Wavelet Transform Learning and Applications

By

Naushad Ansari

under the supervision of

Dr. Anubha Gupta

Indraprastha Institute of Information Technology-Delhi

December, 2017

Wavelet Transform Learning and Applications

By

Naushad Ansari

submitted

in partial fulfillment of the requirements

for the award of the degree of

Doctor of Philosophy

to the

Indraprastha Institute of Information Technology-Delhi
Okhla Industrial Estate, Phase III
New Delhi, India - 110020

December, 2017

Indraprastha Institute of Information Technology, Delhi
Okhla Industrial Estate, Phase III
New Delhi-110020, India

Certificate

This is to certify that the thesis titled **Wavelet Transform Learning and Applications** being submitted by **Mr. Naushad Ansari** to the Indraprastha Institute of Information Technology-Delhi, for the award of the degree of **Doctor of Philosophy**, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

December, 2017

Dr. Anubha Gupta
Associate Professor,
Deptt. of Electronics & Communication Engineering,
Indraprastha Institute of Information Technology-Delhi,
New Delhi-110020, India.

Indraprastha Institute of Information Technology, Delhi
Okhla Industrial Estate, Phase III
New Delhi-110020, India

Declaration

This is certified that the thesis entitled **Wavelet Transform Learning and Applications** being submitted by me to the **Indraprastha Institute of Information Technology-Delhi**, for the award of degree of **Doctor of Philosophy**, is a bonafide work carried out by me. This research work has been carried out under the supervision of **Dr. Anubha Gupta**.

The study pertaining to this thesis has not been submitted in part or in full, to any other University or Institution for the award of any other degree.

December, 2017

Naushad Ansari
PhD student,
Deptt. of Electronics & Communication Engineering,
Indraprastha Institute of Information Technology-Delhi,
New Delhi-110020, India.

Acknowledgement

There are a lot of people who helped me directly or indirectly in my PhD, to whom I wish to say, “thanks”. All of you have made me to reach at this point that I am able to write my PhD thesis.

First of all, I would like to extend my heartiest gratitude towards my supervisor, Dr. Anubha Gupta for her constant support throughout my PhD. She consistently gave me time for detailed technical discussions and for my paper reviews even after her busy schedule. I used to enjoy all the long hours technical discussions with her. I would like to thank her for her continuous technical/non-technical help, her feedback and encouragement all the times during my PhD. I would like to thank her for the semester parties, she used to throw after a long hectic semester. I would like to thank her for guidance and motivation, which enabled me to complete this thesis.

I heartily thank all my family members for their consistent support in every event of my life. I could not reach at this stage without their support.

I would like to thank my lab members, Dr. Hemant Kr. Aggarwal, Dr. Hemanta Kr. Mondal, Wazir Singh, Shiju S. for their help and support in some way or the other. I call them my PhD family members. Dr. Hemant Kr. Aggarwal has been a special part of my PhD time. I thank him for the continuous technical/non-technical discussions and his help at all the odd paths during my PhD. I could not imagine myself reaching at this stage without his support and suggestions. I would like to thank Dr. Hemanta Kr. Mondal for late night tea discussions and his suggestions. I would like to thank Dr. Naveen Gupta for being a regular tea partner and working as a Google for me. I would like to thank Dr. Anupriya Gogna and Ankita Shukla for technical discussions and their feedback at some points. I would like to thank Tanya Shreedhar, Parag Aggarwal, Vijay Sharma, and Kajal Kansal for joining me during my PhD and discussing non-technical but important topics.

I would like to thank all the members of IIIT-Delhi for providing such a great research environment. I would like to thank Priti, Anshu, Prosenjit, Amit and Gaurav for their friendly behavior and fast processing of every admin and finance related issues. I would like to thank Council of Scientific and Industrial Research (CSIR) for their financial support during my PhD.

Naushad Ansari

Abstract

Transform learning (TL) is currently an active research area. It has been explored in several applications including image/video denoising, compressed sensing (CS) of magnetic resonance images (MRI), etc. and is observed to perform better than the existing transforms. However, TL involves non-convex optimization problem with no closed form solution and hence, is solved using greedy algorithms. A large number of variables (transform basis as well as transform coefficients) along with the greedy-based solution makes TL computationally expensive. Also, TL requires a large amount of training data for learning. Hence, it may run with challenges in applications where only single snapshots of short-duration signals such as speech, music or electrocardiogram (ECG) signal are available. Thus, one uses existing transforms that are signal independent. This motivates us to look for a strategy to learn transform in such applications.

Among existing transforms, discrete wavelet transform provides an efficient representation for a variety of multi-dimensional signals. Owing to this, wavelets have been applied successfully in many applications. In addition, wavelet analysis provides an option to choose among existing basis or to learn new basis. This motivates us to learn wavelet transform from a given signal of interest that may perform better than the fixed transforms in an application. The learned wavelet transform is, hereby, called signal-matched wavelet transform. Since the translates of the wavelet filters associated with discrete wavelet transform form the basis in l^2-space, wavelet transform learning implies learning wavelet filter coefficients. This reduces the number of parameters required to be learned with wavelet learning compared to the traditional transform learning. Also, the requirement of learning fewer coefficients allows one to learn basis from a short single snapshot of signal or from the small training data. We also show that closed form solution exists for learning the wavelet transform unlike traditional transform learning.

Although the problem of signal-matched wavelet design/learning has been explored in the literature, there are a number of limitations. Firstly, existing methods require full original signal to learn wavelet transforms and hence, these methods can not be used in inverse problems, where one has access to only the degraded signal and not to the original signal. Secondly, signal-matched wavelet transform learning is not explored for rational wavelets, although rational wavelets are observed to be more effective than dyadic wavelets in audio and speech signal processing. Thus, we note that there is a need for methods to learn signal-matched wavelets that are modular, have compactly supported filters for dyadic or rational wavelet systems, are easily implementable in DSP hardware, and can also be learned from degraded signals. This thesis is motivated to address these limitations and proposes a number of methods along with their utility in applications.

Specifically, we propose methods to learn dyadic as well as rational wavelet transform using the lifting framework. The proposed method inherits all the advantages of lifting, i.e., the learned wavelet transform is always invertible, method is modular, learned transform has compactly supported filters and hence, is DSP hardware friendly, and the corresponding wavelet system can also incorporate non-linear filters, if required. We show that closed form solution exists for learning the wavelet transform with the proposed method. Also, wavelet transform can be learned using the proposed method even when a small amount of data is present. Since the wavelet transform is being learned from the signal itself, one may use the learned wavelet transform in applications instead of struggling to choose from the existing wavelet bases.

For dyadic wavelet transform learning (DWTL), we propose three methods in different scenarios. Particularly, we propose methods to learn dyadic wavelet transform (DWT) from 1) original signal, 2) degraded signal in inverse problems, and 3) a class of signals. We use the learned DWT as the sparsifying transform in the application of 1) Gaussian denoising of speech and music signals, 2) CS based reconstruction of speech, music, and ECG signals, 3) impulse denoising of images, and 4) CS based reconstruction of images. Extensive simulations have been carried out that demonstrate that the learned transforms outperform the standard dyadic wavelet transforms.

We also extend the existing theory of lifting framework from dyadic to rational wavelets and use the extended lifting theory to learn critically sampled signal-matched rational wavelet transform (RWT) with generic decimation ratios from a given signal of interest. We introduce the concept of rate converters in predict and update stages to handle variable subband sample rates. So far, signal-matched rational wavelet learning have remained limited in use because design methods are in general cumbersome. Since our proposed methodology exploits lifting framework, we provide modular, compactly supported, DSP hardware friendly rational wavelet transform learning (RWTL) methods. This may enhance the use of RWT in applications which is so far restricted. We use the learned RWT as the sparsifying transform in CS based reconstruction of 1-D and 2-D signals. The learned RWT is observed to perform better than the existing dyadic as well as rational wavelet transforms.

Apart from the wavelet transform learning methods, we propose a new multi-level wavelet decomposition strategy for images, named as L-Pyramid wavelet decomposition. L-Pyramid wavelet decomposition is observed to perform better in CS based image reconstruction. In addition, we also propose weighted non-convex minimization for CS based recovery. Detailed experiments are provided using the weighted non-convex minimization and the learned wavelet transform for CS based ECG signal recovery with various sensing matrices. The learned wavelet transform along with the proposed weighted non-convex minimization method is observed to provide much better ECG signal reconstruction as compared to existing wavelet transforms as well as existing methods.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BDND	Boundary discriminative noise detection
BP	Basis Pursuit
BPDN	Basis Pursuit Denoising
CI	Compressive Imaging
CS	Compressive/Compressed Sensing
DCT	Discrete Cosine Transform
DL	Dictionary Learning
DSP	Digital Signal Processors
DWT	Dyadic Wavelet Transform
DWTL	Dyadic Wavelet Transform Learning
ECG	Electrocardiogram
FIR	Finite Impulse Response
G	Good
IHT	Iterative Hard Thresholding
IIR	Infinite Impulse Response
IRLS	Iterative Re-weighted Least Squares

ISMF	Ideal Switching Median Filtering
LP	Linear Phase
MRA	Multiresolution Analysis
MRI	Magnetic Resonance Images
NMSE	Normalized Mean Square Error
OMP	Orthogonal Matching Pursuit
PCI	Partial Canonical Identity
PR	Perfect Reconstruction
PRD	Percentage Root Mean Square Difference
PSNR	Peak Signal-to-Noise Ratio
RFB	Rational filterbank
RWT	Rational Wavelet Transform
RWTL	Rational Wavelet Transform Learning
SBM	Sparse Binary Matrix
SMF	Simple Median Filtering
SNR	Signal-to-Noise Ratio
TL	Transform Learning
VG	Very Good
WBAN	Wireless Body Area Network
WTL	Wavelet Transform Learning
WNC	Weighted Non-Convex

Chapter 1

Introduction

1.1 Wavelet Transform Learning

Transform learning (TL) is currently an active research area and is being explored in several applications including image/video denoising, compressed sensing of magnetic resonance images (MRI), etc. [2, 3, 4]. Transform learning has the advantage that it adapts to signals of interest and is often observed to perform better than the existing sparsifying transforms such as total variation (TV), discrete cosine transform (DCT), and discrete wavelet transform in the above-said applications [2, 3, 4].

In general, transform learning is posed as an optimization problem satisfying some constraints that are specific to applications. Transform domain sparsity of signals is a widely used constraint along with some additional constraints on the transform to be learned, say, the minimization of Frobenius norm or the log-determinant of transform [2]. The requirement of joint learning of both the transform basis and the transform domain signal under the constraints renders the optimization problem to be non-convex with no closed form solution. Thus, in general, TL problems are solved using greedy algorithms [2]. A large number of variables (transform basis as well as transform coefficients) along with the greedy algorithm based solutions makes TL computationally expensive.

In general, TL requires a large amount of training data for learning. Hence, TL may run with challenges in applications where only single snapshots of short-duration signals such as speech, music or electrocardiogram (ECG) signal are available because a large amount of training data required for learning transform is absent. Hence, one uses existing transforms that are signal independent. This motivates us to look for a strategy to learn transform in such applications.

Among existing transforms, although Fourier transform and DCT find use in many applications, discrete wavelet transform provides an efficient representation for a variety of multi-dimensional signals [5]. This efficient signal representation stems from the fact that discrete wavelet transform tends to capture signal information into a few significant coefficients. Owing to this advantage, wavelets have been applied successfully in many applications including compression, denoising, biomedical signal and image processing, texture analysis, pattern recognition, etc. [6, 7, 8, 9, 10].

In addition, wavelet analysis provides an option to choose among existing basis or to design new basis, thus motivates one to learn basis from a given signal of interest that may perform better than the fixed basis. Since the translates of the associated wavelet filters form the basis in l^2-space (functional space for square summable discrete-time sequences), wavelet transform learning implies learning wavelet filter coefficients. This reduces the number of parameters required to be learned with wavelet learning compared to the traditional transform learning. Also, the requirement of learning fewer coefficients allows one to learn basis from a short single snapshot of signal or from the small training data. We propose wavelet transform learning with closed form solutions leading to fast implementation without the need to look for greedy solutions.

Various attempts have been made in the past to design wavelets. One of the early attempts was to design smooth orthogonal and compactly supported wavelet [11] with multiresolution analysis (MRA) and to design fast wavelet transform [12, 13]. Later, the concept of wavelet was extended to the design of biorthogonal and semi-orthogonal wavelets. Some of the examples include biorthogonal compactly supported wavelets [14, 15] and construction of semi-orthogonal spline wavelets [16, 17, 18, 19]. Biorthogonal wavelets allow the use of linear phase filters for constructing symmetric wavelets.

In the above works, fixed wavelet bases are designed to represent any signal. While one set of wavelet basis may be very effective in a particular signal processing application on one signal, the same set of wavelet basis may not work that good for another signal or in another application. This motivated researchers to look into multiple basis for best signal representation. For example, wavelet design was proposed in [20] by projecting the signal onto multiple existing basis. In [21], an algorithm is proposed to decompose any given signal into a linear expansion of waveforms that are selected from a redundant dictionary of functions. However, the families of functions over which the signal is expanded are obtained by the translation and dilation of some fixed known functions. In [22], a method is proposed to search for best basis in the families of orthonormal basis constructed with wavelet packets and the reconstruction error of the signal is minimized for selecting the optimal basis.

In all the above works, either one or multiple existing basis are used to represent any signal. However, all these basis are independent of the signal of interest. Wavelet analysis provides an option to choose from existing basis or to design new basis, which motivated researchers to design/learn wavelet basis from a given signal of interest and/or in a particular application. Wavelet learned from a given signal of interest is called as signal-matched wavelet.

Various researchers have attempted to design signal-matched wavelets. For example, in [23], matched wavelet is designed in the time domain. The best approximation of the signal is found upto any desired scale with integer translates of a valid scaling function of fixed support. The upper bound of error norm between the actual signal and its approximation is minimized. As the minimization was complex in the time domain, in [24], minimization is carried out in the frequency domain assuming the signal to be band-limited. The l_p norm of the error in the frequency domain is minimized via a closed form solution in the frequency domain. However, it led to a very complex solution that was difficult to solve. In [25], matched wavelet is designed in the frequency domain. However, the method was computationally expensive. In [26] and [27], signal-matched adaptive filterbank is designed with constrained minimization problems in terms of coding gain criterion that led to complicated solutions. Some methods were proposed in the literature to design signal-matched wavelet [28, 29, 30, 31], where only analysis highpass filter of the wavelet system is designed, while the remaining filters of the corresponding wavelet filterbank were designed using the condition of perfect reconstruction. Hence, the wavelet design is not optimal or fully matched to the given signal.

It is observed that in the above signal-matched wavelet design methods, either the wavelet is not fully signal-matched or the method involves computationally expensive algorithm. Lifting [32, 33] is a powerful tool for either constructing new customized wavelets from existing wavelets or factoring existing wavelet filters into a finite sequence of smaller filtering steps. Lifting provides several advantages for custom design/learning of wavelet such as a) wavelets can be designed in the spatial domain, b) designed wavelet transform is always invertible, c) the design is modular, and d) the design is DSP (digital signal processors) hardware friendly from the implementation viewpoint [32]. With these advantages, several researchers have explored lifting to design signal-matched wavelets [34, 35, 36, 37, 38, 39, 40, 41, 42, 43].

The lifting technique involves alternate predict and update steps. Although it is easy to find the predict filters, finding an update filter offers a real challenge. One of the criteria used in the literature to find the update filters is the minimization of reconstruction error of even and odd indexed samples [34]. In [35, 36], the update first structure with the adaptation of the update step is used. The update filter is changed based on the local gradient information such that sharp variations in the signal get less smoothened than the more homogenous regions. Similar update method is used in [37]. In [38], a nonseparable lifting is used on images with regularity conditions imposed. In [39], directional interpolation is used with coefficients of the interpolation filter to optimize to adapt to statistical property of the image. In [40], wavelets are designed by minimizing the difference between BWT (block wavelet transform) and KLT (Karhunen-Loève transform) of the signal. In [41], orthogonal IIR (infinite impulse response) filterbank is designed using allpass filter in the lifting steps. In [42], the geometry of a given image is used to design new wavelet via lifting leading to local and anisotropic filters. In [43], nonseparable filterbank was designed using pixel-wise adaptation according to local image features.

In this work, we propose to learn signal-matched wavelets using lifting wherein both predict and update stage polynomials are obtained from a given signal. We design predict stage by minimizing the energy in the wavelet subspace domain and update stage by minimizing the difference between signal reconstructed from the coarser subband signal and the original signal. As stated earlier, wavelet transform learning requires fewer coefficients to be learned as compared to the traditional transform learning. This motivates us to explore wavelet transform learning for small data in this work. We also provide closed-form solutions for learning the wavelet transform with the proposed method.

1.2 Wavelet Transform Learning in Inverse Problems

Although there exist a number of methods in the literature to design/learn wavelet as described in the previous section, all these methods require original signal and hence, can be used in only those set of applications where original signal is available. Compression and classification are examples of such applications. On the other hand, these methods cannot be applied in inverse problems, where one does not have access to the original signal. Denoising, compressed sensing (CS), deblurring, inpainting, etc. are some examples of inverse problems. In these problems, degraded signal is observed and the original signal is required to be estimated from the observed degraded signal.

Signal reconstruction in inverse problems is often posed as sparse recovery problem, where the signal is known to be sparse in some transform domain. In this context, the discrete wavelet transform is extensively used as the sparsifying transform for signal reconstruction because of its ability to efficiently represent a signal. Researchers often face challenge in choosing a wavelet because it is not known apriori as to which wavelet will provide the better representation of the signal as compared to others. In general, one uses compactly supported wavelets, either orthogonal Daubechies wavelets or biorthogonal 9/7 or 5/3 wavelets (note that first digit denotes the length of analysis lowpass filter and second digit denotes the length of analysis highpass filter).

As stated earlier, learned wavelet transform may provide better representation of a given signal as compared to existing wavelets and hence, may provide better signal reconstruction in inverse problems compared to any existing wavelet chosen arbitrarily. To the best of our knowledge, there is no method in the literature that addresses the problem of wavelet transform learning in inverse problems. We explore this problem in our work and present a method to learn wavelet transform in inverse problems. The proposed method consists of two stages. In the first stage, a coarse estimate of the signal is obtained using the existing wavelet transform and in the second stage, this coarse estimate of the signal is used to learn signal-matched wavelet. The learned wavelet transform is used as the sparsifying transform for sparse signal recovery of the signal.

1.3 Wavelet Transform Learning for a Class of Signals and CS based ECG Signal Recovery

As stated in the previous section, existing methods require the original signal for learning signal-matched wavelet. However, the full original signal is not available in applications like inverse problems and hence, we propose methods to learn wavelet transform for signals in inverse problems. The proposed method consists of two stages, wherein the coarse estimate of the signal is learned using one of the existing wavelet transform in the first stage. We show that this step of coarse estimation of the signal can be skipped if the wavelet transform is learned for a class of signals with low variability across the signals.

There exist many signal classes that have low variability across their signals, e.g., ECG signals and brain MRI images. For such examples, one can learn wavelet transform matched to the class using an ensemble of signals. Since the wavelet transform is learned for a particular class, it may provide better representation and hence, may perform better in applications for signals of these classes compared to existing wavelets. Also, there is no need to learn wavelets individually for different signals of these classes and in different applications. Further, there is no need of coarse signal estimation (in inverse problems) unlike the method described in the previous section.

ECG signals are very vital because of their ability to diagnose cardiac diseases that is one of the major causes of deaths as per the report of World Health Organization [44]. An estimated 17.5 million people died from cardiovascular diseases in 2012, representing 31% of all global deaths. Given the importance of ECG signal analysis from the health perspective, we focus only on ECG signals in this thesis, although the proposed methods can be explored for signals of other classes, say brain MRI, in different applications.

ECG signals are recorded by wireless body area network (WBAN) in the application of healthcare for continuous monitoring of patients [45]. There are three stages of operations in WBAN: sensing, communication, and processing. Out of these three stages, communication consumes the highest amount of energy [45]. Also, these devices operate by battery. Therefore, to reduce the energy consumption in transmission and hence, to increase battery life, some form of compression is needed. CS [46] based method has been applied in ECG compression in [47] and is shown to be a better alternative in terms of processing complexity of the encoder as compared to the existing methods.

Many researchers have contributed in CS-based ECG reconstruction by constructing measurement matrices [48, 47, 49], by hardware design [50, 51, 52], and by algorithm design for better reconstruction [49, 53, 54]. We contribute in this work toward good quality ECG signal reconstruction in compressive transmission or energy efficient transmission of ECG signals.

Many non-convex regularization/minimization based methods are proposed in the literature [55, 56, 57, 58, 59] for sparse signal recovery. Non-convex regularization strongly promotes sparsity and hence, is used to improve upon the existing l_1 regularization based methods. Among non-convex methods, l_p regularization based methods are most widely used in the literature [58, 59, 60, 61] and are shown to perform best among other methods for sparse signal recovery [57]. To the best of our knowledge, l_p regularization based methods have not been used in CS-based ECG reconstruction. In this work, we propose to use l_p regularization for CS based reconstruction of ECG signal.

Prior information about the signal is generally used to improve sparse signal reconstruction quality [62, 63, 64, 65, 49]. Such methods can be broadly classified into three categories. Methods in the first category use the prior information about the support of the signal [62, 63], while the second category of methods iteratively estimate the support from the signal [64, 65]. The first category of methods cannot be used in applications where one does not know the non-zero components of the signal, while the second category has the challenge of being an iterative method and hence, may take longer time providing trade-offs between the reconstruction quality and the time taken in the reconstruction. These two catergories do not impose sparsity on the known support (non-zero component) of the signal. The third category [49] imposes weighted sparsity on the signal, weighing different components of the signal with different weights unlike l_1 minimization that treats all components of signal equally. These methods use the information about which components of the signal are sparse instead of knowing the exact value at that position. In [49], different weights were imposed on different levels of wavelet detail coefficients. We propose a similar but simpler approach.

Because of the sparsifying ability of wavelet transform, it has been used extensively in CS based reconstruction of ECG signals [66, 67, 47, 48, 49]. However, as stated earlier, existing wavelet basis have a disadvantage of being signal independent and wavelet transform learning has not been explored in CS-based ECG reconstruction so far. Therefore, we propose to learn wavelet for ECG signals in this work and use it as the sparsifying transform for CS-based recovery of ECG signals. Since shape or characteristics of ECG signals do not vary much over time and over different signals, we learn a single wavelet for the class of ECG signals and name the learned wavelet as ECGlet.

In particular, for CS-based ECG signal recovery, we propose weighted nonconvex l_p (WNC) minimization with ECG matched learned wavelet (ECGlet).

1.4 Rational Wavelet Transform Learning using Lifting Framework

Most of the applications of discrete wavelet transform rely on dyadic wavelet transform, where discrete wavelet transform coefficients are obtained by passing an input signal through a 2-channel filterbank consisting of highpass and lowpass filters, followed by downsampling by two. In the frequency domain, this process is equivalent to the decomposition of the signal into two uniform frequency bands. As we move to M-band wavelet system/filterbank, a signal is decomposed into M uniform frequency bands. However, some applications, such as speech and audio signal processing, require non-uniform frequency band decomposition [68, 69, 70]. In such applications, rational wavelet transform (RWT) can prove to be very helpful because RWT provides non-uniform frequency band partition of the signal spectrum. Accordingly, decimation factors of such a rational wavelet system are different in each branch/band and is a non-integer, i.e., the decimation factor is a rational number [71].

RWT has also been explored in applications apart from audio and speech signal processing. For example, RWT is applied in context-independent phonetic classification in [69]. In [72], it is used for synthesizing 10m multispectral image by merging 10m SPOT panchromatic image and a 30m Landsat Thematic Mapper multispectral image. Wavelet shrinkage based denoising is presented in [73] using signal independent rational filterbank designed by [74]. RWT is used in extracting features from images in [75] and in detection of click frauds in [76]. The rational orthogonal wavelet transform is used to design optimum receiver based broadband Doppler compensation structure in [77] and for broadband passive sonar detection in [78].

A number of RWT designs have been proposed in the literature. For example, in [79], a frequency domain approach is proposed for designing IIR (Infinite Impulse Response) rational filterbank (or infinitely supported rational wavelets), while in [80, 81], the method is proposed for designing overcomplete rationally decimated FIR (Finite Impulse Response) filterbank. In [82], a design approach for complex orthogonal rational IIR filterbank is presented. In [83], a method is proposed to design perfect reconstruction FIR rational filterbank with regularity properties. The design involves solving non-convex optimization problem with non-linear constraints. Some other designs include FIR orthonormal rational filterbank design [68], biorthogonal FIR rational filterbank design [83], frequency response masking technique based design of rational FIR filterbank [70] etc. However, so far RWT designed and used in applications are meant to meet certain fixed requirements in the frequency domain or time-domain instead of learning the transform from a given signal of interest. For example, all the above designs are signal independent and hence, the concept of transform learning from a given signal has not been used so far in learning rational wavelets.

As stated earlier, lifting provides several advantages for custom wavelet design/learning. Even with the several advantages, lifting framework is used so far only for the custom design/learning of dyadic (or M-band) wavelets [34, 42, 40, 84] and has not been used to learn the rational wavelet transform to the best of our knowledge. Moreover, the existing architecture of lifting framework cannot be extended directly to rational wavelet because of different sample/signal rates in two subbands. However, similar to dyadic wavelet transform, the lifting framework can help in learning rational wavelets from given signals in a simple modular fashion that will also be easy to implement on hardware. Also, this may lead to the enhanced use of rational wavelet transforms in applications, similar to dyadic wavelets, which is so far restricted.

Motivated by the success of transform learning in applications, the flexibility of rational wavelet transform with respect to non-uniform signal spectral splitting, and the advantages of lifting in learning custom design wavelets, we propose to learn rational wavelet transform from a given signal using the lifting framework in this work. We extend the theory of lifting framework from dyadic to rational wavelets, where dyadic wavelet is a special case. To overcome the problem of different signal/sample rate in different bands, we introduce the concept of rate converters in predict and update stages. We further use the extended lifting theory to learn signal-matched rational wavelet transform from a given signal of interest, where rational wavelets with any decimation ratio can be designed. The learned analysis and synthesis filters are FIR in nature that can be easily implemented in hardware. Similar to dyadic wavelet transform, closed form solution exists for learning rational wavelet and thus, no greedy solution is required making the proposed method computationally efficient. The learned rational wavelet transform is explored as the sparsifying transform in CS based reconstruction of signals.

1.5 Research Contributions

The primary aim of this thesis is to provide methods to learn dyadic as well as rational wavelet transform for a given class of signals or from a given signal, present in the original form or the degraded form. The learned wavelet transform can be used as the sparsifying transform in CS-based applications. Below are the salient contributions of this thesis:

1. We present methods to learn dyadic wavelet transform using the lifting framework from a given signal. FIR analysis and synthesis filters are learned that

can be easily implemented in hardware. Unlike existing learning algorithms, the proposed method has a closed form solution and thus, no greedy solution is required making the proposed method computationally efficient. In addition, the proposed wavelet transform can be learned from a short snapshot of a single signal.

2. The above proposed method requires original signal for learning dyadic wavelet transform. However, in many applications of say inverse problems, only the degraded signal is available instead of the original signal. Hence, we extend the proposed method to learn dyadic wavelet transform from degraded signals in inverse problems. First, we obtain a coarse estimate of the signal using an existing wavelet transform and then use this coarse estimate of the signal to learn wavelet transform. The learned wavelet transform can be used as the sparsifying transform for sparse recovery of the signal. The proposed method is applied in Gaussian denoising and CS-based reconstruction of 1-D signals, and in impulse denoising and CS based reconstruction of images. We also extend the method to learn dyadic wavelet transform for a class of signals. We apply this concept to the class of ECG signals and used the learned transform as the sparsifying transform along with the proposed weighted non-convex minimization for CS-based ECG recovery.

3. The proposed method is extended to learn separable dyadic wavelet transform for images. We present methods to learn separable dyadic wavelet transform from original as well as from degraded images in inverse problems. Learned separable dyadic wavelet transform is used as the sparsifying transform for the CS based reconstruction of images.

4. The theory of lifting is extended from dyadic wavelets to rational wavelets, where dyadic wavelet is a special case. The concept of rate converters is introduced in predict and update stages to handle variable subband sample rates. Extended lifting theory is used to learn rational wavelet transform from a given signal, where rational wavelets with any decimation ratio can be learned. Learned rational filters are FIR that can be easily implemented in hardware. Similar to dyadic wavelet transform, rational wavelet transform learning has a closed form solution and transform can be learned from a short

duration of a signal. The proposed method is extended to images by learning separable rational wavelet transform from the images. The proposed method is applied in the application of CS based reconstruction of 1-D signals and images.

5. For the separable 2D wavelet transform, a new multi-level wavelet decomposition strategy is proposed, named multi-level L-Pyramid wavelet decomposition and its performance is compared with the existing wavelet decomposition in CS based image reconstruction. Also, weighted non-convex minimization is proposed for sparse signal recovery of signals and applied in the CS-based reconstruction of ECG signals.

The work presented in this thesis resulted in some journals and conferences publications. These are presented at the end of the thesis.

1.6 Thesis Organization

In Chapter-2, we present some of the concepts, which are the basic building blocks of our work. Particularly, we present existing lifting framework in dyadic wavelet, rational wavelet and their equivalent M-band structure, polyphase representation and perfect reconstruction condition, the theory of compressed sensing, and denoising concept. We also present evaluation metrics to be used throughout in our work.

In Chapter-3, we present methods to learn dyadic wavelet transform for 1-D signals in various scenarios. Particularly, we present methods to learn dyadic wavelet transform from original signals, from degraded signals in inverse problems, and from a class of signals. The proposed method of dyadic wavelet transform learning is applied in the application of denoising and compressive sensing of 1-D signals. We also present weighted non-convex minimization along with an algorithm to solve it. We apply weighted non-convex minimization in CS-based reconstruction of ECG signals.

In Chapter-4, we present methods to learn separable dyadic wavelet transform for images in two scenarios. One, when the original image is available and another in inverse problem applications, where one has the access to only the degraded image and not the original image. Use of the proposed methods is explored in the applications of compressed sensing and impulse denoising of natural images, where the learned separable dyadic wavelet transform are used as the sparsifying transforms. We also present a new wavelet decomposition method, called as L-Pyramid wavelet decomposition. The performance of L-Pyramid wavelet decomposition is compared with existing R-Pyramid wavelet decomposition in CS based image reconstruction.

In Chapter-5, we present rational wavelet transform learning using the lifting framework. The existing lifting framework is extended from dyadic to rational wavelet case and used to learn signal-matched rational wavelet transform. Variable sample rates in the two branches of rational wavelet decomposition pose difficulty in applying the existing lifting framework. Hence, we introduce the concept of rate-converters to equal the rates of samples in the two branches. Thus, we extend the lifting framework from dyadic wavelet design to rational wavelet design. The theory is developed and presented to learn generic downsampling ratios, say $(N/M, M - N/M)$, where M and N are any integers with $N < M$. The learned rational wavelet transform is used as the sparsifying transform in the application of CS-based reconstruction of 1-D signals. The method of learning rational wavelet transform is extended from 1-D signal to images and is used in the application of compressed sensing of images.

In Chapter-6, we present conclusions of all the work done and present some directions for future work.

Chapter 2

Brief Background

In this chapter, we briefly present some of the important concepts, which are the basic building blocks for this work. Particularly, we present the theory of lifting framework for dyadic wavelet transform, rational wavelet and its equivalence to M-band wavelet, polyphase representation and prefect reconstruction condition for M-band wavelet, compressive sensing, and denoising. We also present evaluation metrics and notations to be used throughout.

2.1 Lifting theory in dyadic wavelet

We present the theory of lifting in this subsection, which is so far restricted to dyadic wavelet case.

Lifting is a technique for either factoring existing wavelet filters into a finite sequence of smaller filtering steps or constructing new customized wavelets from existing wavelets [33]. This design is modular, guarantees perfect reconstruction at every stage, and supports non-linear filters. A general lifting scheme consists of three steps: Split, Predict, and Update (Refer to Figure 2.1). These steps are described below.

Split: In the split step, input signal is split into two disjoint sets of samples, generally, even and odd indexed samples, labeled as $x_e[n]$ and $x_o[n]$, respectively. The original signal can be recovered perfectly by interlacing or combining the two sample streams. The corresponding wavelet system is called as the *Lazy Wavelet* system [32] and is similar to the structure shown in Figure 2.2 with analysis filters

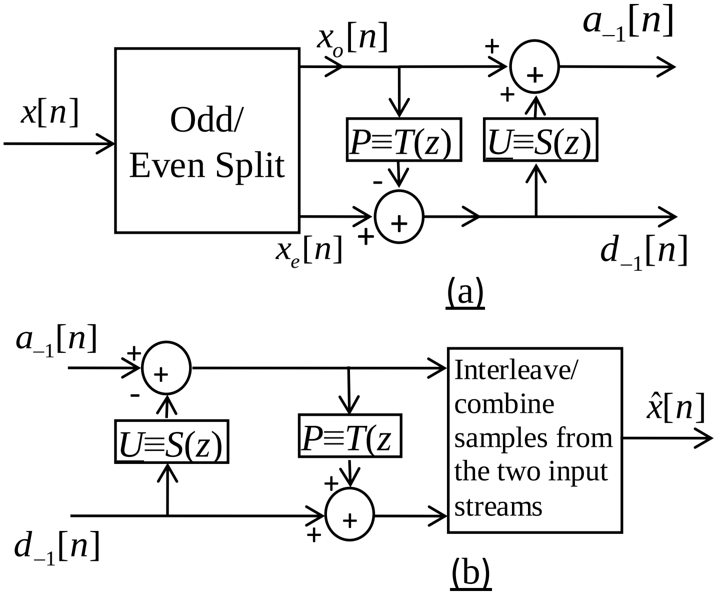


Figure 2.1: Steps of lifting: Split, Predict and Update

labeled as $H_0(z) = Z\{h_0[n]\}$, $H_1(z) = Z\{h_1[n]\}$ and synthesis filters as $F_0(z) = Z\{f_0[n]\}$, $F_1(z) = Z\{f_1[n]\}$.

Predict Step: In the predict step, one of the two disjoint set of samples is predicted from the other set of samples. For example, in Figure 2.1(a), we predict the odd set of samples from the neighboring even samples by using the predictor $P \equiv T(z)$. Predict step is equivalent to applying a highpass filter on the input signal. Predict step modifies the highpass filter of the analysis end and lowpass filter of the synthesis end, without altering other filters, according to the following relations:

$$H_1^{new}(z) = H_1(z) - H_0(z)T(z^2), \tag{2.1}$$

$$F_0^{new}(z) = F_0(z) + F_1(z)T(z^2). \tag{2.2}$$

Update Step: In the update step, predicting samples (even samples in our case) of the predict step are updated with the predicted samples to provide the approximate coefficients of the signal. The signal is updated with $U \equiv S(z)$ (refer to Figure 2.1). This step modifies the analysis lowpass filter and synthesis highpass

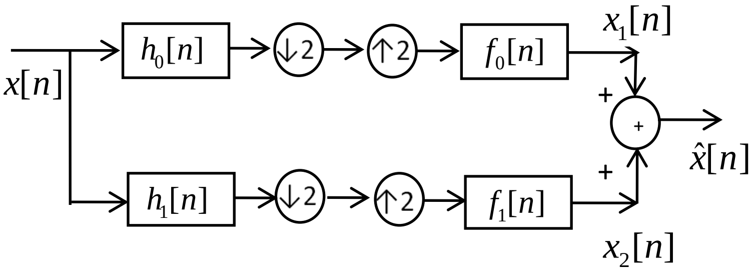


Figure 2.2: Two channel dyadic wavelet system equivalent of system shown in Figure 2.1

filter according to the following relation:

$$H_0^{new}(z) = H_0(z) + H_1(z)S(z^2), \tag{2.3}$$

$$F_1^{new}(z) = F_1(z) - F_0(z)S(z^2). \tag{2.4}$$

Lifting architecture shown in Figure 2.1 can be equivalently converted to dyadic wavelet system shown in Figure 2.2 or similarly, any existing dyadic wavelet system shown in Figure 2.2 can be equivalently broken into lifting steps as shown in Figure 2.1. As each step (predict or update) in lifting is invertible, perfect reconstruction (PR) is guaranteed after every step.

