0.37740285561309	-0.03428376831574
-0.02417772121931	-0.85269867900966	-0.02417772121931
-0.02476852465009	0.37740285561307	0.09921654795315
0.00013347893558	0.11062440441872	0.36889678548531
-0.00112708530786	-0.02384946501973	-0.77203956082396
0	-0.03782845550717	0.36225423798497

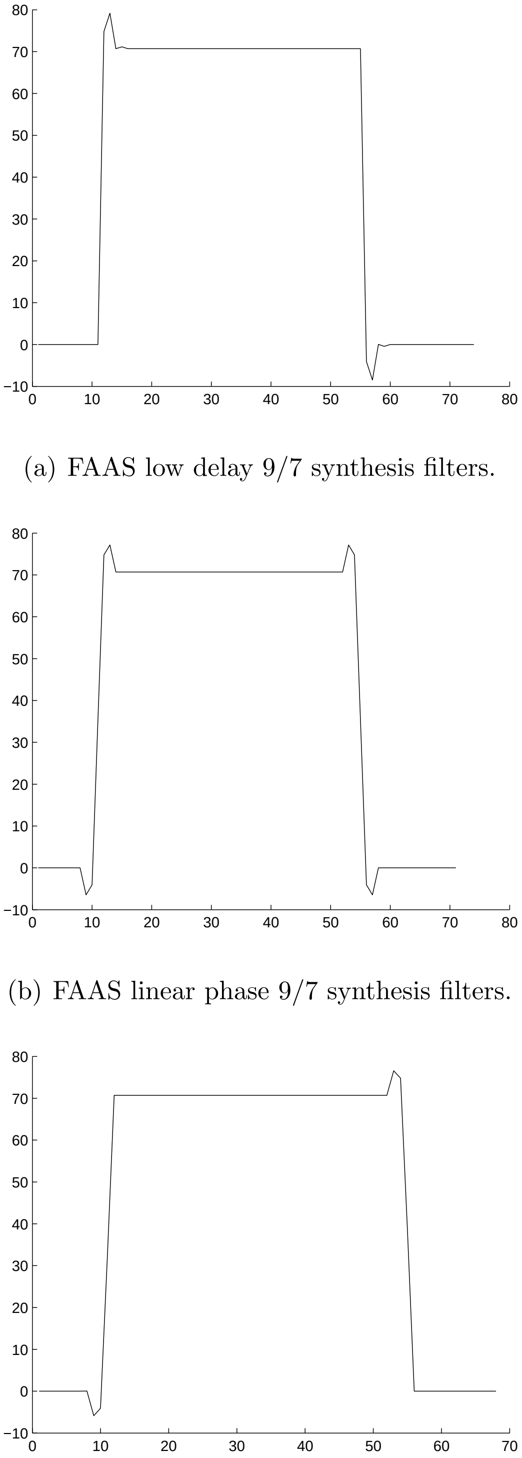
Table 16: Reconstruction errors for the Daubechies 9/7 filters and complementary filters

Filter Pairs (analysis-synthesis)	Reconstruction Error
linear phase - low delay	0.39370928190980
linear phase - linear phase	$1.570612476380791 * 10^{-024}$
linear phase - high delay	0.02794395475609

7.2 New Phase Switching Selection Method

In the prior section, distortion suppression strategies were presented in general terms. What is more important now is to determine when and how often to switch between filters, what constitutes an edge, and what constitutes ringing. Previous methods proposed to make this determination were edge detection and mixed frequency subband thresholding, as mentioned before. In the remainder of this section, we will introduce the new method employed in this work for selecting the filter switching locations. The rationale for expected performance gain articulated earlier in this document was based on the notion of exploiting phase diversity. By considering a multiplicity of reconstruction options involving different synthesis filter group delays, quantization noise can be targeted spatially, enabling higher quality reconstruction. Each of the reconstructions –low delay, linear phase, and high delay –has a unique noise profile in a low bit-rate subband coding environment. This point is illustrated in Figure 31, where a 1-D image block is shown reconstructed with low delay (in blue), linear phase (in green), and high delay (in red) FAAS filters. The selection method employed in this thesis is based on a simple rule-based comparison of the three outputs on a pixel-by-pixel basis. Let $\hat{x}_{LD}, \hat{x}_{LP}, \hat{x}_{HD}$ denote the low delay, linear phase, and high delay reconstructions respectively. Then if

$$\hat{x}_{LD}(i) \approx \hat{x}_{LP}(i) \tag{79}$$



(a) FAAS low delay 9/7 synthesis filters.

(b) FAAS linear phase 9/7 synthesis filters.

(c) FAAS high delay 9/7 synthesis filters.

Figure 27: Unscaled and same length synthesis filters response

and

$$\hat{x}_{HD}(i) \neq \hat{x}_{LP}(i) \tag{80}$$

we use the reconstruction from the high delay synthesis filter. As is evident by examining Figure 31, when this condition is true, we are at the back end of a transition, in which case the high delay filter provides the most accurate reconstruction. Similarly, if,

$$\hat{x}_{HD}(i) \approx \hat{x}_{LP}(i) \tag{81}$$

and

$$\hat{x}_{LD}(i) \neq \hat{x}_{LP}(i) \tag{82}$$

then we are at the onset of a transition and low delay filters provide the most accurate reconstructions. For all other areas in the image, linear phase reconstructions selected. In so doing, the majority of the reconstruction will be identical to that obtained using conventional FAFS filter banks. This method is performed on the image rows and columns within the framework of a separable 2-D filter bank.

This phase switching method has three advantages over the edge detection and the mixed frequency subband thresholding methods employed by Arrowood. First, we do not have to find the explicit location of an edge. Second, we do not have to find an edge and then create a switching mask. Both are accomplished in this one step. Third, edges that occur within close proximity of each other such that the ringing regions overlap are not a problem in the new method. This is because in these cases neither the low delay nor high delay filter will be approximately equal to the linear phase filter and this will not trigger a poor reconstruction choice.

To optimize the effectiveness of the phase selection method, threshold criteria governing the approximations in Equations (79), (80), (81), and (82) must be developed. Such thresholds are best set empirically, but for the purpose of demonstration, we assume equality to within 8-bit numerical precision. Using Equations (7.1)-(7.4) defined in this way, a switching mask is created for both for the rows and column edges. The switching masks for both the row and column edge detection for the cameraman image are shown in Figures 28 and 29. In these images, similar to Figure 15, the green regions represent areas where linear phase filters will be used, the blue are where low delay filters will be used, and the red areas where high delay filters will be used. Control over the switching mask is essentially provided through the implicit threshold used in Equations (7.1) - (7.4). If, for example, we relaxed the precision from 8 bits to 3 bits, the high and low delay filters would be used more frequently. Equivalently, the switching mask would be dilated as illustrated in Figure 34.

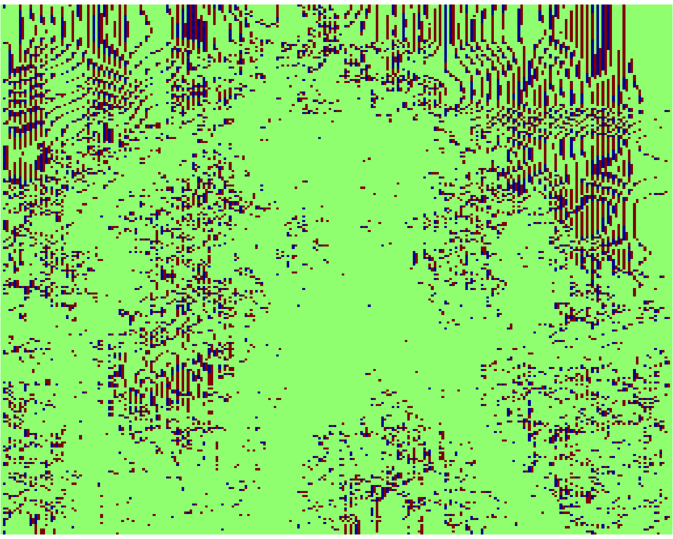


Figure 28: Cameraman phase comparison row mask.

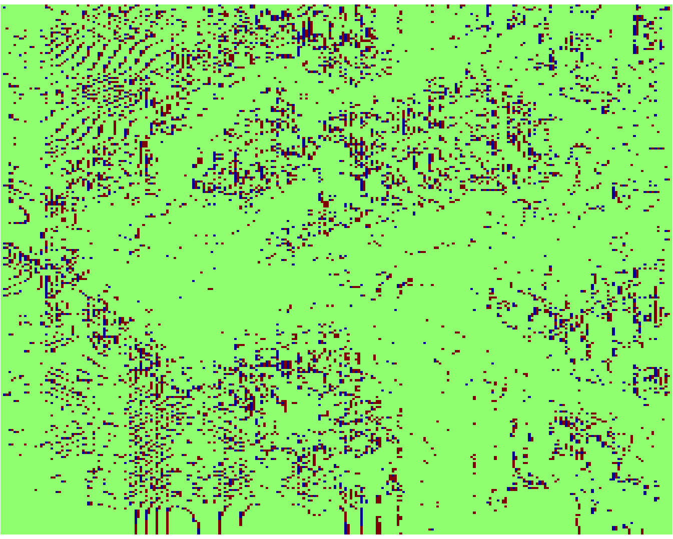


Figure 29: Cameraman phase comparison column mask. Note: Image is transposed.

In closing, we point out that the phase selection method in combination with the FAAS filter design will not reduce ringing distortion completely. However, this method does suppress much of the distortion.

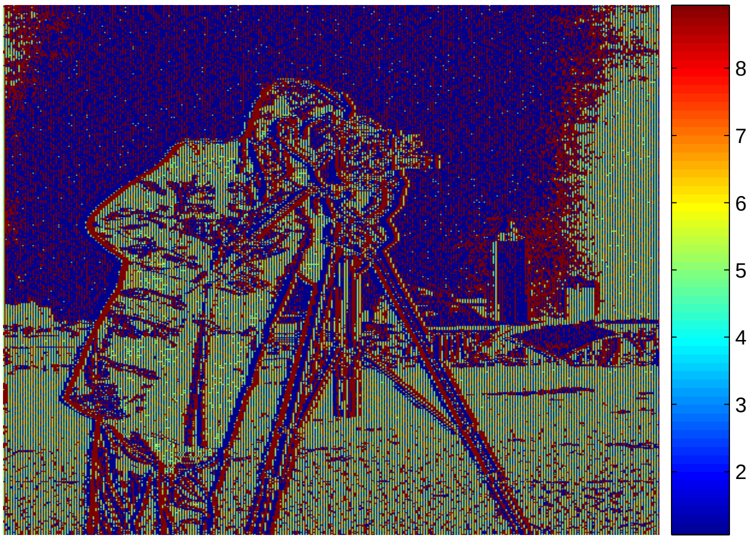


Figure 30: Cameraman phase comparison row mask with accuracy criteria relaxed.

CHAPTER VIII

FAAS FILTER BANK APPLICATION TO SYMMETRIC EXTENSION

In this chapter, we consider the application of FAAS filter banks to symmetric extension-which is now the standard method for implementing subband/wavelet coders. First we provide a summary of the size expansion issue that motivates the use of symmetric extension. This is followed by a discussion of linear phase symmetric extension. In the subsequent sections, the adaptive feature of FAAS filter banks is considered. It is shown that FAAS filter banks provide additional implementation flexibility within the natural decomposition structure that allows *nonlinear* phase in addition to *linear* phase filters to be employed.

8.1 The Boundary Problem

Filter banks were popularized in the 1970s and early 1980s in the context of subband audio coding. In 1986, Woods and O'Neil [47] considered applying subband coding to images. This application caught the eye of researchers all over the world and has been studied extensively since that time.

Extending subband coding to two-dimensions can be done in a straightforward way by applying the filter banks to the rows and then to the columns of the resulting subband images. However, a new issue arises in the context of subband image coding for which there is no 1-D counterpart: the problem of data expansion. Data expansion occurs because a digital image has a finite length, say $N \times M$, and convolution operations inherently expand the length of a finite duration signal. In particular, the convolution of an image row of length M with a filter of length L results in an output of length $M + L - 1$. After performing the 2-band filter bank downsampling operation, the length of the subband signal is $(M + L - 1)/2$.

This is a problem. For subband image coding, the goal is to decompose the original image into subbands, code the subbands efficiently using as few bits as possible, and then reconstruct in the decompression stage. Since the subband decompositions is effectively performing a compaction transform, the resulting subbands should contain no more pixels collectively than the original, if the overall system is to be efficient. Thus, ideally each 2-D subband split should result in a set of four subbands each of quarter size, i.e. $N/2 \times M/2$. Unfortunately, the natural output of the filter bank (for a 2-D split) results in larger subband images of size $(N + L - 1)/2 \times (M + L - 1)/2$. This expansion hurts coding performance, the degree of which is a function of the image size and the filter length "L."

To appreciate the magnitude of the data expansion, consider coding an image of size 256×256 with 32-tap analysis-synthesis filters. A single 2-D split results in data expansion of 25.7 percent. If that same image is split again to form a 16-band subband decomposition, the data expands by 85.9 percent. For larger images and smaller filter lengths, the expansion is less severe. Nonetheless, data expansion is an important issue to address if subband image coding is to be effective.

In the original paper by Woods and O'Neil, implementation of the filter bank was performed using the DFT. That is, convolution was performed in the frequency domain. Their implementation was equivalent to performing circular convolution in the spatial domain. Since circular convolution will result in an output that is of the same length as the input, no data expansion occurs. This would seem to be an ideal solution in that it is simple to implement, it involves no more computation than linear convolution, and it avoids the problem of data expansion. The disadvantage of circular convolution is that it results in boundary distortions when the subbands are quantized. The magnitude of these distortions increases in inverse relation to the bit rate: the lower the bit rate, the more visible the distortions.