2.2 Rational Wavelet and Equivalent M-band structure

M-band wavelet system has integral downsampling ratio, M as shown in Figure 2.3, whereas rational wavelet system has rational down-sampling ratios that allow decomposition of input signals into non-uniform frequency bands. In general, any i^{th} analysis branch of a rational structure is as shown in Figure-2.4, where $G_i(z)$ denotes the analysis filter, q_i denotes the upsampling factor, and p_i denotes the downsampling factor. For example, if $p_i = 3$ and $q_i = 2$, the downsampling ratio in this branch is equal to 3/2. At the synthesis end, the order of downsampler and upsampler are reversed.

An M-channel rational filterbank is said to be critically sampled if the following

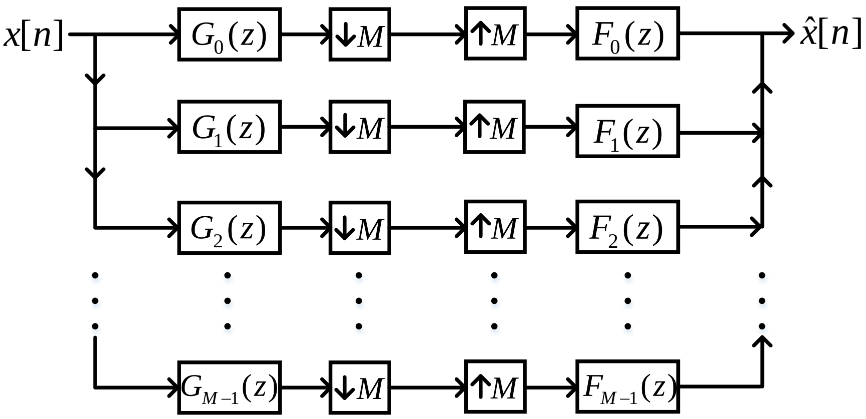


Figure 2.3: M-band wavelet structure

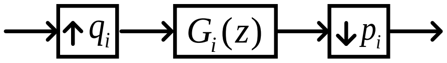


Figure 2.4: i^{th} branch of rationally decimated analysis filterbank

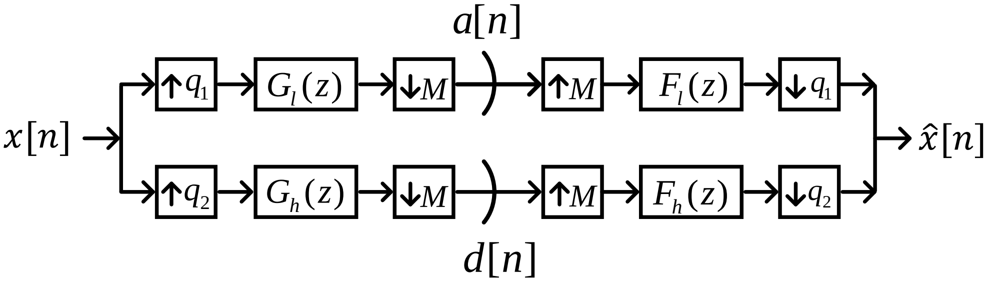


Figure 2.5: General 2-band rational wavelet structure

relation is satisfied

$$\sum_{i=0}^{M-1} \frac{q_i}{p_i} = 1. \tag{2.5}$$

This is to note that, throughout this work, we consider all p_i's to be equal, especially, $p_i = M$ for all i. Also, q_1 and q_2 are relatively prime with each other and with M and $q_1 + q_2 = M$ for a critically sampled rational wavelet system.

In general, a given M-band wavelet system, as shown in Figure 2.3, can be converted into an equivalent 2-band rational wavelet structure of Figure 2.5, having downsampling ratios $\frac{M}{q_1}$ and $\frac{M}{q_2}$ in the two branches. For example, the analysis filters $G_j(z)$ and synthesis filters $F_j(z)$ for $j = 0, 1, 2, ..., q_1 - 1$ as shown in Figure

2.3 can be combined using the following equations:

$$G_l(z) = \sum_{i=0}^{q_1-1} z^{-iM} G_i(z^{q_1}), \tag{2.6}$$

$$F_l(z) = \sum_{i=0}^{q_1-1} z^{iM} F_i(z^{q_1}), \tag{2.7}$$

where $G_l(z)$ and $F_l(z)$ are the corresponding analysis and synthesis lowpass filters of the equivalent 2-band rational wavelet structure of Figure-2.5. Similarly, rest of the filters of both sides can be combined using the following equations:

$$G_h(z) = \sum_{i=0}^{q_2-1} z^{-iM} G_{i+q_1}(z^{q_2}), \tag{2.8}$$

$$F_h(z) = \sum_{i=0}^{q_2-1} z^{iM} F_{i+q_1}(z^{q_2}), \tag{2.9}$$

where $G_h(z)$ and $F_h(z)$ are the corresponding higpass filters of analysis and synthesis ends of Figure-2.5.

2.2.1 Polyphase Representation and Perfect Reconstruction

The polyphase representation of filters is very helpful in filterbank analysis and design [85]. Consider the M-band critically sampled filterbank shown in Figure 2.3. Analysis filter $G_i(z)$ can be written using type-1 polyphase representation as:

$$G_i(z) = \sum_{j=0}^{M-1} z^{j} E_{i,j}(z^M), \tag{2.10}$$

where $E_{i,j}(z) = g_i(j) + g_i(M+j)z^M + g_i(2M+j)z^{2M} + ...$ and $G_i(z) = Z\{g_i[n]\}$.

Synthesis filter $F_i(z)$ can be written using type-2 polyphase representation as:

$$F_i(z) = \sum_{i=0}^{M-1} z^{-j} R_{i,j}(z^M), \tag{2.11}$$

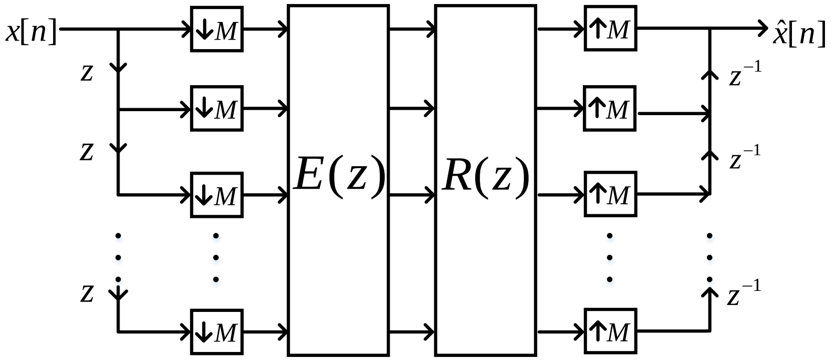


Figure 2.6: M-band wavelet structure with PR

where $R_{i,j}(z) = f_i(j) + f_i(M+j)z^{-M} + f_i(2M+j)z^{-2M} + ...$ and $F_i(z) = Z\{f_i[n]\}$. The M-band filterbank of Figure-2.3 can be equivalently drawn using polyphase matrices as shown in Figure-2.6, where

$$E(z) = \begin{pmatrix} E_{0,0} & E_{0,1} & \cdots & E_{0,M-1} \\ E_{1,0} & E_{1,1} & \cdots & E_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0} & E_{M-1,1} & \cdots & E_{M-1,M-1} \end{pmatrix}, \tag{2.12}$$

and

$$R(z) = \begin{pmatrix} R_{0,0} & R_{0,1} & \cdots & R_{0,M-1} \\ R_{1,0} & R_{1,1} & \cdots & R_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ R_{M-1,0} & R_{M-1,1} & \cdots & R_{M-1,M-1} \end{pmatrix}. \tag{2.13}$$

The below relation of polyphase matrices

$$\mathbf{R}(z)\mathbf{E}(z) = cz^{-n_0}\mathbf{I}, \tag{2.14}$$

yields the condition of perfect reconstruction (PR) stated as

$$\tilde{x}[n] = cx[n - M - n_0], \tag{2.15}$$

where c is a constant and n_0 is a constant delay.

2.3 Compressed Sensing

Classical compression method involves two steps: sensing and compression wherein, first, an analog data is sampled at or above the Nyquist-rate and then, it is compressed by an appropriate transform coding process. In general, natural signals are sparse or compressible in some transform domain. For example, if a signal is smooth, it is compressible in Fourier domain and if it is piece-wise smooth, it is sparse in the wavelet domain. To understand this, let us consider a signal $\mathbf{x}$ of dimension $N \times 1$ that has been sensed by a traditional sensing technique at or above the Nyquist rate. This signal is next transformed to a sparse signal $\mathbf{r}$ with the help of sparsifying basis $\psi_i, i = 1, 2, ..., N$ as below:

$$\mathbf{x} = \mathbf{\Psi r}. \tag{2.16}$$

A signal $\mathbf{r}$ is $K-$ sparse if all but K elements are zero, whereas a signal is compressible if its sorted coefficients obey the power low decay [86]

$$r_j = Cj^{-q},\ j = 1, 2, ..., N, \tag{2.17}$$

where r_j represent the sorted coefficients and q represents decay power parameter. For large value of q, the decay of coefficients is faster and correspondingly, the signal is more compressible. In compression, some of the largest coefficients of the transformed signal are kept and all other coefficients are discarded. These coefficients along with their location information are sent to the receiver. Having the knowledge of the sparsifying basis and signal coefficients along with their positions in the original signal, the signal is reconstructed back at the receiver end. Thus, this process involves sensing the full signal, although most of the samples in the transformed domain are to be discarded.

The above process consisting of first sensing the whole signal and then discarding many of its transform domain coefficients is inefficient. Compressive sampling or sensing [87, 46, 88] combines these two processes. Instead of sampling the signal at or above the Nyquist rate, signal's linear projection on some measurement basis ϕ_i are obtained. If ϕ_i is the i^{th} measurement basis, then i^{th} observation of the projected signal is given by:

$$y[i] = \sum_{j=0}^{N-1} \phi_{i,j} x[j], i = 0, 1, ..., M-1, \tag{2.18}$$

where M is the number of linear projections of the signal. In compact form, this can be written as:

$$\begin{aligned} \mathbf{y}_{M\times 1} =& \mathbf{\Phi}_{M\times N}\mathbf{x}_{N\times 1}, \\ =& \mathbf{\Phi\Psi r} \\ =& \mathbf{Ar}, \end{aligned} \tag{2.19}$$

where i^{th} measurement basis is stacked as a row of the matrix $\mathbf{\Phi}$ and $\mathbf{A} = \mathbf{\Phi\Psi}$. The measurement or the sensing basis $\mathbf{\Phi}$ can be chosen such that it satisfies Restricted Isometry Property (RIP) [89] and coherency property [46]. Some examples of the measurement matrices satisfying these properties are: random matrices with entries from i.i.d. Gaussian distribution [90], random matrices with entries from uniform Bernoulli distributions [88], and Fourier matrices [46]. In addition, several other structured measurement matrices such as Toeplitz and circulant matrices have also been proposed [91, 92, 93].

If the signal is sparse in some transform domain and also if the measurement basis are incoherent with the sparsifying basis, then CS theory states that the original signal of length N can be recovered with very high probability if the number of linear projections M are taken such that [87]

$$M \geq CK \log(N/K), \tag{2.20}$$

where K is the sparsity of the signal, C is some constant, and $M \ll N$ in general.

Equation (2.19) represents under-determined system of linear equations with $\mathbf{y} = \mathbf{\Phi\hat{x}}$ having infinite many solutions $\mathbf{\hat{x}}$. However, if the signal is sparse in some transform domain $\mathbf{\Psi}$, (2.19) can be solved with a unique solution using l_0 minimization as below:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} ||\mathbf{r}||_0 \text{ subject to: } \mathbf{y} = \mathbf{Ar}. \tag{2.21}$$

The above problem is NP-hard to solve. It has been shown in [94] that l_1 minimization

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \, ||\mathbf{r}||_1 \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{r}, \tag{2.22}$$

provides the same solution as l_0 minimization. Here, $||\mathbf{v}||_1$ denotes the l_1 norm or sum of the absolute values of the vector $\mathbf{v}$. Full signal is reconstructed as: $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}}$. l_1 minimization is known as Basis Pursuit (BP) in literature and can be solved by linear programming [95].

As in (2.22), in general, l_1 norm is used as the sparsity promoting penalty function, which is convex and leads to convex objective function. However, it has been shown in the literature that non-convex penalty functions provide better reconstruction accuracy than convex penalty functions as they induce sparsity more effectively than convex penalty functions [96, 97]. In literature, l_p $(0 \leq p < 1)$ norm (Please note that l_p is actually not a norm, rather it is quasi-norm, for $0 \leq p < 1$, as it does not satisfy triangular inequality property. However, with the abuse of notation and its popularity as l_p norm in literature, we call it a norm in this thesis.) is the most widely used non-convex function [58, 59] for CS based recovery. Correspondingly, the resultant optimization problem using l_p norm as the sparsity promoting penalty function, is given as:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \, ||\mathbf{r}||_p^p \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{r}, \tag{2.23}$$

where $0 \leq p < 1$, $||\mathbf{r}||_p^p = |r(1)|^p + |r(2)|^p + ... + |r(N)|^p$, and $|v|$ is the absolute value of the scalar v. The general optimization problem in terms of the penalty function is given by:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \, \theta(\mathbf{r}) \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{r}, \tag{2.24}$$

where $\theta(.)$ is the sparsity promoting penalty function. For l_p minimization, $\theta(\mathbf{r}) = ||\mathbf{r}||_p^p$. The above problem can be converted to unconstrained optimization problem, given by:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{A}\mathbf{r}||_2^2 + \lambda\theta(\mathbf{r}), \tag{2.25}$$

where λ is the regularization parameter. The full signal is obtained as: $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}}$.

Non-convex function as the sparsity promoting penalty function comes with the trade-off that the resultant objective function is non-convex and leads to many sub-optimal local minima [98]. Also, the resulting objective function is sensitive to λ with respect to global minima [98].

Efforts are made to choose non-convex penalty function such that the overall objective function is convex [99, 100]. However, these work use separable penalty function of the form $\theta(\mathbf{r}) = \sum_i q(r_i)$ that does not preserve convexity of the objective function for singular $\mathbf{A}$ [98]. To overcome this problem, non-separable non-convex penalty function is introduced in [57, 101] that leads to convex objective function even when $\mathbf{A}$ is singular. One of such penalty functions, called generalized minimax-concave (GMC) penalty [101], is given as:

$$\theta(\mathbf{r}) = ||\mathbf{r}||_1 - H_B(\mathbf{r}), \tag{2.26}$$

where $H_B(\mathbf{r})$ is the generalized Huber function given by:

$$H_B(\mathbf{r}) = \underset{\mathbf{r}}{\operatorname{argmin}} ||\mathbf{v}||_1 + \frac{1}{2}||B(\mathbf{r} - \mathbf{v})||_2^2. \tag{2.27}$$

Here, $\mathbf{B}$ is a matrix/operator such that $\mathbf{B} : \mathbb{R}^N \rightarrow \mathbb{R}^M$. To preserve the convexity of the objective function, $\mathbf{B}$ is set such that $\mathbf{B}'\mathbf{B} = (\gamma/\lambda)\mathbf{A}'\mathbf{A}, \gamma \leq 1$, where γ controls the non-convexity of the objective function. This non-separable non-convex penalty function leads to the following optimization problem for sparse signal recovery:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \big\{ ||\mathbf{y} - \mathbf{A}\mathbf{r}||_2^2 + \lambda||\mathbf{r}||_1 - \lambda H_B(\mathbf{r}) \big\}. \tag{2.28}$$

The above optimization problem is observed to yield better performance than convex l_1 and other non-convex regularized optimization problems in the case of signal denoising [101].

Compressive sensing has been applied on images, also called as compressive imaging (CI). For example, let us consider an image $\mathbf{X}$ of dimension $N_1 \times N_2$, that is compressively sensed by a measurement matrix $\mathbf{\Phi}$. These measurements are given by

$$\begin{aligned} \mathbf{y} &= \mathbf{\Phi} vec(\mathbf{X}), \\ &= \mathbf{\Phi}\mathbf{x} \end{aligned} \tag{2.29}$$

where $vec(\mathbf{X}) = \mathbf{x}$ denotes the vector of length $N = N_1 N_2$ of image $\mathbf{X}$ and the measured signal $\mathbf{y}$ is of dimension $M \times 1$, where M is the number of compressive measurements. It has been observed that natural images, in general, are compressible in DCT (discrete cosine transform) [102] and wavelet domain [103]. Hence, DCT or wavelet can be applied as separable transforms on images and used as sparsifying basis $\mathbf{\Psi}$ in (2.19) in CS-based image reconstruction.

2.4 Denoising

2.4.1 Gaussian denosing

Gaussian denoising problem can be mathematically modeled as

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{2.30}$$

where $\mathbf{x}$ is the original signal corrupted by additive white Gaussian noise $\mathbf{n}$. Denoising problem aims to recover the approximate of the original signal $\mathbf{x}$ from its noisy measurements $\mathbf{y}$. Since natural signals are generally sparse in the wavelet domain, denoising can be posed as sparse recovery problem solving l_1 minimization problem with quadratic regularization as below:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\text{argmin}} ||\mathbf{r}||_1 \quad \text{subject to } ||\mathbf{y} - \mathbf{\Psi}\mathbf{r}||_2^2 \leq \epsilon, \tag{2.31}$$

where $\mathbf{\Psi}$ represents the sparsifying transform and $\mathbf{r}$ is the transform domain coefficients of $\mathbf{x}$. The above problem is known as basis pursuit denoising (BPDN) [95], where ϵ is the measure of noise in the signal. One may use $\epsilon = \sqrt{N + 2\sqrt{(2N\sigma)}}$ [104], where σ is the standard deviation of noise estimated using a robust median estimator [105]. Once $\tilde{\mathbf{r}}$ is estimated, signal $\mathbf{x}$ can be reconstructed as $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}}$.

2.4.2 Impulse denoising

We pose impulse denoising as the sparse recovery problem, where first the corrupted pixels are identified using impulse noise detection algorithm and then based on the locations of the impulse noise, sparse recovery problem is solved to denoise the image. These two steps are described below.

2.4.2.1 Impulse Detection Algorithm

We use boundary discriminative noise detection (BDND) algorithm for impulse denoising, which is one of the popular methods for impulse noise detection. It classifies each pixel as corrupted or uncorrupted using a specified sized window centered at that pixel [106]. All pixels in a window are grouped into three clusters based on their intensity values using the following steps:

1. All pixels of a window are stored in ascending order in vector $\mathbf{u}$. Median m of $\mathbf{u}$ is computed. Intensity differences of adjacent pixels of $\mathbf{u}$ are stored in $\mathbf{u}_d$.

2. Vector $\mathbf{u}$ is divided in two bins: pixels with intensities in $[0, m)$ and pixels with intensities in $(m, 255]$ (for 8-bit images). From $\mathbf{u}_d$, pixel location with maximum intensity difference in each bin is marked. Corresponding to these maximum intensity difference, pixel intensities of $\mathbf{u}$ are labeled as the boundary values b_1 and b_2.

First, 21 × 21 window is considered for a pixel. If the pixel intensity lies in the cluster $[b_1, b_2]$, the pixel is considered to be uncorrupted and the process stops. Otherwise, the process is repeated over 3 × 3 window. If the pixel falls in $[b_1, b_2]$ cluster over this smaller window, it is labeled as uncorrupted, else it is labeled as corrupted.

2.4.2.2 Impulse Denoising using Sparse Recovery

Let us consider $\mathbf{y}_{uncor}$ be the vectorized form of all the uncorrupted pixels of the corrupted image $\mathbf{y}$, which can be considered as being picked by the operator Φ.

This can be mathematically written as:

$$\begin{aligned}\mathbf{y}_{uncor} =& \boldsymbol{\Phi}\mathbf{y} \\ \approx& \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{r},\end{aligned} \tag{2.32}$$

where $\boldsymbol{\Psi}$ represents the sparsifying transform for the image and $\mathbf{r}$ is the transformed coefficients of original image $\mathbf{x}$ requires to be recovered/denoised. This problem is similar to compressed sensing based reconstruction [46]. Hence, we solve it by using l_1 minimization method as below:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \ \ ||\mathbf{r}||_1 \text{ subject to } \mathbf{y}_{uncor} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{r}. \tag{2.33}$$

Denoised image is obtained as: $\tilde{\mathbf{x}} = \boldsymbol{\Psi}\tilde{\mathbf{r}}$.

2.5 Evaluation metrics

In this subsection, we present relations for the metrics used in this work to evaluate the quality of the reconstructed signal. Let us consider, $x[n]$ be the original signal, $\mathbf{x}$ is the vectorized form of $x[n]$, $\tilde{x}[n]$ is the reconstructed signal, and N is the length of signal $x[n]$.

1. Signal to Noise Ratio (SNR): SNR (in dB) is defined as below:

$$\text{SNR} = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1}(x[n])^2}{\sum_{n=0}^{N-1}(x[n]-\tilde{x}[n])^2}\right), \tag{2.34}$$

2. Peak signal to Noise Ratio (PSNR): PSNR (in dB) is defined as below:

$$\text{PSNR} = 10\log_{10}\frac{(\max(\mathbf{x}))^2}{\sum_{n=0}^{N-1}(x[n]-\tilde{x}[n])^2}, \tag{2.35}$$

   Operation max(.) picks the maximum intensity value of the image.

3. Percentage Root Mean Square Difference (PRD): PRD (in %) is defined as below:

$$\text{PRD} = \frac{\sum_{n=0}^{N-1}(x[n]-\tilde{x}[n])^2}{\sum_{n=0}^{N-1}(x[n])^2} \times 100, \tag{2.36}$$

4. Sampling ratio (S): Sampling ratio (in %) is defined as below:

$$S = \left\lceil \frac{M}{N} \right\rceil * 100, \tag{2.37}$$

where, M represents the number of compressive measurements of the signal.

Chapter 3

DWTL1: Dyadic Wavelet Transform Learning using Lifting for 1-D Signals

Discrete dyadic wavelet transform provides an efficient representation for a variety of multi-dimensional signals [5]. This efficient signal representation stems from the fact that wavelets tend to capture signal information into a few significant coefficients. Owing to this advantage, wavelets have been applied successfully in several applications. One of the advantage with wavelets is that there is no unique basis. One may choose the set of basis depending upon the type of application and signal of interest. Since the wavelet basis are not unique, it is better to learn wavelet basis and hence, wavelet transform matched to a given signal in a particular application. Motivated with this, we propose methods to learn signal-matched dyadic wavelet transform (DWT) using the lifting framework in this chapter. We call the proposed method as dyadic wavelet transform learning (DWTL). We also explore the use of the learned transform in applications.

Section 3.1 provides theoretical contributions of our work related to 1-D signals including various methods to learn DWT. Applications of the learned DWT are presented in Section3.2. We summarize the work of this chapter in Section3.3.

3.1 Theoretical Contributions

We present four theoretical contributions related to wavelet transform learning of 1-D signals. First, in Section 3.1.1, a method is presented for learning DWT when the original signal is known. Section 3.1.2 presents the method to learn DWT in inverse problems. Third method for DWTL is presented in Section 3.1.3 to learn DWT for a class of signals. We also propose weighted non-convex minimization in Section 3.1.4 along with an algorithm to solve it.

3.1.1 DWTL from original signal

Since the translates of the associated wavelet filters form the basis in l^2-space, wavelet transform learning implies learning wavelet filter coefficients.

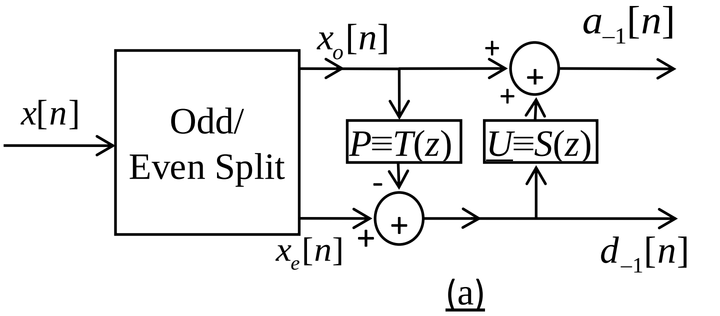
The learning of dyadic wavelet transform or associated wavelet filters is carried out in the lifting framework. Lifting framework is described in Chapter-2. Lifting is modular, and guarantees perfect reconstruction [32] at every stage. It can be effectively utilized for learning new customized wavelet filters [32]. As described in Chapter-2, a general lifting scheme consists of three steps: Split, Predict, and Update as shown in Figure 3.1[1]. These steps can be equivalently converted to a conventional 2-channel wavelet system of Figure 3.2 with analysis filters labeled as $H_0(z) = Z\{h_0[n]\}$, $H_1(z) = Z\{h_1[n]\}$ and the synthesis filters as $F_0(z) = Z\{f_0[n]\}$, $F_1(z) = Z\{f_1[n]\}$. All the three stages of lifting for DWTL are described in the following subsections.

3.1.1.1 Split Step via Lazy Wavelet

As discussed in Chapter-2, in split step, the input signal, $x[n]$ is split into even and odd indexed samples $x_e[n]$ and $x_o[n]$, respectively, on analysis side (Figure 3.1) and the original signal can be recovered perfectly by interlacing or combining these even and odd indexed samples on synthesis side [32]. The operation of split is performed by a wavelet system called *Lazy* wavelet system. We consider following filters for *Lazy* wavelet system:

$$H_0(z) = 1, \quad H_1(z) = z$$

[1] Figure 3.1 and 3.2 are reproduced from Chapter-2 for the ease of understanding.



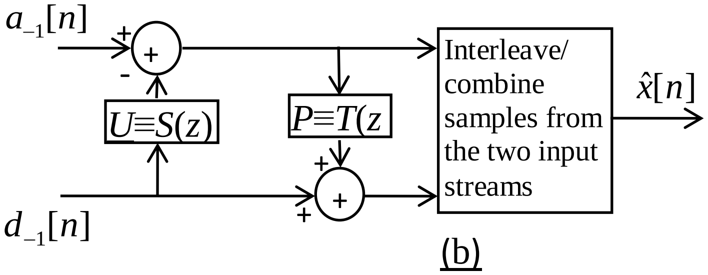


Figure 3.1: Split, predict and update stages of lifting

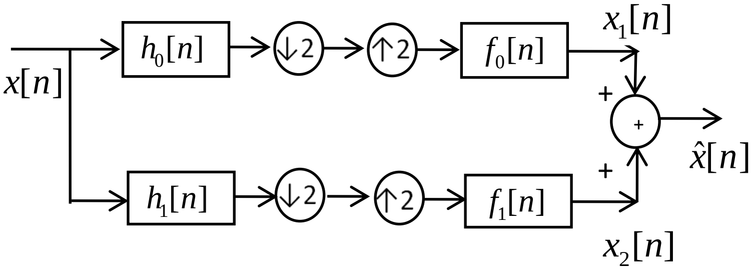


Figure 3.2: Wavelet system equivalent to Figure 3.1

$$F_0(z) = 1, \quad F_1(z) = z^{-1} \tag{3.1}$$

The above choice of wavelet system yields perfect reconstruction, i.e., $\tilde{x}[n] = x[n]$, where $\tilde{x}[n]$ is the reconstructed signal. Starting with this choice of *Lazy* wavelet system, we update all the filters of dyadic wavelet system (Figure 3.2) with the help of predict and update filters, $T(z)$ and $S(z)$, respectively, which we learn from the input signal. This results in the updated wavelet filters, which are learned according to the given input signal. The predict and update filters are learned from the signal in predict and update stages, respectively, as described below.

3.1.1.2 Predict Stage

Once the input signal $x[n]$ is passed through the *Lazy* wavelet system designed above, we obtain $x_e[n]$ and $x_o[n]$, even and odd indexed samples of the original signal, respectively, as:

$$\begin{aligned} x_e[n] =& x[2n] && (3.2) \\ x_o[n] =& x[2n+1] && (3.3) \end{aligned}$$

where $n = 0, 1, ..., \frac{N}{2} - 1$ and N denotes the length of the original signal $x[n]$.

In order to learn predict polynomial/filter $T(z)$ in Figure 3.1, we choose to predict $x_o[n]$ from $x_e[n]$, although the reverse can be done in a similar fashion. The predicted signal is subtracted from the odd indexed samples to obtain prediction error as below:

$$\begin{aligned} d[n] =& x_o[n] - x_e[n] * t[n] \\ =& x_o[n] - p[n] \\ =& x[2n+1] - p[n], && (3.4) \end{aligned}$$

where,

$$\begin{aligned} p[n] &= \sum_{k=0}^{L_t-1} t[k]x_e[n-k] \\ &= \sum_{k=0}^{L_t-1} t[k]x[2n-2k] && (3.5) \end{aligned}$$

where, $T(z)$=Z-transform of $t[n]$, L_t is the length of the predict filter $T(z)$, '$*$' denotes convolution operator and $p[n] = x_e[n] * t[n]$ denotes the predicted signal. For good prediction, the predict polynomial/filter should be chosen such that odd indexed samples are predicted with the nearest neighboring samples only. This is ensured by choosing the structure of $T(z)$ as provided in Theorem-3 below:

Theorem 1. *The following structure of predict filter allows odd-indexed samples to be predicted from their nearest even-indexed samples, i.e., from their immediate* *past and immediate future samples:*

$$T(z) = z^{-(\frac{L_t}{2}-1)} \sum_{i=0}^{L_t-1} t[i]z^i, \tag{3.6}$$

where $T(z)$ is an even-length filter. Even length $T(z)$ will ensure that equal number of past and future samples are used in prediction.

Proof. On using (3.6) in (3.5), we obtain:

$$p[n] = \sum_{k=0}^{L-1} t[k]x[2n - L_t + 2 + 2k]. \tag{3.7}$$

This is to note that the n^{th} sample of $p[n]$ predicts sample $x[2n+1]$. On expanding (3.7), we obtain

$$\begin{aligned} p[n] =& t[0]x[2n - L_t + 2] + ... + t[\frac{L_t}{2} - 1]x[2n] \\ &+ t[\frac{L_t}{2}]x[2n + 2] + ... + t[L_t - 1]x[2n + L_t]. \end{aligned} \tag{3.8}$$

Thus, the sample $x[2n+1]$ is predicted from its exact nearest even neighbors. For better clarity, Figure 3.3 shows even neighboring samples that are being used to predict odd samples with a 2-tap and 4-tap predict filter $T(z)$. This proves the theorem. □

Next, we minimize the energy of prediction error signal ($d[n]$) to learn the predict filter as below:

$$\begin{aligned} \tilde{\mathbf{t}} =& \underset{\mathbf{t}}{\text{argmin}} \sum_{n=0}^{N_d-1} (d[n])^2 \\ =& \underset{\mathbf{t}}{\text{argmin}} \sum_{n=0}^{N_d-1} (x[2n+1] - p[n])^2, \\ =& \underset{\mathbf{t}}{\text{argmin}} \sum_{n=0}^{N_d-1} (x[2n+1] - \sum_{k=0}^{L_t-1} t[k]x[2n-2k])^2, \end{aligned} \tag{3.9}$$

where N_d is the length of the prediction error.

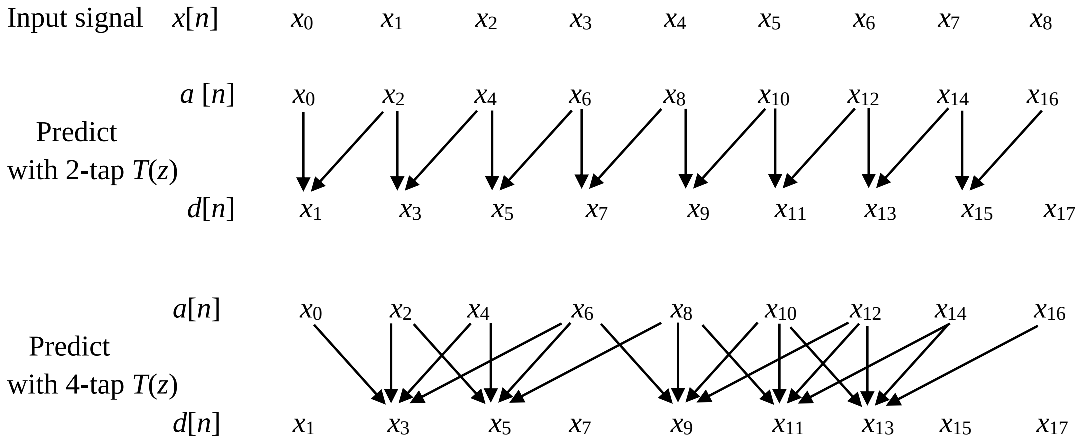

Input signal $x[n]$
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
$a[n]$
x_0 x_2 x_4 x_6 x_8 x_{10} x_{12} x_{14} x_{16}
Predict with 2-tap $T(z)$
$d[n]$
x_1 x_3 x_5 x_7 x_9 x_{11} x_{13} x_{15} x_{17}
$a[n]$
x_0 x_2 x_4 x_6 x_8 x_{10} x_{12} x_{14} x_{16}
Predict with 4-tap $T(z)$
$d[n]$
x_1 x_3 x_5 x_7 x_9 x_{11} x_{13} x_{15} x_{17}

Figure 3.3: An odd sample is being predicted from its neighboring even samples with a 2-tap and a 4-tap filter $T(z)$

The above equation can be written in the vector form as below:

$$\tilde{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmin}} ||\mathbf{b} - \mathbf{A}\mathbf{t}||_2^2, \tag{3.10}$$

where $b(i+1) = x[2i+1]$, $A(i+1, j+1) = x[2i-2j]$ $\forall$ $i \in \{0, 1, ..., N_d - 1\}$ and $j \in \{0, 1, ..., L_t - 1\}$. Here, $b(.)$ and $A(.,.)$ denotes the entries of the column $\mathbf{b}$ and $\mathbf{A}$, respectively. The above formulation leads to the following closed-form solution for $\tilde{\mathbf{t}}$:

$$\tilde{\mathbf{t}} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{b}, \tag{3.11}$$

where $\mathbf{A}'$ denotes the transpose of $\mathbf{A}$.

The learned predict filter $T(z) = Z(t[n])$ is used to update analysis highpass and synthesis lowpass filter using (3.12) and (3.13), respectively.

$$H_1^{new}(z) = H_1(z) - H_0(z)T(z^2). \tag{3.12}$$

$$F_0^{new}(z) = F_0(z) + F_1(z)T(z^2). \tag{3.13}$$

Note that $H_0(z)$, $H_1(z)$, $F_0(z)$ and $F_1(z)$ corresponds to the filters of Lazy wavelet system considered in section 3.1.1.1, whereas $H_1^{new}(z)$ and $F_0^{new}(z)$ are the filters updated with the help of $T(z)$, learned from the input signal $x[n]$.

3.1.1.3 Update Stage

Next, the update filter $S(z)$ is required to be learned to update even indexed samples. To this end, detail coefficients $d[n]$ obtained in (3.4) are passed through the update filter $S(z)$ as shown in Figure 3.1 and added to the even indexed samples to obtain the updated wavelet approximate coefficients as:

$$\begin{aligned} a[n] =& x_e[n] + d[n] * s[n] \\ =& x[2n] + \sum_{k_1=0}^{L_s-1} s[k]d[n-k_1], \end{aligned} \tag{3.14}$$

where $S(z)$ denotes the z-transform of update filter $s[n]$.

Similar to the predict stage filter, we require to choose $s[n]$ such that the elements of the upper branch are updated using nearest neighbors only. The corresponding structure for $s[n]$ is provided by Theorem-2 as below.

Theorem 2. *The following structure of the update filter allows the elements of the upper branch to be updated from nearest neighbors:*

$$S(z) = z^{-(\frac{L_s}{2}-1)} \sum_{i=0}^{L_s-1} s[i]z^{-i}, \tag{3.15}$$

where L_s is the length of filter $S(z)$ or $s[n]$. Note that $S(z)$ is an even-length filter that ensures that sample update is done using equal number of past and future samples.