The reason for these distortions is that often the pixel amplitudes at the beginning and end of an image row or column will be different and hence represents a discontinuity. Circular convolution involves filtering across this boundary discontinuity. Such discontinuities cause ringing distortions that are particularly visible when coding at low bit rates, hence the problem.

To better appreciate the problem, consider the image shown in Figure 32, which has been coded at a low bit rate. Visible ringing distortions are evident at the top and sides of the picture, which are a consequence of circular convolution. In the

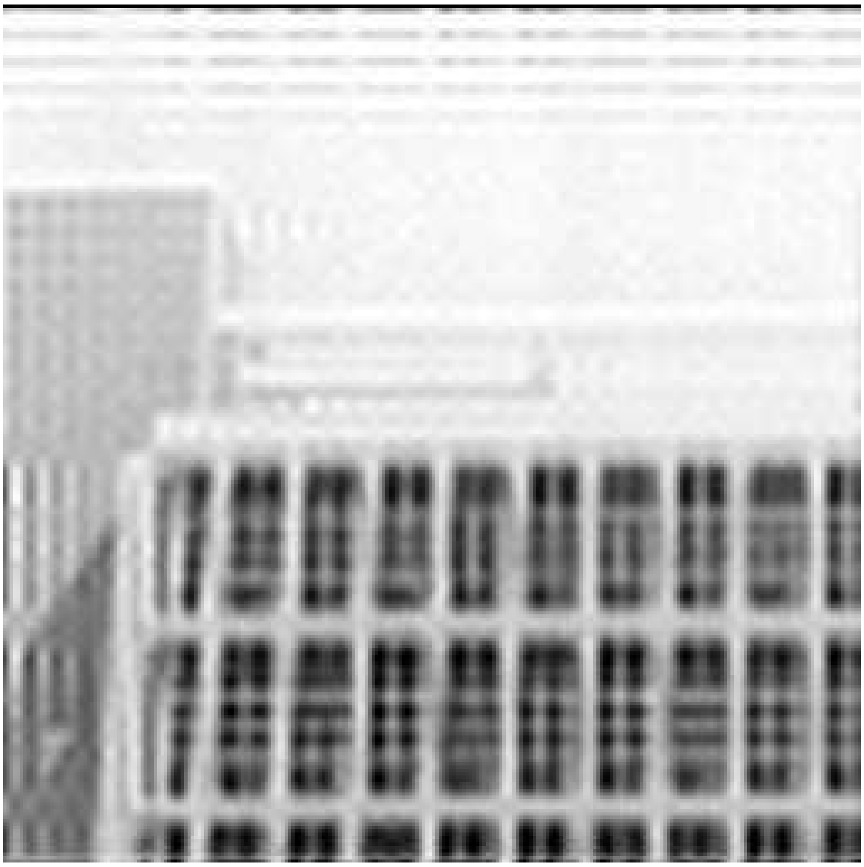


Figure 31: Subband image coding example using circular convolution.

next section, we discuss symmetric extension, the popular solution for addressing this problem, along with its limitations.

8.2 Linear Phase Symmetric Extension

The coding distortion at boundary edges due to circular convolution is well recognized at low bit rates. Smith and Eddins in [?, 12] developed a method to suppress these effects called symmetric extension. Symmetric extension, like the circular convolution method, creates a periodic signal, which is the key to avoiding data expansion, but also avoids creating artificial boundary discontinuities. Symmetric extension is accomplished by reflecting the rows and columns at the image boundaries while circular convolution involves a direct concatenation of the rows and columns. For example, the circular convolution method applied to the sequence a,b,c,d, would create the periodic sequence ...a,b,c,d,a,b,c,d,... . In contrast, the symmetric extension method would create the periodic sequence ...a,b,c,d,d,c,b,a... .

Because artificial boundary discontinuities are avoided the symmetric extension method avoids the migration of quantization noise across beginning and end of the rows and columns. This point can be seen by comparing the circular convolution coding example shown in Figure 32 to the symmetric extension coding example show in Figure 33. In addition, symmetric extension accommodated ER with biorthogonal filters, as highlighted in [16].

Symmetric extension is not as trivial as it may seem. This technique must be done carefully and must obey a clear set of rules. If done correctly, the technique will produce a symmetric output, ensure no expansion, and reduce boundary distortion. Different types of symmetric extension are used for the various combinations of even-length and odd-length image and filter size. Two types of symmetric extensions are noteworthy in particular:

- half-point symmetric extension,

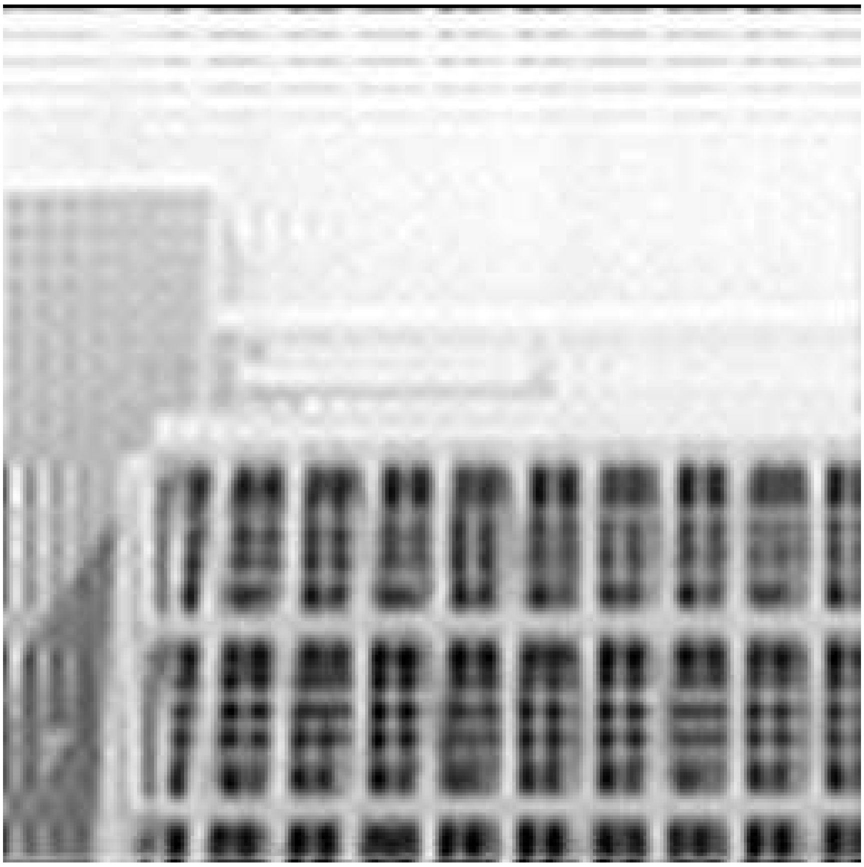


Figure 32: Circularly extended building image that has been filtered and quantized.

- whole-point symmetric extension.

The rules governing the type of symmetric extension used were established and are published in [4] and are outlined in the following two subsections. They have helped to make symmetric extension a much more useful technique.

8.2.1 Linear Phase Half-Point Symmetric Extension

In this subsection, we describe linear phase half-point symmetric extension using a simple example. Consider Figure 34, which has a six-tap (even-length) lowpass filter

$$\mathbf{h}_0 = [h_0[0], h_0[1], \mathbf{h_0[2]}, \mathbf{h_0[3]}, h_0[4], h_0[5]] \tag{83}$$

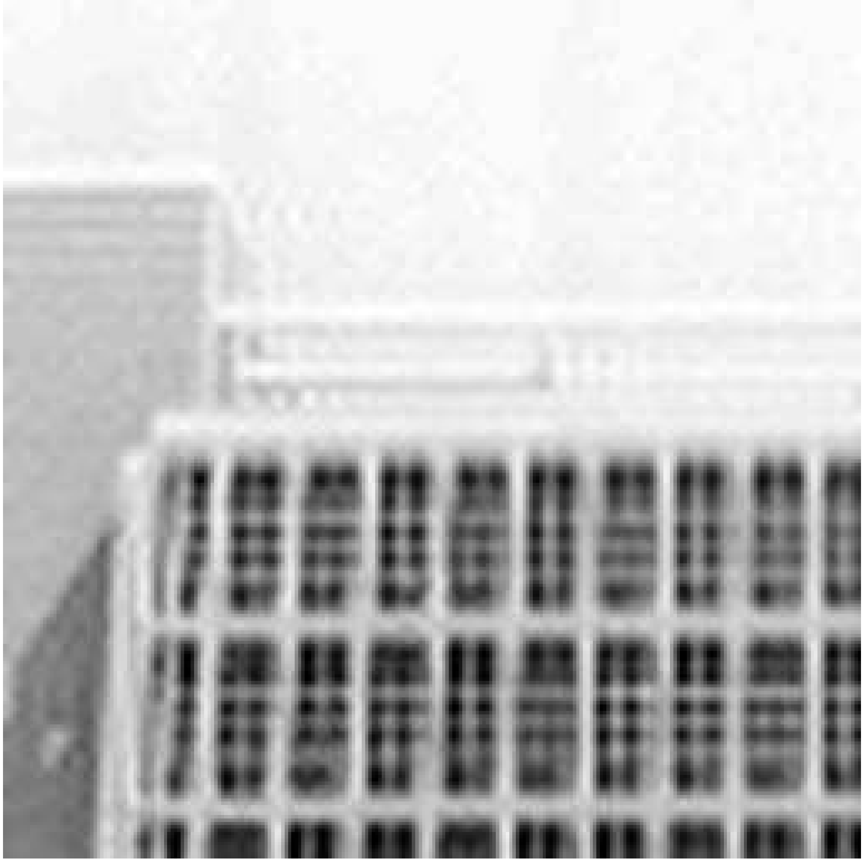


Figure 33: Filtered and quantized symmetrically extended building image.

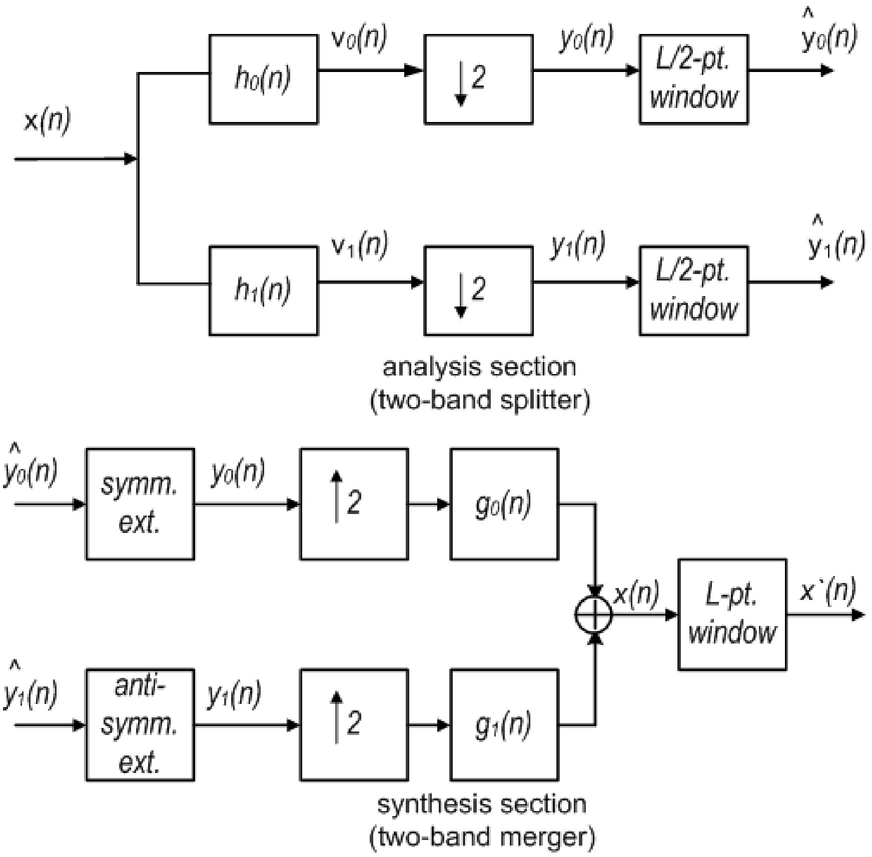


Figure 34: Two-band analysis-synthesis filter bank with symmetric extension.

where $\mathbf{h_0}[\mathbf{2}]$ and $\mathbf{h_0}[\mathbf{3}]$ are the largest coefficients of the mainlobe. Convolution

$$v_k[n] = \sum_{m=0}^{L-1} x[n-m]h_k[m], \qquad k = 0, 1, \ldots, M-1,$$

is performed with an input signal, $\mathbf{x}$, where

$$\mathbf{x} = [x[0], x[1], x[2], \ldots, x[6], x[7]]^T. \tag{84}$$

For convenience, we will describe the filtering and downsampling operations using matrix equations. Combining the filter block structure with the input data will yield $\mathbf{v}_0^{ext}$ which is symmetric at the boundary edges. In this way, each channel can be downsampled and windowed to yield four samples that mitigate the data expansion issue experienced with linear convolution.

If $\mathbf{x}$ is extended prior to entering the system, $\mathbf{x}$ must be extended such that $\mathbf{x}^{ext} = \ldots, x[2], x[1], x[0], x[0], x[1], x[2], \ldots$ at the left boundary of the signal. Likewise, $\mathbf{x}$ must be extended such that $\mathbf{x}^{ext} = \ldots, x[5], x[6], x[7], x[7], x[6], x[5], \ldots$ at the right boundary of the signal. To avoid extending $\mathbf{x}$ prior to it entering the system, a different approach is employed. For simplicity, consider the lowpass channel and the left boundary only. The samples of $\mathbf{v}_0^{ext}$ can be represented as

$$\mathbf{v}_0^{ext} = \mathbf{H}_0 \mathbf{x}^{ext} \tag{85}$$

where $\mathbf{H}_0$ is a filter matrix defined as

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{h_0[3]} & \mathbf{h_0[2]} & h_0[1] & h_0[0] & 0 & 0 & 0 & \ldots \\ h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & h_0[1] & h_0[0] & 0 & 0 & \ldots \\ h_0[5] & h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & h_0[1] & h_0[0] & 0 & \ldots \\ 0 & h_0[5] & h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & h_0[1] & h_0[0] & \ldots \\ 0 & 0 & h_0[5] & h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & h_0[1] & \ldots \\ \vdots & & & & & & & \vdots \end{bmatrix} \tag{86}$$

$$\tag{87}$$

such that convolution is accomplished when the filter matrix is multiplied by an input signal. Equation 85 yields

$$\begin{bmatrix} \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & 0 & 0 & 0 & \cdots \\ h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & 0 & 0 & \cdots \\ h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & 0 & \cdots \\ 0 & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & \cdots \\ 0 & 0 & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & \cdots \\ \vdots & & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[2] \\ x[1] \\ x[0] \\ x[0] \\ x[1] \\ x[2] \\ x[3] \\ \vdots \end{bmatrix} = \begin{bmatrix} v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}. \tag{88}$$

To ensure symmetry at the left boundary we do two things. First, the mainlobe of $\mathbf{h}_0$ must straddle $[\ldots, x[0], x[0], \ldots]$ when the output of $\mathbf{v}_0$ is $v_0[0]$. In addition, $\mathbf{h}_0$ must be linear phase, i.e.

$$\begin{aligned} h_0[0] &= h_0[5] \\ h_0[1] &= h_0[4] \\ h_0[2] &= h_0[3]. \end{aligned} \tag{89}$$

$$\tag{90}$$

Second, the first three (or $L/2$) columns are *overlapped and added* (OLA) onto the proceeding three (or $L/2$) columns. This yields

$$\begin{bmatrix} h_0[0]+h_0[1] & \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{3}] & 0 & 0 & 0 & 0 & \cdots \\ h_0[1]+\mathbf{h_0}[\mathbf{2}] & h_0[0]+\mathbf{h_0}[\mathbf{3}] & h_0[4] & 0 & 0 & 0 & 0 & \cdots \\ \mathbf{h_0}[\mathbf{2}]+\mathbf{h_0}[\mathbf{3}] & h_0[1]+h_0[4] & h_0[0]+h_0[5] & 0 & 0 & 0 & 0 & \cdots \\ \mathbf{h_0}[\mathbf{3}]+h_0[4] & \mathbf{h_0}[\mathbf{2}]+h_0[5] & h_0[1] & h_0[0] & 0 & 0 & 0 & \cdots \\ h_0[4]+h_0[5] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ \vdots \end{bmatrix} = \begin{bmatrix} v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}. \tag{91}$$

which is partially expressed in the time domain due to spatial constraints.

Both the straddling of the mainlobe of $\mathbf{h}_0$ on $\mathbf{x}$ at $[\ldots, x[0], x[0], \ldots]$ when the output of $\mathbf{v}_0$ and the OLA of $\mathbf{x}$ produce the necessary symmetry in $\mathbf{v}^{ext}$ such that

$$\mathbf{v}_0^{ext} = v_0[-2], v_0[-1], v_0[0], v_0[1], v_0[2], v_0[3], \ldots, v_0[6], v_0[7], v_0[8], \ldots, v[10]]^T \quad (92)$$

and is of length 13. Although the region of support is increased as with linear convolution, the added information is not transmitted to the synthesis side because of the symmetry, downsampling (↓ 2), and windowing depicted in Figure 34. If done correctly, the non-expansive transmitted information can be used to reconstruct the original signal on the synthesis side.

Expressed in general equations, the above example can be extended such that the input signal,

$$\mathbf{x} = [x[0], x[1], x[2], \ldots, x[N-2], x[N-1]]^T, \quad (93)$$

where N is the length of the input signal. $\mathbf{x}$ is symmetrically extended to yield

$$\mathbf{x}^{ext} = x[L/2-1], \ldots, x[1], x[0], x[0], x[1], x[2], .., x[N-2], x[N-1], x[N-1], \ldots, x[N-L/2]]^T, \quad (94)$$

where L is the length of the analysis filters. Following the progression of $\mathbf{x}^{ext}$ through Figure 34,

$$\mathbf{v}^{ext} = \mathbf{A}\mathbf{x}^{ext}, \quad (95)$$

where A is the analysis filter matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}, \quad (96)$$

$\mathbf{H}_0$ is a lowpass filter matrix containing the filter coefficients $\mathbf{h}_0$, expressed as

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{h}_0[N/2] & \dots & h_0[0] & 0 & 0 & 0 & 0 & \dots & & 0 \\ h_0[L/2+1] & \dots & h_0[1] & h_0[0] & 0 & 0 & 0 & \dots & & 0 \\ h_0[L/2+2] & \dots & h_0[2] & h_0[1] & h_0[0] & 0 & 0 & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \ddots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & h_0[L/2-1] & h_0[L/2-2] \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & h_0[L/2] & h_0[L/2-1] \end{bmatrix} \tag{97}$$

and $\mathbf{H}_1$ is a highpass filter matrix containing the filter coefficients $\mathbf{h}_1$, expressed as

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{h}_1[N/2] & \dots & h_1[0] & 0 & 0 & 0 & 0 & \dots & & 0 \\ h_1[L/2+1] & \dots & h_1[1] & h_1[0] & 0 & 0 & 0 & \dots & & 0 \\ h_1[L/2+2] & \dots & h_1[2] & h_1[1] & h_1[0] & 0 & 0 & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \ddots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & \mathbf{h}_1[L/2-1] & h_1[L/2-2] \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & \mathbf{h}_1[L/2] & \mathbf{h}_1[L/2-1] \end{bmatrix}. \tag{98}$$

Looking at each channel's output, the filtered output may also be written as

$$\mathbf{v}^{ext} = \begin{bmatrix} \mathbf{v}_0^{ext} \\ \mathbf{v}_1^{ext} \end{bmatrix} \tag{99}$$

where the lowpass filtered output

$$\mathbf{v}_0^{ext} = v_0[-L/2+1], \dots, v_0[-1], v_0[0], v_0[1], \dots, v_0[N-1], v_0[N], v_0[N+1], \dots, v_0[N+L/2]]^T \tag{100}$$

and the highpass filtered output

$$\mathbf{v}_1^{ext} = v_1[-L/2+1], \dots, v_1[-1], v_1[0], v_1[1], \dots, v_1[N-1], v_1[N], v_1[N+1], \dots, v_1[N+L/2]]^T. \tag{101}$$

The region of support in Equation (100) is increased such that,

$$\begin{aligned} v_0[-1] &= v_0[1] \\ v_0[-2] &= v_0[2] \\ &\vdots \end{aligned} \tag{102}$$

$$\begin{aligned} v_0[N-1] &= v_0[N+1] \\ v_0[N-2] &= v_0[N+2] \\ &\vdots, \end{aligned} \tag{103}$$

and likewise the region of support in Equation (101) is increased such that,

$$\begin{aligned} v_1[-1] &= v_1[1] \\ v_1[-2] &= v_1[2] \\ &\vdots \end{aligned} \tag{104}$$

$$\begin{aligned} v_1[N-1] &= v_1[N+1] \\ v_1[N-2] &= v_1[N+2] \\ &\vdots. \end{aligned} \tag{105}$$

This would verify that symmetry is accomplished at the boundary edges. As for expansion, only half the original samples should remain in each channel of a two-band system. Thus, each channel must be downsampled. In the matrix equation representation this results in

$$\mathbf{y}^{ext} = \Lambda \mathbf{A} \mathbf{x}^{ext}, \tag{106}$$

where

$$\Lambda = \begin{bmatrix} \lambda & \\ & \lambda \end{bmatrix} \tag{107}$$

and

$$\lambda = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{108}$$

Looking at each channel's output, the downsampled filtered output may also be written as

$$\mathbf{y}^{ext} = \begin{bmatrix} (\downarrow 2)\mathbf{v}_0^{ext} \\ (\downarrow 2)\mathbf{v}_1^{ext} \end{bmatrix} \tag{109}$$

where

$$\mathbf{y}_0^{ext} = v_0[-L/2{+}1], \dots, v_0[-1], v_0[1], \dots, v_0[N{-}1], v_0[N{+}1], \dots, v_0[N{+}L/2]]^T \tag{110}$$

for the lowpass channel and

$$\mathbf{y}_1^{ext} = v_1[-L/2+1], \dots, v_1[-1], v_1[1], .., v_1[N-1], v_1[N+1], \dots, v_1[N+L/2]]^T \tag{111}$$

for the highpass channel. When both channels are windowed with an $L/2$ point window, the analysis output becomes

$$\hat{\mathbf{y}} = w(\Lambda \mathbf{v}^{ext}) \tag{112}$$

or

$$\hat{\mathbf{y}} = \begin{bmatrix} w(\mathbf{y}_0^{ext}) \\ w(\mathbf{y}_1^{ext}) \end{bmatrix} \tag{113}$$

where

$$w(\mathbf{y}_0^{ext}) = y_0[1], y_0[3], \dots, y_0[N-3], y_0[N-1]^T \tag{114}$$

for the lowpass channel and

$$w(\mathbf{y}_1^{ext}) = y_1[1], y_1[3], \dots, y_1[N-3], y_1[N-1]^T \tag{115}$$

for the highpass channel.

The process is repeated in reverse order in the synthesis subsystem to complete the symmetric extension method. The entire process is called half-point symmetric extension and is viable *if and only if* even-length filters are used [4]. This method, for even-length filters, eliminates expansion and reduces the distortion at the reconstructed image's edge that is seen in systems that use circular convolution.

8.2.2 Linear Phase Whole-point Symmetric Extension

For odd-length linear phase filters, a different procedure is followed. To illustrate this case, consider Figure 34 with a seven-tap (odd-length) lowpass filter

$$\mathbf{h_0} = [h_0[0], h_0[1], \mathbf{h_0[2]}, \mathbf{h_0[3]}, \mathbf{h_0[4]}, h_0[5], h_0[6]] \tag{116}$$

where $\mathbf{h_0}[\mathbf{2}]$, $\mathbf{h_0}[\mathbf{3}]$, and $\mathbf{h_0}[\mathbf{4}]$ are the mainlobe coefficients. This filter is convolved with a signal $\mathbf{x}$, where

$$\mathbf{x} = [x[0], x[1], x[2], \ldots, x[6], x[7]]^T. \tag{117}$$

If odd-length linear phase filters are used, then symmetric extension is done such that

$$x^{ext} = [..., x(2), x(1), x(0), x(1), x(2), ...]. \tag{118}$$

Likewise, $\mathbf{x}$ must be extended such that $\mathbf{x}^{ext} = \ldots, x[5], x[6], x[7], x[6], x[5], \ldots$ at the right boundary of the signal. For simplicity consider the lowpass channel and the left boundary only. We know from Equation (85) that

$$\mathbf{v}_0^{ext} = \mathbf{H}_0\mathbf{x}^{ext} \tag{119}$$

where $\mathbf{H}_0$ is a filter convolution matrix. That is, when expanded, Equation 119 yields

$$\begin{bmatrix} \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & 0 & 0 & 0 & \dots \\ \mathbf{h_0}[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & 0 & 0 & \dots \\ h_0[5] & \mathbf{h_0}[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & 0 & \dots \\ h_0[6] & h_0[5] & \mathbf{h_0}[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & \dots \\ 0 & h_0[6] & h_0[5] & \mathbf{h_0}[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & \dots \\ 0 & 0 & h_0[6] & h_0[5] & \mathbf{h_0}[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & \dots \\ \vdots & \vdots & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[3] \\ x[2] \\ x[1] \\ x[0] \\ x[1] \\ x[2] \\ x[3] \\ \vdots \end{bmatrix} = \begin{bmatrix} v_0[-3] \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix} \quad . \tag{120}$$

To ensure symmetry at the left boundary, ($\mathbf{h_0}[\mathbf{3}]$) is multiplied by $x[0]$ (as a part of convolution) when the output, $\mathbf{v}$, is $v[0]$. In addition, $\mathbf{h}_0$ must be linear phase, i.e.

$$\begin{aligned} h_0[0] &= h_0[6] \\ h_0[1] &= h_0[5] \\ h_0[2] &= h_0[4]. \end{aligned} \tag{121}$$

To eliminate the need to extend $\mathbf{x}$, the first three (or $(rnd(L/2)) - 1$ where rnd means round) columns are overlap and added onto columns 5-7 (or $[(rnd(L/2)) + 1, ..., 2 * (rnd(L/2)) - 1]$). This yields

$$\begin{bmatrix} h_0[0] & h_0[1] & \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{3}] & 0 & 0 & \dots \\ h_0[1] & h_0[0] + \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{4}] & 0 & 0 & \dots \\ \mathbf{h_0}[\mathbf{2}] & h_0[1] + \mathbf{h_0}[\mathbf{3}] & h_0[0] + \mathbf{h_0}[\mathbf{4}] & h_0[5] & 0 & 0 & \dots \\ \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] + \mathbf{h_0}[\mathbf{4}] & h_0[1] + h_0[5] & h_0[0] + h_0[6] & 0 & 0 & \dots \\ \mathbf{h_0}[\mathbf{4}] & \mathbf{h_0}[\mathbf{3}] + h_0[5] & \mathbf{h_0}[\mathbf{2}] + h_0[6] & h_0[1] & h_0[0] & 0 & \dots \\ h_0[5] & \mathbf{h_0}[\mathbf{4}] + h_0[6] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & h_0[1] & h_0[0] & \dots \\ \vdots & \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ \vdots \end{bmatrix} = \begin{bmatrix} v_0[-3] \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix} \tag{122}$$

which is partially expressed in the time domain. Once again this processes produces the necessary symmetry in $\mathbf{v}^{ext}$ such that

$$\mathbf{v}_0^{ext} = v_0[-3], v_0[-2], v_0[-1], v_0[0], v_0[1], v_0[2], v_0[3], \ldots, v_0[6], v_0[7], v_0[8], \ldots, v[10], v[11]]^T \tag{123}$$

and is of length 14. Similar to the half-point procedure, this is sufficient information for reconstruction.