Proof. For the sake of simplicity, let us consider two tap predict filter that provides the following detail coefficients:

$$d[n] = -t[0]x[2n] + x[2n+1] - t[1]x[2n+2] \tag{3.16}$$

Once this signal is passed through the update stage filter in the update branch, we obtain:

$$u[n] = d[n] * s[n]. \tag{3.17}$$

With the choice of filter in (3.15), we obtain:

$$\begin{aligned} u[n] =& s[0]d[n+\frac{L_s}{2}-1] + s[1]d[n+\frac{L_s}{2}-2] + ... \\ &+ s[L_s-2]d[n-\frac{L_s}{2}+1] + s[L_s-1]d[n-\frac{L_s}{2}]. \end{aligned} \tag{3.18}$$

On expanding (3.18) and rearranging, we obtain

$$\begin{aligned} u[n] = & -s[0]t[1]x[2n+L_s] + s[0]x[2n+L_s-1]+ \\ & (-s[0]t[0]-s[1]t[1])x[2n+L_s-2] + ...+ \\ & (-s[L_s-2]t[0]-s[L_s-1]t[1])x[2n-L_s+2]+ \\ & s[L_s-1]x[2n-L_s+1] - s[L_s-1]t[0]x[2n-L_s] \end{aligned} \tag{3.19}$$

The above signal updates approximate coefficients, i.e., $x[2n]$. It can be clearly noticed that coefficients $x[2n]$ are updated using the nearest neighbors from $x[2n+L_s]$ to $x[2n-L_s]$. For better clarity, Figure 3.4 shows the neighboring samples that are being used to update even samples with a 2-tap predict filter $T(z)$ and a 2-tap update filter $S(z)$. This proves the theorem. □

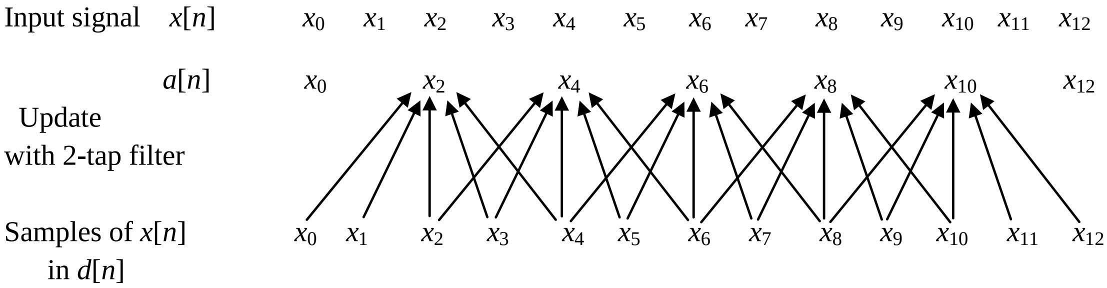


Figure 3.4: Even samples being updated from neighboring samples with a 2-tap $T(z)$ and 2-tap update filter $S(z)$

The updated approximate coefficients $a[n]$ obtained above are passed through the upper branch of the synthesis side of Figure 3.2. First, it is passed through a

2-fold upsampler that provides:

$$x_{1u}[n] = \begin{cases} a[\frac{n}{2}] & \text{if } n \text{ is a multiple of 2} \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

Next, signal $x_{1u}[n]$ is passed through the synthesis lowpass filter $f_0^{new}[n]$ that was updated in the predict stage mentioned earlier. This provides us the signal $x_1[n]$ (shown in Figure 3.2) reconstructed from the upper subband (branch) only and is given by

$$\begin{aligned} x_1[n] =& x_{1u}[n] * f_0^{new}[n] \\ =& \sum_{k_2=0}^{L_{f_0}-1} f_0^{new}[k_2] x_{1u}[n-k_2], \end{aligned} \tag{3.21}$$

where L_{f_0} is the length of the filter f_0^{new}.

Since detail coefficients represent prediction error having very less energy, most of the energy of the input signal should move to the updated (upper) branch signal. Hence, the energy of signal $x_1[n]$ should be close to the original signal. This allows us to define the objective function of (3.22), minimization of which via least squares yields the solution $s[n]$.

$$\tilde{\mathbf{s}} = \underset{\mathbf{s}}{\text{argmin}} \sum_{n=0}^{N-1} (x[n] - x_1[n])^2. \tag{3.22}$$

It can be noted from (3.14), (3.20) and (3.21), that $x_1[n]$ can be written in terms of update filter $s[n]$ in the above equation as follows:

$$x_1[n] = \sum_{k_2=0}^{L_{f_0}-1} f_0^{new}[k_2](x[n-k_2] + \sum_{k_1=0}^{L_s-1} s[k_1]d[\frac{n-k_2-2k_1}{2}]). \tag{3.23}$$

From above two equations, we obtain:

$$\tilde{\mathbf{s}} = \underset{\mathbf{s}}{\text{argmin}} \sum_{n=0}^{N-1} (x[n] - \sum_{k_2=0}^{L_{f_0}-1} f_0^{new}[k_2] x[n-k_2] - \sum_{k_2=0}^{L_{f_0}-1} \sum_{k_1=0}^{L_s-1} f_0^{new}[k_2] s[k_1] d[\frac{n-k_2-2k_1}{2}]). \tag{3.24}$$

The above relation can be written in matrix form as below:

$$\tilde{\mathbf{s}} = \underset{\mathbf{s}}{\text{argmin}}||\mathbf{b} - \mathbf{A}\mathbf{s}||_2^2, \tag{3.25}$$

where $b(i{+}1) = x[i] - \sum_{k_2=0}^{L_{f_0}-1} f_0^{new}[k_2]x[i-k_2]$, $A(i{+}1, j{+}1) = \sum_{k_2=0}^{L_{f_0}-1} f_0^{new}[k_2]d[\frac{i-k_2-2j}{2}])$ $\forall\ i \in \{0, 1, ..., N-1\}$ and $j \in \{0, 1, ..., L_s - 1\}$. The above problem has following closed form solution:

$$\tilde{\mathbf{s}} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{b}. \tag{3.26}$$

Please note that we use the following relation throughout the thesis:

$$u_2[n] = \begin{cases} u_1[\frac{n}{c}] & \text{if } n \text{ is a multiple of c} \\ 0 & \text{otherwise,} \end{cases} \tag{3.27}$$

where $u_1[n]$ is the signal that is upsampled by a factor of c to provide the signal $u_2[n]$.

Once, update filter $S(z) = Z(s[n])$ is learned as above, the analysis lowpass and synthesis highpass filters are updated using (3.28) and (3.29), respectively.

$$H_0^{new}(z) = H_0(z) + H_1(z)S(z^2). \tag{3.28}$$

$$F_1^{new}(z) = F_1(z) - F_0(z)S(z^2). \tag{3.29}$$

Note that $H_0(z)$ and $F_1(z)$ corresponds to the filters of Lazy wavelet system considered in section 3.1.1.1, $H_1(z)$ and $F_0(z)$ are analysis highpass and synthesis lowpass filters, already updated in predict stage (section 3.1.1.2 above), and $H_0^{new}(z)$ and $F_1^{new}(z)$ are analysis lowpass and synthesis highpass filters, which are being updated with the help of $S(z)$, learned from the input signal.

This completes the process of learning all the filters of 2-channel dyadic wavelet system (Figure 3.2) from the input signal. These steps of learning filters of dyadic wavelet systems are repeated in successive stages in multi-level wavelet decomposition.

Note that the wavelet learned with the proposed method is matched to the input signal where energy in the lower branch and hence, one of the branch (or subband) is minimized. At the same time, the update step tries to push more energy in the other branch (subband) ensuring perfect reconstruction. This implies that signal energy gets concentrated in the wavelet coefficients of one of the subband. Since these steps are repeated in successive stages in multi-level wavelet decomposition, the number of coefficients with significant magnitude decreases enforcing sparsity with the learned matched wavelet in the lifting framework.

This is to be noted that, since the lifting scheme is modular, more number of such predict and update stage filters can be learned and appended in order to design higher order or larger length filters.

DWTL with Linear Phase filters

A linear phase filter is symmetric or anti-symmetric about the center weight. On expanding filters $h_0[n]$ and $h_1[n]$ in terms of polynomials $T(z)$ and $S(z)$, it is noted that the learned wavelet filters can be made linear phase, if $T(z)$ and $S(z)$ are chosen appropriately. For example, the following choice of 2-tap $T(z)$ and $S(z)$ leads to linear phase wavelet filters

$$T(z) = t[0](1 + z) \text{ and } S(z) = s[0](1 + z^{-1}). \tag{3.30}$$

The above relation can be extended to L_t and L_s-tap predict and update filters, respectively to learn larger length linear phase wavelet filters.

3.1.2 DWTL in inverse problems

In applications such as denoising, CS, deconvolution etc., the degraded signal is present and problem is to estimate the original signal. These problems are known as inverse problems. Mathematically, inverse problems can be written as:

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \boldsymbol{\eta}, \tag{3.31}$$

where, $\mathbf{y}$ is the degraded form of the original signal $\mathbf{x}$ and is observed with the observation process $\mathbf{D}$ in the presence of additive white Gaussian noise $\boldsymbol{\eta}$. For example, $\mathbf{D}$ is the identity matrix in Gaussian denoising, measurement/sensing matrix in CS problem and is convolution operator in deblurring problems. If the signal is known to be sparse in some transform domain $\boldsymbol{\Psi}$, then the original signal can be estimated/reconstructed from the degraded form by solving the following l_1 minimization problem (or it's equivalent form):

$$\tilde{\mathbf{r}} = \min_{\mathbf{r}} ||\mathbf{y} - \mathbf{D}\boldsymbol{\Psi}\mathbf{r}||_2^2 + \lambda||\mathbf{r}||_1, \tag{3.32}$$

where $\mathbf{r}$ represent the transform domain signal, λ is the regularization parameter and original signal is estimated as $\tilde{\mathbf{x}} = \boldsymbol{\Psi}\tilde{\mathbf{r}}$.

The method of DWTL presented in the previous subsection requires full original signal to learn DWT, whereas full original signal is not accessible in inverse problems, due to which the method cannot be applied for wavelet transform learning there. Hence, we present another method in this subsection, where we learn DWT from degraded signals in inverse problems. The method is divided into two stages. In stage-1, we obtain a coarse signal estimate from the observed degraded signal using a standard wavelet. We call this a coarser estimate because the wavelet used is not signal-matched and hence, the original signal may not be that sparse over this wavelet compared to the signal-matched wavelet. This will impact the reconstruction performance. In stage-2, we learn all the filters of a matched perfect reconstruction filterbank or wavelet system. We can reconstruct/estimate original signal from the observed signal using learned DWT as the sparsifying transform.

3.1.2.1 Stage 1: Coarser Signal Estimation

In this stage, we reconstruct a coarser estimate of the signal from it's degraded form $\mathbf{y}$ by solving (3.32). We use any standard wavelet as the sparsifying transform $\boldsymbol{\Psi}$. We use biorthogonal 5/3 wavelet in our experiments. The coarser approximation of the signal is obtained as $\tilde{\mathbf{x}} = \boldsymbol{\Psi}\tilde{\mathbf{r}}$ after solving (3.32). We solve the above optimization problem using MATLAB solver spgl1 [107, 108]. We will show later in applications section that the reconstruction accuracy with (3.32) using standard wavelet is not good as compared to the learned wavelet transform. Hence, we call the estimated signal $\tilde{\mathbf{x}}$ as the coarser estimate of the original signal $\mathbf{x}$.

3.1.2.2 Stage 2: Wavelet Transform Learning

We use the signal reconstructed in the previous stage to learn signal-matched wavelet system by learning predict and update filters of the lifting structure. This is described below.

Predict Step

For learning signal-matched wavelet in lifting framework, first, we consider Lazy wavelet with $H_0(z) = 1$, $H_1(z) = z$, $F_0(z) = 1$, and $F_1(z) = z^{-1}$, respectively. We start with this Lazy wavelet structure and proceed with learning the predict stage filter $T(z)$.

We apply coarser version of original signal $\tilde{\mathbf{x}}$ estimated in Stage-1 as input to the *Lazy* wavelet system and obtain even and odd sampled streams $\tilde{\mathbf{x}}_e(n)$ and $\tilde{\mathbf{x}}_o(n)$, respectively as shown in Figure 3.1(a). We pass even indexed samples $\tilde{\mathbf{x}}_e(n)$ through the predict stage filter and write the output of the lower subband signal, as below:

$$\begin{aligned} \tilde{d}[n] &= \tilde{x}_o[n] - \tilde{x}_e[n] * t[n], \\ &= \tilde{x}[2n+1] - \sum_{k=0}^{L_t-1} t[k]\tilde{x}_e[n-k],, \end{aligned} \tag{3.33}$$

We use (3.6) for the structure of predict filter.

Subband signal $\tilde{\mathbf{d}}$ can be considered as the noisy version of detail coefficients $\mathbf{d}$ that could be obtained by passing the original signal $\mathbf{x}$ through the learned signal-matched analysis wavelet branch. This can be written as:

$$\tilde{\mathbf{d}} = \mathbf{d} + \boldsymbol{\eta}_1, \tag{3.34}$$

where $\boldsymbol{\eta}_1$ is the corresponding error.

Further, we analyze this coarser approximate signal $\tilde{\mathbf{x}}$ using the same biorthogonal 5/3 wavelet with which it had been reconstructed and obtain detail coefficients (subband coefficients of highpass filtered branch) $\hat{\mathbf{d}}$. Again, this signal can be considered as the noisy version of detail coefficients $\mathbf{d}$ and can be written as below:

$$\hat{\mathbf{d}} = \mathbf{d} + \boldsymbol{\eta}_2, \tag{3.35}$$

where $\boldsymbol{\eta}_2$ is the corresponding error.

From (3.34) and (3.35), we obtain:

$$\hat{\mathbf{d}} = \tilde{\mathbf{d}} + \boldsymbol{\eta}, \tag{3.36}$$

where $\boldsymbol{\eta}$ is the error that consists of two components: 1) because we are using approximate signal $\tilde{\mathbf{x}}$ instead of the original signal $\mathbf{x}$ and 2) because we are using biorthogonal 5/3 wavelet instead of the signal-matched wavelet.

On substituting for $\tilde{\mathbf{d}}$ from (3.33) and (3.6) in (3.36) and writing in the matrix form, we obtain

$$\hat{\mathbf{d}} = \mathbf{At} + \boldsymbol{\eta}, \tag{3.37}$$

where $\mathbf{A}$ is the convolution matrix consisting of even and odd indexed samples of $\tilde{\mathbf{x}}$ and $\mathbf{t}$ denotes the vectorized form of predict stage filter $t[n]$ or $T(z)$. We solve for $\mathbf{t}$ in (3.37) using least squares method and substitute in (3.12) and (3.13) to update the analysis highpass and synthesis lowpass filters and obtain new filters $H_1^{new}(z)$ and $F_0^{new}(z)$, respectively. This ends the predict stage.

Update Stage

In the update stage, update polynomial $S(z)$ is required to be computed. In order to do this, we write the output of the upper subband signal using the lower subband signal, $\tilde{d}[n]$ as below:

$$a[n] = x_e[n] + \tilde{d}[n] * s[n], \tag{3.38}$$

where $s[n]$ is the time domain description of the update stage filter $S(z)$. The structure of the update filter, $S(z)$ is provided by (3.15).

This subband signal $a[n]$ is passed through a 2-fold upsampler that provides:

$$\tilde{x}_{1u}[n] = \begin{cases} a[\frac{n}{2}] & \text{if } n \text{ is a multiple of 2} \\ 0 & \text{otherwise.} \end{cases} \tag{3.39}$$

Next, signal $\tilde{x}_{1u}[n]$ is passed through the synthesis lowpass filter $f_0^{new}[n]$ that was updated in the predict stage mentioned earlier. This provides us the signal

$\tilde{x}_1[n]$ reconstructed from the upper subband only and is given by

$$\tilde{x}_1[n] = \tilde{x}_{1u}[n] * f_0^{new}[n]. \tag{3.40}$$

Assuming that the original signal of interest is rich in low frequency content, signal $\tilde{x}_1[n]$ reconstructed in the upper subband should be in close approximation to the input signal $\tilde{x}[n]$. This allows us to solve for the update stage filter as below:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \sum_n (\tilde{x}_1[n] - \tilde{x}[n])^2. \tag{3.41}$$

It can be noted from (3.38), (3.39) and (3.40), that $\tilde{\mathbf{x}}_1$ can be written in terms of update stage filter $s[n]$ obtained on solving (3.41) using least squares method. Correspondingly, analysis lowpass filter $H_0(z)$ and synthesis highpass filter $F_1(z)$ are updated to $H_0^{new}(z)$ and $F_1^{new}(z)$ using (3.28) and (3.29), respectively. This completes learning of all the filters of wavelet system.

Signal Reconstruction using Learned DWT

Once we have learned DWT using the above method, we can estimate the original signal from the it's degraded form $\mathbf{y}$ by solving (3.32) with learned DWT as the sparsifying transform $\mathbf{\Psi}$.

3.1.3 DWTL from a class of signals

As discussed in the previous subsection, the method presented in section3.1.1 cannot be applied in inverse problems as the full original signal is not available. On the other hand, the method presented in section 3.1.2 to learn DWT in inverse problem requires coarse estimate of the signal. This increases complexity in the overall learning method. There exist many classes of signals in nature where the variation is very small over different signals in the class. ECG (electrocardiogram) and brain MRI (magnetic resonance images) are two examples of such classes. We use low variation property of such classes of signals in our work to learn DWT for them. For this, we form an ensemble of a class of signals, for which the DWT is to be learned. We use this ensemble of signals in the method presented in section 3.1.1 to learn DWT corresponding to this class of signals. It is assumed here that the ensemble captures most of the variations in the class, which is possible if the ensemble is formed carefully.

The limitation of the proposed method is that one has to identify the class of signal having low variation to apply this method. Although this kind of classes can be identified using the statistical properties of the signals of the class under consideration, we have not used this approach in this work. We would like to explore this problem in future.

The learned DWT for a class of signals can be used as the sparsifying transform in any application related to them. Also, the learned DWT for a particular class of signals can be applied to any signal of the class. We present DWTL method only for ECG signals in this work, although the same method can be extended to other similar class of signals (having low variation in the class) as well. The proposed method for ECG signals is presented in the next paragraph.

We consider MIT-BIH Arrhythmia ECG dataset [109] most commonly used by the ECG community. We use 10 ECG signals (100, 102, 105, 107, 112, 115, 117, 118, 124 and 222), having several variations, of Arrhythmia database and construct an ensemble shown in Figure 3.5. We consider only 3 seconds duration of each signal that includes roughly 4 beats of each signal. We use this ensemble of ECG signals as the input signal $x[n]$ in the method presented in section 3.1.1 to learn DWT corresponding to them. We call the learned DWT as ECGlet and use it in CS based reconstruction of ECG signals, which is presented in section 3.2.3.

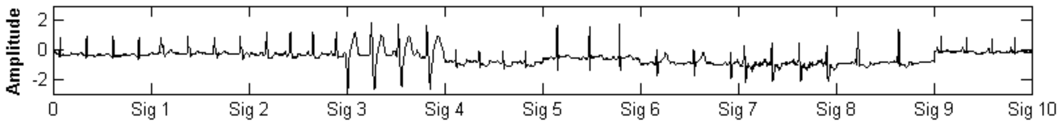


Figure 3.5: Ensemble of signals used for learning ECGlet: segment Sig(i-1) to Sig(i) on x-axis represents the i^{th} ECG signal.

3.1.4 Weighted Non-Convex (WNC) Minimization for CS based recovery

As discussed in section 2sec:CS Theory, CS problem can be mathematically written as:

$$\mathbf{y} = \mathbf{\Phi x}, \tag{3.42}$$

where $\mathbf{x}$ is the original signal of dimension $N \times 1$, $\mathbf{y}$ is the compressed signal of dimension $M \times 1$ and $\mathbf{\Phi}$ corresponds to the sensing/measurement matrix of dimension $M \times N$.

As non-convex functions strongly promote sparsity than convex functions and also, they generally leads to better reconstruction accuracy as compared to convex functions, we can use non-convex function as the sparsity promoting penalty function to recover full signal. Since l_p $(0 \leq p < 1)$ norm is the most widely used non-convex function in literature [58, 59], we use l_p norm as non-convex penalty function in this section. Correspondingly, the resultant optimization problem is given as:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} ||\mathbf{r}||_p^p \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{r}, \tag{3.43}$$

where $\mathbf{A} = \mathbf{\Phi}\mathbf{\Psi}$.

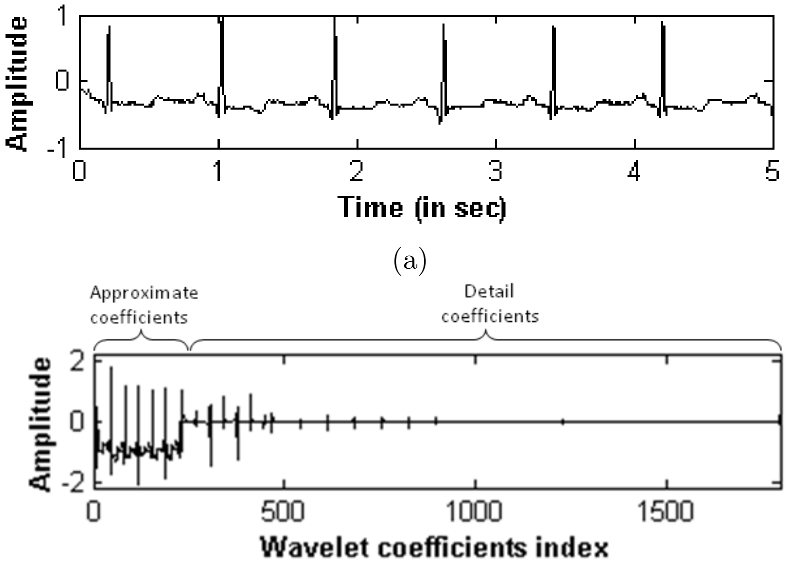
Proposed weighted non-convex minimization: Since most of the signals are rich in low frequency content, the approximate wavelet coefficients are less sparse compared to the detail wavelet coefficients. For example, Figure 3.6a shows one of the MIT-BIH signal (signal-100) for 5-seconds of duration while Figure 3.6b shows its three level wavelet decomposition with approximate and detail coefficients highlighted. It is observed that approximate coefficients contain more than 95% of signal energy. This fact, that signal energy is concentrated in a limited number of transform coefficients, encourages us to impose weighted sparsity on the wavelet transform of the signal. For this, we introduce an operator $\mathbf{\Theta}$, defined as below:

$$\mathbf{\Theta}\mathbf{r} = \begin{cases} \alpha_1 r_i & \text{if } r_i \in \text{ approximate coefficients} \\ \alpha_2 r_i & \text{if } r_i \in \text{ detail coefficients} \end{cases} \tag{3.44}$$

where α_1 and α_2 are weights for approximate and detail coefficients, respectively. Without loss of generality, we choose $\alpha_2 = 1$ and $\alpha_1 = \alpha < \alpha_2$ because the approximate coefficients are not sparse and hence, should be given less weights compared to detail coefficients.

With the above weights, we propose to solve the following optimization problem to reconstruct a signal:

$$\tilde{\mathbf{r}} = \min_{\mathbf{r}} ||\mathbf{\Theta}\mathbf{r}||_p^p \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{r}. \tag{3.45}$$



(a)

(b)

Figure 3.6: (a) 5-seconds of ECG signal-100 (b) Wavelet transform of the signal in (a) with 'db4' wavelet.

We call the above formulation of l_p minimization problem with weighting as the *weighted non-convex (WNC)* minimization problem. This is to note that this problem reduces to the conventional l_1 minimization problem when $\alpha_1 = \alpha_2 = 1$ and $p = 1$. The above problem cannot be solved directly. Although iterative reweighted least squares (IRLS) method can be used to solve it (similar to [61]), it leads to an extra parameter that needs tuning for optimal reconstruction accuracy. We propose a parameter free method that solves (3.45) iteratively by converting this weighted l_p norm to weighted l_1 norm as described next.

The l_p norm of weighted signal, $\mathbf{\Theta r}$ can be approximated to the weighted l_1 norm of the signal as $||\mathbf{\Theta r}_{(i)}||_p^p = ||\mathbf{W}_{(i)}\mathbf{\Theta r}_{(i)}||_1$. Here, subscript i denotes the iteration number and $\mathbf{W}_{(i)}$ is the weighting matrix used in the i^{th} iteration and is obtained using the $(i-1)^{\text{th}}$ estimate of the signal as:

$$\mathbf{W}_{(i)} = diag(|\mathbf{\Theta r}_{(i-1)}|_1^{p-1}). \tag{3.46}$$

Note that $\mathbf{\Theta}$ is fixed for all iterations. Considering $\mathbf{\Theta}$ to be an identity matrix that leads to l_p minimization problem, it is known that

$$\begin{aligned}||\mathbf{r}_{(i)}||_p^p &= \sum_{k=0}^{N-1} |r_{(i)}(k)|^p = \sum_{k=0}^{N-1} |r_{(i)}(k)|^{p-1}|r_{(i)}(k)| \\ &= \sum_{k=0}^{N-1} ||r_{(i)}(k)|^{p-1}r_{(i)}(k)| \\ &\sim \sum_{k=0}^{N-1} ||r_{(i-1)}(k)|^{p-1}r_{(i)}(k)| \text{ as } n \to \infty \\ &= \sum_{k=0}^{N-1} |W_{(i)}(k,k)r_{(i)}(k)| = ||\mathbf{W}_{(i)}\mathbf{r}_{(i)}||_1, \end{aligned} \tag{3.47}$$

where $\sim$ denotes asymptotic equality as $n \to \infty$ or when the estimate converges to some value. Equation (3.47) shows that l_p norm can be solved using l_1 norm in an iterative manner till the solution converges. Similar relation holds with weighted l_p norm introduced in (3.45). Hence, (3.45) is solved using weighted l_1 norm as formulated below:

$$\tilde{\mathbf{r}}_{(i)} = \underset{\mathbf{r}_{(i)}}{\operatorname{argmin}} ||\mathbf{W}_{(i)}\mathbf{\Theta}\mathbf{r}_{(i)}||_1 \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{r}_{(i)}, \tag{3.48}$$

where, $\tilde{\mathbf{r}}_{(i)}$ denotes the solution of i^{th} iteration. Considering $\mathbf{v}_{(i)} = \mathbf{W}_{(i)}\mathbf{\Theta}\mathbf{r}_{(i)}$ leads to

$$\tilde{\mathbf{v}}_{(i)} = \underset{\mathbf{v}}{\operatorname{argmin}} ||\mathbf{v}||_1 \text{ subject to: } \mathbf{y} = \mathbf{A}\mathbf{\Theta}^{-1}\mathbf{W}_{(i)}^{-1}\mathbf{v}. \tag{3.49}$$

Note that $\mathbf{\Theta}$ and $\mathbf{W}_{(i)}$ are diagonal matrices and hence, their inverses can be computed inexpensively by simply computing the inverse of the diagonal elements, e.g. $\mathbf{W}_{(i)}^{-1} = diag(1./|\mathbf{\Theta}\mathbf{r}_{(i)}|^{p-1})$, where '.' represents the element-wise operation on the vector and $diag(.)$ represents the diagonalization operator. we note from (3.46) that if any diagonal entry in $(i-1)^{th}$ iteration estimate of $\mathbf{\Theta}\mathbf{r}$ approaches to zero, the corresponding entry in $\mathbf{W}_{(i)}^{-1}$ approaches to infinity. Hence, we introduce ϵ close to zero in (3.46) as below:

$$\mathbf{W}_{(i)} = diag(|\mathbf{\Theta}\mathbf{r}_{(i-1)}|^{p-1} + \epsilon). \tag{3.50}$$

Algorithm 1: Algorithm to solve weighted l_p minimization problem of (3.48)

```
Input  : Compressed measurements ($\mathbf{y}$) and Sensing matrix ($\mathbf{A}$), weight
         matrix $\mathbf{\Theta}$, maxIter, $\epsilon$, tol
Output: Reconstructed signal ($\tilde{\mathbf{x}}$)
Initialization: initialize $\tilde{\mathbf{r}}_{(0)}$ to be a zero vector, $i = 1$ and $\delta = 1$.
while {$i <=$ maxIter or $\delta >$ tol} do
    1. Form matrix $\mathbf{W}_{(i)} = diag(|\mathbf{\Theta r}_{(i-1)}|^{p-1} + \epsilon)$,
       its inverse, $\mathbf{W}_{(i)}^{-1} = diag(1./(|\mathbf{\Theta r}_{(i-1)}|^{p-1} + \epsilon))$
    2. Solve:
       $\tilde{\mathbf{v}_{(i)}} = \min_{\mathbf{v}} ||\mathbf{v}||_1$ subject to: $\mathbf{y} = \mathbf{A\Theta}^{-1}\mathbf{W}_{(i)}^{-1}\mathbf{v}$.
    3. Compute $\tilde{\mathbf{r}_{(i)}}$ using $\tilde{\mathbf{r}_{(i)}} = \mathbf{\Theta}^{-1}\mathbf{W}_{(i)}^{-1}\tilde{\mathbf{v}_{(i)}}$
    4. Compute $\delta = \frac{||\tilde{\mathbf{r}}_{(i)} - \tilde{\mathbf{r}}_{(i-1)}||}{||\tilde{\mathbf{r}}_{(i-1)}||}$
end
```

With the above $\mathbf{W}_{(i)}$ matrix, (3.49) is solved iteratively until the solution converges or until the maximum number of iterations are reached. The reconstructed signal is obtained using $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}_{(i)}}$. Note that (3.49) can be solved by any of the existing solvers. We used MATLAB solver SPGL1 [107], [108] to solve the problem. The complete algorithm to solve for non-convex weighted l_p minimization problem is presented in Algorithm 1.

We use WNC minimization in CS based recovery of ECG signals in section 3.2.3. We set maximum number of iterations maxIter= 10, tol = 10^{-4} (tolerance) and $\epsilon = 10^{-4}$ in those experiments.

Please note that the prior knowledge about the signal can be employed in our algorithm for better reconstruction of the signal. For example, if we know about the support of the signal, then weighting can be done on only the known support instead of the whole approximate coefficients.

3.2 Applications

In this section, we present the application of the proposed method of DWTL in denoising and compressive sensing based reconstruction in Section 3.2.1 and in

Table 3.1: Learned wavelet filters for two signals

S.No.	Input Signal	Filter Coefficients
1	Speech-1 Sampling frequency: f_s =11.025 KHz Number of samples = 2712	9/7 Filters
		h_0=[0 -0.0004 0.0007 -0.1225 0.2578 0.7108 0.3126 -0.1593 0.0007 -0.0003] h_1=[0 0.0089 -0.0183 -0.5360 1.0016 -0.4642 0.0167 -0.0086 0 0]
		5/3 Filters
		h_0=[0 -0.1296 0.2380 0.7336 0.2648 -0.1360] h_1=[0 0 0 -0.5445 1.0000 -0.5136]
2	Music-1 Sampling frequency: f_s = 11.025 KHz Number of samples = 10000	9/7 Filters
		h_0=[0 -0.0004 0.0007 -0.1192 0.2414 0.7091 0.3293 -0.1615 0.0008 -0.0004] h_1=[0 0.0090 -0.0178 -0.5569 0.9997 -0.4432 0.0181 -0.0090 0 0]
		5/3 Filters
		h_0=[0 -0.1404 0.2791 0.7183 0.2841 -0.1413] h_1=[0 0 0 -0.5029 1.0000 -0.4973]

section 3.2.2, respectively. Section3.2.3 presents CS of ECG signals using the DWTL method presented in Section 3.1.3 via weighted non-convex minimization proposed in Section3.1.4.

3.2.1 Denoising of 1-D signals

Discrete wavelet transform has been proved to be a powerful tool for signal denoising. In this subsection, we apply the proposed method of DWTL presented in section 3.1.1 for the denoising of speech and music signals. Music signals are picked randomly from [110]. One stage and two stages of 2-tap predict and update filters are learned (using the modular property of lifting) with and without linear phase conditions. The resulting wavelet system corresponds to the synthesis filters of lengths 5/3 (highpass/lowpass) and 9/7 (highpass/lowpass) with one and two stages, respectively. Analysis side filters for one speech and one music signal are presented in Table 3.1. Since the resulting wavelet system are signal-matched biorthogonal 5/3 and 9/7, it is appropriate to compare results with the standard biorthogonal 5/3 and 9/7 wavelets.

Since this method of Section 3.1.1 requires the original signal, it cannot be applied directly in denoising application where noisy version of the signal is present instead of the original signal. Hence, we modify the method of 3.1.1 as presented below.

We learn the analysis lowpass filter in the update stage considering that most of the signal energy will move to the low frequency branch of the filterbank. This makes the proposed scheme of matched wavelet learning suitable for signals rich in low frequencies. However, noisy signals will be rich in high frequency content. Thus, we use an accumulator, a discrete time counterpart of an integrator, on the given noisy signal $x(n)$ as below:

$$y[n] = \sum_{k=0}^{n} x[k] \tag{3.51}$$

where $x[n] = 0$, when $n < 0$. This step converts the input noisy signal $x[n]$ into dominantly lowpass signal $y[n]$. Resulting dominantly lowpass signal $y[n]$ is fed as input to the above method of DWTL of section 3.1.1 and wavelet filterbank is learned matched to this signal $y[n]$ [30]. After denoising as discussed in the next paragraph, we apply first difference on the successive samples of the output signal $s[n]$ to obtain the actual denoised signal $\hat{x}[n]$ according to the following relation:

$$\hat{x}[n] = s[n] - s[n-1] \tag{3.52}$$

We add white Gaussian noise at 5dB SNR per sample. After learning DWT, soft-thresholding is applied on the wavelet coefficients. We have applied 3-level wavelet decomposition for denoising. All the subband coefficients are thresholded using *Bayes Shrink* threshold strategy [6] except coarsest approximation coefficients. Table 3.2 shows the comparison of the denoised results of speech and music signals between learned DWT and standard biorthogonal wavelets. Peak signal to noise ratio (PSNR) is used as the performance measure for denoising. Each experiment is performed with 30 runs and the results shown here are the average over all runs.

Discussion: From Table 3.2, the following observations are in order:

- Signal-matched learned DWT without LP (linear phase) constraint gives

Table 3.2: Results of Denoising

Signal	PSNR (in dB)						
	Noisy	DWT 9/7LP	DWT 9/7	Standard 9/7 LP	DWT 5/3LP	DWT 5/3	Standard 5/3 LP
Speech-1	12.28	13.14	14.68	14.61	13.37	12.98	**14.82**
Speech-2	11.85	12.24	**14.08**	11.84	11.85	12.11	12.76
Music-1	12.39	13.93	**16.06**	14.16	13.73	13.98	14.65
Music-2	11.81	12.44	**14.26**	11.94	12.34	12.48	12.98
Music-3	11.87	13.78	**15.42**	13.43	13.22	13.41	13.86

better results of denoising compared to the one with the LP constraint for both the 5/3 and the 9/7 wavelet.

- Signal-matched 9/7 wavelet without LP constraint is working best on most of the signals considered. The results are better compared to the standard 9/7 and 5/3 wavelets.

The above results are obvious because with linear phase condition, we are imposing constraints on the learned wavelet and hence, it may deviate from the exact signal-matched wavelet. However, the LP matched wavelet learning may be useful in applications requiring linear phase conditions. This is to note that denoising results using the standard 9/7 and 5/3 wavelets with the method of an accumulator and first difference are observed to be inferior. Hence, these have not been included in the text.

3.2.2 Compressive Sensing of 1-D signals

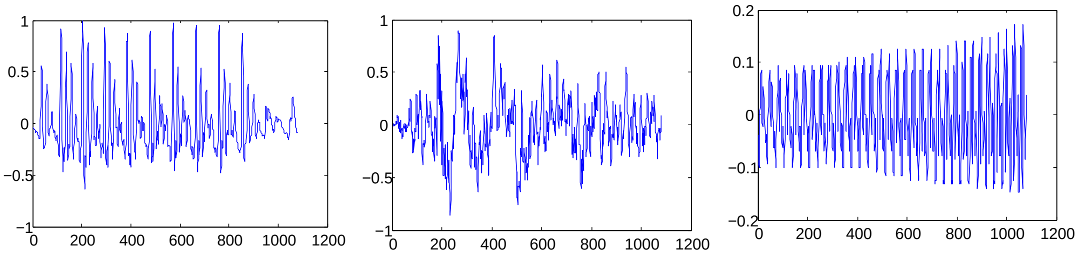
We present CS based reconstruction using the learned DWT as the sparsifying transform in this section. We use the method presented in section 3.1.2 for DWTL.