The above example can be recast in the following way. Given the input signal

$$\mathbf{x} = [x[0], x[1], x[2], \ldots, x[N-2], x[N-1]]^T, \tag{124}$$

the symmetric extension results in

$$\mathbf{x}^{ext} = [x[rnd(L/2)-1], \ldots, x[1], x[0], x[1], x[2], .., x[N-2], x[N-1], \ldots, x[N-(rnd(L/2))]]^T. \tag{125}$$

Following the progression of $\mathbf{x}^{ext}$ through the diagram in Figure 34,

$$\mathbf{v}^{ext} = \mathbf{A}\mathbf{x}^{ext}, \tag{126}$$

when

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}, \tag{127}$$

$\mathbf{H}_0$ is a lowpass filter matrix containing the filter coefficients $\mathbf{h}_0$,

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{h}_0[rnd(L/2)-1] & \ldots & h_0[0] & 0 & 0 & \ldots & & 0 \\ \mathbf{h}_0[rnd(L/2)] & \ldots & h_0[1] & h_0[0] & 0 & \ldots & & 0 \\ h_0[rnd(L/2)+1] & \ldots & h_0[2] & h_0[1] & h_0[0] & 0 & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & \mathbf{h}_0[rnd(L/2)-1] & \mathbf{h}_0[rnd(L/2)-2] \\ 0 & \ldots & 0 & 0 & 0 & \ldots & \mathbf{h}_0[rnd(L/2)] & \mathbf{h}_0[rnd(L/2)-1] \end{bmatrix}, \tag{128}$$

and $\mathbf{H}_1$ is a highpass filter matrix containing the filter coefficients $\mathbf{h}_1$,

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{h}_1[rnd(L/2)-1] & \dots & h_1[0] & 0 & 0 & \dots & & 0 \\ \mathbf{h}_1[rnd(L/2)] & \dots & h_1[1] & h_1[0] & 0 & \dots & & 0 \\ h_1[rnd(L/2)+1] & \dots & h_1[2] & h_1[1] & h_1[0] & 0 & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & \mathbf{h}_1[rnd(L/2)-1] & \mathbf{h}_1[rnd(L/2)-2] \\ 0 & \dots & 0 & 0 & 0 & \dots & \mathbf{h}_1[rnd(L/2)] & \mathbf{h}_1[rnd(L/2)-1] \end{bmatrix}. \tag{129}$$

The filtered output may be written as

$$\mathbf{v}^{ext} = \begin{bmatrix} \mathbf{v}_0^{ext} \\ \mathbf{v}_1^{ext} \end{bmatrix}, \tag{130}$$

where the lowpass filtered output is

$$\mathbf{v}_0^{ext} = v_0[-(rnd(L/2)+1)], .., v_0[0], v_0[1], .., v_0[N-1], v_0[N], .., v_0[N+(rnd(L/2))]]^T, \tag{131}$$

and the highpass filtered output is

$$\mathbf{v}_1^{ext} = v_1[-(rnd(L/2)+1)], .., v_1[0], v_1[1], .., v_1[N-1], v_1[N], .., v_1[N+(rnd(L/2))]]^T. \tag{132}$$

If the regions of support in Equation (128) and (129) are increased, Equations (102), (103), (104), and (105) will be true for filter banks with odd-length filters also. This would verify that symmetry is accomplished at the boundary edges. As for expansion, again, only half the original samples should remain in each channel of a two-band system. Thus, each channel must be downsampled which in the matrix equation representation results in

$$\mathbf{y}^{ext} = \Lambda \mathbf{A} \mathbf{x}^{ext} \tag{133}$$

where

$$\Lambda = \begin{bmatrix} \lambda & \\ & \lambda \end{bmatrix} \tag{134}$$

and

$$\lambda = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{135}$$

Through careful evaluation one notices that each channel's downsampled filtered output can be written using Equation (109) where

$$\mathbf{y}_0^{ext} = v_0[-(rnd(L/2)+1)], \dots, v_0[-1], v_0[1], \dots, v_0[N-1], v_0[N+1], .., v_0[N+(rnd(L/2))]]^T \tag{136}$$

for the lowpass channel and

$$\mathbf{y}_1^{ext} = v_1[-(rnd(L/2)+1)], \dots, v_1[-1], v_1[1], .., v_1[N-1], v_1[N+1], .., v_1[N+(rnd(L/2))]]^T \tag{137}$$

for the highpass channel. When both channels are windowed with an $L/2$ point window, the analysis output becomes

$$\hat{\mathbf{y}} = w(\Lambda \mathbf{v}^{ext}) \tag{138}$$

or

$$\hat{\mathbf{y}} = \begin{bmatrix} w(\mathbf{y}_0^{ext}) \\ w(\mathbf{y}_1^{ext}) \end{bmatrix}. \tag{139}$$

This means

$$w(\mathbf{y}_0^{ext}) = y_0[1], y_0[3], \dots, y_0[N-3], y_0[N-1]^T \tag{140}$$

for the lowpass channel and

$$w(\mathbf{y}_1^{ext}) = y_1[1], y_1[3], \dots, y_1[N-3], y_1[N-1]^T \tag{141}$$

for the highpass channel despite the fact that Equations 135 and 108 are not the same. This difference is necessary to guarantee symmetry throughout the system.

The process is repeated in reverse order in the synthesis subsystem. This entire process is referred to as whole-point extension and is viable *if and only if* odd-length filters are used [4]. This odd-length method eliminates expansion and reduces distortion at the reconstructed image's edge.

Despite symmetric extension's apparent non-expansive properties and its ability to mitigate boundary distortion, it can only be used in limited situations. Symmetric extension can only be used for linear phase filters. This means that it can only be used with QMF filters and other linear phase filters. Thus, CQF filters, which have been proven to have better frequency responses than QMF and have been proven to achieve ER [40, 41, 38], are unusable in the filter bank framework.

8.3 Adaptive Boundary Symmetric Extension

Symmetric extension is useful for reducing boundary distortion. Despite this, symmetric extension is not useable for nonlinear phase filters. This means that many high performing filters cannot be used with symmetric extension with a filter bank framework. Arrowood in [2] suggested a way to apply symmetric extension to nonlinear phase filters. Despite this, no simulations were ever conducted. One of the objectives of this dissertation is to develop an even-length a proof of concept for adaptive boundary symmetric extension and assess the method. We will also develop the method for the odd-length case, which was not done by Arrowood. Finally, in this section we will evaluate the method for image coding, which was not done by Arrowood. Through Arrowood's example and Figure 35, we will review his work and then proceed to the contributions of this dissertation. The following example is summarized from [2].

8.3.1 Adaptive Boundary Half-Point Symmetric Extension

Consider again a lowpass six-tap filter. This time, the filter is nonlinear phase. This nonlinear phase filter is shown in Equation (142).

$$h_0 = [h_0[0], \mathbf{h_0[1]}, \mathbf{h_0[2]}, h_0[3], h_0[4], h_0[5]] \tag{142}$$

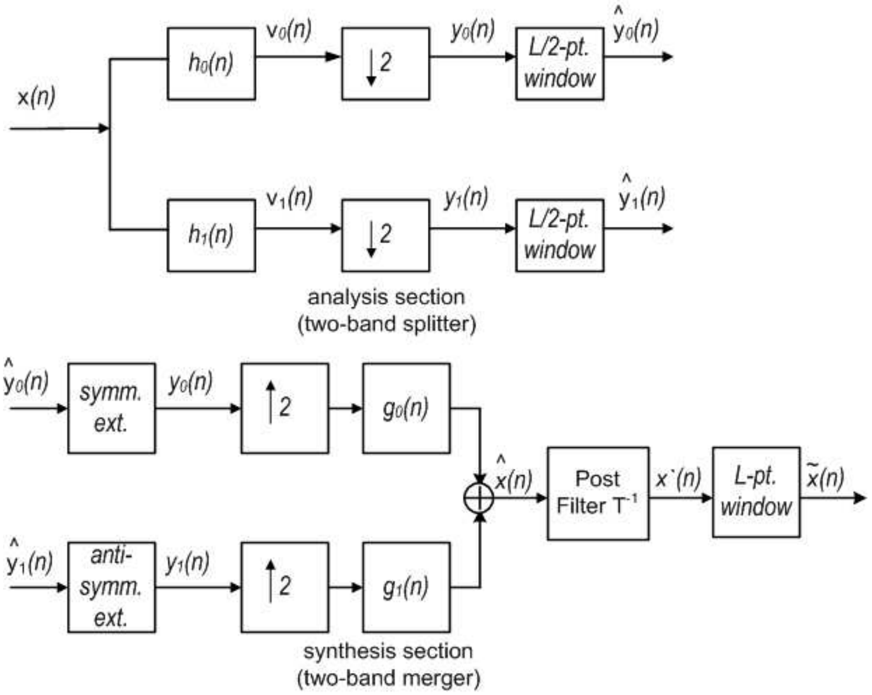


Figure 35: New two-band analysis-synthesis filter bank for nonlinear phase symmetric extension.

In this equation $\mathbf{h_0}[\mathbf{1}]$ and $\mathbf{h_0}[\mathbf{2}]$ are the largest coefficients of the mainlobe. Because the filter's coefficients are anti-symmetric and therefore nonlinear, this filter cannot be used for traditional symmetric extension. Thus, the approaches outlined previously must be reformulated. Since this filter is even-length, $\mathbf{x}$ must have a half-point symmetric extension, as indicated by [4]. This makes

$$\mathbf{x}^{ext} = [x[3], x[2], x[1], x[0], x[0], x[1], x[2], \ldots, x[N-2], x[N-1], x[N-1], x[N-2]]^T \tag{143}$$

where, because of the filter's anti-symmetry characteristics, the signal is extended by four taps on the left boundary and two taps on the right boundary. We know from Figure 35 that

$$\mathbf{v}_0^{ext} = \mathbf{H}_0 \mathbf{x}^{ext}. \tag{144}$$

Written in the time domain, and extended for demonstration purposes, Equation (144) is

$$\begin{bmatrix} \vdots & \vdots & & & & & & & & \vdots \\ h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & 0 & 0 & \dots \\ 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & 0 & \dots \\ 0 & 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & \dots \\ 0 & 0 & 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & \dots \\ 0 & 0 & 0 & 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & \dots \\ \vdots & \vdots & & & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[5] \\ x[4] \\ x[3] \\ x[2] \\ x[1] \\ x[0] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix} \tag{145}$$

where $\mathbf{h}_0$ is placed such that the mainlobe straddles $[\dots, x[0], x[0], \dots]$ when the output for $\mathbf{v}$ is $v[0]$. For the sake of tracking this row, we will call it the boundary row. Symmetry at the boundary is ensured by time-reversing all filter coefficients in $\mathbf{H}_0$ above the boundary row. The time-reversing point in these rows is the half-point between the two largest coefficients of the mainlobe. This yields

$$\mathbf{v}_0^{ext} = \mathbf{H}_0^{tv} \mathbf{x}^{ext} \tag{146}$$

where Equation (146) is expressed in the time domain as

$$\begin{bmatrix} \vdots & \vdots & & & & & & & & \vdots \\ 0 & 0 & h_0[0] & \mathbf{h_0[1]} & \mathbf{h_0[2]} & h_0[3] & h_0[4] & h_0[5] & 0 & \dots \\ 0 & 0 & 0 & h_0[0] & \mathbf{h_0[1]} & \mathbf{h_0[2]} & h_0[3] & h_0[4] & h_0[5] & \dots \\ 0 & 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & \dots \\ 0 & 0 & 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & \dots \\ 0 & 0 & 0 & 0 & h_0[5] & h_0[4] & h_0[3] & \mathbf{h_0[2]} & \mathbf{h_0[1]} & \dots \\ \vdots & \vdots & & & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[5] \\ x[4] \\ x[3] \\ x[2] \\ x[1] \\ x[0] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}. \tag{147}$$

Overlapping and adding the extended region in $\mathbf{H}_0$ yields

$$\mathbf{v}_0^{ext} = \mathbf{H}_0^{tv}\mathbf{x} \tag{148}$$

where it is expressed in the time domain as

$$\begin{bmatrix} \vdots & \vdots & & & & & \vdots \\ h_0[4]+h_0[3] & h_0[5]+\mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & 0 & \dots \\ h_0[3]+\mathbf{h_0[2]} & h_0[4]+\mathbf{h_0[1]} & h_0[5]+h_0[0] & 0 & 0 & 0 & \dots \\ \mathbf{h_0[1]}+\mathbf{h_0[2]} & h_0[0]+h_0[3] & h_0[4] & h_0[5] & 0 & 0 & \dots \\ \mathbf{h_0[2]}+h_0[3] & \mathbf{h_0[1]}+h_0[4] & h_0[0]+h_0[5] & 0 & 0 & 0 & \dots \\ h_0[3]+h_0[4] & \mathbf{h_0[2]}+h_0[5] & \mathbf{h_0[1]} & h_0[0] & 0 & 0 & \dots \\ \vdots & \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}. \tag{149}$$

The procedure above illustrates adaptive boundary symmetric extension at the left boundary. Comparable action is taken to accomplish symmetry at the right boundary.

It is clear by using Equations (142) through (148) that symmetry is preserved, as can be seen in Equation (149). This means that

$$\begin{aligned} v_0[-1] &= v_0[1] \\ v_0[-2] &= v_0[2] \\ &\vdots \end{aligned} \tag{150}$$

and

$$\begin{aligned} v_0[N-1] &= v_0[N+1] \\ v_0[N-2] &= v_0[N+2] \\ &\vdots. \end{aligned} \tag{151}$$

Incorporating downsampling into the process yields

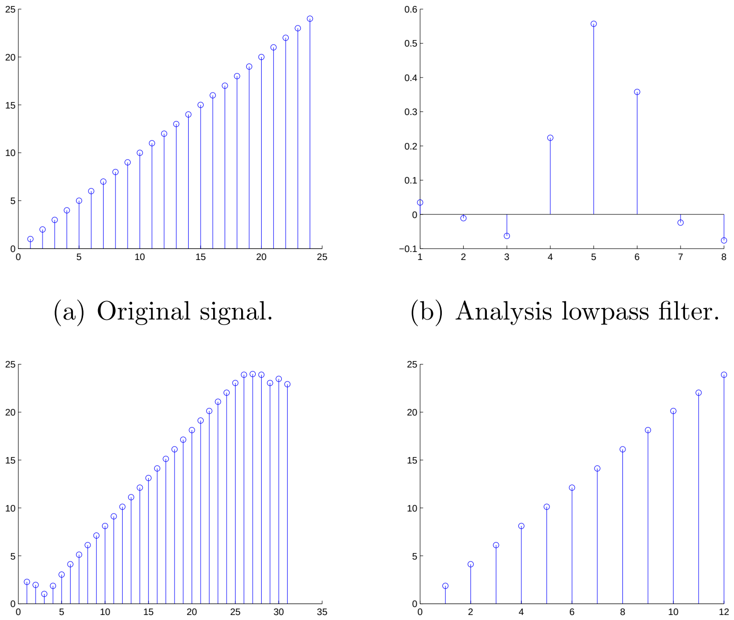
$$\mathbf{y}_0^{ext} = [\ldots, v[3], v[1], v[1], v[3], \ldots, v[N-3], v[N-1], v[N-3], \ldots]. \tag{152}$$

Repeating the procedure for the highpass channel completes the technique and preserves symmetry in the highpass channel.

The adaptive boundary symmetric extension resolves the issues surrounding symmetry at image boundaries. Adaptive boundary symmetric extension also allows the use of nonlinear phase filters with symmetric extension. Despite this, flipping the filter's coefficients at the boundary to preserve symmetry leads to imperfect reconstruction. To correct this, a postfilter is used to restore PR for this region.

8.3.2 Adaptive Boundary Half-point Symmetric Extension Results

To prove that adaptive boundary half-point symmetric extension works, we start by tracing the progression of a signal through the low frequency channel of an adaptive boundary symmetric extension filter bank. Figure 36 shows the progression through the analysis subsystem, while Figure 37 shows the progression through the synthesis subsystem. More specifically, Figure 36(a) shows the input signal. This signal is convolved in the low frequency channel by $h_0[n]$, shown in Figure 36(b). The particular filter $h_0[n]$ is an eight-tap near linear phase CQF filter in [41]. The filtered output $\hat{v}_0[n]$ is shown in Figure 36(c). The lowpass filter's result is then downsampled and windowed and the resulting signal is shown in Figure 36(d).



(a) Original signal.

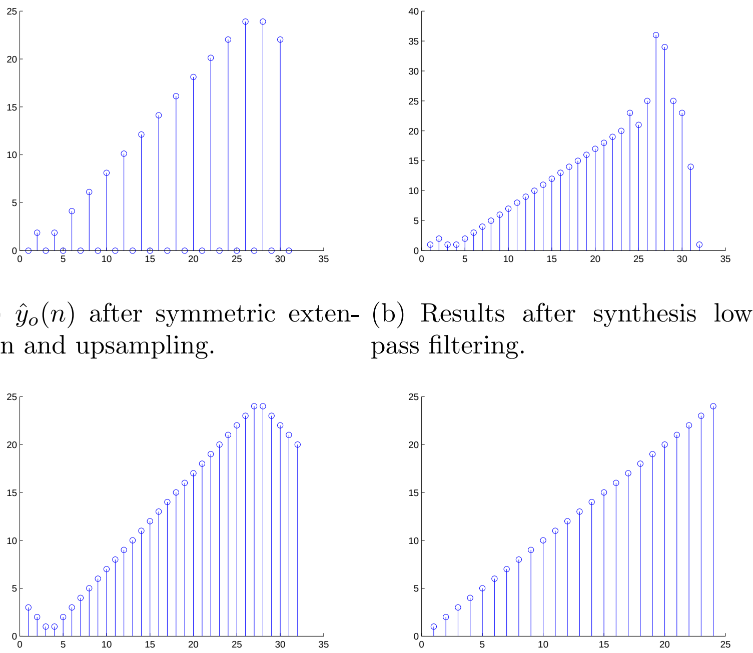
(b) Analysis lowpass filter.

(c) $\hat{v}_o(n)$.

(d) $\hat{y}_o(n)$ after window function.

Figure 36: Results from a filter bank with nonlinear symmetric extension.

Figure 37(a) shows the results of the analysis output after it has been symmetrically extended and upsampled in the synthesis subsystem. This signal is then convolved in the low frequency channel by $G_0(z)$, which is equal to $z^L H_0(z^{-1})$, resulting in the signal shown in Figure 37(b). Inspecting the results from Figure 37(b) one quickly notices that the boundary edges contain imperfections, as expected. To rectify this distortion, postfilters are applied resulting in the signal shown in Figure 37(c). Windowing is performed next to reconstruct the original signal. This result is shown in Figure 37(d).



(a) $\hat{y}_o(n)$ after symmetric extension and upsampling.

(b) Results after synthesis lowpass filtering.

(c) Results after postfiltering.

(d) Results after windowing.

Figure 37: Illustration of filter bank stage outputs for adaptive boundary half-point symmetric extension.

8.3.3 Nonlinear Phase Whole-Point Symmetric Extension

A technique for adaptive boundary half-point symmetric extension was explored in [2]. Arrowood, however, never explored odd-length adaptive boundary symmetric extension. In this section, we extend the theory to the odd-length case. Consider once again the lowpass seven-tap filter shown below

$$h_0 = [h_0[0], \mathbf{h_0[1]}, \mathbf{h_0[2]}, \mathbf{h_0[3]}, h_0[4], h_0[5], h_0[6]], \tag{153}$$

where $\mathbf{h_0[1]}$, $\mathbf{h_0[2]}$, and $\mathbf{h_0[3]}$ are the largest coefficients of the mainlobe. Because the filter coefficients again are not symmetric, they cannot be used for traditional symmetric extension. Since this filter is odd-length, we must employ whole-point symmetric extension of the input resulting in

$$\mathbf{x}^{ext} = [x[5], x[4], x[3], x[2], x[1], x[0], x[1], \ldots, x[N-2], x[N-1], x[N-2], x[N-3], x[N-4]]^T. \tag{154}$$

In addition, because of the filter's anti-symmetric characteristics, the signal is extended this time by five taps on the left boundary and three taps on the right boundary. Note that the region of support is $N+1$ instead of N, as with the half-point example.

We know from Figure 35 that

$$\mathbf{v}_0^{ext} = \mathbf{H}_0\mathbf{x}^{ext}. \tag{155}$$

Written in the time domain and extended for illustration purposes, Equation (155) can be expressed as

$$\begin{bmatrix} \vdots & \vdots & \vdots & & & & & & & \vdots \\ h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{1}] & h_0[0] & 0 & 0 & \ldots \\ 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{1}] & h_0[0] & 0 & \ldots \\ 0 & 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{1}] & h_0[0] & \ldots \\ 0 & 0 & 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & \mathbf{h_0}[\mathbf{1}] & \ldots \\ 0 & 0 & 0 & 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0}[\mathbf{3}] & \mathbf{h_0}[\mathbf{2}] & \ldots \\ \vdots & \vdots & \vdots & & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[5] \\ x[4] \\ x[3] \\ x[2] \\ x[1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ v_0[-3] \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}, \tag{156}$$

where $\mathbf{h}_0$ is placed such that the center coefficient of the mainlobe ($\mathbf{h_0}[\mathbf{2}]$) is multiplied with $x[0]$ (as a part of convolution) when the output for $\mathbf{v}$ is $v[0]$. Again, for the sake of tracking this critical row, we will call it the boundary row. Symmetry at the boundary is ensured by time-reversing all filter coefficients in $\mathbf{H}_0$ above the boundary row. The time-reversing point of these rows are the $\mathbf{h_0}[\mathbf{2}]$ coefficients of the mainlobe. This yields

$$\mathbf{v}_0^{ext} = \mathbf{H}_0^{tv}\mathbf{x}^{ext}, \tag{157}$$

where it is expressed in the time domain as

$$\begin{bmatrix} \vdots & \vdots & & & & & & & & \vdots \\ 0 & h_0[0] & \mathbf{h_0[1]} & \mathbf{h_0[2]} & \mathbf{h_0[3]} & h_0[4] & h_0[5] & h_0[6] & 0 & \dots \\ 0 & 0 & h_0[0] & \mathbf{h_0[1]} & \mathbf{h_0[2]} & \mathbf{h_0[3]} & h_0[4] & h_0[5] & h_0[6] & \dots \\ 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & \dots \\ 0 & 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & \dots \\ 0 & 0 & 0 & h_0[6] & h_0[5] & h_0[4] & \mathbf{h_0[3]} & \mathbf{h_0[2]} & \mathbf{h_0[1]} & \dots \\ \vdots & \vdots & & & & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[5] \\ x[4] \\ x[3] \\ x[2] \\ x[1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ v_0[-3] \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}. \tag{158}$$

Overlapping and adding the extended region in $\mathbf{H}_0$ yields

$$\mathbf{v}_0^{ext} = \mathbf{H}_0^{tv}\mathbf{x}, \tag{159}$$

or, equivalently

$$\begin{bmatrix} \vdots & \vdots & & & & & \vdots \\ h_0[4] & h_0[5] + \mathbf{h_0[3]} & h_0[6] + \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & \dots \\ \mathbf{h_0[3]} & h_0[4] + \mathbf{h_0[2]} & h_0[5] + \mathbf{h_0[1]} & h_0[6] + h_0[0] & 0 & 0 & \dots \\ \mathbf{h_0[2]} & \mathbf{h_0[1]} + \mathbf{h_0[3]} & h_0[0] + h_0[4] & h_0[5] & h_0[6] & 0 & \dots \\ \mathbf{h_0[3]} & \mathbf{h_0[2]} + h_0[4] & \mathbf{h_0[1]} + h_0[5] & h_0[6] + h_0[0] & 0 & 0 & \dots \\ h_0[4] & \mathbf{h_0[3]} + h_0[5] & h_0[6] + \mathbf{h_0[2]} & \mathbf{h_0[1]} & h_0[0] & 0 & \dots \\ \vdots & \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ v_0[-3] \\ v_0[-2] \\ v_0[-1] \\ v_0[0] \\ v_0[1] \\ v_0[2] \\ \vdots \end{bmatrix}. \tag{160}$$

These procedures illustrate adaptive boundary symmetric extension at the left boundary for odd-length filters. Comparable action is taken to accomplish symmetry at the right boundary for odd-length filters.

It is clear by using Equations (153) - (159) that symmetry is preserved, as can be seen in Equation (160). This means that

$$\begin{aligned} v_0[-1] &= v_0[1] \\ v_0[-2] &= v_0[2] \\ &\vdots \end{aligned} \tag{161}$$

and

$$\begin{aligned} v_0[N-1] &= v_0[N+1] \\ v_0[N-2] &= v_0[N+2] \\ &\vdots \end{aligned} \tag{162}$$

After downsampling we obtain

$$\mathbf{y}_0^{ext} = [\ldots, y[3], y[1], y[1], y[3], \ldots, y[N-3], y[N-1], y[N-3], \ldots]. \tag{163}$$

The same procedure is performed on the high frequency channel.

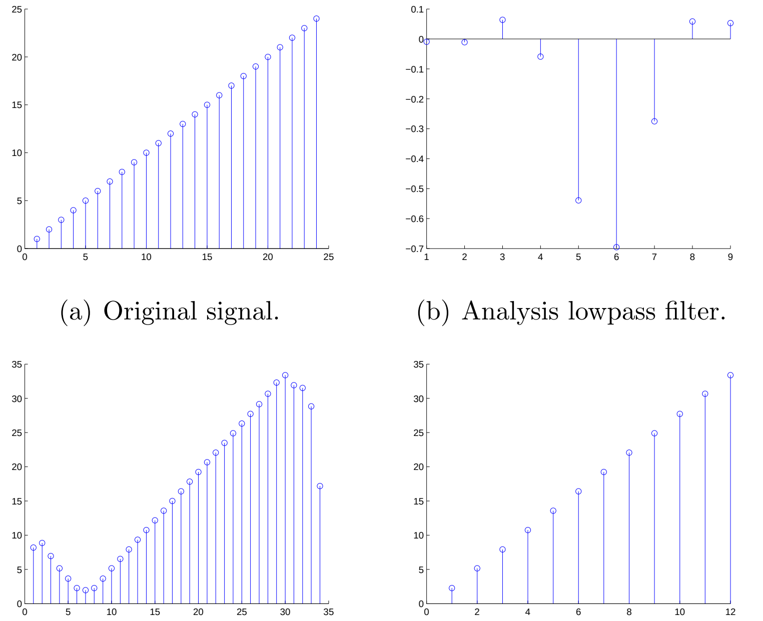
Using adaptive boundary symmetric extensions allows odd-length nonlinear phase filters to be used within the filter bank framework. Although flipping the filter coefficient at the boundary creates imperfections, postfilters can be used in these regions, leading to exact reconstruction.