The method is applied on ECG, speech, and music signals. Sampling ratios ranging from 10% to 90% are considered in the experiments. Sampling ratio of 100% is not included as it does not imply compressively sensed signal. Experiments are performed for 30 independent trials for every sampling ratio. We compare reconstruction results of the learned wavelet with those using standard wavelets. Reconstruction performance is measured in terms of PSNR. Results are presented

Table 3.3: Reconstruction results of music and speech signals with standard and learned wavelets

Sampling	Average PSNR (in dB)								
Ratio	Speech 1			Music 1			Music 2		
(in %)	Bi5/3	db4	WTL	Bi5/3	db4	WTL	Bi5/3	db4	WTL
90	35.44	39.01	**40.8**	28.12	28.85	**29.94**	23.55	25.66	**29.16**
80	29.45	32.62	**35.92**	24.07	24.95	**26.5**	18.29	20.88	**24.76**
70	24.82	27.9	**32.27**	21.37	22.03	**24.03**	14.63	16.78	**20.91**
60	20.81	23.51	**28.75**	19.13	19.75	**22.09**	11.77	13.92	**17.58**
50	17.41	20.21	**25.27**	16.96	17.72	**20.42**	9.69	11.64	**14.33**
40	14.31	17.03	**21.69**	14.99	15.81	**18.57**	8.29	9.78	**11.28**
30	12.22	14.27	**17.89**	13.1	13.88	**16.55**	7.35	8.5	**9.32**
20	10.72	11.96	**13.17**	11.28	11.99	**13.86**	6.69	7.49	**7.69**
10	9.789	10.64	**10.69**	9.72	10.4	**10.75**	6.32	6.85	**6.89**

on average PSNR with 30 iterations as mentioned above.



(a)

(b)

(c)

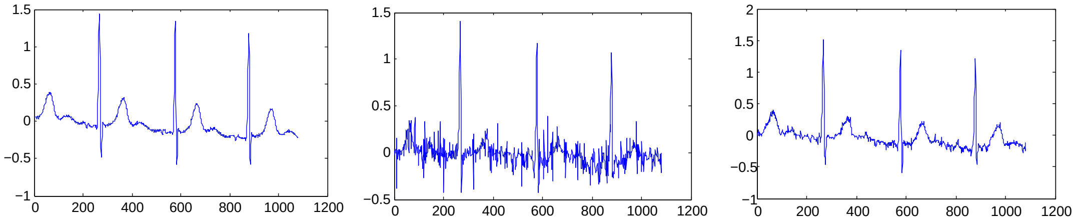
Figure 3.7: (a). Speech signal (b). Music signal 1 (c). Music signal 2

Table 3.3 presents signal reconstruction results on one speech and two music signals. The original signals are shown in Figure 3.7. We compare the reconstruction performance of the learned DWT with standard biorthogonal 5/3 (bior2.2) and orthogonal Daubechies-4 wavelets, shown as Bi5/3 and db4, respectively, in the table. From Table-3.3, we observe that reconstruction results with the learned wavelet outperform standard wavelets based signal recovery at every sampling ratio.

In particular, we observe that performance with the learned wavelet is very good at sampling rations ranging from 20% to 90%, while we obtain improved but comparable results at 10%. Since we are learning matched wavelet in the second stage using the coarser reconstructed signal using standard wavelet, it is

Table 3.4: Reconstruction results on ECG signal with standard and learned (from full and compressively sensed signal) wavelet

Sampling Ratio (in %)	Average PSNR (in dB)				
	Bi5/3	db4	coiflet4	WTL using original (Full) signal	WTL with Proposed method using compressively sensed signal
90	50.27	50.52	**51.69**	**51.83**	51.39
80	46.19	46.58	47.23	**48.33**	**47.97**
70	43.38	43.73	44.26	**45.89**	**45.58**
60	39.7	40.73	41.28	**43.71**	**43.4**
50	35.11	36.6	36.85	**41.92**	**41.77**
40	27.58	28.65	30.37	**39.66**	**39.74**
30	22.32	23.26	23.62	**33.95**	**33.75**
20	19.22	19.71	19.9	**23.31**	**22.9**
10	16.55	**17.70**	17.51	16.52	16.3



(a) (b) (c)

Figure 3.8: (a). Original ECG Signal (b). ECG signal reconstructed with bior2.2 (c). ECG signal reconstructed with proposed method

possible that at such lower sampling ratios, not much can be recovered in the second refinement stage from this reconstructed signal.

It will also be interesting to look at the reconstruction results if wavelet transform is learned from the original (full) signal. In Table 3.4, we present reconstruction results on ECG signal using standard wavelets bior2.2, db4 and coiflet4, the proposed method, and perfectly matched wavelet (wavelet learned from original signal).

From Table 3.4, we observe that the learned wavelet outperforms existing wavelets including Coiflet-4 that is known to match best, among all standard wavelets, with the PQRST wave of the ECG signal. In addition, we observe that the performance of wavelet learned from sparsely observed data is as good as wavelet matched to the fully available input signal. Both these results show the efficacy of the proposed method.

In order to ascertain the quality of the reconstructed signal visually, we present original and reconstructed results in Figure 3.8. We consider 30% compressively sensed samples of the original signal. Figure 3.8a shows original ECG signal. Figure 3.8b shows the signal reconstructed using bior2.2 wavelet (5/3 wavelet) and Figure 3.8c shows the signal reconstructed with the learned wavelet. Indeed, reconstructed signal quality is better with the learned wavelet compared to that reconstructed using bior2.2 wavelet.

3.2.3 CS of ECG signals via weighted non-convex minimization

In this subsection, we present CS based reconstruction of ECG signals via weighted non-convex minimization proposed in Section3.1.4 using the learned DWT as the sparsifying transform. The method proposed in Section3.1.3 is used for DWTL. Also, various experiments with different sensing matrices, optimization method, and wavelet transform are presented.

3.2.3.1 Dataset

We considered MIT-BIH Arrhythmia ECG dataset [109] most commonly used by the ECG community. This data-set is captured via Holter 2-channel recorder by Beth Israel Hospital Arrhythmia Laboratory between 1975 and 1979 and consists of 24 hour ECG recordings of 47 subjects. Of these, a total of 48 number of half an hour duration recordings were uploaded in the database. Here, 23 number of recordings were from a mixed population of inpatients (about 60%) and outpatients (about 40%) and 25 number of recordings were of clinically significant arrhythmia subjects.

The ECG signal was recorded for modified limb lead II (MLII) channel and modified lead V1 (sometimes V2, V4 or V5) channel. Original analog recordings of the ECG signals were made by using Del Mar Avionics model 445 two-channel recorders. The obtained analog signal was sampled at the sampling frequency of

360 Hz per channel with 11-bit resolution over 10 mV voltage range.

Results in this subsection are shown over one minute recordings and similar performance is observed on CS based reconstruction for different duration clips owing to the repetitive patterns in ECG signals. Since QRS complexes are more prominent in the MLII lead channel compared to the V1 lead channel [111], we considered only MLII lead signal in all our experiments. ECG signals 100, 105, 107, 109, 112, 115, 117, 118, 119, 124 were considered for demonstrating the performance of the proposed method of DWTL and WNC based ECG signal reconstruction, while ECG signals 100, 102, 105, 107, 112, 115, 117, 118, 124 and 222 were used for offline learning of the DWT (ECGlet) as discussed in Section 3.1.3.

3.2.3.2 Results and Discussion

As described in Section 3.1.3, we from an ensemble of 10 ECG signals (100, 102, 105, 107, 112, 115, 117, 118, 124 and 222) of Arrhythmia dataset (Section 3.2.3.1) and use the ensemble to learn DWT using the method presented in Section 3.1.1. As the wavelet is learned from the class of ECG signal, we call the learned wavelet as ECGlet. We use ECGlet as the sparsifying transform for the CS based reconstruction of ECG signals. We also use WNC method proposed in Section 3.1.4. We carry out wavelet transform learning or ECGlet learning offline using an ensemble of some ECG signals and test in the proposed WNC based CS reconstruction on another set of transmitted ECG signals, showing the utility of the proposed work in real inverse problems.

The complete block-diagram of the proposed methodology is shown in Figure 3.9. Each ECG recording $\mathbf{x}$ is segmented into small length segments $\mathbf{x}_i$ and compressively sensed using sensing matrix $\mathbf{\Phi}$ to obtain compressive measurements $\mathbf{y}_i$. These compressively sensed segments are transmitted to the receiver. Assuming noise-free channel, the receiver reconstructs the original signal $\mathbf{x}$ using the proposed methodology.

We conducted a number of experiments on the ECG signals collected from the Arrhythmia database as discussed in Section 3.2.3.1. First, we present the wavelet filters of ECGlet learned on the ensemble ECG signal of Figure 3.5 using the method presented in Section 3.1.3. Since wavelet transform is generally applied at three-level decomposition, we learned wavelet filters at all the three successive levels

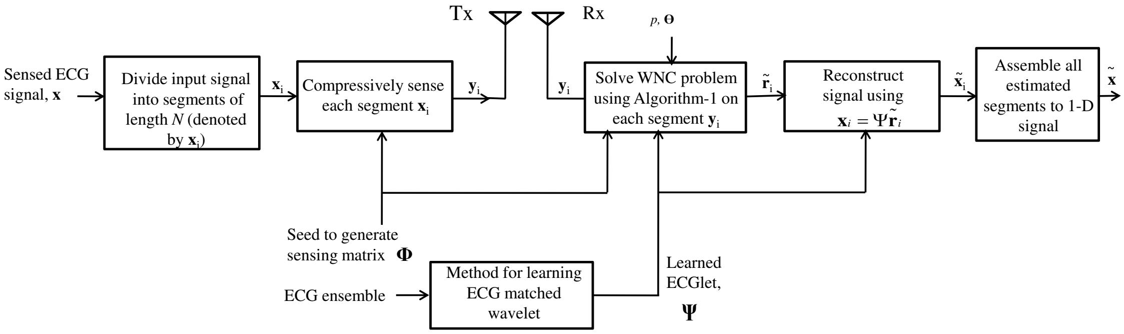

Tx
Rx
p, Θ
Sensed ECG signal, $\mathbf{x}$
Divide input signal into segments of length N (denoted by $\mathbf{x}_i$)
$\mathbf{x}_i$
Compressively sense each segment $\mathbf{x}_i$
$\mathbf{y}_i$
$\mathbf{y}_i$
Solve WNC problem using Algorithm-1 on each segment $\mathbf{y}_i$
$\tilde{\mathbf{r}}_i$
Reconstruct signal using $\mathbf{x}_i = \Psi\tilde{\mathbf{r}}_i$
$\tilde{\mathbf{x}}_i$
Assemble all estimated segments to 1-D signal
$\tilde{\mathbf{x}}$
Seed to generate sensing matrix $\mathbf{\Phi}$
ECG ensemble
Method for learning ECG matched wavelet
Learned ECGlet, $\mathbf{\Psi}$

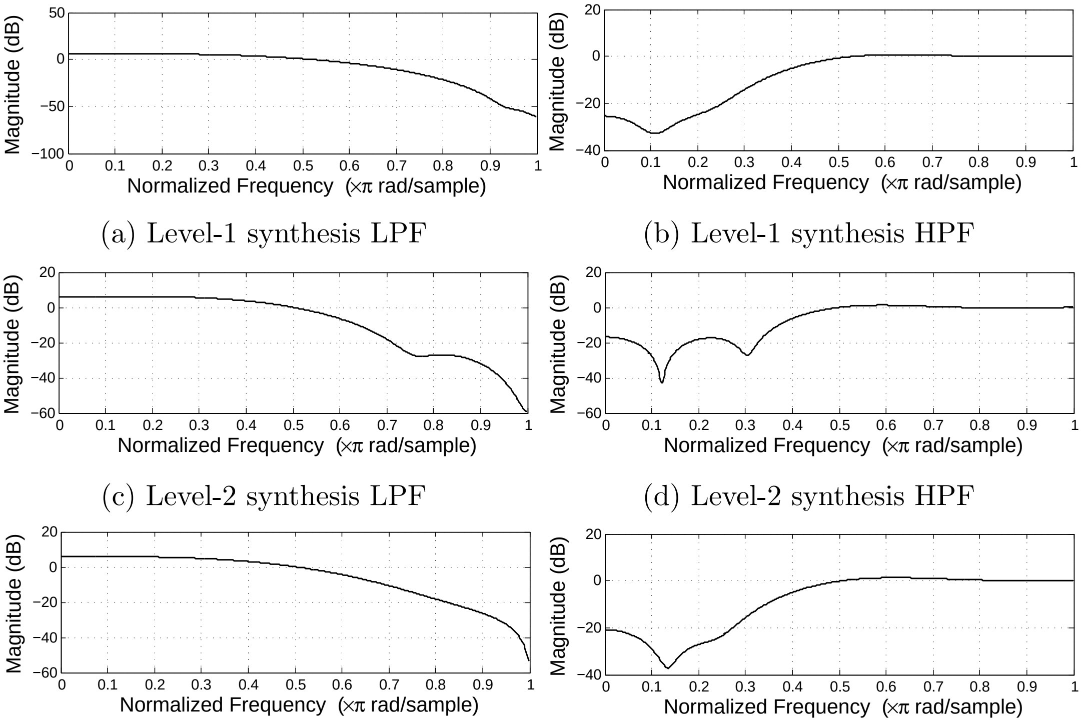
Figure 3.9: Block diagram of the proposed WNC-ECGlet based reconstruction of compressively transmitted ECG signals

Table 3.5: Synthesis filters along with predict/update filters at all 3 levels of wavelet decomposition

Level	1	2	3
$t[n]$	t_1=[-0.1333 0.7369 0.4135 -0.0181]	t_2=[-0.0918 0.5593 0.6371 -0.1057]	t_3=[-0.0455 0.4656 0.6338 -0.0519]
$s[n]$	s_1=[-0.1136 0.4135 0.1892 -0.0156]	s_2=[-0.0814 0.2785 0.3351 -0.1040]	s_3=[-0.0577 0.2366 0.3636 -0.0886]
$f_0[n]$	f_{01}=[0 -0.0181 0 0.4135 1.0000 0.7369 0 -0.1333]	f_{02}=[0 -0.1057 0 0.6371 1.0000 0.5593 0 -0.0918]	f_{03}=[0 -0.0519 0 0.6338 1.0000 0.4656 0 -0.0455]
$f_1[n]$	f_{11}=[0 -0.0020 0 0.0544 0.1136 -0.0839 -0.4135 0.6017 -0.1892 -0.0779 0.0156 0.0367 0 -0.0021]	f_{12}=[0 -0.0086 0 0.0813 0.0814 -0.0965 -0.2785 0.6123 -0.3351 -0.0956 0.1040 0.0889 0 -0.0095]	f_{13}=[0 -0.0030 0 0.0489 0.0577 -0.1042 -0.2366 0.6522 -0.3636 -0.1024 0.0886 0.0578 0 -0.0040]

of wavelet decomposition. Table 3.5 shows 4-tap predict/update filters learned with the proposed method along with the synthesis filters learned at all the three levels. Figure 3.10 shows the frequency response of the corresponding synthesis filters.

We carried out a number of experiments to understand the significance of l_p norm, weighted l_p norm (or weighted non-convex minimization), and ECGlet in the context of CS-based reconstruction of ECG signals. Reconstruction accuracy is computed in terms of SNR (Signal to Noise ratio, in dB) and PRD (Percentage Root Mean Square Difference, in %) presented in Section 2.5. In [112], as assessed by the specialists, a PRD value of less than 9% is labeled as 'good' quality reconstruction and a PRD value of less than 2% is labeled as 'very good' quality reconstruction of ECG signals. The corresponding SNR values for 'good' and 'very



(a) Level-1 synthesis LPF

(b) Level-1 synthesis HPF

(c) Level-2 synthesis LPF

(d) Level-2 synthesis HPF

(e) Level-3 synthesis LPF

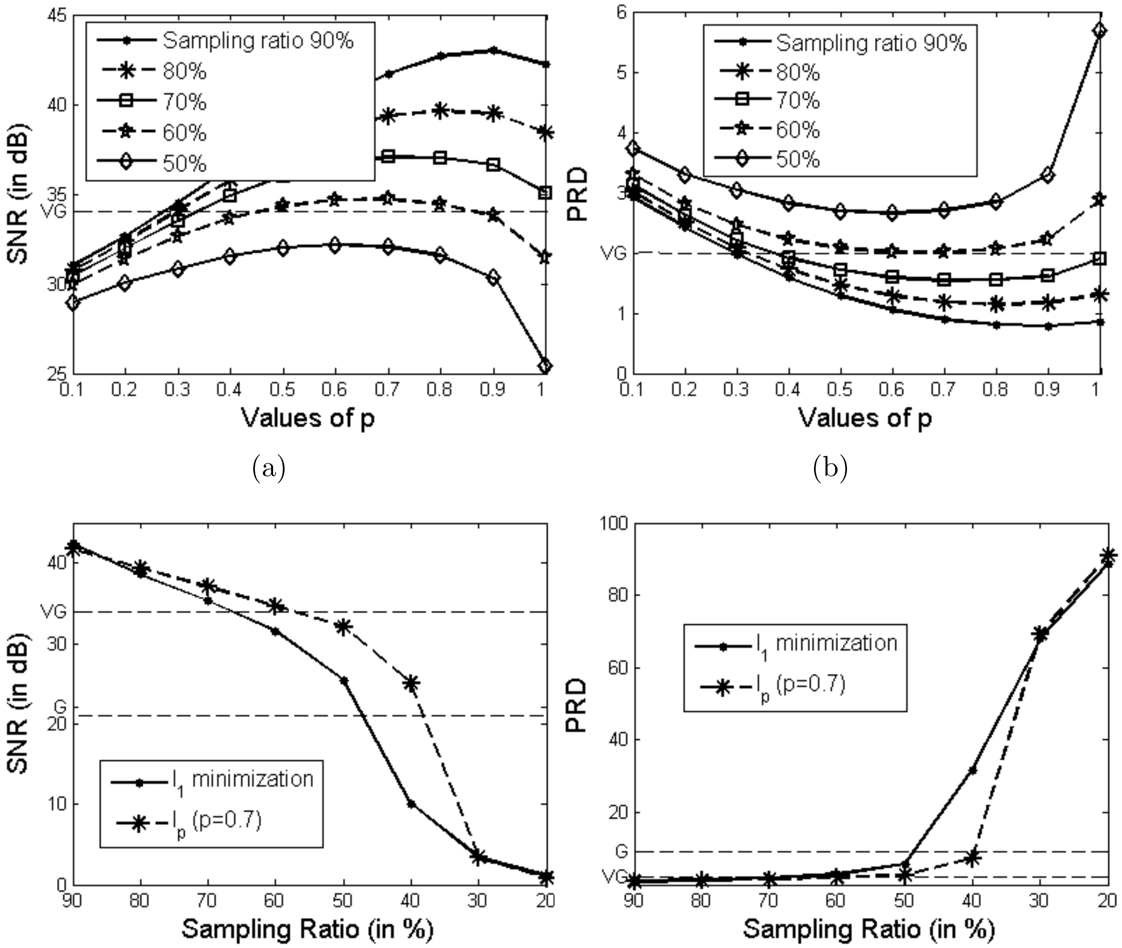
(f) Level-3 synthesis HPF

Figure 3.10: Frequency response of synthesis filters of learned wavelet transform

good' quality reconstruction of ECG signals are 21 dB and 34 dB, respectively. We denote good and very good quality reconstruction in every result with symbols 'G' and 'VG', respectively. In all the experiments, SNR and PRD values imply mean SNR and mean PRD values calculated over the reconstructed ECG signals of records 100, 105, 107, 109, 112, 115, 117, 118, 119, 124 of the dataset. Gaussian sensing matrix has been considered in CS, unless otherwise specified, with sampling ratios ranging from 90% to 20%, where expression for sampling ratio (S) is given in Section 2.5. Experiments at each sampling ratio are carried out for 10 iterations.

Experiment-1: CS-based reconstruction of ECG signal using non-convex l_p minimization with 'db4' wavelet

In this experiment, we evaluate the performance of CS-based ECG reconstruction using l_p minimization of (3.43) with orthogonal 'db4' wavelet used as the spar-



(a) (b)

(c) (d)

Figure 3.11: Performance of l_p minimization in CS-based ECG reconstruction with 3-level 'db4' wavelet; (a) SNR and (b) PRD, for a range of p-values at different sampling ratios. (c) SNR and (d) PRD, for a range of sampling ratios at $p = 0.7$ and $p = 1$.

sifying transform (implemented with 3-level wavelet decomposition). Although $l_p(0 \leq p < 1)$ minimization may lead to better reconstruction accuracy compared to l_1 minimization, this may not be true for all values of p. Thus, we compare the reconstruction performance for values of p ranging from 0 to 1 with an interval of 0.1 at sampling ratios of 50% to 90% with an interval of 10%. Results are shown in Figure 3.11a-3.11b.

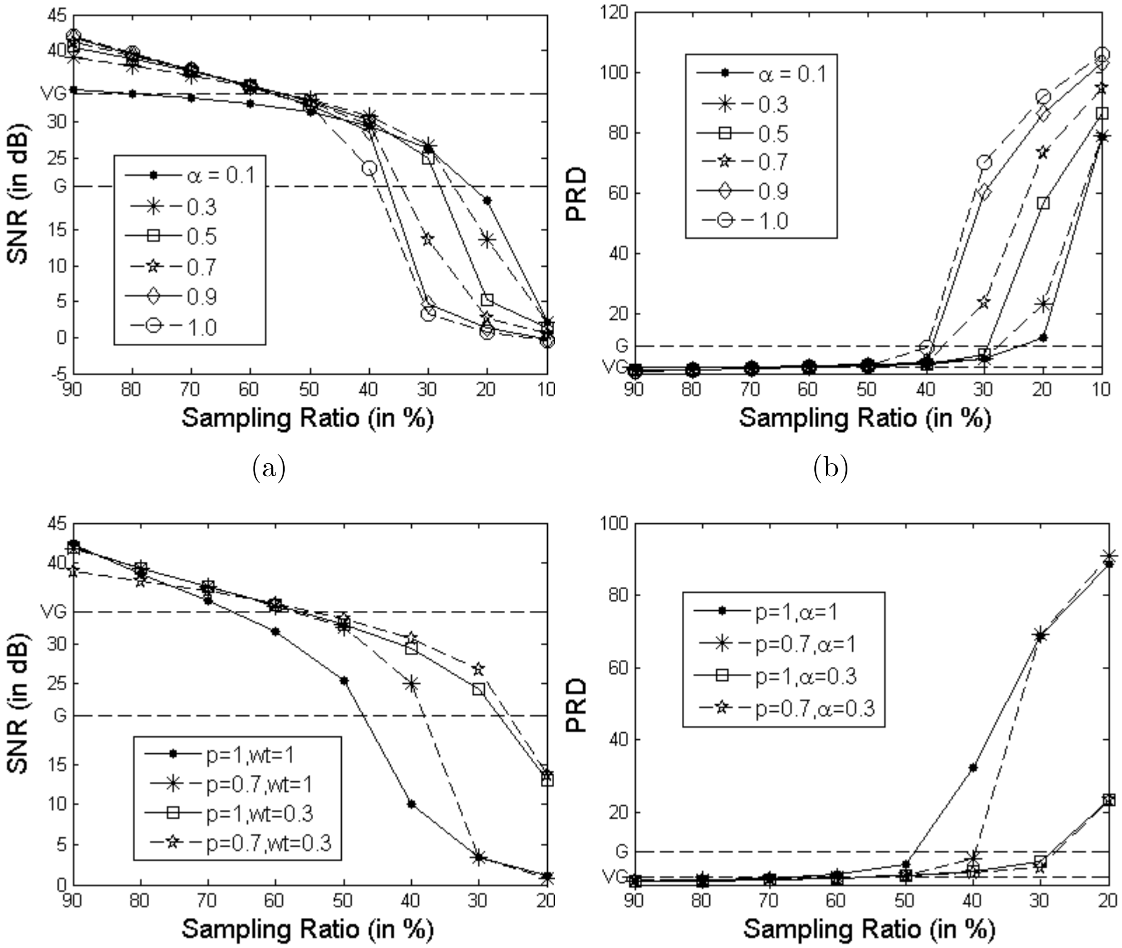
Figure 3.11a and Figure 3.11b show SNR and PRD performance curves for different values of p at different sampling ratios. From these graphs, it is noted that the reconstruction accuracy is maximum and nearly constant for p between 0.6 and 0.8 at all sampling ratios considered. Hence, we consider $p = 0.7$ in l_p minimization with Gaussian sensing matrix in rest of the experiments. The comparison of CS-based reconstruction with l_p minimization ($p = 0.7$) and l_1 minimization is presented in Figure 3.11c and 3.11d. Results are reported with sampling ratios ranging from 90% to 20%. It is noted that l_p minimization provides better accuracy at sampling ratios from 80% to 30% with a fairly high improvement at 50% to 40% sampling ratio.

Although the optimal value of p is found experimentally in this work, there exist some works in the literature, where optimal value is obtained from the statistics of the data [113, 114]. We would like to explore this statistical based approach for ECG signals in the near future.

Experiment-2: CS-based ECG reconstruction using the proposed weighted non-convex l_p (WNC) minimization

We evaluated the CS-based reconstruction performance of the proposed weighted non-convex l_p minimization (WNC) with 3-level 'db4' wavelet at different sampling ratios with $\alpha_2 = 1$ and $\alpha_1 = \alpha$ varying from 0.1 to 0.9 with an interval of 0.2. Better reconstruction accuracy is observed in Figure 3.12 at higher sampling ratios with the greater value of weight α. At higher sampling ratio, least accuracy is observed at $\alpha = 0.1$. This gap in accuracy is decreased as the sampling ratio is decreased with a flip in performance after 50% sampling ratio, where performance decreases rapidly, while better performance is observed at lower values of weight α. At 20% sampling ratio, best accuracy is observed at $\alpha = 0.1$.

These curves indicate that as the sampling ratio is decreased, i.e., less number of samples are picked up, all approximate wavelet coefficients become important and lesser sparsity penalty is required on the approximate wavelet coefficients. After 50% sampling ratio is crossed, the performance crosses 'very good' quality and remains almost constant and high for all values of α except for $\alpha = 0.1$ indicating that sparseness is required to be imposed. The value of α becomes insignificant at 10% sampling ratio with all weight values providing the poor performance of nearly 0 dB SNR. This indicates that the number of samples picked up are too insufficient



(a) (b)

(c) (d)

Figure 3.12: Effect of weight α in WNC l_p minimization in CS-based ECG reconstruction with 3-level 'db4' wavelet; (a) SNR and (b) PRD, with various values of α at different sampling ratios; (c) SNR and (d) PRD comparing l_1 with $\alpha = 1$ to l_p with $p = 0.7$ and $\alpha = 0.3$.

to reconstruct the good quality ECG signal and even the proposed WNC fails to reconstruct ECG signals at 10% sampling ratio. Using these curves, we decide to choose $\alpha = 0.3$ because it provides fairly good and consistent performance at all sampling ratios.

Next, we compared the performance of WNC with l_1 (conventional CS) at all sampling ratios in Figure 3.12c-3.12d. It is noted that WNC performs best and provides 'good' quality reconstruction at all sampling ratios above 25%. At 20% sampling ratio, the conventional CS fails completely providing an SNR of approx.

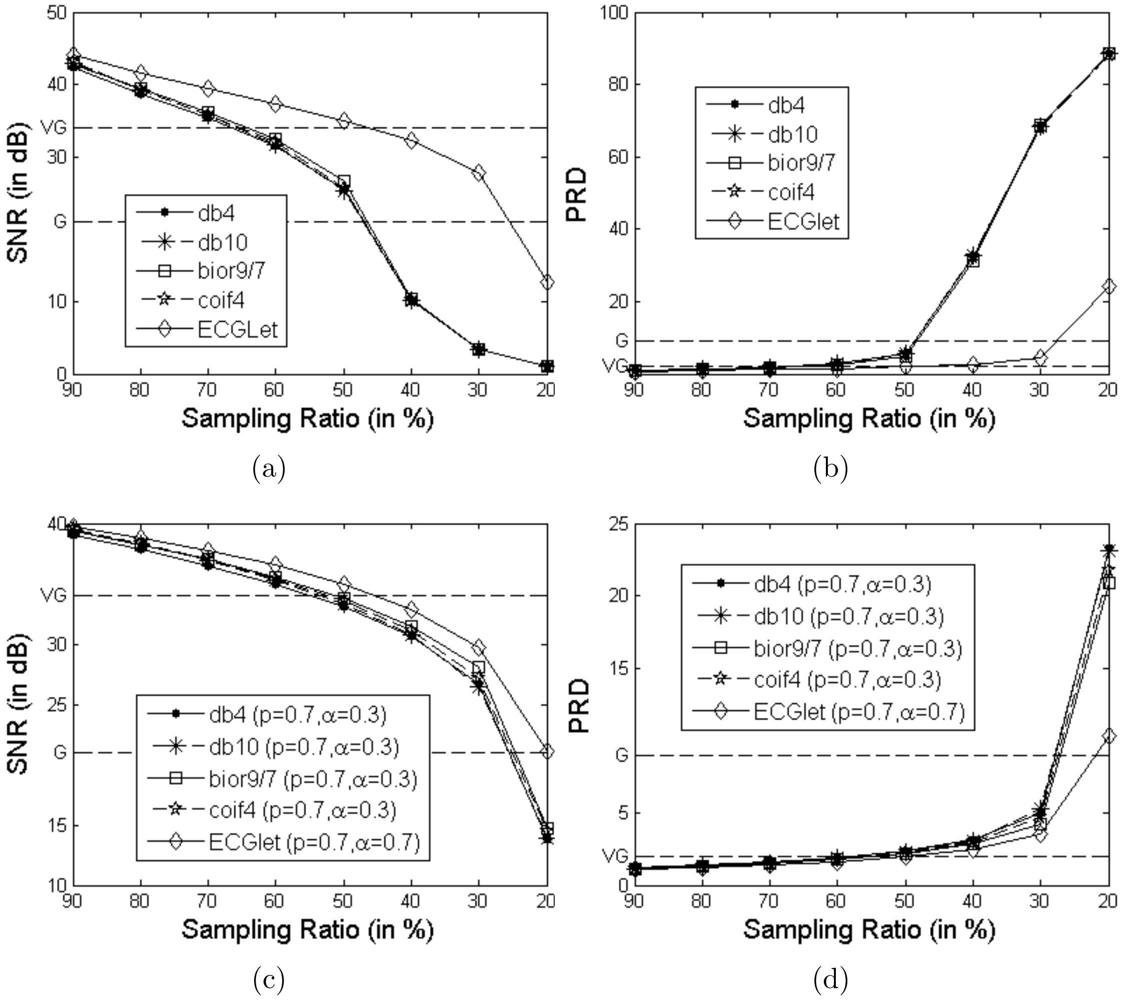


Figure 3.13: Effect of different wavelets in CS based ECG reconstruction; (a) SNR and (b) PRD, with various wavelets using conventional CS ($p = \alpha = 1$); (c) SNR and (d) PRD, with various wavelets using WNC ($p = 0.7$ and $\alpha = 0.3$) at different sampling ratios.

2 dB, while WNC provides an SNR of 15 dB that is about a 13 dB higher than conventional CS indicating superior performance of WNC method.

Experiment-3: Performance comparison of ECGlet with existing standard wavelets with conventional CS and with WNC

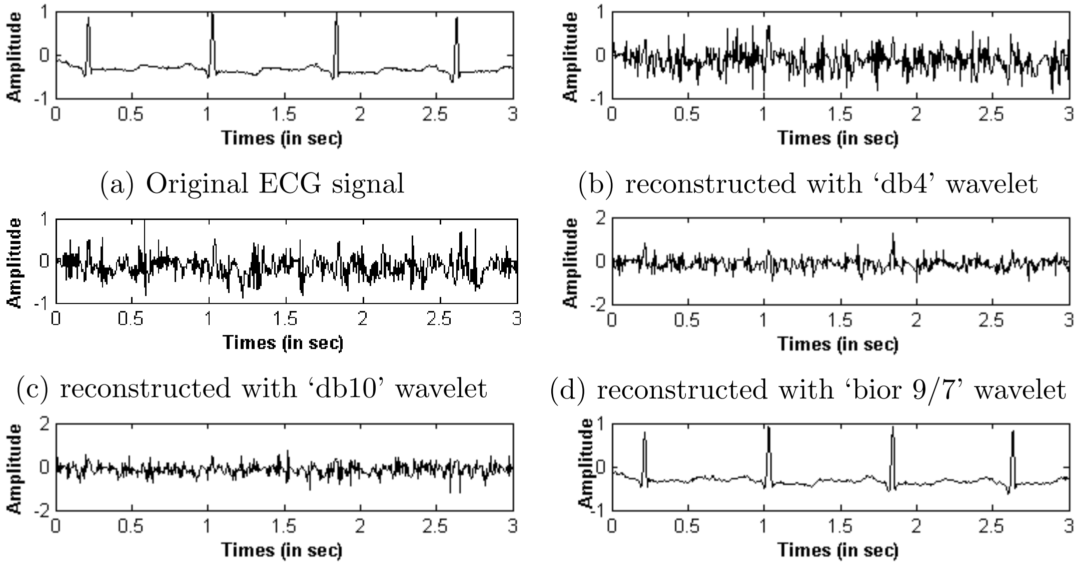
Both the previous experiments tested WNC with 'db4' wavelet. Next, we evaluated the performance of ECGlet as the sparsifying transform vis-à-vis existing standard wavelets in conventional CS ($p = \alpha = 1$). Four standard wavelets are considered: orthogonal 'db4' and 'db10' wavelet, bi-orthogonal 'bior 9/7' wavelet, and orthogonal coiflet ('coif4') wavelet. Coiflet-4 is chosen because its shape matches the QRS shape of the ECG signals.

Conventional CS-based reconstruction results are presented in Figure 3.13(a) and 3.13(b). It is observed that all standard wavelet perform almost similarly, while ECGlet shows marked improved performance with approx. 24 dB higher SNR at 30% sampling ratio compared to standard wavelets. While ECGlet reconstructs signals with an SNR of around 12 dB at 20% sampling ratio, existing wavelets fail to reconstruct the signals with an SNR of approx. 2 dB.

The performance with ECGlet remains 'very good' above 50% sampling ratio and is 'good' or above beyond 25% sampling ratio. On the other hand, standard wavelets cross 'very good' performance at 65% sampling ratio and 'good' performance near 50% sampling ratio. This shows that ECGlet is indeed matched to ECG signals and works as a good sparsifying basis. The performance with ECGlet drops very rapidly after 30% sampling ratio implying that there is not enough information in the measurements to recover ECG signals.

Qualitative reconstruction results are shown over a 3-second ECG segment of ECG signal record-100 in Figure 3.14 at 30% sampling ratio. It is observed that while QRS complex is not visible in reconstruction with any of the existing wavelets, it is not only visible but looks similar to the original ECG signal with ECGlet reconstruction.

CS-based reconstruction results with WNC are presented in Figure 3.13(c) and 3.13(d) with best performing p and α values. Standard wavelets cover up the performance gap with ECGlet in the WNC method as seen from these figures. However, the performance of ECGlet still excels compared to all standard wavelets. Interestingly, while the WNC method and the ECGlet had individually failed to meet the 'good' quality mark individually at 20% sampling ratio by more than 5 dB, this 'good' quality performance is met when both these are used in conjunction. This shows that both the contributions of the paper ECGlet design and WNC methods contribute significantly to the CS-based ECG reconstruction.



(a) Original ECG signal

(b) reconstructed with ‘db4’ wavelet

(c) reconstructed with ‘db10’ wavelet

(d) reconstructed with ‘bior 9/7’ wavelet

(e) reconstructed with ‘coif4’ wavelet

(f) reconstructed with ECGlet

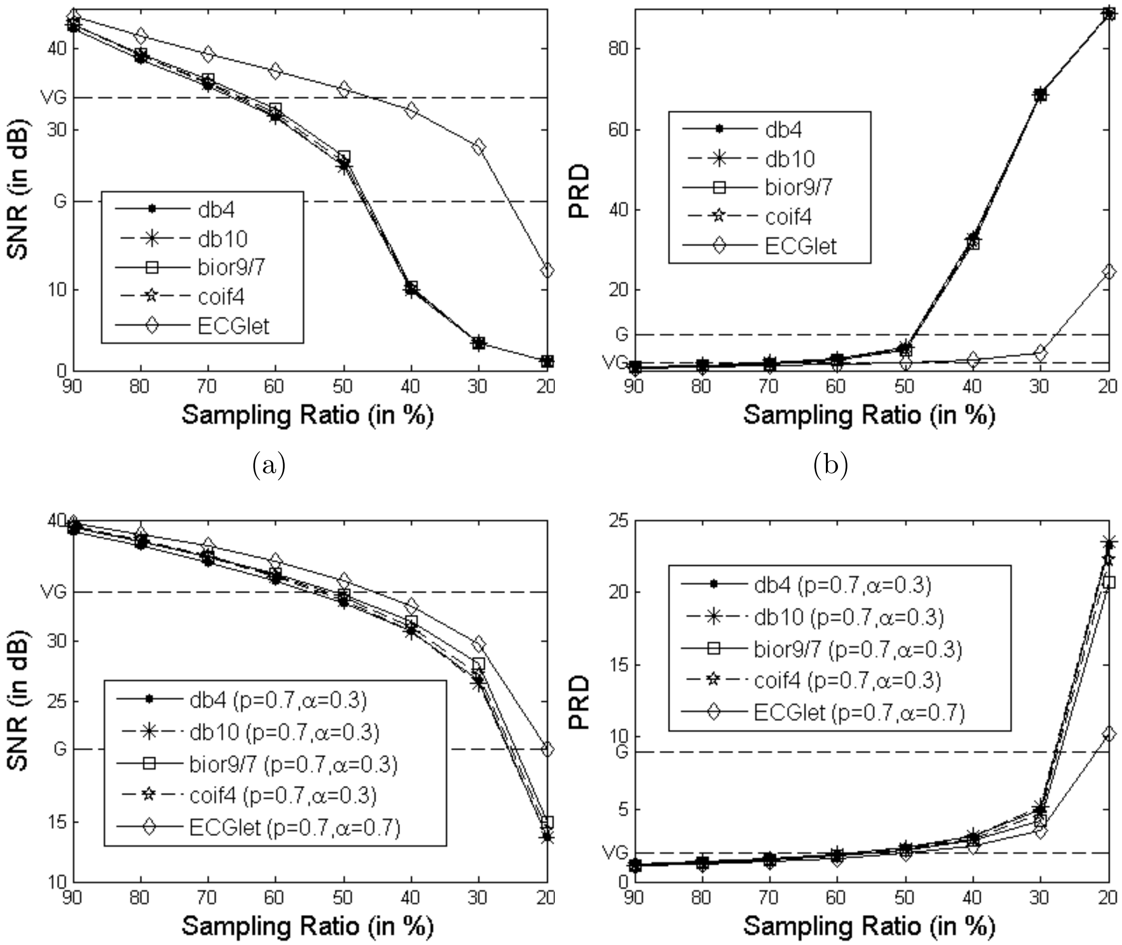
Figure 3.14: 3 seconds segment of ECG signal record no.100 and its reconstruction with various wavelets from 30% measurements

Experiment-4: Comparative performance of Gauusian with Bernoulli, and Sparse Binary Sensing matrices

So far, we focused only on CS-based ECG reconstruction with random Gaussian sensing matrices that are popular in CS applications because they provide ease of theoretical analysis. However, these matrices are not hardware friendly because (1) Gaussian sensing matrices are dense having value at each matrix position, (2) each entry is a floating point number and hence, a large amount of memory is required to store these matrices, and (3) operations with floating point numbers lead to increased complexity of the hardware. Also, most of the microprocessor in WBAN applications are not capable for floating point operations. Hence, Gaussian sensing matrix is not a good choice for CS-based reconstruction of ECG signals.

A good alternative of Gaussian matrices is Bernoulli sensing matrices [88], used widely in CS applications because of their ease in hardware implementation [115]. Entries in such a sensing matrix are taken from Bernoulli distribution with either ± 1 or $\{0, 1\}$ values. Because of binary entries, these matrices are easy to implement on hardware compared to Gaussian sensing matrices. Hence, we apply the proposed WNC method with ECGlet in CS-based ECG reconstruction with

Bernoulli sensing matrix. The accuracy of the reconstructed ECG signal in SNR and PRD are shown with conventional CS method ($p = 1$, $\alpha = 1$) in Figure 3.15a and 3.15b and are shown with WNC method in Figure 3.15c and 3.15d. Results of WNC method with Bernoulli matrix are shown with best performing p and α values for all wavelets including learned ECGlet.



(a) (b)

(c) (d)

Figure 3.15: Accuracy of CS based ECG signal reconstruction with Bernoulli sensing matrix in terms of SNR and PRD with different wavelets. (a) SNR (b) PRD, with conventional CS ($p = 1$ and $\alpha = 1$); (c) SNR (d) PRD, with WNC method.

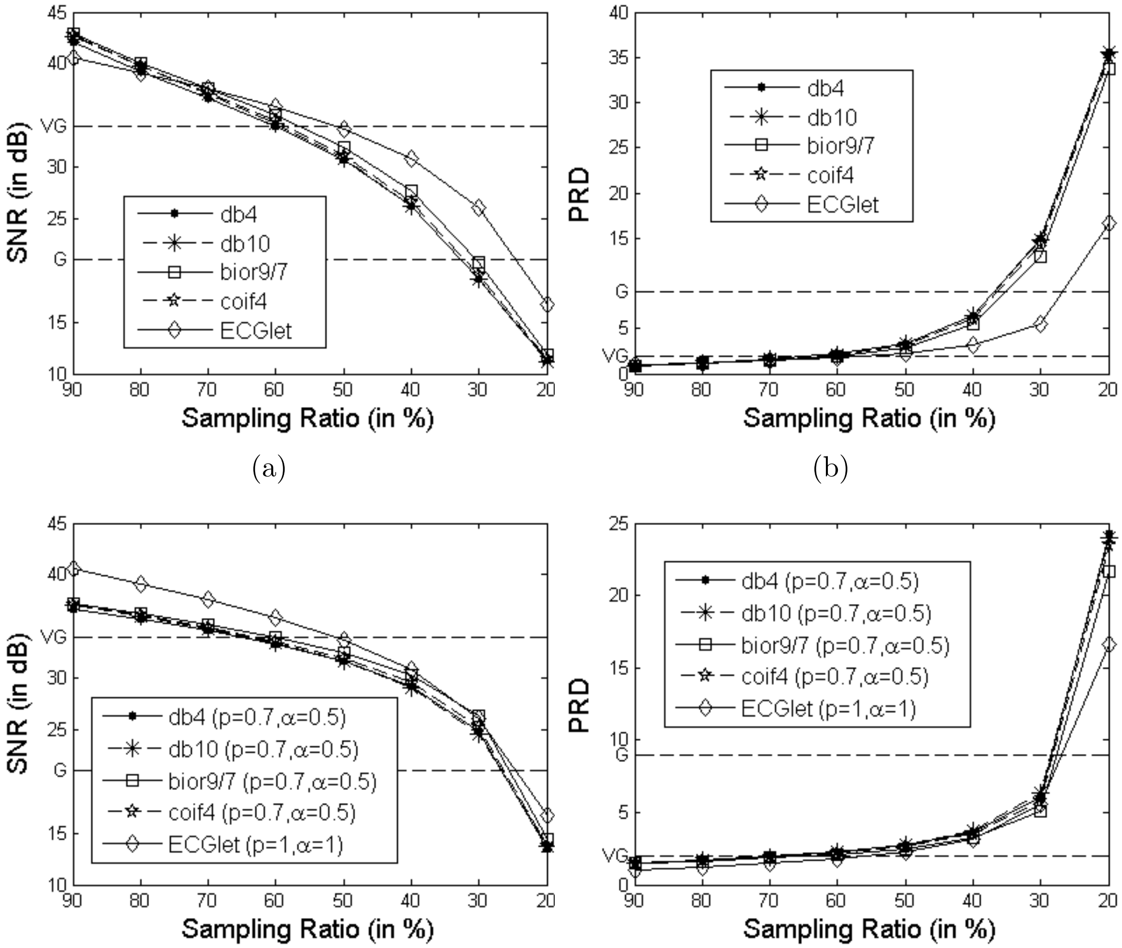
From Figure 3.15, it is noted that ECGlet outperforms the existing transforms in both the conventional CS and WNC based CS. Also, the difference in performance is covered up by standard transforms with WNC method because WNC method increases the reconstruction accuracy of the existing wavelet transform by a good amount as seen in 3.15c and 3.15d. Since the learned ECGlet is already able to reconstruct ECG signals with good quality, the performance gain of WNC over conventional CS is not as high with ECGlet as with existing transforms. This again validates the superior performance of both the contributions. Overall, the performance with Bernoulli sensing matrices is similar to that of with Gaussian sensing matrix.

Another hardware efficient sensing matrix, namely, sparse binary matrix (SBM) is proposed in [47]. SBM matrix contains very less non-zeros as compared to Bernoulli sensing matrix that has 50% non-zero entries (implemented as {0,1}). In SBM, only d element in a column are non-zero (equal to 1). Similar to [47], we chose $d = 12$, implying 12% non-zero entries in the sensing matrix. Performance results are shown with conventional CS and WNC based CS in Figure 3.16. Results of WNC method with SBM matrix are shown with best performing p and α values for all wavelets including learned ECGlet.

From Figure 3.16a and 3.16b, it is observed that the learned ECGlet outperforms all existing wavelets with conventional CS at sampling ratios lower than 70%. At sampling ratio higher than 70%, SNR is more than 35 dB with all wavelets and hence, reconstruction performance is sufficiently good. The reconstruction accuracy of the existing wavelets is enhanced by WNC method at lower sampling ratios below 50% as seen from Figure 3.16(a) versus 3.16(c) and 3.16(b) versus 3.16(d). Further, it is observed that best reconstruction accuracy with learned ECGlet is achieved with $p = 1$ and $\alpha = 1$ indicating that the performance of ECGlet renders WNC minimization irrelevant with SBM matrices.

Next, we compared the reconstruction accuracy with Gaussian, Bernoulli and SBM sensing matrices in one Figure 3.17. While SBM outperformed Gaussian and Bernoulli matrices with 'db4' in conventional CS (Figure13 (a)-(b)), WNC method pushed the performance with Gaussian and Bernoulli above SBM (Figure13 (c)-(d)). In the case of ECGlet, Gaussian and Bernoulli performed above SBM with both conventional CS and WNC CS (Figure13 (e) to (h)). Further, both 'good' and 'very good' performance bars are crossed at lower sampling ratios with ECGlet vis-à-vis standard wavelet and with WNC method vis-à-vis conventional CS.

Clearly, WNC (Figure 13(c)-(d)), ECGlet (Figure 13(e)-(f)) and WNC-ECGlet

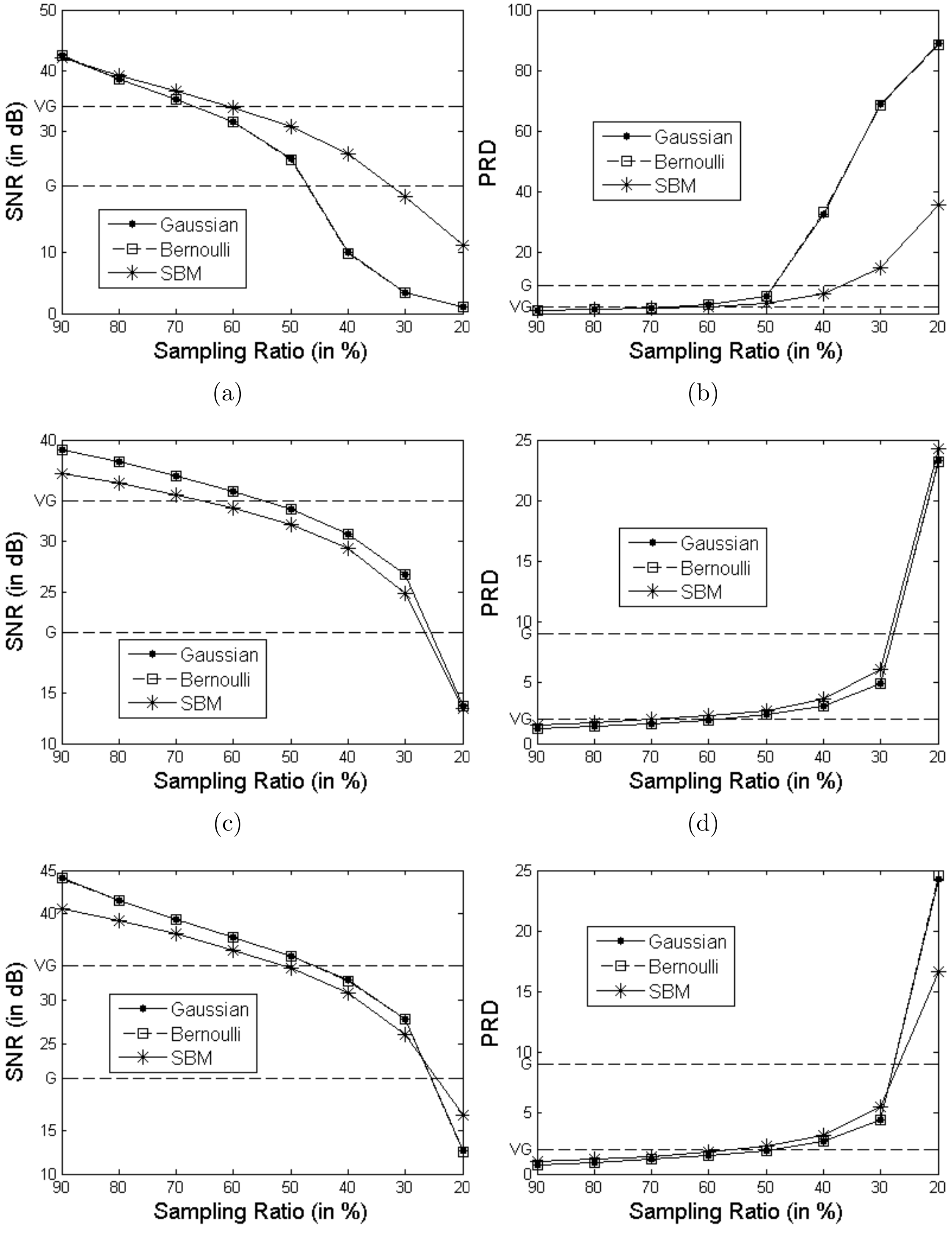


(a) (b)

(c) (d)

Figure 3.16: Accuracy of CS based ECG reconstruction with 'sparse binary matrix' (SBM) in terms of SNR and PRD with different wavelets. (a) SNR (b) PRD, with conventional CS ($p = 1$ and $\alpha = 1$); (c) SNR (d) PRD, with WNC method.

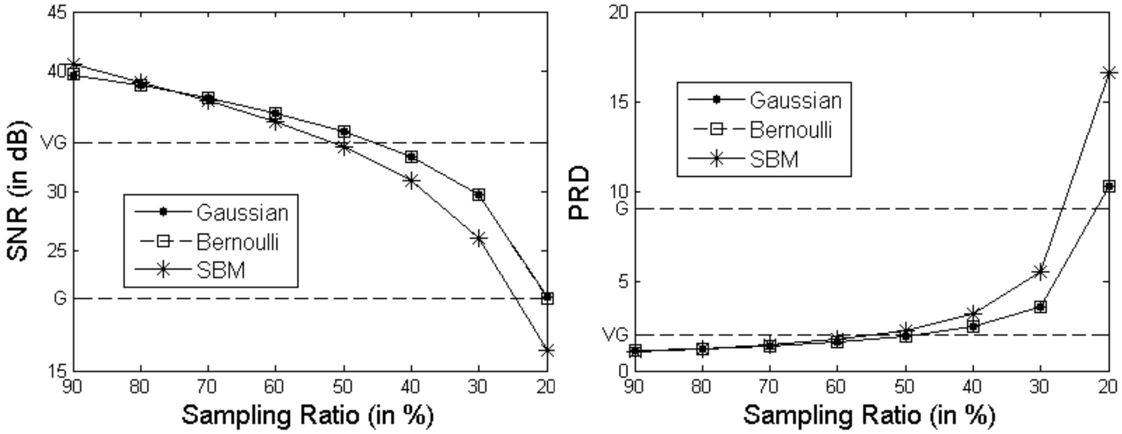
(Figure 13(g)-(h)), all improved the reconstruction performance of conventional CS (Figure 13(a)-(b)) with all the sensing matrices. Also, Gaussian and Bernoulli are better choices of sensing matrix over SBM with WNC, ECGlet and WNC-ECGlet in terms of reconstruction performance, however, the gap in reconstruction performance is small. As the SBM sensing matrix is much easier to implement on hardware in comparison to Gaussian and Bernoulli sensing matrices, one may choose SBM sensing matrix implemented with WNC, ECGlet or WNC-ECGlet over Gaussian and Bernoulli sensing matrices with slight compromise in recon-


SNR (in dB)
50
40
VG
30
G
10
0
Gaussian
Bernoulli
SBM
90
80
70
60
50
40
30
20
Sampling Ratio (in %)
PRD
100
80
60
40
20
G
VG
Gaussian
Bernoulli
SBM
Sampling Ratio (in %)
SNR (in dB)
40
VG
30
25
G
15
10
Gaussian
Bernoulli
SBM
Sampling Ratio (in %)
PRD
25
20
15
10
G
5
VG
0
Gaussian
Bernoulli
SBM
Sampling Ratio (in %)
SNR (in dB)
45
40
VG
30
25
G
15
10
Gaussian
Bernoulli
SBM
Sampling Ratio (in %)
PRD
25
20
15
10
G
5
VG
0
Gaussian
Bernoulli
SBM
Sampling Ratio (in %)

(a) (b)

(c) (d)

(e) (f)



(g) (h)

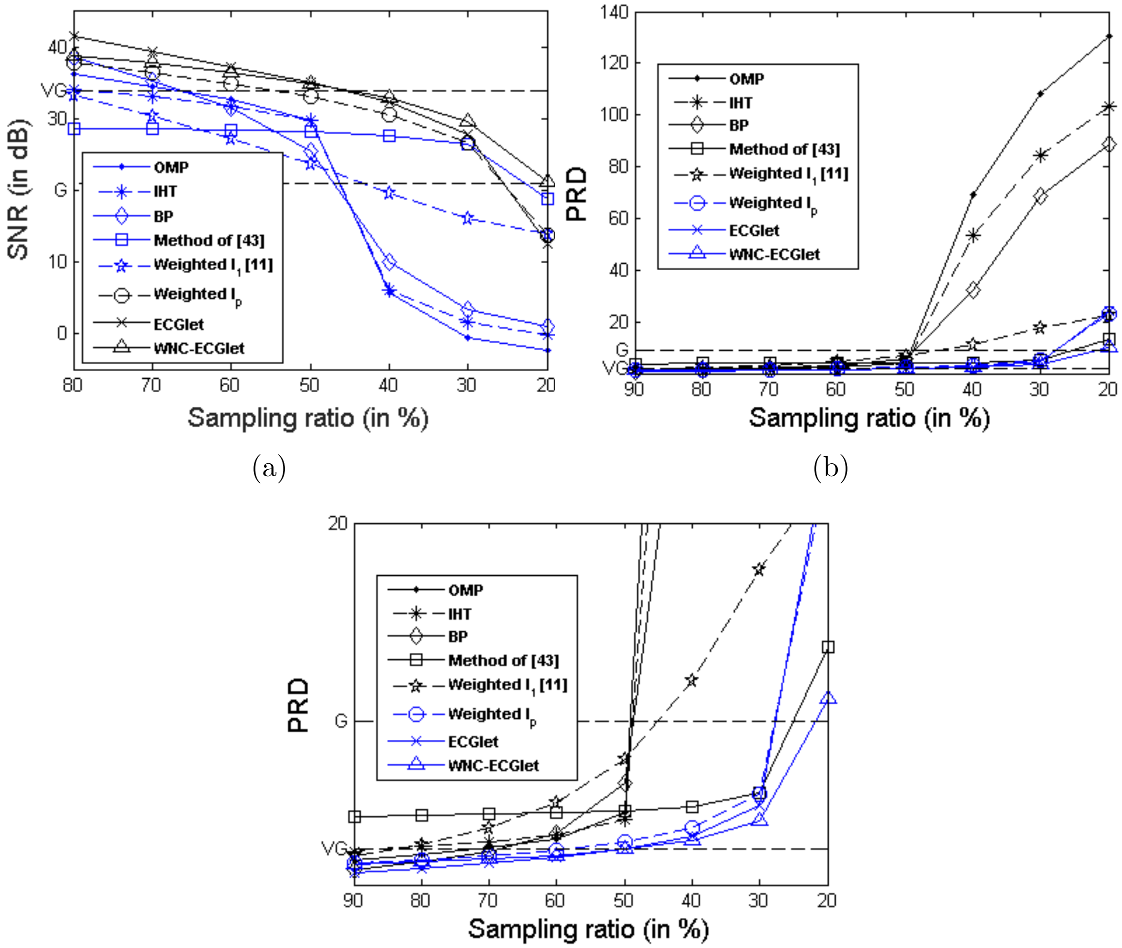
Figure 3.17: Accuracy of CS-based ECG reconstruction with different sensing matrices. Reconstruction accuracy in terms of (a) SNR and (b) PRD with $p = 1$, $\alpha = 1$ and 'db4' wavelet; (c) SNR and (d) PRD with WNC method and 'db4' wavelet; (e) SNR and (f) PRD with $p = 1$, $\alpha = 1$ and ECGlet; (g) SNR and (h) PRD with WNC method and ECGlet.

struction performance.

Experiment-5: Comparative performance of WNC-ECGlet and existing methods

Lastly, we compared WNC-ECGlet with other CS-based ECG reconstruction methods. Some of these methods are based on greedy algorithms such as orthogonal matching pursuit (OMP)[116] and iterative hard thresholding (IHT) [117] that have been widely used in ECG reconstruction. Both these algorithms solve for l_0 minimization. On the other hand, basis pursuit solve for l_1 minimization by posing the problem as a linear program. We compared all these three methods in Figure 3.18 with 3-level 'db4' wavelet as the sparsifying wavelet transform as used generally in the literature.

We also compared WNC-ECGlet with two more methods, proposed sepcifically for ECG reconstruction [118, 49]. In [118], distributed CS is applied by exploiting the common support between adjacent heart beats over a period of six consecutive heartbeats with 'db4' wavelet and random Gaussian sensing matrix, while [49]



(a)

(b)

(c)

Figure 3.18: Comparison of proposed scheme with others in case of Gaussian sensing matrix in terms of: (a) SNR and (b) PRD; (c) zoomed part of results in part (b).

used weighted l_1 minimization for CS-based ECG reconstruction. In [49], sparsity constraint is imposed only on detail coefficients.

WNC-ECGlet and methods of [118, 49] perform better than other existing methods that fail below 50% sampling ratios. Further, WNC-ECGlet performs consistently best at all sampling ratios. It performs above ‘good’ quality mark from 20% to 90% sampling ratio and above ‘very good’ quality mark at 50% sampling ratio and above. The performance of [118] remains almost steady and above ‘good’ quality mark from 30% to 90% but below ‘very good’ quality mark throughout.

The performance of [49] remains above ‘good’ quality mark from 42% and above, but falls consistently from its peak performance at 90% sampling ratio. Overall, WNC-ECGlet is a good choice for CS-based ECG reconstruction.

3.3 Summary

Methods are presented in this chapter to learn DWT for 1-D signals. Particularly, three methods are presented. The first method can be used when the original signal is available. This method can be applied in applications like compression and classification, where one has the access of full original signal. However, this method cannot be used in inverse problems like denoising, compressive sensing, deconvolution etc., where the degraded signal is present instead of the original signal. We present the second method to learn DWT in inverse problems, where coarse estimate of the signal is obtained in the first stage and this coarse estimate is used to learn DWT in the second stage. The third method is presented to learn DWT for a class of signals, which can be used for a class of signals where signal characteristics do not vary much in different signals of the class, e.g. ECG signals and brain MRI. Unlike the second method, this method does not require to compute coarse estimate of the signal. Hence, the third method is more computationally efficient as compared to the second method. Applications of these proposed methods are also presented. Denoising and CS of ECG, music and speech signal is presented. The first method of DWTL is used in denoising application. The second method is used for CS of ECG, music and speech signals. The learned wavelet transform is observed to outperform the existing dyadic wavelet transforms. Also, it was observed that the signal reconstruction performance of the wavelet transform, learned from the degraded signals is slightly inferior than the performance of the wavelet transform learned from the original signal (assuming that the original signal is present). This proves that the proposed method of wavelet transform learning is as good as the wavelet transform learning method from the original signal. Extensive experiments are presented for CS based ECG reconstruction using the third method of DWTL with various sensing matrices via proposed weighted non-convex minimization. The learned wavelet transform along with the proposed weighted non-convex minimization is observed to provide much better ECG signal reconstruction as compared to existing wavelet transforms as well as existing methods.

Chapter 4

DWTL2: Dyadic Wavelet Transform Learning for Images

It has been observed that natural images, in general, are compressible in wavelet domain [103]. Hence, Wavelet transform is extensively applied as separable transforms on images. Also, it was discussed in chapter-3 that wavelet transform does not have unique basis and hence, it is better to learn wavelet basis from the signal itself. Various methods were presented in chapter-3 for dyadic wavelet transform learning (DWTL) for 1-D signals in different situations/applications using the lifting framework. It was also shown that the learned dyadic wavelet transform (DWT) when used as the sparsifying transform in the application of denoising and compressive sensing, performs better than the existing DWTs. In this chapter, we extend the idea to images and present methods to learn separable DWT for them.

Theoretical contributions of our work related to images are presented in Section 4.1. Applications of the proposed method of separable DWTL are presented in Section 4.2. The work of this chapter is summarized in Section 4.3.

4.1 Theoretical Contributions

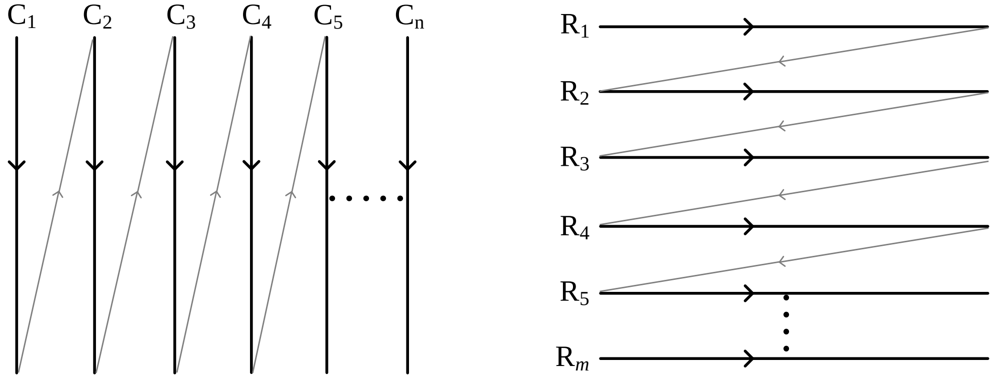
Theoretical contributions of our work related to images are presented in this Section. We present two methods to learn separable DWT in this Section. The first method, presented in Section 4.1.1, can be used in applications where the original image is known. The second method, presented in Section 4.1.2, can be used in inverse problems where the degraded image is present instead of the original image. A new wavelet decomposition method is proposed for images, presented in Section 4.1.3.

4.1.1 DWTL from original image

We propose to learn separable DWT for the images in this Section. Since the proposed work is on separable DWT, we require to learn wavelet for row and column directions separately. Thus, before proceeding with the matched wavelet learning, we present the scanning mechanism of rows and columns data in images as used in this work.

4.1.1.1 Scanning Mechanism for Row- and Column-wise Data

As stated earlier, we require to learn matched wavelet for both the row and column directions. One easier method can be learning matched wavelet on row- or column-vectorized image and use the same wavelet, later, along both the columns and rows as a separable wavelet. Instead, we propose to learn matched wavelet separately for the row- and column-directions using the following two scanning patterns:

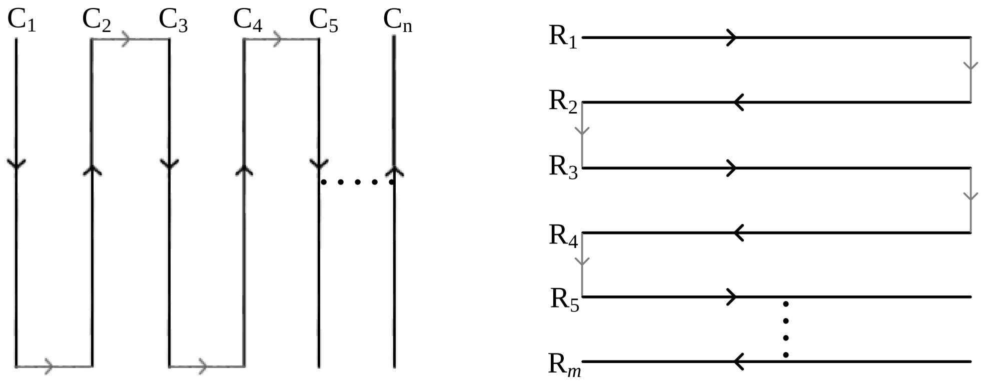


(a) Column wise raster scanning

(b) Row wise raster scanning

Figure 4.1: Raster scanning pattern
R_m denotes the m^{th} row and C_n denotes the n^{th} column

- Raster scanning pattern: The image is scanned according to the scanning pattern shown in Fig. 4.1, wherein rows or columns are stacked one after the



(a) Column wise serpentine scanning

(b) Row wise serpentine scanning

Figure 4.2: Serpentine scanning pattern
R_m denotes the m^{th} row and C_n denotes the n^{th} column

other to obtain 1-D signal for both the directions. However, this will cause discontinuity in the 1-D signal at the transitions when one column ends and another starts and likewise, for the rows.

- Serpentine scanning pattern: In order to avoid this discontinuity, an alternate way is to scan all even rows or columns in the reverse direction as shown in Fig. 4.2.

Since serpentine scanning pattern is robust to sudden transitions at the row- or column-endings, we use it in all our experiments.

4.1.1.2 Proposed Methodology of DWTL

With the serpentine scanning discussed above, we convert a given image into two 1-D signals: one with column-wise scanning and another with row-wise scanning. Now, we need to learn DWT matched to these column-wise and row-wise scanned 1-D signals. Note that, we have already presented the method to learn DWT corresponding to the 1-D signal in Section 3.1.1. We apply this method for the column and row-wise scanned 1-D signals to learn separable DWT to be used on images.

4.1.2 DWTL for images in inverse problems

The proposed method in this Section is the extended form of the method presented in Section 3.1.2 to images. The proposed methodology has two stages. In stage-1, we obtain a coarse estimate of the image from the degraded image using a standard wavelet. We call this a coarser estimate because the wavelet used is not matched to the given image and hence, the original image may not be that sparse over this wavelet compared to that with the matched wavelet (here matched wavelet for images implies learned separable DWT). In stage-2, we learn DWT corresponding to column-wise and row-wise scanned signal and use them as separable DWT. One can reconstruct the full image from the observed degraded image using the image-matched separable DWT learned in stage-2.

Mathematical formulation for the inverse problem is given by (3.31) and the original signal can be reconstructed by solving (3.32).

4.1.2.1 Stage-1: Coarser Image Estimation

In this stage, we reconstruct a coarser estimate of the image from compressively sensed image $\mathbf{y}$ by solving (3.32) and using any standard separable wavelet transform as the sparsifying transform $\mathbf{\Psi}$. We use biorthogonal 5/3 wavelet in our work. The coarser approximation of the vectorized version of the image is obtained as $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{s}}$. The coarser image is obtained as $\tilde{\mathbf{X}}$.

4.1.2.2 Stage-2: Separable Dyadic Wavelet Transform Learning

We use the serpentine scanning as discussed in Section 4.1.1.1 to convert the estimated image ($\tilde{\mathbf{X}}$) in the previous stage to column-wise and row-wise scanned 1-D signals, $\tilde{\mathbf{x}}_c$ and $\tilde{\mathbf{x}}_r$, respectively. We use these 1-D signals to learn separable DWT. Note that we already presented the method to learn DWT from the degraded 1-D signal in Section 3.1.2. We use stage-2 of the method presented in Section 3.1.2 to learn DWT corresponding to the column and row-wise scanned 1-D signals. These learned DWT are used as separable DWT for the images. This completes the learning of image-matched separable DWT in inverse problems.

4.1.2.3 Stage-3: Image Reconstruction using Learned Wavelet Transform

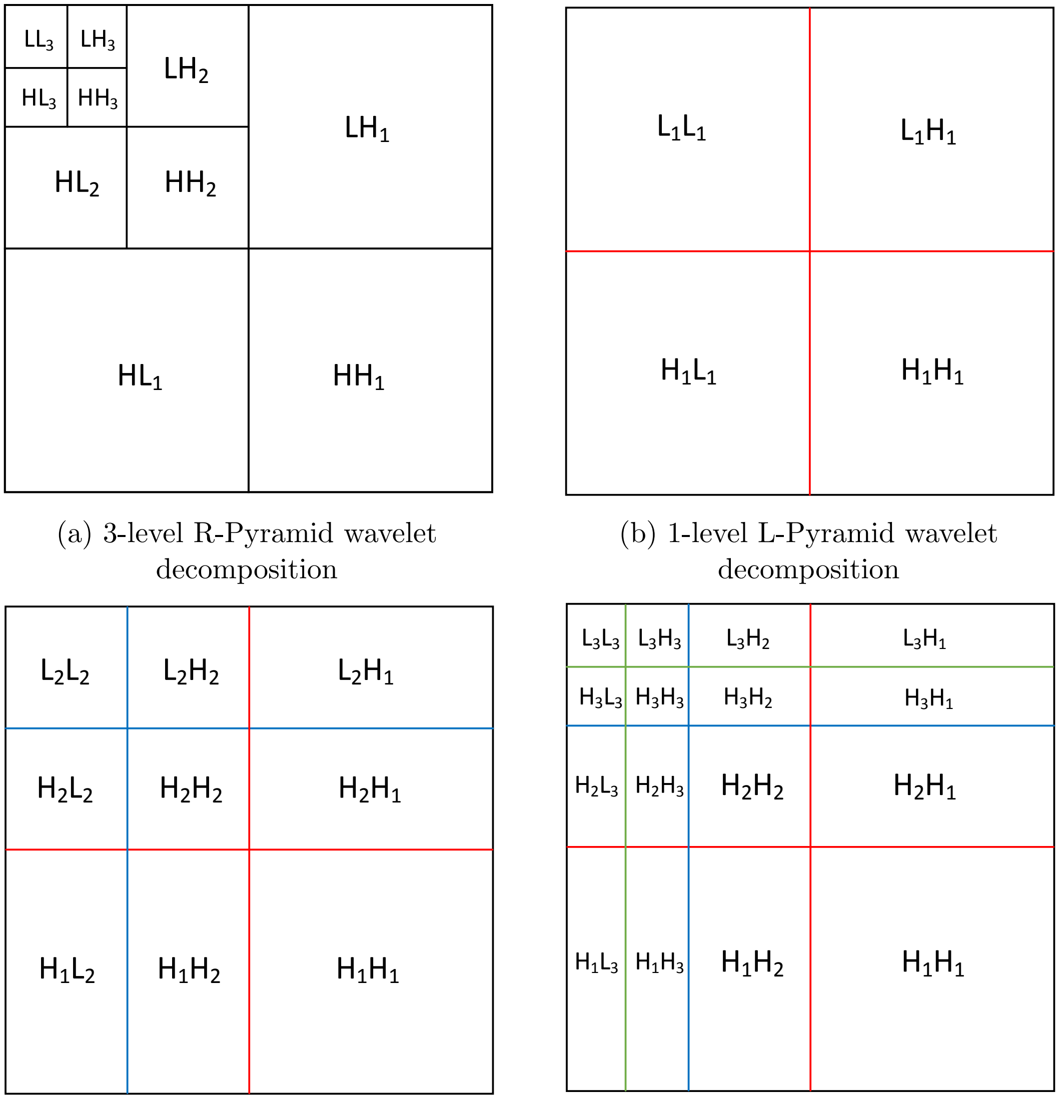
Once we have learned separable DWT, we employ (3.32) on observed degraded signal $\mathbf{y}$ with learned image-matched separable DWT as the sparsifying transform $\mathbf{\Psi}$ and estimate sparse coefficients $\tilde{\mathbf{r}}$. This is used to recover the original signal $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}}$ and hence, image $\tilde{\mathbf{X}}$

4.1.3 Proposed L-Pyramid Wavelet Decomposition Method for Images

In this Section, we propose a new strategy of multi-level wavelet decomposition on images.