8.3.4 Adaptive Boundary Whole-point Symmetric Extension Results

Now that we have developed the method to perform nonlinear phase whole-point symmetric extension, we need to verify that it works. We will trace the progression of a signal through the low frequency channel of an adaptive boundary symmetric extension filter bank. Figure 38 shows the progression through the analysis subsystem, while Figure 39 shows the progression through the synthesis subsystem.

The input shown in Figure 38(a) is convolved in the low frequency channel by the 9-tap filter $h_0[n]$, shown in Figure 38(b). The resulting output is shown in Figure

38(c). If the region of support were made larger, samples at the exterior of the original signal edge samples would be mirrored in the extended area. The lowpass filter's result is then downsampled and windowed, and the resulting signal is shown in Figure 38(d).



(a) Original signal.

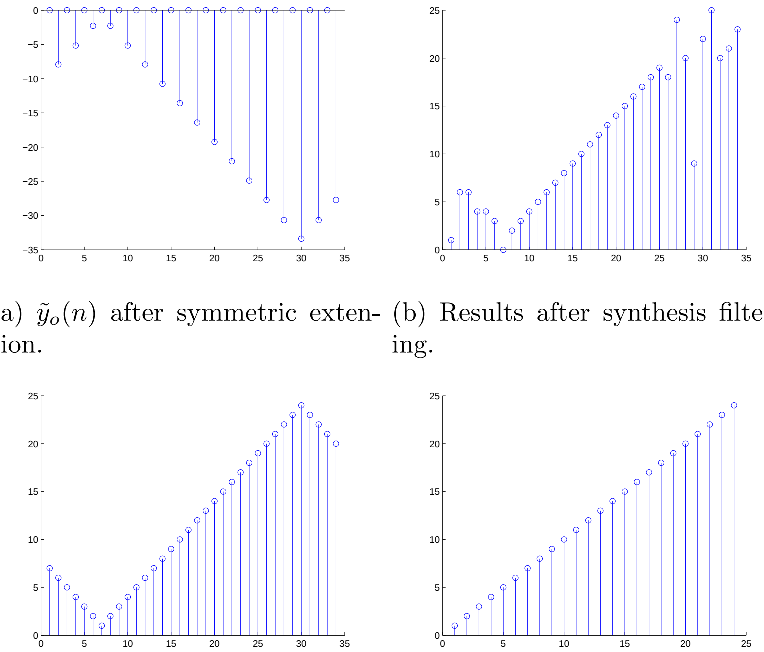
(b) Analysis lowpass filter.

(c) $\tilde{v}_o(n)$.

(d) $\tilde{y}_o(n)$ after window function.

Figure 38: Results from a filter bank with nonlinear symmetric extension.

Figure 39(a) shows the results of the analysis' output after it has been symmetrically extended and upsampled in the synthesis subsystem. This signal is then convolved in the low frequency channel by $G_0(z)$, which is equal to $z^L H_0(z^{-1})$. This results in the signal shown in Figure 39(b). Similar to the half-point results, Figure 39(b) boundary edges contain imperfections. Once again postfilters are applied at these regions resulting in the signal shown in Figure 39(c). Windowing is performed next to reconstruct the original signal. This result is shown in Figure 39(d).



(a) $\tilde{y}_o(n)$ after symmetric extension.

(b) Results after synthesis filtering.

(c) Results after synthesis lowpass filter.

(d) Results after windowing.

Figure 39: Results from a filter bank with nonlinear symmetric extension.

CHAPTER IX

APPLICATION OF FAAS SYSTEMS TO INTERPOLATION AND CODING

In this chapter, we apply the FAAS filter bank to two important problem areas: image compression and image interpolation. In the first section of this chapter, we describe the compression system employed in our evaluation. We comment on some of the particulars associated with the filters and the implementation method and conclude with some comparative results.

In the second section of this chapter, we discuss the application of the FAAS filter bank to image interpolation (or resizing). This is a very common application that appears in virtually all computer programs that manage digital images, from Microsoft Word to Photoshop. Comparative results are presented at the end of the section that illustrate the improvement that can be obtained by employing FAAS filter banks.

9.1 Application to Image Compression

The SPIHT coder developed by Said and Pearlman [36] is a very popular subband image coder that is widely used for benchmarking. Thus, we use the SPIHT coder as the baseline system to compare the performance of the FAAS filter bank against the conventional FAFS currently employed in the industry.

The analysis section of the SPIHT coder is shown in Figure 40. The SPIHT coder uses the Daubechies 9/7 lowpass and highpass filters to perform its subband decomposition. These filter coefficients are shown in Table 17, and are designed for a two-band system that is maximally decimated. Because these filters are linear phase

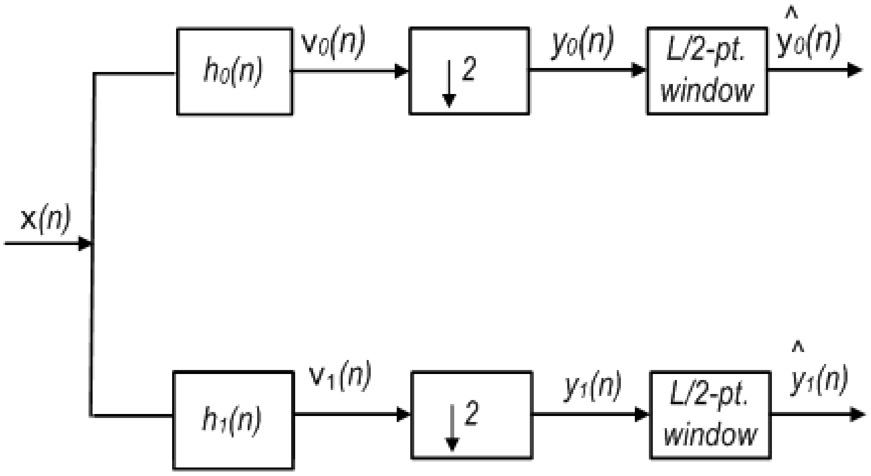


Figure 40: Block diagram of the analysis system of the SPIHT coder.

biorthogonal filters, symmetric extension is used to ensure exact reconstruction. For implementational convenience we combine the filtering, downsampling, and symmetric extension functions into two convolution matrices as illustrated in Section 8.2.2. One convolution matrix contains the lowpass channel structure and the other contains the highpass channel structure.

After the filtering and downsampling, the subband images are quantized and entropy coded using a progressive adaptive coder which was pioneered originally by Shapiro [37] and refined by Said and Pearlman [36]. Quantization in the SPIHT coder is performed on the bit-plane level, which allows bits to be allocated efficiently in order to reduce quantization error. The quantized values are then coded with an adaptive arithmetic coder. In this work, we use the conventional SPIHT encoder. The deployment of the FAAS filter-bank results in a modification to the decoder's filter bank synthesis section and to the symmetric extension implementation.

In the decoder synthesis filter bank, we start by symmetrically extending the input signal. Since linear phase filters were used in the analysis filter, the signal is extended by L/2 samples on either side of the signal. For implementational convenience, we

Table 17: Daubechies 9/7 analysis filter coefficients

H_0	H_1
0.03782845550726	-0.06453888262870
-0.02384946501956	0.04068941760916
-0.11062440441844	0.41809227322162
0.37740285561283	-0.78848561640558
0.85269867900889	0.41809227322162
0.37740285561283	0.04068941760916
-0.11062440441844	-0.06453888262870
-0.02384946501956	0
0.03782845550726	0

incorporated symmetric extension into each convolution matrix. The symmetric extension used is dependent on the filters used in each matrix. If low delay or high delay filters are in the matrix, we must use the adaptive boundary symmetric extension as described in Section 8.3.4. Otherwise, traditional symmetric extension is used as described in Section 8.2.2. Unlike the analysis convolution matrices, we do not overlap and add matrices in the synthesis convolution matrices. If the overlap and add is done, boundary values in the subsystem output would be doubled.

9.1.1 Filtering Upsampling and Phase Selection

The remaining components of the convolution matrices are upsampling and filtering. To perform upsampling, the convolution matrices are simply transposed after the filter coefficients are incorporated. The filters for the linear phase portion of the FAAS system use the Daubecies 9/7 synthesis filters. Complementary low delay and high delay filters were designed for use in the FAAS filter bank. These filters are listed in Table 18 for the lowpass filters and in Table 19 for the highpass filters.

The final step in the FAAS system is to window the output and determine which of the three reconstructions to use in the final synthesis process. For this, we will use the phase selection method described in Section 7.2. The final synthesis subsystem is shown in Figure 41.

Table 18: Daubechies 9/7 synthesis lowpass filter and complementary synthesis lowpass filters.

G_{00}	G_{01}	G_{02}
0.74789653543220	-0.06453888262897	0
0.79166508748431	-0.04068941760930	0.00022205891726
-0.04101181316292	0.41809227322225	-0.05865186334346
-0.08020455820952	0.78848561640516	-0.04101181316292
0.00022205891726	0.41809227322225	0.37817288321209
-0.00435374808825	-0.04068941760931	0.74789653543220
0	-0.06453888262897	0.38758576131792

Table 19: Daubechies 9/7 synthesis highpass filter and complementary synthesis highpass filters.

G_{10}	G_{11}	G_{12}
0.36225423798497	-0.03782845550717	0
-0.75371272132055	-0.02384946501974	0
0.36889678548531	0.11062440441872	0.00013347893558
0.07250155009196	0.37740285561309	-0.03428376831574
-0.02417772121931	-0.85269867900966	-0.02417772121931
-0.02476852465009	0.37740285561307	0.09921654795315
0.00013347893558	0.11062440441872	0.36889678548531
-0.00112708530786	-0.02384946501973	-0.77203956082396
0	-0.03782845550717	0.36225423798497

In the next section we present comparative results that highlight the improvements possible with FAAS filter banks.

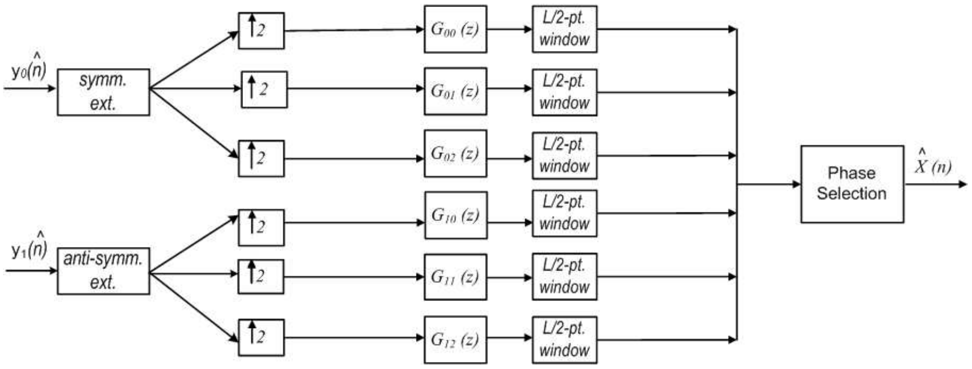


Figure 41: Block diagram of the Adaptive-Synthesis Filter Bank.

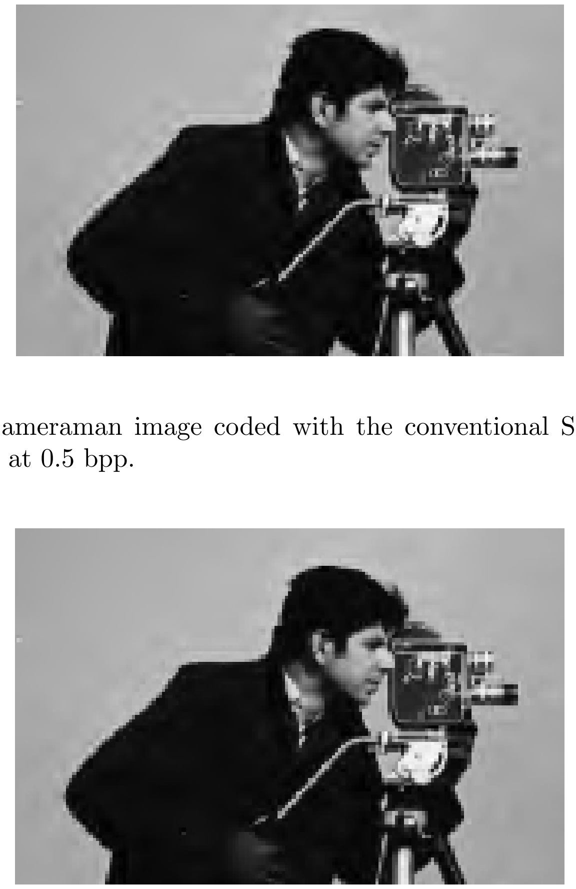
9.2 Image Compression Results

The fixed-analysis-adaptive-synthesis filter bank was incorporated into the popular SPHIT coder and compared against the conventional SPHIT coder that uses fixed analysis-synthesis filter banks. In this particular implementation, the adaptive synthesis filters are used only in the base level of the subband tree, to which to the most dominant visual coding distortions are attributable.

Several images were tested. Both the conventional SPHIT and FAAS SPHIT coders were evaluated. At this point, we are using a simple empirical rule to determine which pixel from among the three reconstructions to choose for reconstruction. In each case, the FAAS SPHIT algorithm outperformed the conventional. However, the improvements both subjectively and in terms of SNR are very modest (less than one-tenth dB in PSNR). The marginal improvement may be attributed to the simplicity of the selection algorithm, which we recognize needs to be improved. The potential for high performance using the FAAS SPHIT algorithm is actually quite significant. If we employ an optimal selection algorithm the improvement is both visible and measurable in terms of PSNR.