A separable DWT is implemented on images by first applying the 1-D wavelet transform along the columns and then along the rows of an image. This provides 1-level wavelet decomposition that consists of four components labeled as LL, LH, HL, and HH, respectively. The same procedure is repeated on the LL part of the wavelet transform k-times to obtain k-level decomposition of an image (Fig. 4.3a). We call this decomposition as Regular Pyramid (R-Pyramid) wavelet decomposition.

In general, k-level wavelet decomposition of an image consists of the following components: LL$_k$, LH$_i$, HL$_i$ and HH$_i$, where $i = 1, 2, ..., k-1$. LH$_\mathrm{i}$, HL$_\mathrm{i}$ and HH$_\mathrm{i}$ components are obtained by applying wavelet transform on the columns and rows of LL$_{i\text{-}1}$ component. LH$_i$ is obtained by filtering LL$_{i\text{-}1}$ column-wise using a lowpass filter and filtering it row-wise using a highpass filter. Thus, the current nomenclature of labeling subbands is: first character represents operation on columns and second character represents operation on rows, where operation implies highpass or lowpass filtering denoted by symbols ‘H’ and ‘L’, respectively. In the conventional 2-D wavelet transform (Fig. 4.3a), wavelet decomposition is applied on LL$_i$ part only to obtain the $(i+1)^{\text{th}}$ level coefficients. Since it is a separable transform, similar to the 1-D wavelet transform wherein wavelet is applied repeatedly on lowpass filtered branches, we propose to apply wavelet in the lowpass filtered directions of LH$_{i\text{-}1}$ and HL$_{i\text{-}1}$ subbands in contrast to the conventional decomposition strategy wherein these subbands are left unaltered. Thus, the proposed 2$^{\text{nd}}$ level wavelet



(a) 3-level R-Pyramid wavelet decomposition

(b) 1-level L-Pyramid wavelet decomposition

(c) 2-level L-Pyramid wavelet decomposition

(d) 3-level L-Pyramid wavelet decomposition

Figure 4.3: Multi-level wavelet decomposition of image

decomposition is as shown in Fig. 4.3c.

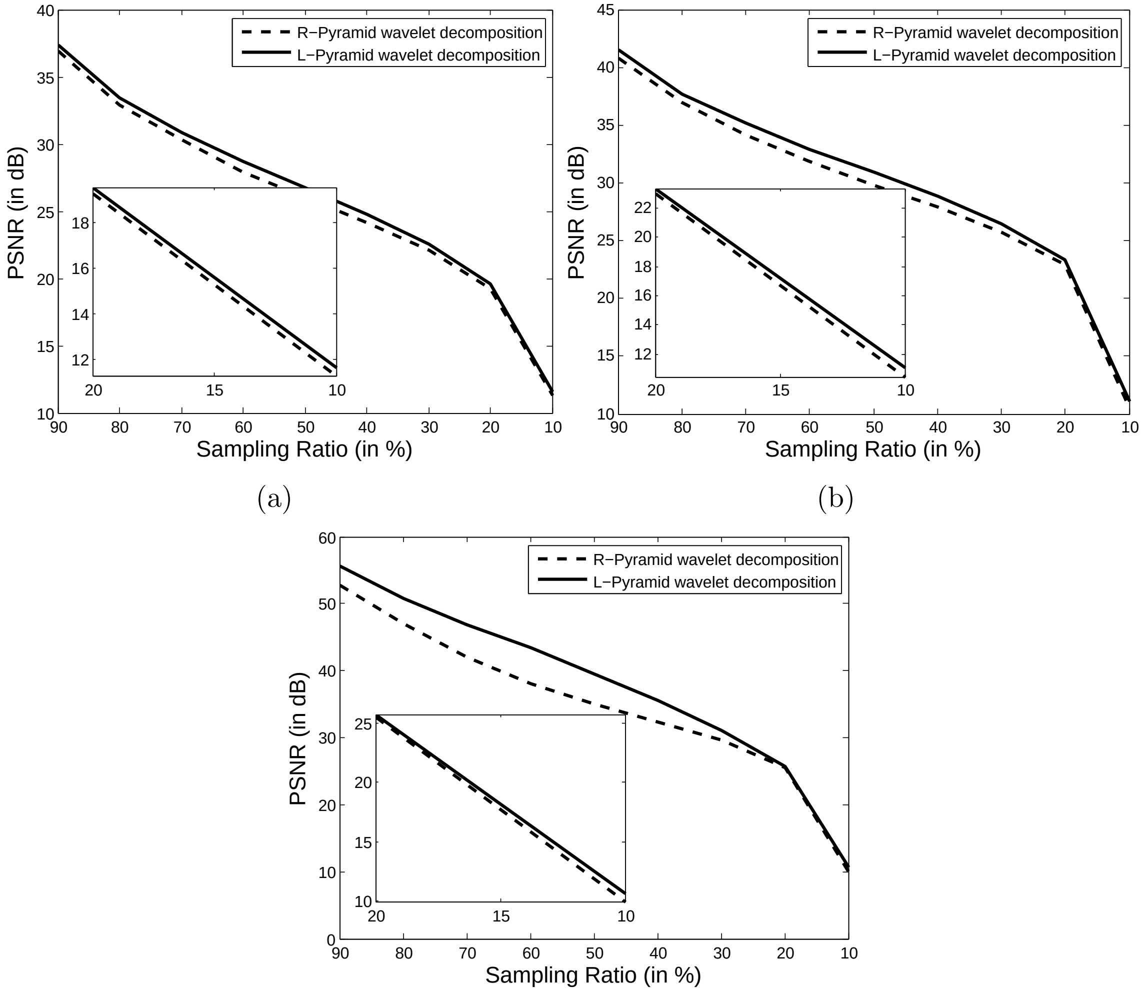
Since we apply wavelet in only one direction of $LH_{i\text{-}1}$ and $HL_{i\text{-}1}$ subbands, we notate these subbands differently compared to the conventional scheme. We assign subscript with both 'L' and 'H' symbols of every subband to denote the no. of times wavelet has been applied in that direction. In order to understand this, let us first consider 1-level wavelet decomposition as shown in Fig. 4.3b that is similar to the conventional scheme shown in Fig. 4.3a. However, the subbands are labeled as L_1L_1, L_1H_1, H_1L_1, and H_1H_1.

In the 2nd level wavelet decomposition, wavelet is applied in both directions of L_1L_1 subbands leading to L_2L_2, L_2H_2, H_2L_2, and H_2H_2 subbands. But in addition, wavelet is applied on the columns of L_1H_1 yielding two subbands L_2H_1 and H_2H_1. Also, wavelet is applied on the rows of H_1L_1 subband yielding two subbands H_1L_2 and H_1H_2. Applying similar strategy for the 3rd level decomposition, we obtain subbands as shown in Fig. 4.3d. We name this wavelet decomposition as L-shaped Pyramid (L-Pyramid) wavelet decomposition.

The efficacy of the proposed L-Pyramid wavelet decomposition is shown in CS-based image reconstruction with orthogonal Daubechies wavelet 'db4' and PCI (partial canonical identity) sensing matrix (discussed in next Section). Fig. 4.4 shows reconstruction accuracy in terms of PSNR with sampling ratios ranging from 10% to 90% averaged over 10 iterations. We compare reconstruction accuracy at different sampling ratios with the existing R-Pyramid wavelet decomposition and with the proposed L-Pyramid wavelet decomposition on the same three images: 'Beads', 'Lena' and 'House'. From Fig.4.4, we note better results with L-Pyramid wavelet decomposition compared to R-Pyramid wavelet decomposition at sampling ratios from 90% to 30%. There is considerably less improvement at lower sampling ratios of 20% and 10% (refer to the enlarged view in Fig.4.4). This may be due to the reason that the number of samples acquired at such lower sampling ratios do not contain enough information for good image reconstruction. In addition, we note that performance is particularly improved for image 'House' that is rich in low frequencies. Since the lowpass bands are repeatedly broken in all the subbands in the proposed L-Pyramid unlike the R-Pyramid, images rich in lower frequencies are benefited more. This, further, establishes the significance of the proposed decomposition strategy.

4.2 Applications

Applications of the proposed method of separable DWTL is presented in this Section. Compressive sensing of images is presented in Section 4.2.1 using partial canonical identity (PCI) sensing matrix. Impulse denoising of the images is pre-



(a)

(b)

(c)

Figure 4.4: CS-based reconstruction accuracy with the existing R-Pyramid and the proposed L-Pyramid wavelet decomposition with ‘db4’ wavelet on image (a) ‘Beads’ (b) ‘Lena’ (c) ‘House’

sented in Section4.2.2. Both of these problems are posed as sparse recovery problems. Separable DWT is learned in both of these applications and the learned separable DWT is used as the sparsifying transform for image recovery.

4.2.1 CS of images via L-Pyramid Wavelet Decomposition using PCI Sensing Matrix

4.2.1.1 Compressive Sensing of Images using Proposed Sensing Matrix

In this Section, first, we establish the need for a different, other than conventional, sensing matrix. Next, we discuss the proposed matrix. And later, we present results to show the comparison of time complexity and reconstruction performance with the proposed matrix in CS based image reconstruction.

Context

In today's world, size of images are increasingly large and N generally approaches to millions of samples. This large size imaging poses challenges for CS-based image reconstruction. The first challenge is the huge size of measurement matrix Φ that poses problems with storage and computation. Other problems include the design of imaging system with larger space bandwidth product (SBP) and difficult calibration requirements [119].

In an attempt to overcome the above challenges, single pixel camera hardware architecture has been proposed in [115]. It replaces the traditional camera architecture and captures the inner product between the scene under view and measurement basis. Thus, the camera captures one pixel at a time that is a linear combination of all pixel samples of the image. This process is repeated M number of times with $M \ll N$. These are called the compressive measurements and are transmitted to the receiver where full sized image is reconstructed by employing the theory of CS-based reconstruction. For more information on single pixel camera, reader may refer to [115].

Although the above hardware architecture implements compressive imaging (CI) nicely, it suffers with some difficulties including sensor dynamic range, A/D quantization, and photon counting effects [115]. Also, this process is time consuming as one has to wait for M samples that are captured sequentially. This is a serious problem in real-time applications, say, when one has to record a video using the camera as the scene may change while capturing samples sequentially of the current scene under view. Also, since M linear projections are captured instead of N pixel samples, it "effectively" samples the image at sub-Nyquist rate instead of “actually” sampling it at the sub-Nyquist rate.

Proposed Use of Partial Canonical Identity (PCI) Sensing Matrix

We propose to use PCI sensing matrix that, to our understanding, is the simplest sensing matrix proposed so far and “actually” senses the image at the sub-Nyquist rate by capturing less number of pixels without sensing information about every pixel [120]. This is explained as below.

Consider an image $\mathbf{X}$ of dimension $N_1 \times N_2$. Instead of sampling all the $N(N = N_1N_2)$ pixels of the image using the sensor array of the traditional camera, we capture M samples of the image using the proposed measurement matrix $\mathbf{\Phi}^p$, where $M \ll N$. The measurement matrix $\mathbf{\Phi}^p$ has the entries shown below:

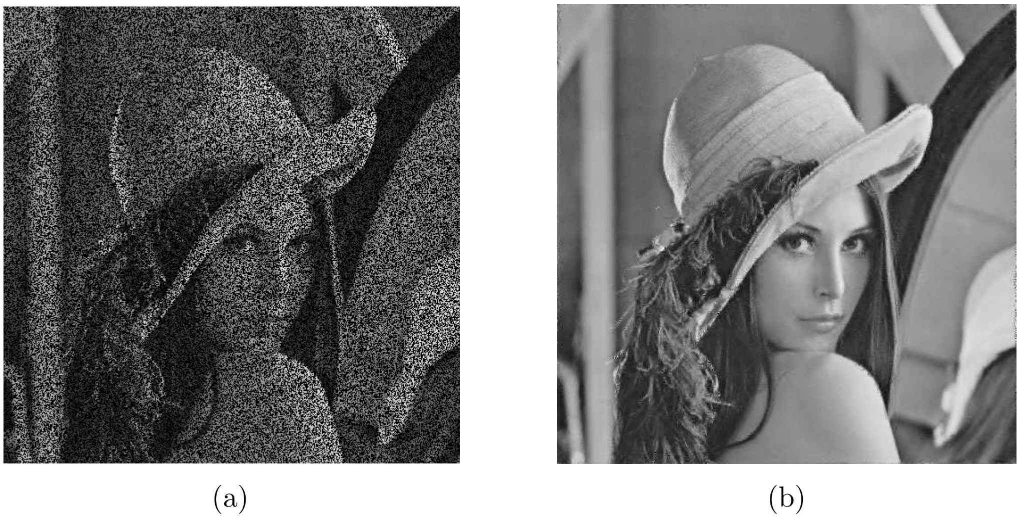
$$\mathbf{\Phi}^p_{i,j} = \begin{cases} 1 & \text{if } i \in \{1, 2, ..., M\} \text{ and } j \in \Omega \\ 0 & \text{otherwise} \end{cases}, \tag{4.1}$$

where $\Omega \subset \{1, 2, ..., N\}$ such that $|\Omega| = M$ and $|.|$ denotes the cardinality of the the set. This sensing matrix is known as partial canonical identity (PCI) matrix because it consists of partially selected and permuted rows of the identity matrix.

The PCI sensing matrix captures only M samples of the actual image; thus, actually sub-samples the original image. This can be accomplished by using existing cameras by switching ON only M sensors of the sensor array. This is unlike the single pixel camera where every captured pixel is the linear combination of the entire image pixel set. Also, in single pixel camera, one has to wait for M units of time to sense M number of samples, whereas all M samples are sensed in one unit of time in the case of PCI sensing matrix. Thus, PCI sensing matrix reduces the sensing time by a factor of M in comparison to a single pixel camera.

Results using PCI Sensing Matrix

Consider a sub-sampled Lena image (original image dimension is 512×512), shown in Fig. 4.5a, with only 50% samples captured via PCI sensing matrix. The uncaptured positions are filled with zeros. Fig. 4.5b shows the image reconstructed from this sub-sampled image using (2.22) with standard wavelet ‘db4’ as the sparsifying basis. Since images are, in general, correlated in the spatial domain, full



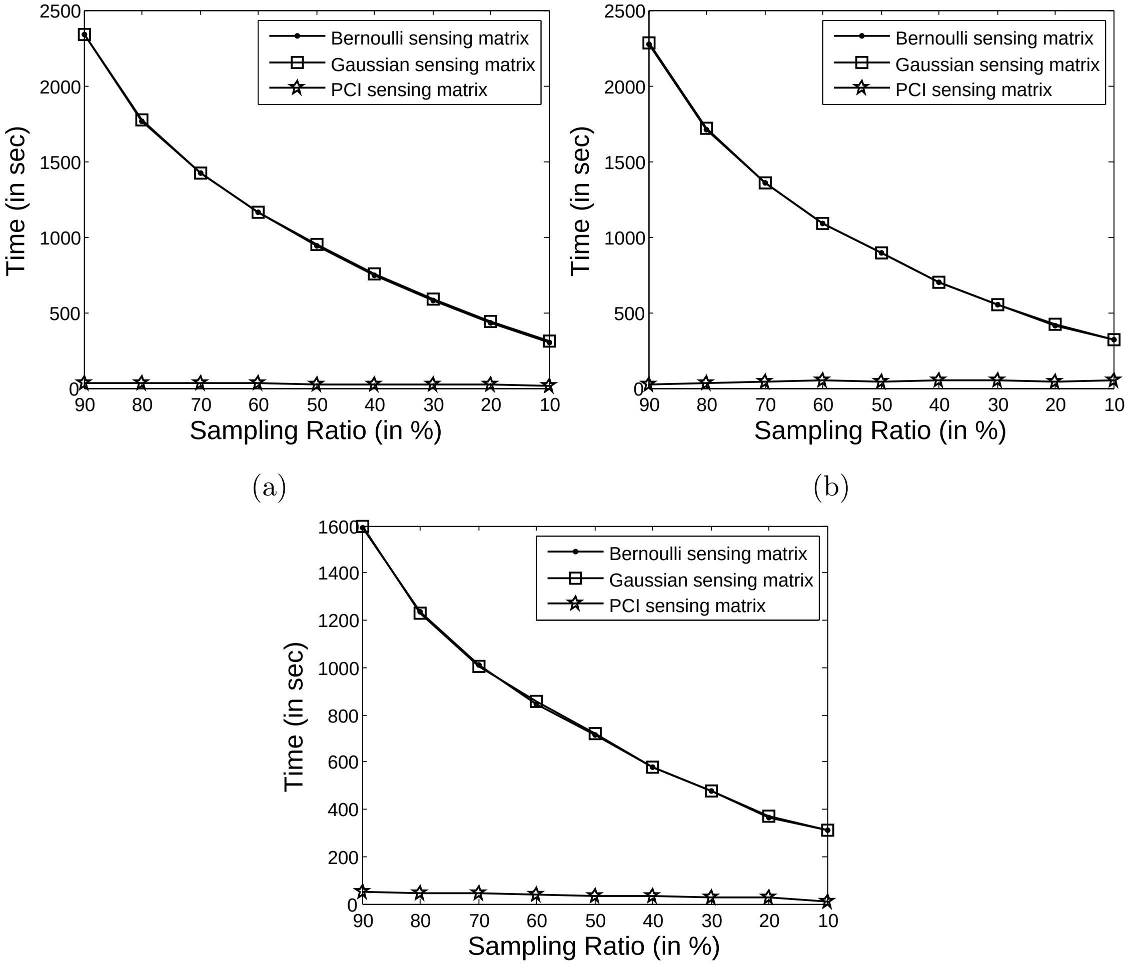
(a) (b)

Figure 4.5: (a) Image (with dimension 512 × 512) captured using PCI sensing matrix with 50% sampling ratio and with zeros filled at positions not sampled (b) Image reconstructed from subsampled image using 'db4' wavelet as the sparsifying basis in (2.22).

image can be reconstructed using partial samples collected through the PCI sensing matrix. From Fig. 4.5b, we indeed note good reconstruction quality of the image sensed partially with PCI sensing matrix. This makes PCI sensing matrix as the candidate for the measurement matrix. Fig. 4.6 and 4.7 provide detailed results.

Fig. 4.6 compares the time taken in image reconstruction from the measured samples with sampling ratios varying from 10% to 90% using different measurement matrices, where sampling ratio is defined as the ratio of the number of samples captured to the total number of samples in the image (M/N). We compare the reconstruction time taken using the PCI sensing matrix, random Gaussian matrix, and Bernoulli random matrix with ±1 as its entries. Random Gaussian matrix is preferred in a wide range of applications because of the ease in theoretical analysis, while the Bernoulli matrix depicts the physical implementation of single pixel camera. The image is reconstructed using equation (2.22), where we have used standard Daubechies orthogonal wavelet 'db4' as the sparsifying basis. We have used MATLAB solver spgl1 [107], [108] to solve (2.22) that implements Basis Pursuit (BP) [95].

Compressed sensing based reconstruction with Gaussian and Bernoulli sensing



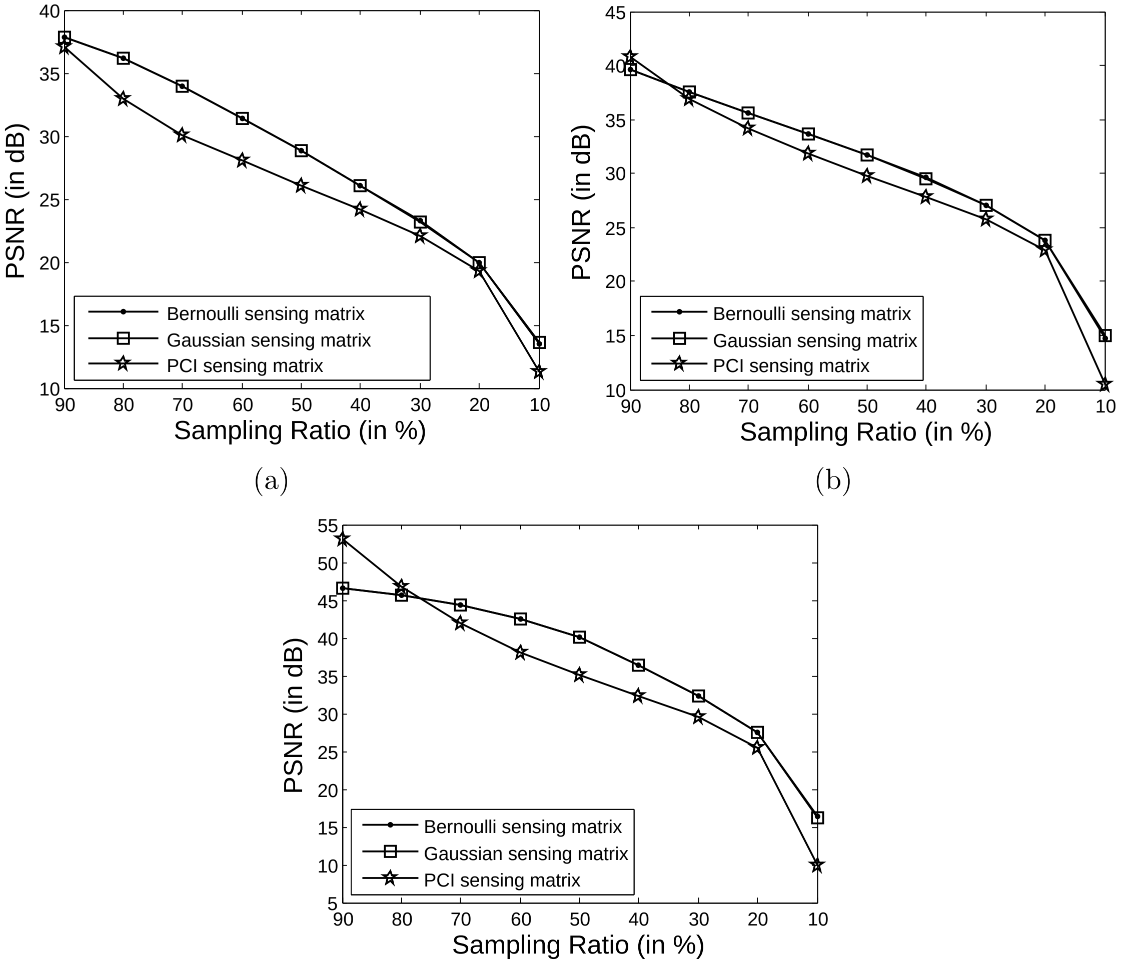
(a)

(b)

(c)

Figure 4.6: Time comparison in CS-based image reconstruction with different measurement matrices on (a) Image 'Beads' (b) Image 'Lena' (c) Image 'House'

matrices is implemented using block compressed sensing [121]. This is to note that reconstruction with PCI sensing matrix requires only the position information of the sampled pixels instead of the information of all entries of $M \times N$ sensing matrix that simplifies reconstruction with PCI sensing matrix. We compare reconstruction results on three images: 'Beads', 'Lena', and 'House' as shown in Fig. 4.8a, 4.8c and 4.8h, respectively. The size of each image is 512×512. We have chosen these images because they exhibit different spectral properties. For example, 'Beads' is rich in high frequencies, 'Lena' contains both low and high frequency contents, while 'House' is rich in low frequencies.



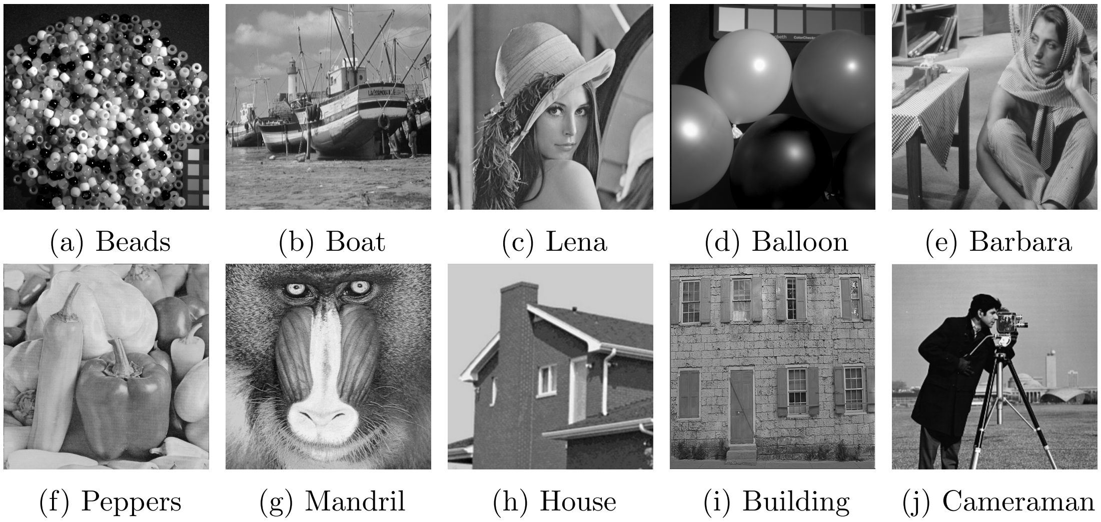
(a)

(b)

(c)

Figure 4.7: Reconstruction accuracy in terms of PSNR (in dB) with different measurement matrices on (a) Image 'Beads' (b) Image 'Lena' (c) Image 'House'

From Fig. 4.6, we note that the reconstruction time with Gaussian and Bernoulli sensing matrices is almost the same, whereas reconstruction time with the PCI sensing matrix is extremely less. This huge reduction in time is owing to the implementation simplicity of PCI sensing matrix. However, there is a trade-off between the reconstruction time and the accuracy. Fig. 4.7 compares reconstruction accuracy of images in terms of PSNR (2.35). Results shown are averaged over 10 iterations. From Fig. 4.7, we observe that reconstruction accuracy with PCI sensing matrix is 2-5 dB lower than that with Bernoulli or Gaussian sensing matrices. This gap in reconstruction accuracy is not negligible in practical applications. Similar to this observation, comparatively inferior CS-based reconstruction results have been noted with PCI sensing matrix in [120]. Hence, reconstruction with PCI sensing matrix is better than Gaussian or Bernoulli (and hence, single pixel camera) sensing matrix only in terms of time taken in reconstruction and not in the reconstruction quality. In Section 4.2.1.2, we will show that this gap in reconstruction accuracy can be not only bridged, but rather enhanced to a large extent with the learned matched wavelet. Since the use of PCI sensing matrix expedites the complete pipeline, its use is overall useful if the loss in quality can be covered up via better signal processing methodology.



(a) Beads (b) Boat (c) Lena (d) Balloon (e) Barbara

(f) Peppers (g) Mandril (h) House (i) Building (j) Cameraman

Figure 4.8: Images used in experiments. These images are referred as Im1-Im10 in Table-4.2

4.2.1.2 Experiments and Results

In this Section, we present CS-based reconstruction results of images using image-matched separable DWT learned from compressively sensed images. We apply the proposed method on ten natural images shown in Fig. 4.8. Images with different spectral contents have been selected. For example, Balloon, House and cameraman are rich in low frequencies, Beads, Barbara and Mandrill are rich in high frequencies, and rest of the images have varied lower and higher frequency content.

Table 4.1: Analysis filters along with predict/update filters at all 3 levels of wavelet decomposition learned with image ‘Lena’ compressively sensed at a sampling ratio of 50% with PCI sensing matrix

Column wise matched			
Level	1	2	3
$t[n]$	[0.2875 0.2380 0.3946 0.0812]	[0.1642 0.3317 0.5096 -0.0041]	[0.1877 0.3166 0.5390 -0.0401]
$s[n]$	[0.1708 0.1184 0.2688 0.0039]	[0.0694 0.1763 0.3439 -0.0677]	[0.0884 0.1620 0.3615 -0.0890]
$h_0[n]$	[-0.0139 0 -0.0770 0.1708 -0.1092 0.1184 0.8163 0.2688 -0.0996 0.0039 -0.0782 0 -0.0011]	[0.0003 0 -0.0346 0.0694 -0.1114 0.1763 0.7546 0.3439 -0.1085 -0.0677 -0.0340 0 0.0111]	[0.0035 0 -0.0411 0.0884 -0.1008 0.1620 0.7337 0.3615 -0.0969 -0.0890 -0.0397 0 0.0167]
$h_1[n]$	[-0.0812 0 -0.3946 1.0000 -0.2380 0 -0.2875]	[0.0041 0 -0.5096 1.0000 -0.3317 0 -0.1642]	[0.0401 0 -0.5390 1.0000 -0.3166 0 -0.1877]
Row wise matched			
Level	1	2	3
$t[n]$	[0.1987 0.3041 0.4723 0.0270]	[0.1487 0.3489 0.5114 -0.0065]	[0.2324 0.2649 0.5287 -0.0205]
$s[n]$	[0.0955 0.1615 0.3231 -0.0446]	[0.0586 0.1883 0.3426 -0.0691]	[0.1211 0.1302 0.3593 -0.0705]
$h_0[n]$	[-0.0026 0 -0.0494 0.0955 -0.1140 0.1615 0.7805 0.3231 -0.1093 -0.0446 -0.0506 0 0.0089]	[0.0004 0 -0.0288 0.0586 -0.1145 0.1883 0.7499 0.3426 -0.1122 -0.0691 -0.0268 0 0.0103]	[0.0025 0 -0.0614 0.1211 -0.0935 0.1302 0.7460 0.3593 -0.0882 -0.0705 -0.0648 0 0.0164]
$h_1[n]$	[-0.0270 0 -0.4723 1.0000 -0.3041 0 -0.1987]	[0.0065 0 -0.5114 1.0000 -0.3489 0 -0.1487]	[0.0205 0 -0.5287 1.0000 -0.2649 0 -0.2324]

Learned filter Coefficients

We learn image-matched wavelets for all the images shown in Fig. 4.8. Since wavelet transform is generally applied at three-level decomposition on images, we learned wavelet filters for rows and columns separately at all the three successive levels of wavelet decomposition. Table-5.2 shows the analysis filters designed with the image ‘Lena’ compressively sensed at a sampling ratio of 50% with PCI sensing matrix. Coefficients for filters designed are shown in Table-5.2 for both column matched and row matched wavelets along with the coefficients of predict and update stage filters $T(z)$ and $S(z)$, respectively.

Experiment-1: Comparison of existing CS-based image reconstruction methodology with the proposed methodology

In the application of CS-based image reconstruction, the proposed methodology of this work has three contributions:

1. *The proposed use of PCI sensing matrix*: that is computationally inexpensive compared to the existing sensing matrices (Fig. 4.6), but provides approx. 2-5 dB lower performance compared to the existing sensing matrices (Fig. 4.7).

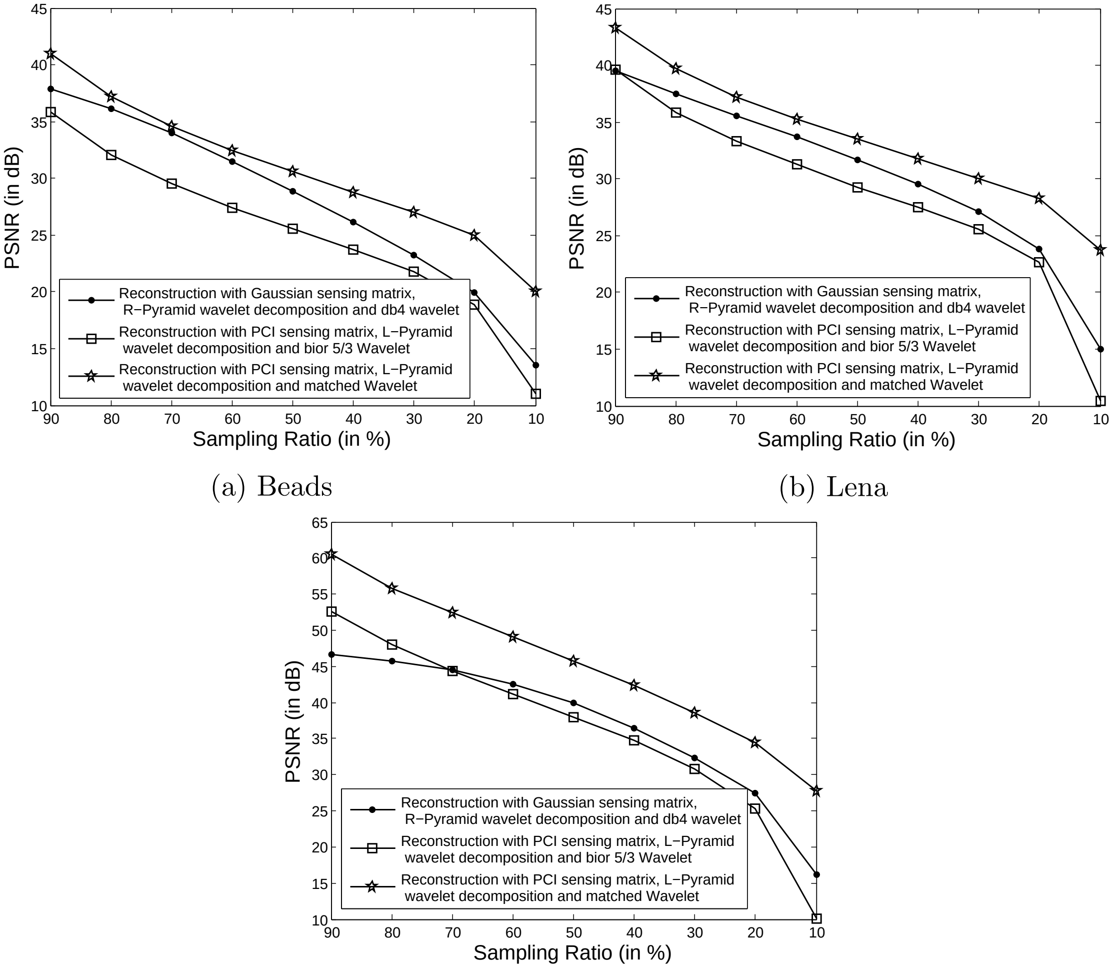
2. *Proposed L-Pyramid wavelet decomposition* (Fig. 4.3d): that provides better results in CS-based image reconstruction (Fig. 4.4) compared to the existing R-Pyramid wavelet decomposition (Fig. 4.3a).

3. *Learning of image-matched wavelets*: wherein separable DWT is learned from a compressively sensed image and is used for the reconstruction of the same. Hence, an image is recovered by employing a wavelet matched to it.

Based on the above observations, we would like to compare the performance of the proposed CS-based image reconstruction (with all three novelties: proposed use of PCI sensing matrix, proposed L-Pyramid wavelet decomposition, and proposed matched wavelet learning) with the existing CS-based reconstruction (Gaussian sensing matrix, existing R-Pyramid wavelet decomposition, and db4 wavelet). Fig. 4.9 shows CS-based reconstruction results in terms of PSNR averaged over 10 iterations on three images ‘Beads’, ‘Lena’, and ‘House’. In addition, Fig. 4.9 also presents results of the coarse estimate obtained after stage-1 of the proposed method, i.e., the image reconstructed with PCI sensing matrix, L-Pyramid wavelet decomposition, and bior 5/3 wavelet.

Discussion: From Fig. 4.9 we observe that the proposed methodology performs consistently better than the existing methodology and the coarse estimate at all sampling ratios with the following observations:

1. As the image changes from being rich in high frequency (‘Beads’) to mid-frequency (‘Lena’) to low frequency (‘House’) content, better and better reconstruction performance is observed with the proposed methodology com-



(a) Beads

(b) Lena

(c) House

Figure 4.9: CS-based reconstruction comparison of 1) the existing methodology: Gaussian sensing matrix, R-Pyramid wavelet decomposition, and db4 wavelet; 2) coarse estimate: PCI sensing matrix, L-Pyramid wavelet decomposition, and bior 5/3 wavelet; and 3) our proposed methodology: PCI sensing matrix, L-Pyramid wavelet decomposition, and proposed matched-wavelet design; on image a) Beads, b) Lena, and c) House

pared to the existing methodology. This is owing to the fact that the proposed methodology works best for signals rich in low frequency contents.

2. The quality of images reconstructed with the proposed methodology is consistently better in terms of PSNR than the coarse estimate. This is to note that the difference between the coarse estimate and the proposed solution is only in using standard bior 5/3 wavelet versus learned image-matched

wavelet, while keeping the sensing matrix and wavelet decomposition strategy as same. This further establishes the significance of the learned image-matched wavelet vis-*à*-vis standard bior 5/3 wavelet in the proposed work.

3. At higher sampling ratio of 90%, performance gain with the proposed methodology over the existing methodology is 3 dB with ‘Beads’, 4 dB with ‘Lena’, and 13 dB with ‘House’. Since at higher sampling ratios, most of the input image samples are available upfront, hence, learned matched wavelet is optimum. This provides very good performance and a huge improvement over the existing methodology, particularly, for images rich in low frequency content.

4. At lower sampling ratio of 10%, performance gain with the proposed methodology over the existing methodology is 6.5 dB with ‘Beads’, 9 dB with ‘Lena’, and 11 dB with ‘House’. In fact, standard methodology with standard wavelets almost fails in reconstructing images with any good quality at very low sampling ratios, while the proposed methodology still performs well. For visual clarity, Fig. 4.10 shows reconstructed images with the existing methodology, coarse estimate, and the proposed methodology at 10% sampling ratio.

Experiment-2: Comparison of standard wavelets and learned matched wavelet with PCI sensing matrix in CS-based image reconstruction

In this subsection, we present CS-based image reconstruction results using the PCI sensing matrix and the proposed L-Pyramid wavelet decomposition. Table-4.2 shows the comparison on reconstruction accuracy in terms of PSNR averaged over 10 iterations obtained with standard wavelets and that obtained with image-matched separable DWT learned from compressively sensed images. Table-4.2 shows reconstruction accuracy on all the 10 images used in the experiments along with the average accuracy (averaged over these ten images) in the last column.

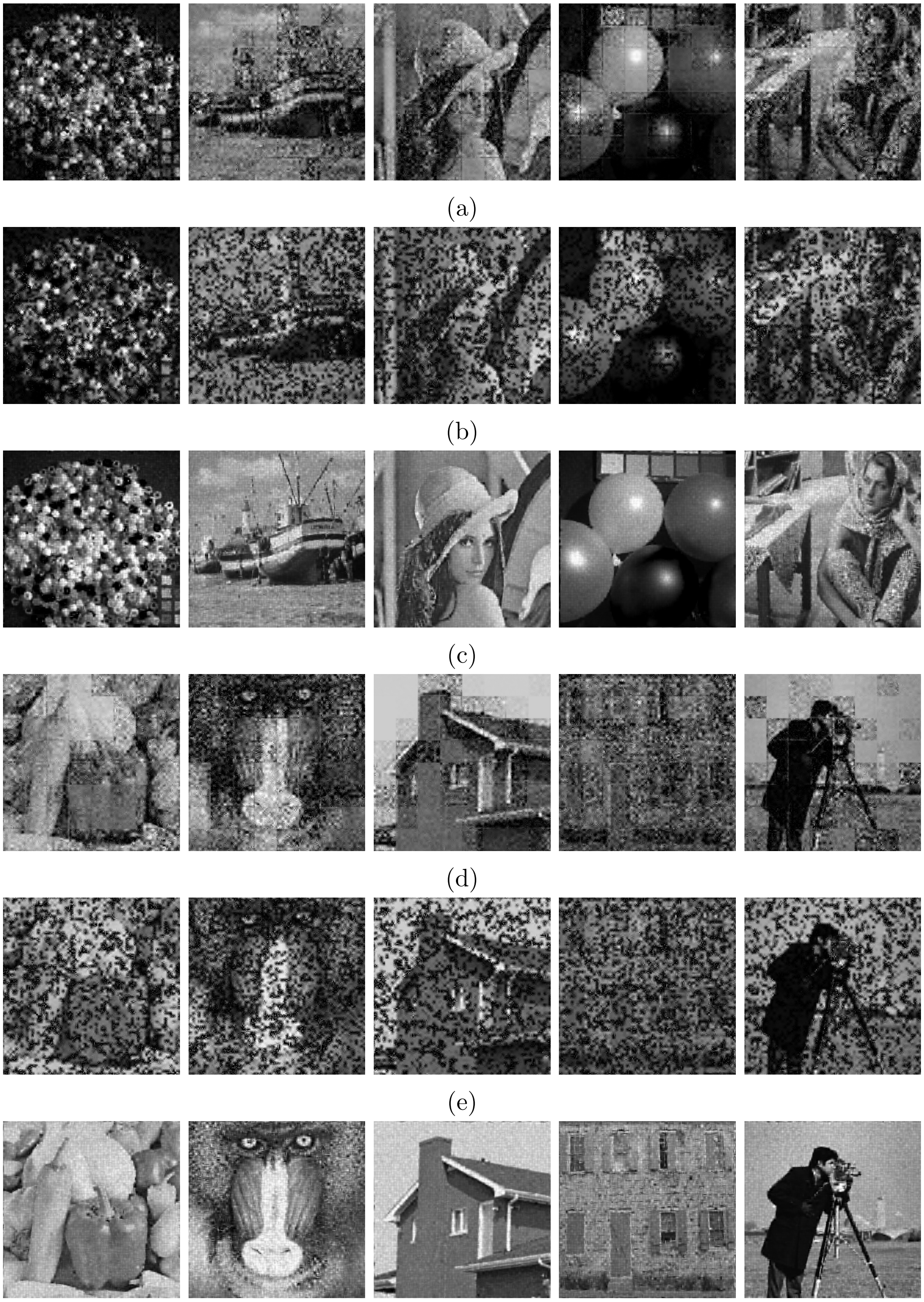
We have considered three standard wavelets: orthogonal Daubechies’ wavelets ‘db2’, ‘db4’, and biorthogonal standard 5/3 wavelet (represented as Bi 5/3 in the table). While orthogonal wavelets are widely used in applications, biorthogonal

Table 4.2: Reconstruction accuracy on CS-based image reconstruction with standard wavelets and with image-matched wavelets learned from compressively sensed images with 5/3 length filters. PCI sensing matrix and the proposed L-Pyramid wavelet decomposition have been used to generate these results.

SR	Wavelet used	Im1	Im2	Im3	Im4	Im5	Im6	Im7	Im8	Im9	Im10	Average
90	db2	36.1	38.4	40.2	49.7	35.8	38.3	30.4	52.5	34.9	43.5	40.0
	db4	37.2	39.5	41.5	51.0	**37.8**	**38.6**	30.7	55.7	35.0	45.5	41.3
	Bi 5/3	35.9	37.8	39.6	49.4	35.0	37.7	29.7	52.7	34.4	43.2	39.5
	DWTL	**41.0**	**41.1**	**43.3**	**52.4**	36.4	38.0	**31.6**	**60.5**	**37.3**	**50.5**	**43.2**
80	db2	32.3	34.6	36.5	46.3	31.8	35.0	27.1	47.5	31.4	39.0	36.2
	db4	33.5	35.5	37.7	47.2	**33.5**	**35.3**	27.4	50.8	31.5	40.9	37.3
	Bi 5/3	32.1	33.9	35.8	45.7	31.1	34.4	26.3	48.0	31.0	38.7	35.7
	DWTL	**37.2**	**37.5**	**39.7**	**49.0**	32.4	35.0	**28.3**	**55.8**	**33.8**	**46.1**	**39.5**
70	db2	29.9	32.0	34.0	43.5	29.2	32.9	25.0	44.1	29.3	35.7	33.6
	db4	30.9	32.8	35.1	44.5	**30.7**	33.1	25.3	46.9	29.3	37.5	34.6
	Bi 5/3	29.5	31.3	33.3	43.1	28.5	32.3	24.3	44.4	28.8	35.2	33.1
	DWTL	**34.6**	**35.1**	**37.2**	**46.5**	29.7	**33.3**	**26.2**	**52.4**	**31.5**	**42.5**	**36.9**
60	db2	27.8	30.0	31.9	41.2	27.2	31.2	23.5	41.0	27.5	33.2	31.4
	db4	28.6	30.6	33.0	42.1	**28.4**	31.4	23.7	43.3	27.5	34.6	32.3
	Bi 5/3	27.4	29.2	31.3	40.9	26.5	30.6	22.8	41.1	27.0	32.4	30.9
	DWTL	**32.5**	**33.2**	**35.3**	**44.6**	27.5	**32.0**	**24.7**	**49.1**	**29.8**	**39.7**	**34.8**
50	db2	26.0	28.0	30.0	39.1	25.5	29.7	22.2	37.9	25.9	30.7	29.5
	db4	26.7	28.6	30.9	39.7	**26.3**	29.9	22.3	39.6	25.8	31.7	30.2
	Bi 5/3	25.5	27.4	29.3	38.9	24.8	29.2	21.5	37.9	25.4	29.8	29.0
	DWTL	**30.6**	**31.5**	**33.5**	**42.7**	25.9	**30.9**	**23.4**	**45.7**	**28.3**	**37.0**	**33.0**
40	db2	24.1	26.1	28.0	36.5	23.9	28.0	21.0	34.7	24.1	28.1	27.5
	db4	24.7	26.5	28.8	36.6	**24.5**	28.3	21.0	35.6	24.0	29.0	27.9
	Bi 5/3	23.8	25.6	27.5	36.6	23.4	27.7	20.4	34.7	23.8	27.2	27.0
	DWTL	**28.8**	**29.8**	**31.8**	**40.3**	24.4	**29.9**	**22.3**	**42.3**	**27.0**	**34.4**	**31.1**
30	db2	21.9	24.0	25.9	33.3	22.2	26.2	19.7	30.7	22.2	25.4	25.2
	db4	22.5	24.3	26.4	32.7	22.5	26.4	19.7	31.0	22.1	26.0	25.4
	Bi 5/3	21.8	23.7	25.5	33.7	21.8	26.0	19.2	30.7	21.9	24.7	24.9
	DWTL	**27.0**	**28.2**	**30.1**	**37.1**	**23.2**	**28.9**	**21.2**	**38.5**	**25.6**	**31.7**	**29.2**
20	db2	18.8	21.1	22.8	28.7	19.7	23.0	17.8	25.4	19.6	21.8	21.9
	db4	19.6	21.4	23.3	28.0	20.0	23.6	17.9	25.8	19.6	22.3	22.2
	Bi 5/3	18.8	20.8	22.7	29.0	19.4	23.0	17.5	25.4	19.3	21.5	21.7
	DWTL	**25.0**	**26.3**	**28.3**	**33.1**	**22.2**	**27.7**	**20.2**	**34.4**	**24.0**	**28.8**	**27.0**
10	db2	10.8	9.8	10.0	13.5	9.7	8.2	8.9	9.6	10.2	10.3	10.1
	db4	11.6	10.6	11.1	14.5	10.6	9.3	9.6	10.7	10.9	11.3	11.0
	Bi 5/3	11.0	10.2	10.5	13.8	10.1	8.6	9.3	10.1	10.7	10.6	10.5
	DWTL	**20.0**	**22.3**	**23.7**	**30.9**	**20.1**	**23.0**	**18.1**	**27.7**	**20.3**	**23.0**	**22.9**

wavelets take care of boundary effects and provide better compression results as compared to orthogonal wavelet [122], [123]. This is to note that our learned image-matched wavelets are biorthogonal by design.

Discussion: From Table-4.2, we observe much better reconstruction accuracy with the learned matched wavelets compared to standard wavelets in terms of average PSNR over all the sampling ratio ranging from 90% to 10%. Orthogonal db4 wavelet performs better than other standard wavelets but its performance is still inferior to our learned matched wavelets. At the sampling ratio of 90%,



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.10: Visual comparison of CS-based image reconstruction on images sensed at 10% sampling ratio with a) the existing methodology: Gaussian sensing matrix, R-Pyramid wavelet decomposition, and db4 wavelet; b) PCI sensing matrix, L-Pyramid wavelet decomposition, and bior 5/3 wavelet (coarse estimate); c) proposed methodology: PCI sensing matrix, L-Pyramid wavelet decomposition, and proposed matched-wavelet design; (d), (e), and (f): same as (a) (b) and (c)

learned matched wavelet provides an improvement upto 3 dB, while we observe an improvement upto 11.5dB at 10% sampling ratio.

Further, we observe that while standard wavelets fail almost completely at lower sampling ratio of 10% with a reconstruction PSNR of approximately 10dB only, learned matched wavelets are able to reconstruct images with a PSNR above 20dB.

For better visual clarity, we have also shown images reconstructed from compressively sensed images at 10% sampling ratio using the PCI sensing matrix with the learned matched wavelet and standard bior 5/3 wavelet (coarse estimate) for all ten images in Fig. 4.10. From the figure, it can be clearly noticed that the existing wavelet bior 5/3 is not able to reconstruct full images whereas the learned matched wavelets provide good reconstruction quality.

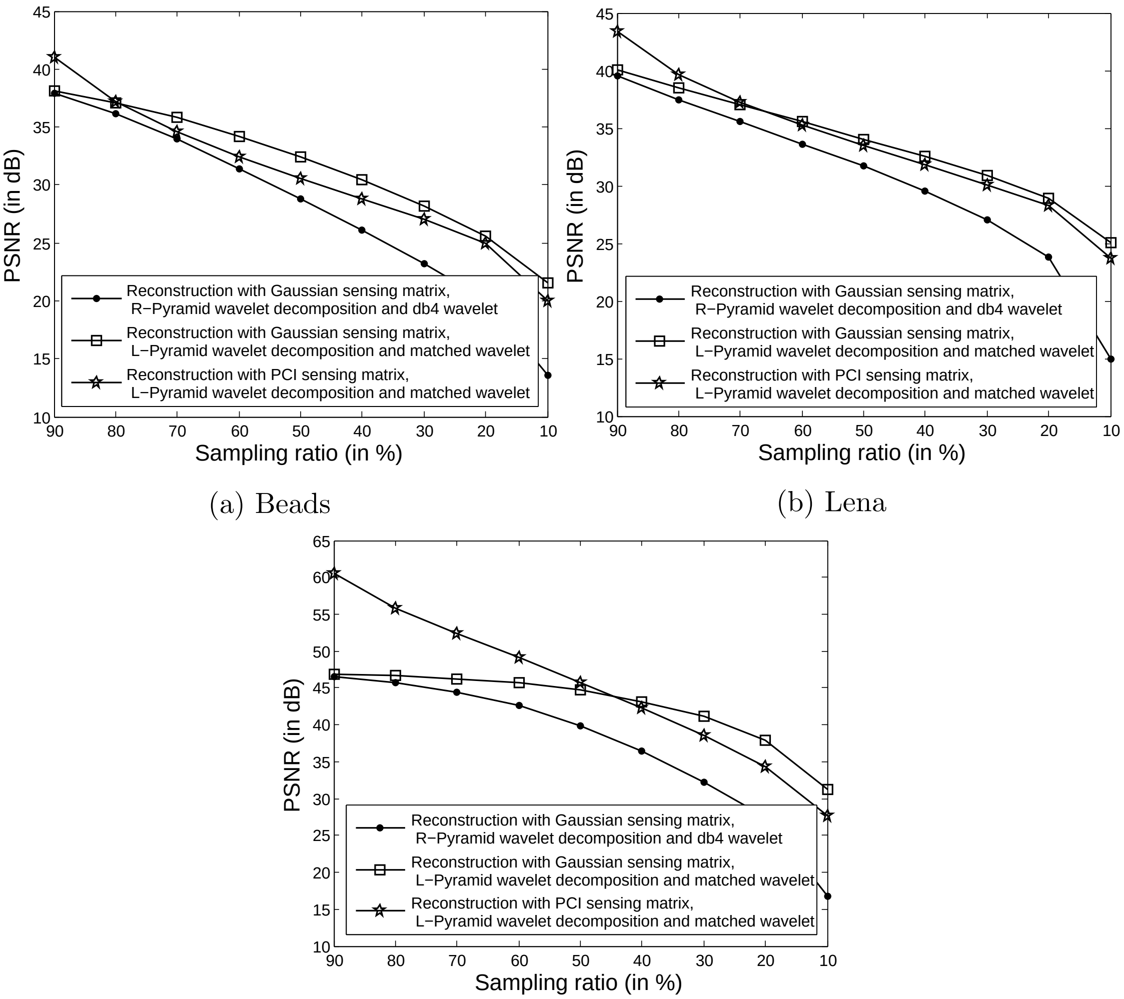
Experiment-3: Comparison of standard wavelet vis-*à*-vis learned matched wavelet with Gaussian sensing matrix in CS-based image reconstruction

In this part, we would like to understand the performance of the proposed L-Pyramid wavelet decomposition method and the learned matched wavelet compared to the existing R-Pyramid wavelet decomposition and standard wavelet. This performance comparison is shown with the conventional Gaussian sensing matrix in Fig. 4.11 on three images ‘’Beads’, ‘Lena’ and ‘House’. As stated earlier, the PCI sensing matrix has been introduced in this work for reducing the hardware complexity. However, its performance is inferior to the Gaussian matrix as noted in Section 4.2.1.1. In this experiment, we would like to see whether the poor performance of PCI sensing matrix can be bridged with the proposed novelties of matched wavelet learning and better multi-level decomposition strategy.

From Fig. 4.11, following observations are in order:

1. The performance of the existing methodology- CS with the conventional regular wavelet decomposition (R-Pyramid), db4 wavelet, and the Gaussian matrix is the least.

2. Learned matched wavelet improves the reconstruction accuracy over the existing (db4) wavelet with Gaussian sensing matrix. Further, this improvement increases as the sampling ratio decreases establishing the significance



(a) Beads

(b) Lena

(c) House

Figure 4.11: CS-based reconstruction comparison of 1) Gaussian sensing matrix, R-Pyramid wavelet decomposition, and db4 wavelet; 2) Gaussian sensing matrix, L-Pyramid wavelet decomposition, and proposed matched-wavelet design; and 3) PCI sensing matrix, L-Pyramid wavelet decomposition, and proposed matched-wavelet design.

of the learned matched wavelet.

3. On comparing the reconstruction accuracy of the Gaussian sensing matrix with the PCI sensing matrix (both with L-Pyramid wavelet decomposition and the learned matched wavelet), it is observed that the Gaussian sensing matrix mostly performs better than the PCI sensing matrix. However, reconstruction accuracy with Gaussian is inferior at higher sampling ratio. This

may possibly be due to the block artifacts occurring owing to the implementation of block compressed sensing with Gaussian sensing matrix. Implementation with PCI sensing matrix is simple and does not require block CS implementation. Although block artifacts with Gaussian sensing matrix are also present at lower sampling ratios, but reconstruction with PCI sensing matrix at lower sampling ratio is poorer because only fewer samples (subsamples) of the images are present for image reconstruction. Hence, Gaussian sensing matrix performs better.

From the above, we note that the learned matched wavelet with L-Pyramid performs better than standard wavelet with R-Pyramid at all sampling ratios with both the PCI sensing matrix and with the Gaussian sensing matrix. Further, we may improve the reconstruction with Gaussian sensing matrix. However, there is a trade-off on time complexity and ease of hardware implementation versus accuracy in using PCI versus Gaussian sensing matrix. Since there is not much difference in the reconstruction ability of the two sensing matrices, while much improved performance is obtained with both while using learned matched wavelet and L-Pyramid over standard wavelet and R-Pyramid, it may be worthwhile to use the learned matched wavelet, L-Pyramid, and PCI sensing matrix. Thus, the proposed methodology will provide both improved accuracy and lesser computational time compared to the existing methodology (standard wavelet, R-Pyramid, and Gaussian sensing matrix).

4.2.2 Impulse Denoising of Natural Images

Impulse denoising of natural images is presented in this subsection. Impulse denoising is posed as a sparse recovery problem. First, an impulse detection algorithm is employed to detect corrupted (from impulse noise) and uncorrupted pixels in the image. Method to learn separable DWT is presented, where separable DWT is learned using the uncorrupted pixels identified by the impulse detection algorithm. The learned separable DWT is used as the sparsfiying transform to reconstruct full image.

4.2.2.1 Proposed DWTL based impulse denoising

The proposed method of impulse denoising consists of three stages. First, impulse locations are detected using BDND algorithm described in Section 2.4.2.1. Next, uncorrupted pixels are used to learn wavelet basis and hence, wavelet transform for the given image. In the end, the image is denoised using the sparse recovery method and the wavelet learned in the second stage.

Stage 1: Detection of Impulse Locations

Let us consider an unknown image $\mathbf{X}$ of dimension $N_1 \times N_2$ and its measured impulse noise corrupted version $\mathbf{X}_n$. The BDND algorithm discussed in Section 2.4.2.1 is applied on the noisy input image $\mathbf{X}_n$ to locate impulse positions. These indices are stored in set ω_c, while indices of pixels identified as uncorrupted are stored in set ω_u. Next, three 1-D signals are constructed: a) the uncorrupted pixels (belonging to ω_u) of $\mathbf{X}_n$ are scanned row-wise using serpentine scanning shown in Fig. 4.2b and labeled as $\mathbf{y}_r$; b) the uncorrupted pixels (belonging to ω_u) of $\mathbf{X}_n$ are scanned column-wise using serpentine scanning shown in Fig. 4.2a and labeled as $\mathbf{y}_c$; and c) the uncorrupted pixels of $\mathbf{X}_n$ are raster scanned as shown in Fig. 4.1a and labeled as $\mathbf{x}_{n,uncor}$. Signals $\mathbf{y}_r$ and $\mathbf{y}_c$ are used for learning wavelet basis for the row-space and the column space of the image in Stage-2. Since noisy pixels will adversely impact the learning of matched wavelet basis, noisy pixels are dropped before basis learning. Signal $\mathbf{x}_{n,uncor}$ is used for sparse recovery in Stage-3.

Stage 2: Learning Image Matched DWT

In this stage, we learn wavelet basis for the row-space and the column space of the image, $\mathbf{y}_r$ and $\mathbf{y}_c$, respectively, using the method proposed in Section 3.1.1. This results in learned separable DWT corresponding to the image corrupted with impulse noise. This learned separable DWT is used as the sparsifying transform in stage-3 below for the recovery of full image or impulse denoising of the image.

Stage 3: Impulse denoising using Sparse Recovery

Once all the wavelet filters for the row-space and the column-space are learned using Stage-2, the learned wavelet transform is used to recover the unknown image

$\mathbf{X}$. For this, we use the vector $\mathbf{x}_{n,uncor}$ (picked by the operator Φ) formed in stage-1 of the pixels identified as uncorrupted and equate it to

$$\begin{aligned} \mathbf{x}_{n,uncor} =& \mathbf{\Phi}\mathbf{x}_n \\ \approx& \mathbf{\Phi}\mathbf{\Psi}\mathbf{r}, \end{aligned} \tag{4.2}$$

where $\mathbf{\Psi}$ represents the learned wavelet basis operator and $\mathbf{r}$ are the wavelet transformed coefficients of original signal $\mathbf{x}$ requires to be recovered/denoised. This problem corresponds to compressed sensing [46] and can be solved by using sparse recovery method as below:

$$\tilde{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \, ||\mathbf{r}||_1 \text{ subject to } \mathbf{x}_{n,uncor} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{r}. \tag{4.3}$$

Full image is reconstructed as: $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}}$.

Table 4.3: Predict/Update stage filter with analysis side filters learned with columns of image ‘Boat’ at 90% impulse noise

$t[n]$	0.0281 0.4731 0.4545 0.0426]
$s[n]$	[-0.0369 0.2900 0.2721 -0.0240]
$h_0[n]$	[0.0016 0 0.0044 -0.0369 -0.1260 0.2900 0.7412 0.2721 -0.1260 -0.0240 0.0037 0 0.0007]
$h_1[n]$	[-0.0426 0 -0.4545 1.0000 -0.4731 0 -0.0281]

4.2.2.2 Experiments and Results

We apply the proposed method of wavelet transform learning based impulse denoising on two standard test images ‘Lena’ and ‘Boat’ shown in Fig. 4.12a and 4.12g respectively. We have considered 4-tap predict and update filters. The learned predict and update filters along with the analysis filters ($\mathbf{h_0}$ and $\mathbf{h_1}$) obtained with the proposed method at 90% noise ratio for ‘Boat’ are provided in Table-4.3.

We use 3-level L-Pyramid wavelet decomposition in these experiments. The impulse denoising results are quantitatively measured in terms of PSNR and tabulated in Table-4.4 and 4.5. Fixed valued impulse noise ratio of 10% to 90% is considered.

The proposed impulse denoising is compared with simple median filtering [124]

Table 4.4: Impulse denoising results in PSNR (dB) on 'Lena'

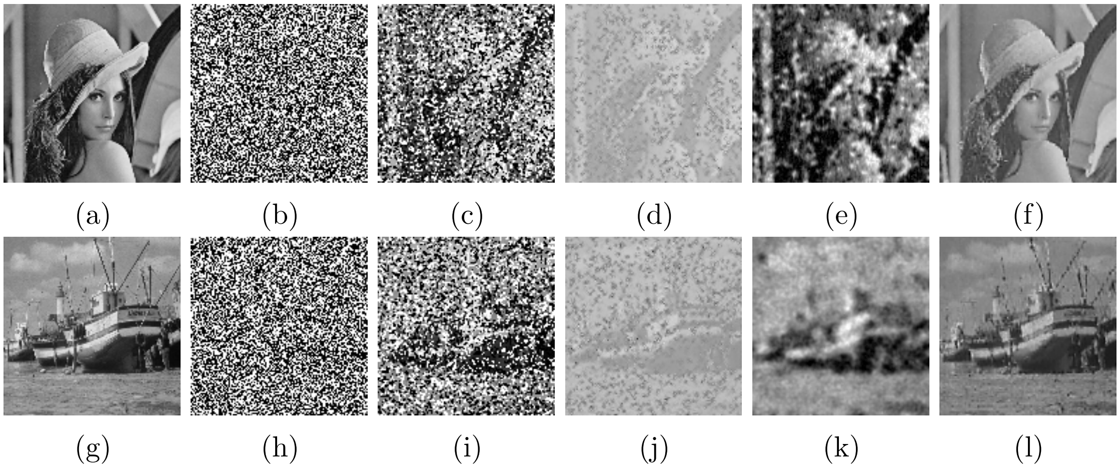
Noise ratio (in %)	Methods used					
	Median filtering based		Sparse recovery based			
	SMF	ISMF	[1]	db4	Bi 9/7	WTL-I
10	28.2	37.8	43.7	41.3	42.4	**43.9**
30	25.2	29.7	37.0	35.3	36.2	**37.9**
50	23.2	26.0	31.1	31.0	32.0	**33.8**
70	17.9	19.1	26.2	26.4	27.5	**30.1**
90	8.1	8.5	15.7	11.0	10.3	**25.6**

Table 4.5: Impulse denoising results in PSNR (dB) on 'Boat'

Noise ratio (in %)	Methods used					
	Median filtering based		Sparse recovery based			
	SMF	ISMF	[1]	db4	Bi 9/7	WTL-I
10	25.0	34.7	34.8	39.3	40.1	**41.6**
30	23.2	28.0	29.0	32.8	33.9	**35.5**
50	21.8	24.5	26.2	28.5	29.6	**31.5**
70	17.4	18.7	22.2	24.4	25.2	**28.0**
90	8.4	8.8	18.5	10.6	10.1	**23.3**

(SMF), ideal switching median filtering (ISMF), and sparse recovery based denoising method [1]. ISMF assumes all the impulse noise locations to be known apriori and applies median filtering only to the corrupted impulse locations. Thus, ISMF indicates the best performance among all possible variants of switching median filtering. A 7 × 7 window size has been used in both SMF and ISMF.

To ascertain the performance of learned wavelet transform, results have also been tabulated by replacing Stage-2 of the proposed method with standard orthogonal (db4) and biorthogonal (Bi-9/7) wavelets. From Tables 4.4 and 4.5, it is noted that ISMF provides a huge improvement over SMF especially at lower noise ratios. However, these results are inferior to sparse recovery based denoising methods showing the superiority of sparse recovery based methods over median filtering based methods. Also, it is observed that the learned wavelet transform outperforms the method of [1] and the standard wavelet transforms at all noise ratios. At large noise ratio of 90%, ISMF, the method of [1], and standard wavelets are not able to retain the structure of the images, whereas learned wavelet transform recovers good quality of images as shown visually in Fig. 4.12.



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

(g) (h) (i) (j) (k) (l)

Figure 4.12: Qualitative comparison of denoising of images corrupted with 90% impulse noise. First row: (a) Original Lena image, (b) Noisy image, (c) Image reconstructed with ISMF, (d) with Bi 9/7, (e) with [1], and (e) with learned separable DWT. Second row: similar to first row on 'Boat'.

4.3 Summary

Methods are presented in this chapter to learn separable dyadic wavelet transform from the images. Particularly, Two methods are presented. The first method can be used in applications, where one has the access of full original image. The second method can be employed in inverse problems, where the degraded image is present. A new multi-level wavelet decomposition strategy for the images is also presented, named as L-Pyramid wavelet decomposition. L-Pyramid wavelet decomposition is observed to perform better than the existing wavelet decomposition strategy in CS based image reconstruction. The applications of the proposed method of learning separable DWT is presented in CS based image reconstruction and in impulse denoising. Compressive sensing based image reconstruction of images is presented using partial canonical identity (PCI) matrix as the sensing matrix. PCI sensing matrix is observed to be much faster in terms of both sensing as well as reconstruction of the images than existing sensing matrices but its reconstruction performance is inferior than the existing sensing matrices when existing wavelet transform is used. However, the use of learned wavelet transform enhances the capability of PCI sensing matrix and overall, PCI sensing matrix outperforms the existing sensing matrices both in terms of time in reconstruction as well as reconstruction quality. Impulse denoising of natural images is also explored using the proposed method of DWTL. Impulse denoising is posed as sparse recovery problem and the learned separable DWT is used as the sparsifying transform for the sparse recovery of the images. The proposed method is observed to outperform the existing methods of impulse denoising.

Chapter 5

RWTL: Rational Wavelet Transform Learning using Lifting framework

Rational wavelet transform (RWT) can prove helpful in applications requiring non-uniform partitioning of the signal spectrum. Speech or audio signal processing are examples of these applications [68, 69, 70]. Decimation factors of the corresponding rational filterbank (RFB) may be different in each subband and are rational numbers [71].

As discussed in chapter-2, lifting has been shown to be a simple yet powerful tool for custom design/learning of wavelet [32] and has several advantages. However, the lifting framework is used only for the custom design/learning of dyadic (or M-band) wavelets [34, 42, 40, 84] and has not been explored to learn RWT to the best of our knowledge. Moreover, the existing architecture of lifting framework cannot be extended directly to rational wavelet system because of different sample/signal rates in subband branches.

Motivated with the above discussion, we propose to use lifting framework to learn signal-matched RWT in this chapter. For this, we extend the existing lifting framework from dyadic to rational wavelet and then use it to learn signal-matched RWT. As variable sample rate is present in different branches, we introduce the concept of rate-converters to equal the rates in two branches. We use the learned RWT as the sparsifying transform in applications.

Method to learn signal-matched RWT is presented in section5.1 for 1-D signal along with the application of the learned RWT in CS based reconstruction of speech, music, and ECG signals. Method to learn separable rational wavelet transform from the images is presented in section 5.2 and the learned RWT is used in the application of CS based reconstruction of natural images.

5.1 RWTL for 1-D signals

In this section, we propose the method of RWTL for 1-D signals along with its application in CS based reconstruction of signals.

5.1.1 Proposed Method of RWTL

In this section, we present the proposed RWTL method of learning signal-matched 2-channel/band rational wavelet system using the lifting framework. First, we propose the extension of 2-band dyadic Lazy wavelet transform to M-band Lazy wavelet and find its equivalent 2-channel rational Lazy filterbank. Both these structures will be used in the proposed work.

5.1.1.1 *M*-band and Rational Lazy Wavelet System

As explained earlier, a 2-band *Lazy* wavelet system divides an input signal into two disjoint signals. Similarly, on the analysis side, an M-band Lazy wavelet system divides an input signal $x[n]$ into M disjoint sets of data samples, given by $v_i[n], \;\; i = 0, 1, 2, ..., M-1$, where $v_i[n] = x[Mn + i]$. At the synthesis end, these M disjoint sample sets are combined or interlaced to reconstruct the signal at the output. An M-band Lazy wavelet can be designed with the following choice of analysis and synthesis filters in Figure-5.1:

$$G_i(z) = z^{i} \qquad i = 0, 1, 2, ..., M-1, \tag{5.1}$$

$$F_i(z) = z^{-i} \qquad i = 0, 1, 2, ..., M-1. \tag{5.2}$$

In order to obtain the corresponding 2-band rational Lazy wavelet system with dilation factor $\frac{M}{q_1}$ (Figure-5.2) from the M-band Lazy wavelet, we use (2.6) and

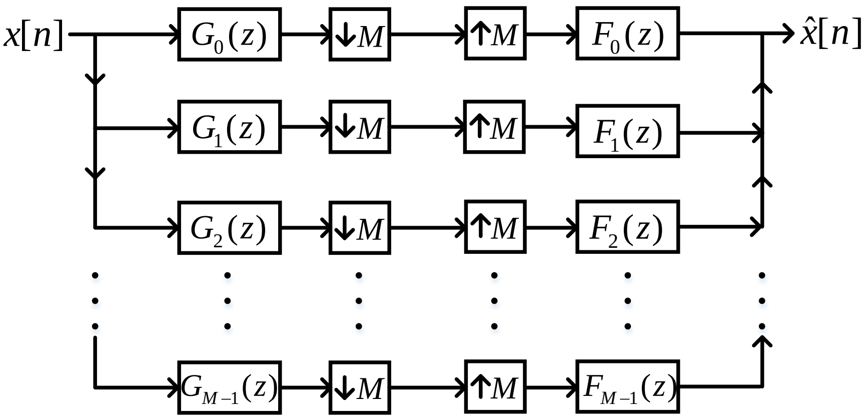


Figure 5.1: M-band wavelet structure

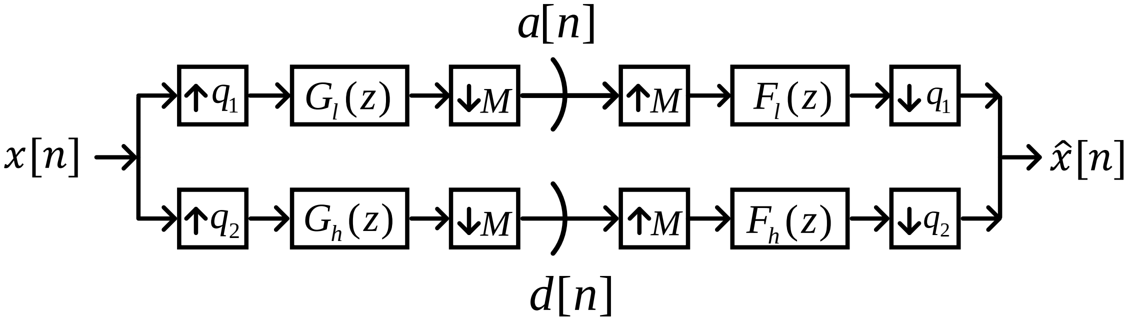


Figure 5.2: General 2-band rational wavelet structure

(2.7) to obtain lowpass analysis and synthesis filters as:

$$
\begin{aligned}
G_l(z) &= \sum_{i=0}^{q_1-1} z^{-iM} z^{iq_1} = \sum_{i=0}^{q_1-1} z^{-iq_2}, \\
F_l(z) &= \sum_{i=0}^{q_1-1} z^{iM} z^{-iq_1} = \sum_{i=0}^{q_1-1} z^{iq_2}.
\end{aligned}
\tag{5.3}
$$

Similarly, we use (2.8) and (2.9) to obtain the corresponding highpass analysis and synthesis filters of rational Lazy wavelet as:

$$
\begin{aligned}
G_h(z) &= \sum_{i=0}^{q_2-1} z^{-iM} z^{(i+q_1)q_2} = z^{q_1 q_2} \sum_{i=0}^{q_2-1} z^{-iq_1}, \\
F_h(z) &= \sum_{i=0}^{q_2-1} z^{iM} z^{-(i+q_1)q_2} = z^{-q_1 q_2} \sum_{i=0}^{q_2-1} z^{iq_1}.
\end{aligned}
\tag{5.4}
$$

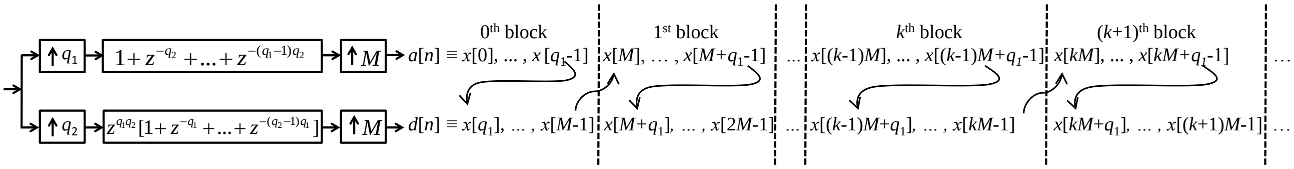

↑q_1
$1+z^{-q_2}+\ldots+z^{-(q_1-1)q_2}$
↑M
↑q_2
$z^{q_1q_2}[1+z^{-q_1}+\ldots+z^{-(q_2-1)q_1}]$
↑M
0th block
1st block
kth block
(k+1)th block
$a[n] \equiv x[0], \ldots, x[q_1\text{-}1]$
$x[M], \ldots, x[M+q_1\text{-}1]$
$x[(k\text{-}1)M], \ldots, x[(k\text{-}1)M+q_1\text{-}1]$
$x[kM], \ldots, x[kM+q_1\text{-}1]$
$d[n] \equiv x[q_1], \ldots, x[M\text{-}1]$
$x[M+q_1], \ldots, x[2M\text{-}1]$
$x[(k\text{-}1)M+q_1], \ldots, x[kM\text{-}1]$
$x[kM+q_1], \ldots, x[(k+1)M\text{-}1]$
…

Figure 5.3: Analysis side of rational Lazy wavelet; Each block consists of M samples of input signal $x[n]$ divided into $a[n]$ and $d[n]$.