As an illustration we show in Figure 42(a) a section of the cameraman image coded at a bit rate of 0.5 bpp. Distortions can be observed in a number of places, particularly around the body outline. Shown in Figure 42(b) is the same image coded at the same bit rate using the FAAS SPHIT with optimal selection. It is evident by comparing the two coded images that the FAAS SPIHT coder has less distortion. This improvement is also reflected in the PSNR, which turns out to be an improvement of approximately $1.5dB$. The conventional SPHIT coder achieves $31.47dB$ while the adaptive SPHIT achieves $32.95dB$.

Although we cannot implement an optimal selection algorithm in practice, a slightly more sophisticated selection approach can be expected to better approximate the optimal.



(a) Cameraman image coded with the conventional SPIHT coder at 0.5 bpp.

(b) Cameraman image coded with FAAS SPHIT coder using optimal selection at 0.5 bpp.

Figure 42: SPIHT coder comparisions.

9.3 Interpolation

Many methods have been proposed for enlarging digital images. One of the simplest methods is zero-order (nearest-neighbor) interpolation. It requires the least amount of digital processing. In this method, each pixel is replicated as a 2×2 pixel in the case of a 1:2 interpolation. Digital zooming is often performed using this method, however, the output often acquires undesirable blocking artifacts known as pixellation. While there are many other methods that have been proposed, the most popular continue to be those based on simple low order upsampling and filtering. In particular, bilinear and bicubic interpolation are most prevalent in practice, owing to their relative simplicity and good performance.

Adaptive interpolation filters, like the ones developed for FAAS filter banks, provide an alternative method for interpolation. This approach involves switching among a set of filters dynamically in a way intended to improve reconstruction quality. One of the key issues previously associated with adaptive filter banks is the synchrony between the analysis and synthesis filters. For the interpolation application, this is not of consequence, since we are only concerned with the synthesis section. The key issue in our case is how to design the filters and orchestrate the dynamic filtering to maximize interpolation quality.

In this section, we apply the FAAS synthesis section to maximize interpolation quality. We have limited the filters to three lowpass filters: a low delay, a linear phase, and a high delay filter. Each of these filters individually can interpolate the image with relatively good quality. Together these filters remove ringing that typically accompanies conventional interpolation. In addition, together they provide a large range of methods to improve interpolation quality. We will discuss in detail the methods we have considered for performing the interpolation and show examples of our interpolation results on standard database images and video frames.

9.3.1 Odd-Length Lowpass Adaptive Filters For Interpolation

FAAS filter banks could be applied with even-length filters or with odd-length filters. Because even-length linear phase filters have a fractional sample delay, we have decided to focus on odd-length lowpass filters. The interpolation system is comprised of components of the low frequency channel of the FAAS synthesis filter bank. This portion is shown in Figure 43. Interpolation is performed by switching among the lowpass filters on a pixel-by-pixel basis.

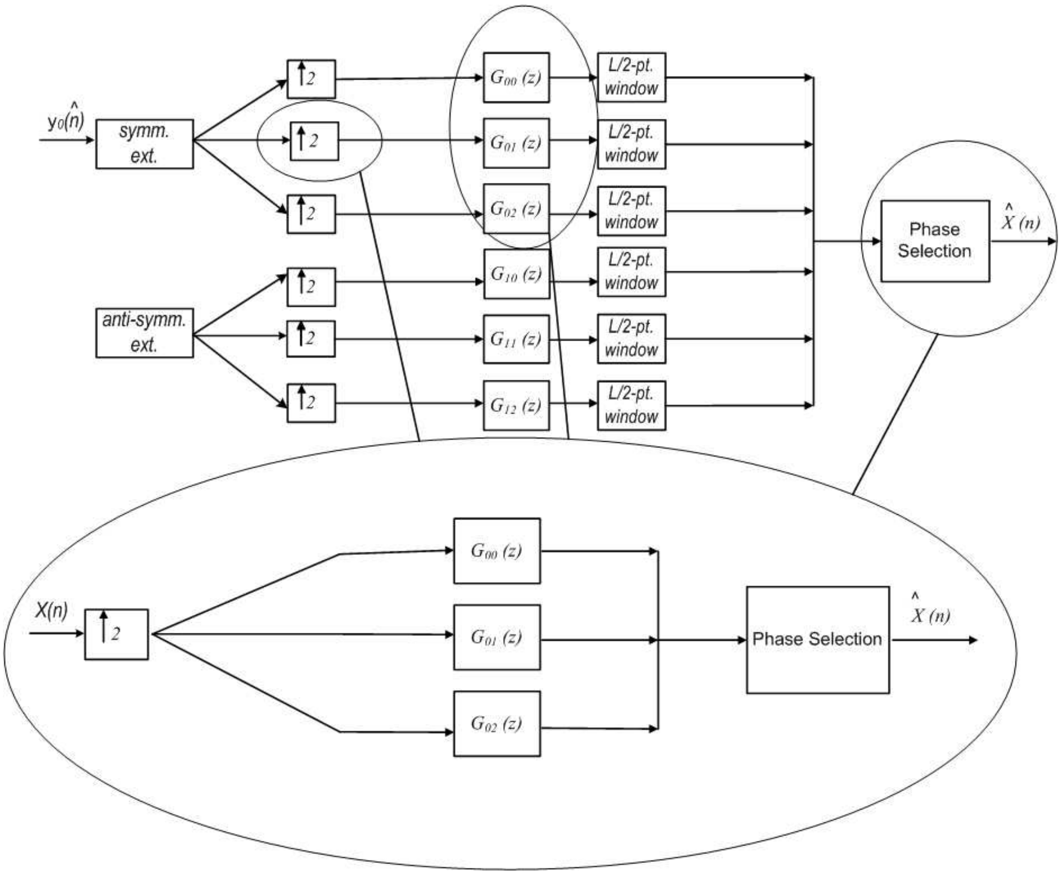


Figure 43: Block diagram of the low-frequency synthesis channel of the FAAS Filter Bank.

For the Daubechies 9/7 synthesis filters, the associated frequency response for the low delay, high delay, and linear phase synthesis filters are shown in Figures 44, 45, and 46.

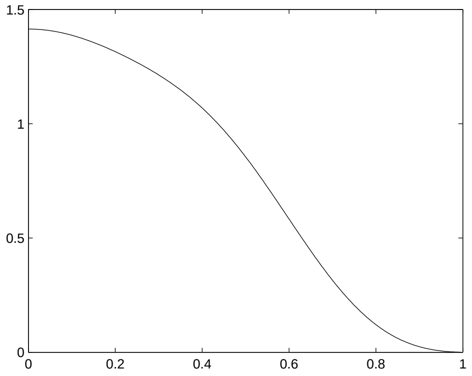


Figure 44: Frequency response for the Daubechies 9/7 lowpass low delay complementary synthesis filter.

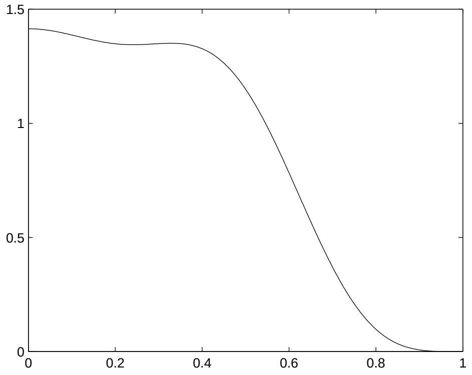


Figure 45: Frequency response for the Daubechies 9/7 lowpass linear phase complementary synthesis filter.

9.3.2 Results

For image reconstruction, we use separable interpolation where we interpolate the rows followed by the columns of the result. For the proposed adaptive synthesis filter bank, this gives us nine reconstructions: low delay on the rows, low delay on the columns; low delay on the rows, high delay on the columns; low delay on the rows,

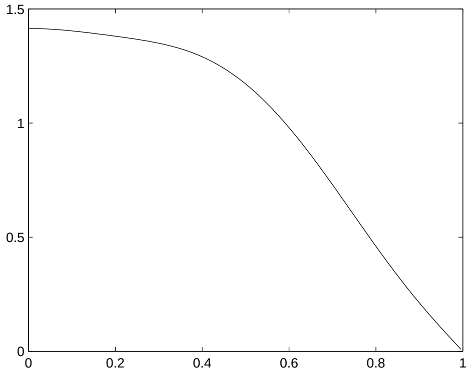


Figure 46: Frequency response for the Daubechies 9/7 lowpass high delay complementary synthesis filter.

linear phase on the columns; and so on. From this set of nine reconstructions, we choose the best of the nine candidates on a pixel-by-pixel basis. A simple empirical rule is applied that determines how pixel selection is performed. First we identify all the dominant edges in the image. When we detect the onset of a positive going edge, we use the result from the low delay interpolation. Similarly, when we detect the onset of a negative edge (i.e. high to low), we use the result from the high delay reconstruction. In all other cases, we use the linear phase reconstruction. It turns out that this simple selection process yields improvement and suggests that even better results could be obtained with refinements to the selection algorithm.

Several images were tested. Both the bicubic and proposed adaptive filter interpolation were evaluated. In each case, the adaptive interpolation algorithm outperformed the conventional bicubic algorithm. However, the improvements subjectively and in terms of PSNR are very modest (less than half a dB PSNR). The marginal improvement may be attributed to the simplicity of the selection algorithm, which we recognize needs to be improved. The potential for high performance using the adaptive interpolation algorithm is actually quite significant. If we employ an optimal selection algorithm the improvement is both visible and measurable in terms of

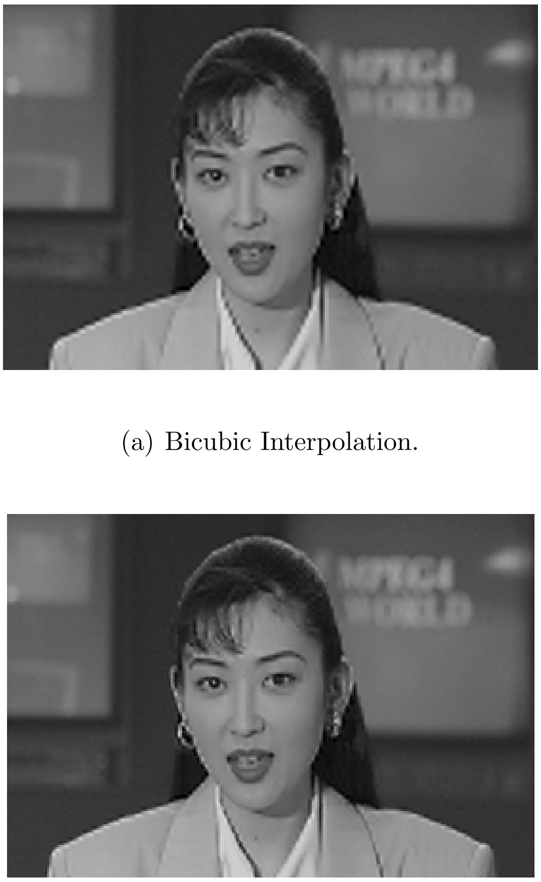
PSNR.

As an illustration we interpolated a 144 by 176 version of the Akiyo image to size 288 by 352 image using bicubic interpolation and the new FAAS interpolation. It is evident by comparing the two interpolated images shown in Figure 47 that the adaptive filter interpolation is significantly sharper. The gains we see are evident both visually and in terms of PSNRs. Table 20 shows a comparison of bicubic interpolation and the proposed adaptive interpolation method for several standard sequences: Stefan, Akiyo, Bus, Football, and Carphone. In all cases the adaptive approach yields improvement of several dB. The minimum performance gain is 2 dB, while the highest is 8 dB.

Table 20: PSNR for Bicubic and Optimal Adaptive Filter Interpolations

Image name	Bicubic	Optimal Adaptive
Stefan	24.56	28.6
Akiyo	28.37	36.52
Bus	25.18	28.08
Football	31.05	33.23
Carphone	27.47	32.86

Although we cannot implement an optimal selection algorithm in practice, a slightly more sophisticated selection approach can be expected to better approximate the optimal. In future work, we will consider the use of an algorithm to better utilize the nine interpolated images available to adaptively interpolate images so that the image is better perceptually than interpolated images using other methods. Currently, the adaptation is based on a simple selection process that suppresses overshoots and undershoots at the edge boundaries.



(a) Bicubic Interpolation.

(b) Adaptive Filter Interpolation.

Figure 47: Interpolated Images

CHAPTER X

CONCLUSION AND FUTURE WORK

This dissertation expands the evolution of filter bank (and wavelet) theory to include new classes of adaptive systems. Four filter bank classes were identified: FAFS, AAAS, FAAS, and AAFS. The focus of this thesis is on the theory, design, and application of FAAS filter banks, which have not been explored previous.

At the onset, FAAS filter banks were hypothesized to be interesting because of their potential to exploit phase diversity in a coding application. It is well known that at high bit rates the output of quantization can be modeled as the signal plus the noise associated with the signal at that spatial location. If reconstructions are performed on the same quantized signal based on a diversity of synthesis filters with different phases, the resulting reconstructions will each contain the signal plus the associated noise spatially displaced. Consequently, since spatial regions with high amplitude changes generate proportionately higher quantization noise and since this noise is now isolated and spatially shifted across the diverse reconstructions, that noise in theory can be suppressed as part of the process of merging the images together. We were successful in demonstrating the proof of this concept using a three phase reconstruction filter bank involving a low delay, a linear phase, and a high delay filter set.

Furthermore, FAAS filter banks are attractive in the sense that they are compatible with existing subband/wavlet systems. Among the most popular applications of analysis/synthesis filter banks are image compression, decimation/interpolation (resizing), and denoising. In virtually all of these cases, the existing systems employ fixed filters. The work introduced here could be applied to any existing system by simply employing the adaptive synthesis structure introduced in this thesis in conjunction with appropriately designed synthesis filters.

Experimental results support the potential utility of FAAS systems in the important areas of compression and interpolation. The theoretical performance gains for the three-phase system we investigated are dramatic in terms of signal-to-noise ratio. The practical gains that were realized in this work are less dramatic but are positive gains. Consequently, we expect that future work using a higher diversity of synthesis filters and/or a more sophisticated recombination model would further improve output reconstruction quality.

Two applications were considered in this thesis: image compression and image interpolation. We expect that FAAS filter banks could also net improvement in denoising applications. It is noteworthy that Arrowood showed improved denoising results using AAAS filter banks. Similar improvement is expected for the FAAS case but with much lower complexity. Such an investigation we believe would be an interesting project for the future.

10.1 Itemized Contributions

The contributions of this thesis are outlined concisely below, broken down in terms of theory and application.

- The FAAS filter bank was evaluated and compared against conventional FAFS systems for image compression.

  The popular SPHIT algorithm was used in the comparison. Performance improvement can be achieved with FAAS filter banks using simple synthesis filter selection algorithms. Experimental results suggest that the potential for dramatic performance improvement is possible through the development of more sophisticated synthesis filter selection algorithms.

- The FAAS filter bank was applied to symmetric extension. Symmetric extension is the preferred method for handling boundary effects in image compression, but is also used to handle boundary effects associated with other image processing tasks. For example, it can be used to handle boundary effects when performing interpolation, for performing decimation, and for image analysis (such as target detection where target chips are employed).

- The FAAS filter bank was examined as part of a video compression algorithm. The video compression algorithm uses an adaptive warping algorithm in lieu of motion compensated prediction. The FAAS was used as a component in the video coder to perform still image frame interpolation.

- The FAAS filter bank was examined for image enlargement and resizing. Experimental results show that improvement can be achieved over the conventional FAFS system.

- Performed a proof of concept for even-length adaptive boundary symmetric extension. Experimental results show that even-length adaptive boundary symmetric extension aid in image reconstruction while not increasing the region of support of an image.

- Developed the method for odd-length adaptive boundary symmetric extension. Because odd-length nonlinear phase complementary filters were needed for the FAAS system, the odd-length adaptive boundary symmetric extension was developed. Through experimentation then application, we were able to show that this method aided in image reconstruction while not increasing the region of support of an image.

Conference Publications:

- Ying Chen, Clyde Lettsome, Mark Smith and Edward Delp, "A Low Bit-rate Video Coding Approach Using Modified Adaptive Warping and Long-Term Spatial Memory" in *Visual Communications and Image Processing.* (VCIP'2007), San Jose, California, Jan. 2007.
- Clyde A. Lettsome, Mark J.T. Smith, and Russell Mersereau, "Fixed Analysis Adaptive synthesis Filter Banks" in *SPIE Defense and Security Conference.*, Orlando, Florida, March. 2008.
- Clyde A. Lettsome and Mark J.T. Smith, "Image Interpolation Exploiting Phase Diversity" in *IEEE DSP Workshop.*, Marco Island, Florida, Jan. 2009.

Journal Publications:

- Jienyu lIn, Clyde Lettsome and Mark Smith, "Optimized Non-linear phase Filters for Subband/Wavlet Coding," in *Transactions on Image Processing (In Preparation).*
- Clyde A. Lettsome, Mark J.T. Smith, and Russell Mersereau "Fixed-Analysis Adaptive-Synthesis Filter Banks: Theory and Applications," in *Transactions on Image Processing (In Preparation).*

10.2 Future Work

The investigation of FAAS filter banks reported in this thesis suggest a number of projects for future investigation. Some of these were mentioned in earlier sections of this chapter.

- Investigate the use of the adaptive synthesis filters on the subsequent levels in the subband tree. Currently, the adaptation is being confined to the base level. Adding multi-layer adaptive filtering should further improve performance results. FAAS filter banks were shown to achieve performance gains at low bit rates.

- Determine the compression rate at which FAAS systems achieve significant gain is a topic worth examining.

- Investigate alternative selection algorithms for exploiting phase diversity.

- Consider the use of a rich set of synthesis filters such as 6, 9, or 12 and investigate the extent to which performance can be improved.

- Investigate and analyze the results of using FAAS high delay filters for image interpolation.

REFERENCES

[1] Arrowood, J. and Smith, M., "Exact reconstruction analysis/synthesis filter banks with time-varying filters," *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 233–236, Apr 1993.

[2] Arrowwood, J., *Theory and Application of Adaptive Filter Banks*. PhD thesis, Georgia Institute of Technology, 1999.

[3] Bellanger, M., Bonnerot, G., and Coudreuse, M., "Digital filtering by polyphase network: Application to sampling rate alteration and filter banks," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, pp. ASSP–24:109–114, April 1976.

[4] Bislawn, C. M., "Preservation of subband symmetry in multirate signal coding," *IEEE Trans. On Signal Processing*, vol. 43, pp. 3046–3050, Dec. 1995.

[5] Chu, P., "Quadrature mirror filter design for an arbitrary number of equal bandwidth channels.," *IEEE Trans. Acoust. Speech Signal Process*, vol. 33, pp. 203–218, February 1985.

[6] Chung, W. and Smith, M., "Spatially-varying iir filter banks for image coding," in *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, (Minneapolis, Minnesota), pp. 570–573, Apr 1993.

[7] Cox, R., "The design of uniformly and nonuniformly spaced pseudoquadrature mirror filters.," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 1090–1096, October 1986.

[8] Croisier, A., Estaban, D., and Galand, C., "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques," in *In Proc. Conf on Inf. Sciences and Systems*, pp. 191–195, May 1976.

[9] Daubechies, I., "Orthonormal bases of compactly supported wavelets," *Comm. on Pure and Applied Math.*, vol. 41, pp. 909–996, November 1988.

[10] Daubechies, I. and Sweldens, W., "Factoring wavelet transforms into lifting steps," tech. rep., 1996.

[11] Dudley, H., "The vocoder," *Bell Labs Rec.*, vol. 177, pp. 122–126, 1939.

[12] Eddins, S., *Subband Analysis-Synthesis and Edge Modeling Methods for Image Coding*. PhD thesis, Georgia Institute of Technology, 1990.

[13] Gckler, H. and Scheuermann, H., "A modular approach to a digital 60-channel transmultiplexer using directional filters.," *IEEE Transactions on Communications*, vol. 30, pp. 1598–1613, July 1982.

[14] Ikehara, M., Tran, T., and Nguyen, T., "Linear phase paraunitary filter banks with unequal-length filters," *Proc. IEEE Int. conf. on Image Processing*, vol. 2, pp. 223 – 226, Oct. 1997.

[15] Johnston, J., "A filter family designed for use in quadrature mirror filter banks," in *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 291–294, April 1977.

[16] Kiya, H., Yae, M., and Iwahashi, M., "A linear phase two-channel filter bank allowing perfect reconstruction," *IEEE ASSP Magazine*, pp. 951–954, May 1992.

[17] Koilpillai, R. and Vaidyanathan, P., "New results on cosine modulated FIR filter banks satisfying perfect reconstruction.," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1793–1796, 1991.

[18] Lettsome, C. A., Smith, M. J., and Mersereau, R., "Fixed analysis adaptive synthesis filter banks," *SPIE defense and security conference*, pp. Proc. of SPIE Vol. 6979 697902–3, March 2008.

[19] Malvar, H. S., "Modulated qmf filter banks with perfect reconstruction.," *Electronics Letters*, vol. 26, pp. 906 – 907, June 1990.

[20] Malvar, H. S., *Signal Processing with Lapped Transforms.* Artech House, 1991.

[21] Malvar, H. S. and Staelin, D., "The (lot) transform coding without blocking effects," *assp*, vol. ASSP-37, pp. 553–559, 1989.

[22] Meier, T., Ngan, K. N., and Crebbin, G., "Reduction of coding artifacts at low bit rates.," *SPIE Proc. Visual Communications and Image Processing*, pp. 241–251, January 1998.

[23] Mitra, S. K. and Sherwood, R. J., "Canonic realizations of digital fibers using the continued fraction expansion," *IEEE Trans. on Audio and Electroacoustics*, pp. 185–194, August 1972.

[24] Narasimha, M. and , A. P., "Design of a 24-channel transmultiplexer.," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, pp. 752 – 762, December 1979.

[25] Nayebi, K., Barnwell, T. P., and Smith, M. J. T., "Design and implementation of computationally efficient, modulated filter banks.," *IEEE International Sympoisum on Circuits and Systems*, vol. 34, pp. 650 – 653, June 1991.

[26] Nayebi, K., Smith, M. J. T., and Barnwell, T. P., "Analysis-synthesis systems based on time varying filter banks," in *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 617–620, March 1992.

[27] Nayebi, K., Barnwell, T. P., and Smith, M. J. T., "Low delay fir filter banks: Design and evaluation," *IEEE Trans. On Signal Processing*, vol. 42, pp. 24–32, Jan. 1994.

[28] Nayebi, K., Barnwell, T. P., and Smith, M. J., "Time-domain conditions for exact reconstruction in analysis/synthesis systems based on maximally decimated filter banks," *Proc. Southeastern Symp. syst. theory.*, pp. 489–503, March 1987.

[29] Nayebi, K., Barnwell, T. P., and Smith, M. J., "General time-domain analysis and design framework for exactly reconstructing fir analysis/synthesis filter banks," *Proc. Int. Symp. Circuits Syst.*, pp. 2022–2025, May 1990.

[30] Nayebi, K., Barnwell, T. P., and Smith, M. J., "Time-domain filter bank analysis: A new design theory," *IEEE Transacton on signal processing*, pp. 1412–1429, June 1992.

[31] Nayebi, K., Smith, M. J. T., and Barnwell, T., "The time domain analysis and design of exactly reconstructing fir analysis/synthesis filter banks," *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1735–1738, Mar. 1990.

[32] Portnoff, M. R., "Time-frequency representation of digital signals and systems based on short-time Fourier analysis.," *IEEE Trans. Acoust. Speech Signal Process*, vol. 28, pp. 55–69, February 1980.

[33] Princen, J. P. and Bradley, A. B., "Analysis/synthesis filter bank design based on time domain alias cancellation.," *IEEE Trans. Acoust. Speech Signal Process*, pp. 1153–1161, 1986.

[34] Ramstad, T. and Tanem, J., "Cosine-modulated analysis synthesis filter banks with critical sampling and perfect reconstruction.," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, p. 46.D4b.4, 1991.

[35] Rothweiler, J., "Polyphase quadrature mirror filters – a new sub-band coding technique.," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 285–288, 1983.

[36] Said, A. and Pearlman, W. A., "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, Jun. 1996.

[37] Shapiro, J., "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions Signal Processing*, pp. 3445 – 3462, December 1993.

[38] Smith, M. J. T. and Barnwell, T. P., "A procedure for designing exact reconstruction filter banks for tree structured subband coders," in *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1984.

[39] Smith, M. J. T. and Eddins, S. L., "Analysi/synthesis techniques for subband image coding," *IEEE ASSP Magazine*, pp. 1446–1456, August 1990.

[40] Smith, M. J. T., *Exact Reconstruction Analysis/synthesis Systems and Their Applications to Frequency Domain Coding.* PhD thesis, Georgia Institute of Technology, 1984.

[41] Smith, M. J. and Barnwell, T., "Exact reconstruction techniques for tree-structured subband coders," *IEEE ASSP Magazine*, pp. 434–441, June 1986.

[42] Sodagar, I., Nayebi, K., Barnwell, T. P., and Smith, M. J. T., "A new approach to time-varying fir filter banks," in *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pp. 1271–1275, Nov 1993.

[43] Sodagar, I., Nayebi, K., and Smith, M. J. T., "Time-varying analysis-synthesis systems based on filter banks and post filtering.," *IEEE Trans. On Signal Processing*, vol. 43, pp. 2512 – 2524, Nov. 1995.

[44] Vaidyanathan, P., "Theory and design of m-channel maximally decimated quadrature mirror filter banks with arbitrary m, having the perfect reconstruction property," *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 476–492, Apr. 1987.

[45] Vaidyanathan, P. and Hoang, P., "Lattice structures for optimal design and robust implementaion of two-channel perfect reconstruction qmf banks," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, pp. ASSP–36:97–112, January 1988.

[46] Vetterli, M., "Multi-dimensional sub-band coding: Some theory and algorithms.," *Signal Processing*, pp. 6(2):97–112, April 1984.

[47] Woods, J. W. and O'Neil, S. D., "Subband coding of images.," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1278–1288, October 1986.

VITA

Clyde Alphonso Lettsome was born on August 18, 1974 on the island of Saint Thomas in the United States Virgin Islands. He graduated from the Charlotte Amalie High School in June of 1992 in the top ten of his high school class. After graduating, he enrolled in the freshman class of the Florida Institute of Technology in the fall. He received his BSEE, graduated with honors in August 1996, and enrolled in graduate school at Florida Tech. in the fall of 1996. While obtaining his MS in Engineering Management and MSEE at Florida Tech., he worked at Sigma Electrical and Environmental Engineering, Computer Sciences Raytheon, and Rockwell Collins. Clyde received his MS in Engineering Management in 1998 and his MSEE in 1999.

Clyde then moved on to the Georgia Institute of Technology to work on his doctorate degree. During his early years at Georgia Tech., he worked as a graduate teaching assistant and graduate research assistant. The School of Electrical and Computer Engineering recognized him as an outstanding graduate teaching assistant in 2004. In addition, he earned his professional engineering license during this time. In his later years at Georgia Tech., Clyde co-founded Calabrix Corporation before defending his thesis and receiving a Doctorate of Philosophy degree in Electrical Engineering under the guidance of Dr. Mark J.T. Smith and Dr. Russell M. Mersereau on March 31, 2009.