The above filters form the rational Lazy wavelet system equivalent of M-band Lazy wavelet transform. To learn signal-matched rational wavelet system, we start with the rational Lazy wavelet. This provides us initial filters $G_l(z)$, $G_h(z)$, $F_l(z)$, and $G_h(z)$. These filters are updated according to signal characteristics to obtain the signal-matched rational wavelet system. Analysis highpass and synthesis lowpass filters are updated in the predict stage, whereas analysis lowpass and synthesis highpass filters are updated in the update stage. These stages are described in the following subsections.

5.1.1.2 Predict Stage

As discussed in section 2.1, a 2-band Lazy wavelet system divides the input signal into two disjoint sample sets, wherein one set is required to be predicted using the other set of samples. In the conventional 2-band lifting framework with integer downsampling ratio of M=2 in both the branches (refer to Figure 2.1 and 2.2), the output sample rate of $x_e[n]$ and $x_o[n]$ is equal. However, in a 2-band rational wavelet system, the output sample rate of two branches is unequal. Hence, the predict polynomial branch of a conventional 2-band dyadic design cannot be used.

For example, from Figure-5.2, we note that in a 2-band rational wavelet system, higher (lower) rate branch samples used in prediction should be downsampled (upsampled) by a factor of k_p to equal the rate to the lower (higher) predicted branch samples, defined as:

$$k_p = \frac{q_1}{q_2}, \tag{5.5}$$

where q_1/M is the rate of '*predicting*' branch samples and q_2/M is the rate of 'to be *predicted*' branch samples.

We propose to predict the lower branch samples with the help of upper branch samples. Here, input signal $x[n]$ is divided into two disjoint sets, $x[kM+i], i = 0, 1, ..., q_1 - 1$ and $x[kM+j], j = q_1, q_1+1, ..., M-1$. We label these outputs as $\boldsymbol{a}$ and $\mathbf{d}$, respectively (Figure 5.3). Here, $k = 0, 1, ..., L-1$, where $N = LM$ is the length of input signal $x[n]$ and, without loss of generality, is assumed to be a multiple of M. Thus, a given input signal is divided into L blocks of M size each at the subband output of 2-channel rational Lazy wavelet system. Here, first q_1 samples of every block move to the upper branch as a block of $\boldsymbol{a}$ and next q_2 samples (such that $q_1 + q_2 = M$) move to the lower branch as a block of $\mathbf{d}$. Or in other words, the rate of upper branch output is q_1 samples per block and the rate of lower branch output is q_2 samples per block. Figure-5.3 shows these blocks of outputs $\boldsymbol{a}$ and $\mathbf{d}$ explicitly.

This motivates us to introduce the concept of *rate converter* that equals the output sample rate of the upper *predicting* branch to that of the lower *predicted* branch to enable predict branch design. In other words, the output of upper branch $a[n]$ is upsampled by q_2 and downsampled by q_1 to match the rate of lower branch output $d[n]$. It should be noted that the above downsampler and upsampler cannot be connected consecutively. Since an upsampler introduces spectral images in the frequency domain, it is generally preceded by a filter, also known as interpolator or anti-imaging filter. On the other hand, a downsampler stretches the frequency spectrum of the signal, that's why it is followed by a filter known as anti-aliasing filter. Both these conditions imply that if a rate converter is required with downsampling factor of $k_p = \frac{q_1}{q_2}$, a polynomial $R_p(z)$ should be incorporated which acts as the anti-imaging filter for the upsampler and at the same time as the anti-aliasing filter for the downsampler. The structure of this polynomial is $R_p(z) = 1 + z^{-1} + ... + z^{-(q_2-1)}$. The placement of this polynomial is shown in Figure 5.4. We call the complete branch with rate converter k_p and polynomial $R_p(z)$ as the predict rate converter.

Definition 1. *Predict Rate Converter: It is a combination of the polynomial $R_p(z) = 1 + z^{-1} + ... + z^{-(q_2-1)}$ preceded by a q_2-fold upsampler and followed by a q_1-fold downsampler as shown in Figure-5.4.*

The predict rate converter with this choice of $R_p(z)$ will lead to an appropriate repetition or drop of samples of $a[n]$ such that the total number of samples in $a_4[n]$ and $d[n]$ in any k^{th} block contains an equal number of samples.

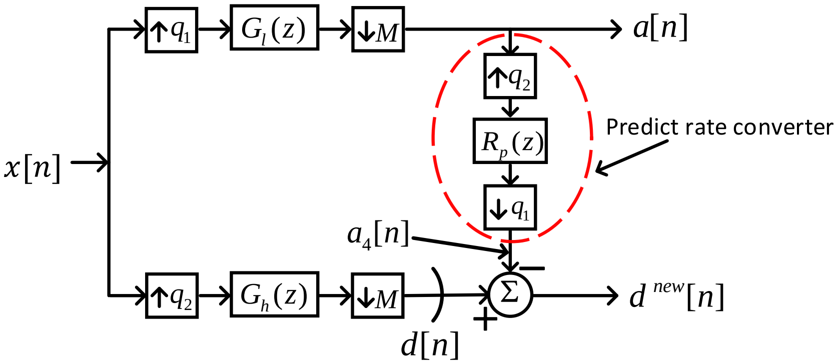


Figure 5.4: Illustration of Predict Rate Converter

Next, we present three subsections: 1) on the structure of predict stage filter $T(z)$, 2) how to learn $T(z)$ from a given signal, and 3) how to update all the filters of the corresponding 2-channel rational filterbank using the learned $T(z)$.

Structure of predict stage filter $T(z)$

Once the outputs of two analysis filterbank branches are equal, lower branch samples are predicted from the upper branch samples with the help of predict stage filter $T(z)$. This filter is introduced after the polynomial $R_p(z)$ as shown in Figure-5.5. For good prediction, the current sample of input signal $\mathbf{x}$ contained in $d[n]$ should be predicted from its past and future samples. The k^{th} block of $a[n]$ has preceding samples and $(k+1)^{th}$ block of $a[n]$ has future samples of the k^{th} block of $d[n]$ as is evident from Figure-5.3. Thus, the structure of the predict polynomial $T(z)$ should be chosen appropriately. This is presented with a theorem for 2-tap $T(z)$ as below.

Theorem 3. *A 2-tap predict stage filter $T(z) = z^{q_2 q_1}(t[0]z^{-q_2} + t[1])$ ensures that every sample in the k^{th} block of $d[n]$ branch is predicted from the k^{th} and/or the $(k+1)^{th}$ block samples of $a[n]$.*

Proof. As discussed in section 5.1.1.2, on passing the input signal $x[n]$ through the Lazy filterbank of analysis side, we obtain approximate and detail coefficients ($\mathbf{a}$ and $\mathbf{d}$, respectively) in the form of blocks as shown in Figure-5.3. The k^{th} block of approximate and detail signals are given by:

$$a^k[m] = a[(k-1)q_1 + m] = x[(k-1)M + m], \quad m = 0, 1, ..., q_1 - 1,$$

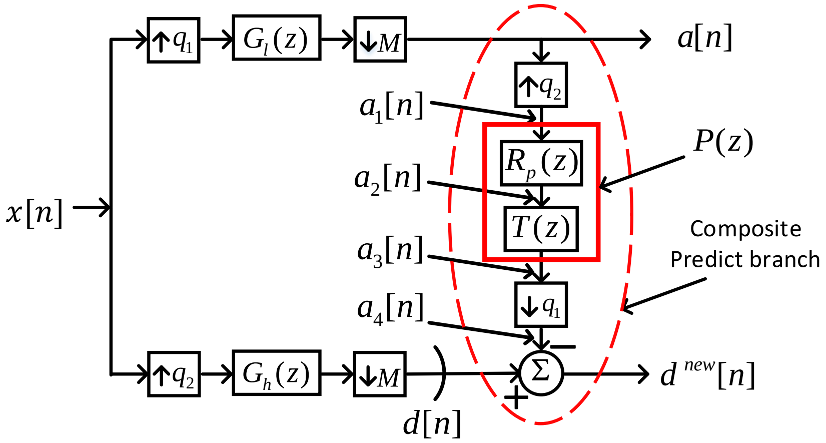


Figure 5.5: Predict Rate Converter

$$d^k[m] = d[(k-1)q_2 + m] = x[(k-1)M + q_1 + m], \quad m = 0, 1, ...q_2 - 1. \tag{5.6}$$

For prediction, first the upper branch signal $a[n]$ is passed through q_2-fold upsampler and polynomial $R_p(z)$. This leads to signal $a_2[n]$, as shown in Figure-5.5, that contains every element of $a[n]$ repeated q_2 number of times. Mathematically, the k^{th} block of signal $a_2[n]$ is given by:

$$a_2^k[m] = a\left[(k-1)q_1 + \left\lfloor \frac{m}{q_2} \right\rfloor\right], \quad m = 0, 1, ..., q_1q_2 - 1, \tag{5.7}$$

where $\lfloor . \rfloor$ denotes the floor function. Predict filter $T(z)$ is defined as a product of two polynomials $z^{q_2q_1}$ and $t[0]z^{-q_2} + \ t[1]$ and is applied to this signal $a_2[n]$. Intuitively, the first polynomial $z^{q_2q_1}$ will position the $(k+1)^{th}$ block of $a[n]$ over the k^{th} block of $d[n]$ since every block of $a_2[n]$ contains q_1q_2 number of elements. The second polynomial $(t[0]z^{-q_2} + t[1])$ chooses two consecutive samples of $a[n]$ because every sample of $a[n]$ is repeated q_2 times as explained earlier. Thus, the second term will help in choosing either both the elements of $(k+1)^{th}$ block of $a[n]$ that are future samples of $d[n]$ or in choosing one element of $(k+1)^{th}$ block and one of k^{th} block of $a[n]$. Mathematically, this is seen as below.

On passing $a_2[n]$ through the predict filter $T(z)$, the k^{th} block of signal $a_3[n]$ is obtained as:

$$a_3^k[m] = t[0]a\left[(k-1)q_1 + \left\lfloor \frac{m}{q_2} \right\rfloor + (q_1 - 1)\right] + t[1]a\left[(k-1)q_1 + \left\lfloor \frac{m}{q_2} \right\rfloor + q_1\right], \tag{5.8}$$

where $m = 0, 1, ..., q_1q_2 - 1$. This signal is passed through the q_1-fold downsampler resulting in the k^{th} block of signal $a_4[n]$ as:

$$\begin{aligned} a_4^k[m] =& t[0]a\left[(k-1)q_1 + \left\lfloor \frac{q_1 m}{q_2} \right\rfloor + (q_1 - 1)\right] + t[1]a\left[(k-1)q_1 + \left\lfloor \frac{q_1 m}{q_2} \right\rfloor + q_1\right], \\ =& t[0]a\left[kq_1 - 1 + \left\lfloor \frac{q_1 m}{q_2} \right\rfloor\right] + t[1]a\left[kq_1 + \left\lfloor \frac{q_1 m}{q_2} \right\rfloor\right], \end{aligned} \tag{5.9}$$

where $m = 0, 1, ..., q_2 - 1$. The block size or the rate of $a_4[n]$ is same as that of $d[n]$ and hence, it predicts $d[n]$ providing the k^{th} block prediction error given by:

$$d^{new,k}[m] = d^k[m] - a_4^k[m], \;\; m = 0, 1, ..., q_2 - 1. \tag{5.10}$$

From (5.6), (5.9), (5.10), and Figure-5.3, it can be noted that

(i) The first element of k^{th} block of $d[n]$, i.e., $d^k[0]$ for $m = \; 0$ is $x[(k-1)M + q_1]$ and this sample is predicted from $a[kq_1 - 1] = x[(k-1)M + q_1 - 1]$ and $a[kq_1] = x[kM]$ that are the elements of the k^{th} and $(k+1)^{th}$ blocks of $a[n]$, respectively. Also, these are the past and future samples of $x[(k-1)M + q_1]$.

(ii) The last element of k^{th} block, i.e., $d[kq_2 - 1] = x[kM - 1]$ for $m = q_2 - 1$ is predicted from $x\left[kM + q_1 - 2 - \left\lfloor \frac{q_1}{q_2} \right\rfloor\right]$ and $x\left[kM + q_1 - 1 - \left\lfloor \frac{q_1}{q_2} \right\rfloor\right]$ that are the elements of the $(k+1)^{th}$ block of $a[n]$.

(iii) From (i) and (ii) above, it is clear that in between elements of the k^{th} block of $d[n]$ will be predicted from only the k^{th} and $(k+1)^{th}$ block elements of $a[n]$.

In fact, $\lceil \frac{q_2}{q_1} \rceil$ elements of a block of $d[n]$ (where $\lceil . \rceil$ denotes the ceil function) are predicted using the past and future samples, i.e., from the k^{th} and $(k+1)^{th}$ block elements of $a[n]$, while rest of the elements are predicted from the elements of $(k+1)^{th}$ block of $a[n]$. This proves Theorem-1. □

Although, it is not possible to illustrate the prediction of samples in rational wavelet similar to dyadic wavelet case (as shown in Fig. 3.3) because of the generic sampling ratio in two branches, we present a specific case of rational wavelet with sampling ratio of $(\frac{2}{3}, \frac{1}{3})$ in Fig. 5.6 using 2-tap predict stage filter $T(z)$.

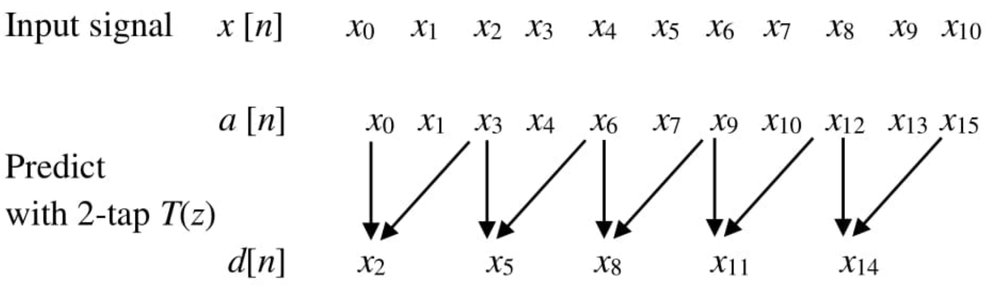

Input signal $x[n]$
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}
$a[n]$
x_0 x_1 x_3 x_4 x_6 x_7 x_9 x_{10} x_{12} x_{13} x_{15}
Predict with 2-tap $T(z)$
$d[n]$
x_2 x_5 x_8 x_{11} x_{14}

Figure 5.6: Illustration of predict in rational wavelet with sampling ratio $(\frac{2}{3}, \frac{1}{3})$ using 2-tap filter $T(z)$

We name the modified rate converter branch incorporating polynomial $T(z)$, shown in Figure-5.5, as the '*Composite Predict Branch*'.

This is to note that the above choice of polynomial provides the best possible generic solution for prediction using nearest neighbors for different values of q_1 and q_2. For example, if the first polynomial $z^{q_2 q_1}$ of $T(z)$ is omitted, one may note that samples in $d[n]$ will be predicted from far away past samples or far away future samples depending on the second polynomial. Thus, although $T(z)$ can be chosen in many ways, we choose to use the polynomial provided in Theorem-1.

The above theorem of 2-tap predict filter can be easily extended to obtain an N_p-tap filter with even N_p. For example, a 4-tap $T(z)$ can be given by

$$T(z) = z^{q_2 q_1}(t[0]z^{-2q_2} + t[1]z^{-q_2} + t[2] + t[3]z^{q_2}), \tag{5.11}$$

that will choose four consecutive samples of $a[n]$ and are from the immediate neighboring blocks of $d[n]$. In general, an even length N_p-tap filter $T(z)$ is given by

$$T(z) = z^{q_2 q_1} z^{-\frac{N_p}{2} q_2} \sum_{k=0}^{N_p - 1} t[k] z^{k q_2}. \tag{5.12}$$

Estimation of predict stage filter $T(z)$ from a given signal

In order to learn an N_p-length predict stage filter $T(z)$, we consider the prediction error $d^{new}[n]$ shown in Figure-5.5 as below:

$$\begin{aligned} d^{new}[n] =& d[n] - a_4[n] \\ =& d[n] - \sum_{j=0}^{N_p-1} \sum_{k=0}^{q_2-1} t[j] a_1[q_1 n - (j+k)]), \end{aligned} \tag{5.13}$$

and minimize it using the Least Squares (LS) criterion yielding the solution of $T(z)$ as below:

$$\begin{aligned} \hat{\mathbf{t}} =& \underset{\mathbf{t}}{\text{argmin}} \sum_{n=0}^{N_d-1} (d^{new}[n])^2 \\ =& \underset{\mathbf{t}}{\text{argmin}} \sum_{n=0}^{N_d-1} (d[n] - \sum_{j=0}^{N_p-1} \sum_{k=0}^{q_2-1} t[j] a_1[q_1 n - (j+k)]))^2 \end{aligned} \tag{5.14}$$

where N_d is the length of the difference signal $\mathbf{d}^{new}$ and $\mathbf{t} = [t[0]t[1] \ldots t[N_p - 1]]'$ is the column vector of elements of the polynomial $T(z)$. The above equation can be written in the vector form as below:

$$\tilde{\mathbf{t}} = \underset{\mathbf{t}}{\text{argmin}} ||\mathbf{b} - \mathbf{A}\mathbf{t}||_2^2, \tag{5.15}$$

where $b(i+1) = d[i]$, $A(i+1, l+1) = \sum_{k=0}^{q_2-1} a_1[q_1 i - (l+k)] \;\; \forall \;\; i \in \{0, 1, ..., N_d - 1\}$ and $l \in \{0, 1, ..., N_p - 1\}$. The above equation can be solved using the following closed-form solution:

$$\tilde{\mathbf{t}} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{b}. \tag{5.16}$$

Update of RFB filters using learned $T(z)$

For the dyadic wavelet design using lifting as discussed in section 2.1, predict polynomial is used to update analysis highpass and synthesis lowpass filters using (2.1) and (2.2). However, we require to derive similar equations for a rational filterbank. Before we present this work, let us look at a Lemma that will be helpful in defining these equations.

Lemma 5.1. *A structure containing a filter $H(z)$ followed by an M-fold downsampler and preceded by an M-fold upsampler (Figure 5.7a) can be replaced by an equivalent filter $\tilde{H}(z)$ (Figure 5.7b), which is given by:*

$$\tilde{H}(z) = \frac{1}{M}\sum_{r=0}^{M-1} H(z^{\frac{1}{M}}W_M^r), \tag{5.17}$$

where $W_M = \exp(-j\frac{2\pi}{M})$.

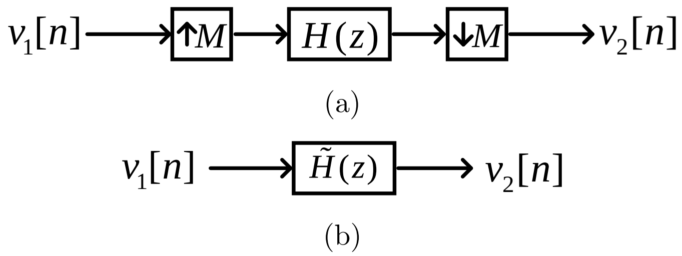
Proof. Consider the structure in Figure 5.7a, which shows the filter $H(z)$ followed by an upsampler preceded by an downsampler both M fold. Let a signal $v_1[n]$ is passed to the structure. Let Z-transform of signal be $V_1(z)$. Z-transform of the signal obtained after passing $V_1(z)$ to M-fold upsampler is $V_1(z^M)$. This upsampled signal is passed through the filter $H(z)$, which gives the signal $V_1(z^M)H(z)$. Let this signal be $\tilde{V}_1(z)$. This signal, when passed through the downsampler, gives us the following signal:

$$\begin{aligned} V_2(z) =& \frac{1}{M}\sum_{r=0}^{M-1} \tilde{V}_1(z^{\frac{1}{M}}W_M^r) \\ =& \frac{1}{M}\sum_{r=0}^{M-1} V_1(zW_M^{rM})H(z^{\frac{1}{M}}W_M^r) \\ =& \frac{1}{M}\sum_{r=0}^{M-1} V_1(z)H(z^{\frac{1}{M}}W_M^r) \\ =& V_1(z)\frac{1}{M}\sum_{r=0}^{M-1} H(z^{\frac{1}{M}}W_M^r) \\ =& V_1(z)\tilde{H}(z), \end{aligned} \tag{5.18}$$

where $W_M^{rM} = 1$ is used in third step above and $\tilde{H}(z) = \frac{1}{M}\sum_{r=0}^{M-1} H(z^{\frac{1}{M}}W_M^r)$. This is shown in Figure 5.7b.

The above relation (5.18) is equivalent to applying a single filter $\tilde{H}(z)$ to the input signal $v_1[n]$. Hence, the two structures in Figure 5.7 are equivalent. □

Next, we present Theorem-4 that provides the structure of updated analysis



(a)

(b)

Figure 5.7: Filter structure in (a) is equivalent to filter in (b)

highpass filter $G_h^{new}(z)$ using $T(z)$ for an RFB.

Theorem 4. *The analysis highpass filter of a 2-channel rational filterbank can be updated using the predict polynomial $T(z)$ used in the 'composite predict branch' via the following relation:*

$$G_h^{new}(z) = G_h(z) - \frac{1}{q_1}\sum_{k=0}^{q_1-1} G_l(z^{\frac{q_2}{q_1}} W_{q_1}^{q_2 k}) P(z^{\frac{M}{q_1}} W_{q_1}^{Mk}), \tag{5.19}$$

where $P(z) = R_p(z)T(z)$.

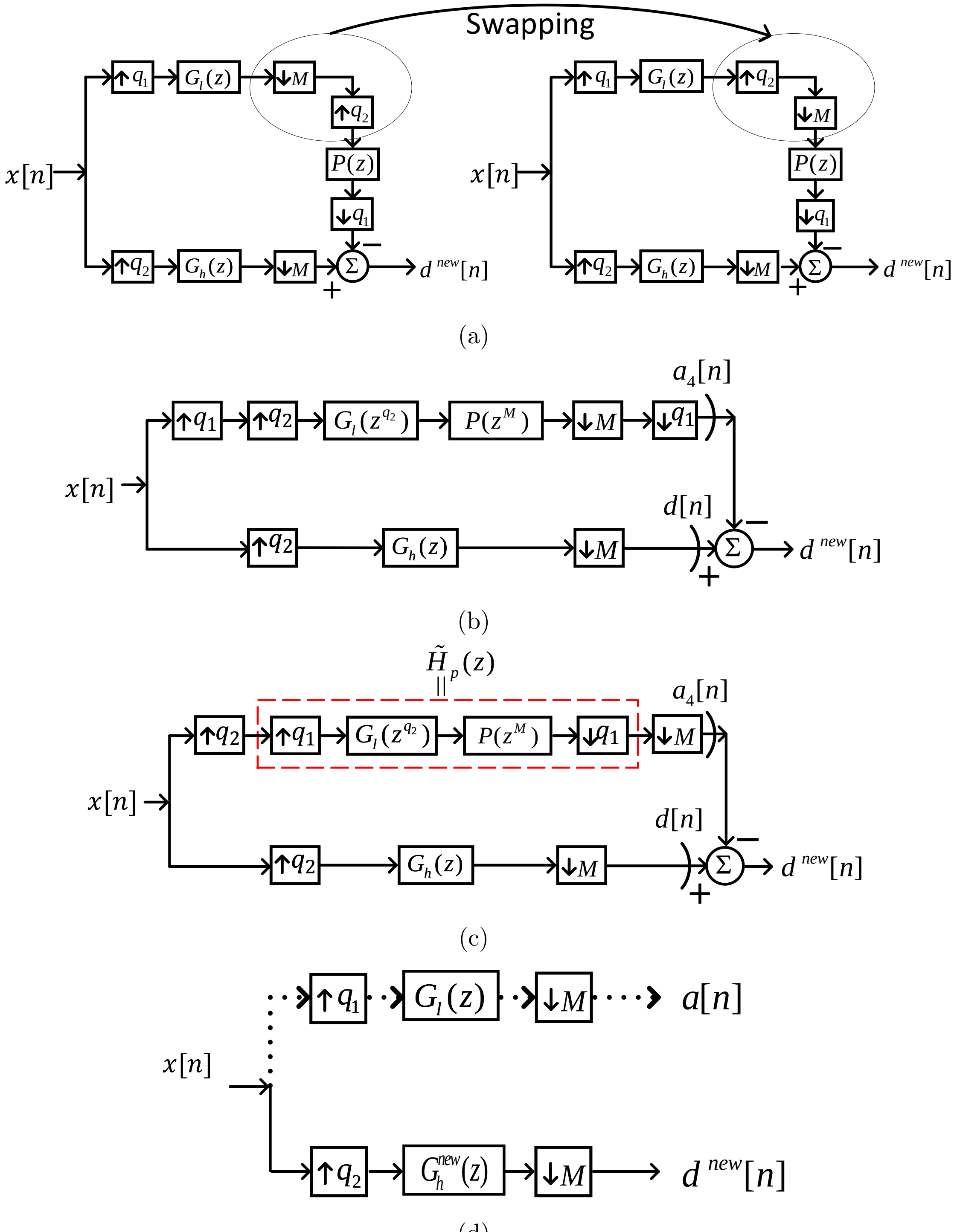
Proof. Refer to Figure-5.8. Since M and q_2 are relatively prime, the corresponding downsampler and upsampler can be swapped to simplify the structure (refer Figure 5.8a). Using noble identities [85] and further simplification, we obtain the structure in Figure 5.8b. As two downsampler or upsampler can swap each other, we obtain the structure in Figure 5.8c. The part in red dotted rectangle in the figure can be replaced by an equivalent filter, $\tilde{H}_p(z)$ using (5.17) of Lemma-1 and is given by:

$$\tilde{H}_p(z) = \frac{1}{q_1}\sum_{r=0}^{q_1-1} G_l(z^{\frac{q_2}{q_1}} W_{q_1}^{q_2 r}) P(z^{\frac{M}{q_1}} W_{q_1}^{Mr}). \tag{5.20}$$

Considering the structure in Figure 5.8c, signals $a_4[n]$, $d[n]$, and $d^{new}[n]$ can be written in Z-domain as:

$$A_4(Z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{q_2}{M}} W_M^{q_2 k}) \tilde{H}_p(z^{\frac{1}{M}} W_M^{k}), \tag{5.21}$$

$$D(Z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{q_2}{M}} W_M^{q_2 k}) G_h(z^{\frac{1}{M}} W_M^{k}), \tag{5.22}$$



(a)

(b)

(c)

(d)

Figure 5.8: Illustration for the proof of Theorem-4. Structures in (a)-(e) are equivalent of each other.

$$\begin{aligned} D^{new}(z) &= X_l(z) - X_u(z) \\ &= \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{q_2}{M}} W_M^{q_2 k})\big[G_h(z^{\frac{1}{M}} W_M^{k}) - \tilde{H}_p(z^{\frac{1}{M}} W_M^{k})\big]. \end{aligned} \tag{5.23}$$

The above relation is equivalent to applying a new filter $G_h^{new}(z)$ to the lower branch of the rational wavelet system (Figure 5.8d) and is given by

$$\begin{aligned} G_h^{new}(z) =& G_h(z) - \tilde{H}_p(z) \\ =& G_h(z) - \frac{1}{q_1}\sum_{k=0}^{q_1-1} G_l(z^{\frac{q_2}{q_1}} W_{q_1}^{q_2 k}) P(z^{\frac{M}{q_1}} W_{q_1}^{Mk}) \end{aligned} \tag{5.24}$$

This proves Theorem-2. □

Next, the synthesis lowpass filter $F_l(z)$ is updated to $F_l^{new}(z)$ as follows. First, the analysis RFB containing $G_l(z)$ and $G_h^{new}(z)$ is converted into an equivalent M-band analysis filterbank structure as shown in Figure-5.1 using (2.6) and (2.8), where lowpass filter $G_l(z)$ is transformed to q_1 upper filters of M-band analysis filters and highpass filter $G_h^{new}(z)$ of rational wavelet transforms to $M - q_1(= q_2)$ lower filters of the M-band analysis filterbank. Next, the polyphase matrix $R^{new}(z)$ is obtained by using (2.10), (3.33), and (2.14). On using $R^{new}(z)$ from (2.13) in (2.11), we obtain all M updated synthesis filters of uniformly decimated filterbank. Out of these M synthesis filters, lower $M - q_1$ filters are unchanged, while the upper q_1 filters are updated because of the predict branch. On using these filters in (2.7), we obtain $F_l^{new}(z)$.

5.1.1.3 Update Stage

In this subsection, we present the structure of update branch to be used in lifting structure of an RFB, present the learning of the update branch polynomial from the given signal, and the theorem for the update of corresponding filters of RFB.

Structure of update branch

In the predict stage, we used upper branch samples to predict the lower branch samples. In the update stage, we update the upper branch samples using the lower branch samples. Since the output sample rate of two branches is unequal, we require to downsample the lower branch samples by a factor of k_u given by:

$$k_u = \frac{q_2}{q_1}, \tag{5.25}$$

before adding this signal to $a[n]$ as shown in Figure-5.10.

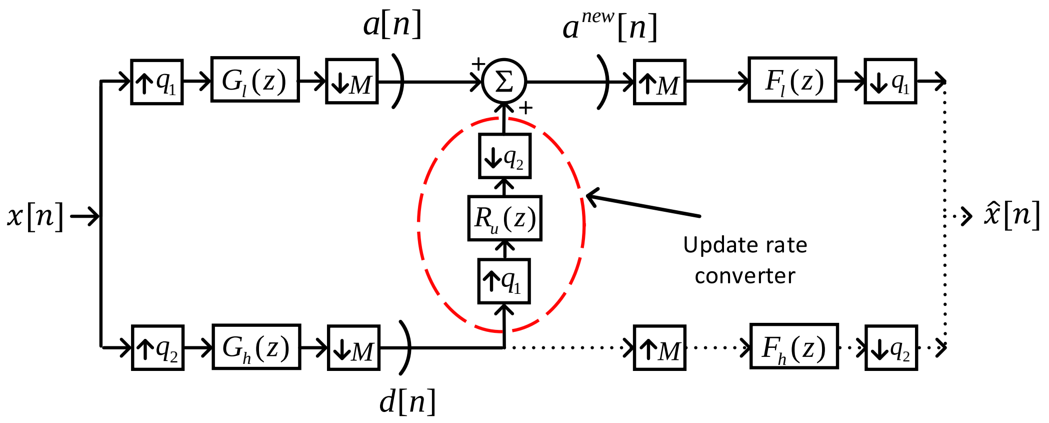


Figure 5.9: Illustration of Update Rate Converter

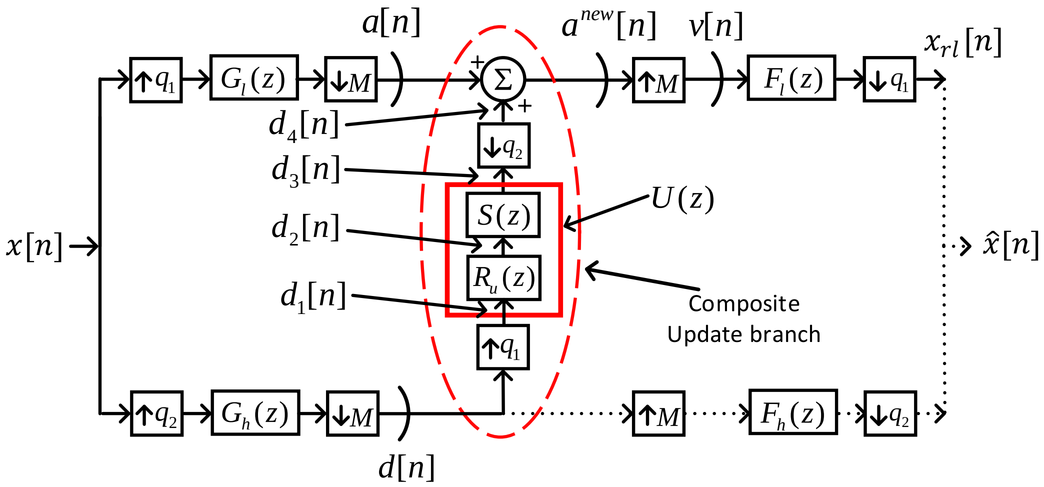


Figure 5.10: Update Stage

As explained earlier, the upsampler is required to be followed by an anti-imaging filter and downsampler should be preceded by an anti-aliasing filter. We use filter $R_u(z) = 1 + z^{-1} + ... + z^{-(q_1-1)}$ that accomplishes the same and its placement is shown in Figure-5.9. Similar to the *predict stage*, we call the update branch of Figure-5.10 as the *composite update branch* that includes *update rate converter*, update polynomial $S(z)$, and a summer.

Definition 2. *Update rate converter: It is a combination of the polynomial* $R_u(z) = 1 + z^{-1} + ... + z^{-(q_1-1)}$ *preceded by a* q_1*-fold upsampler and followed by a* q_2*-fold downsampler as shown in Figure-5.9.*

Similar to the structure of predict stage filter $T(z)$, the structure of update stage filter $S(z)$ should also be chosen carefully, so that the elements in the upper branch samples are updated only with the nearest neighbors. Below we present the theorem on the structure of a 2-tap update stage filter that ensures this.

Theorem 5. *A 2-tap update stage filter* $S(z) = s[0] + s[1]z^{-q_1}$ *ensures that every sample in the* k^{th} *block of* $a[n]$ *branch is updated from the* $(k-1)^{th}$ *and/or* k^{th} *block samples of* $d[n]$*.*

Proof. After passing the signal through the Lazy wavelet, blocks of output signal can be formed as described in section 5.1.1.2. The k^{th} block of approximation and detail coefficients is given by

$$\begin{aligned} a^k[m] =& a[(k-1)q_1 + m] = x[(k-1)M + m], \quad m = 0, 1, ..., q_1 - 1, \\ d^k[m] =& d[(k-1)q_2 + m] = x[(k-1)M + q_1 + m], \quad m = 0, 1, ...q_2 - 1, \end{aligned} \tag{5.26}$$

where superscript denoted the k^{th} block. For update, first the detail coefficients are passed through a q_1-fold upsampler and the polynomial $R_u(z)$. This leads to signal $d_2[n]$ that contain every element of $d[n]$ repeated q_1 times. Mathematically, k^{th} block of signal $d_2[n]$ is given by:

$$d_2^k[m] = d\left[(k-1)q_2 + \left\lfloor \frac{m}{q_1} \right\rfloor\right], \quad m = 0, 1, ..., q_1q_2 - 1, \tag{5.27}$$

Update filter $S(z) = s[0]+s[1]z^{-q_1}$ is applied on this signal $d_2[n]$. Unlike the predict stage, we do not require advancement of any block of $d_2[n]$ because $a_2[n]$ requires to be updated from the past and future samples of its block that are contained in the $(k-1)^{th}$ and the k^{th} blocks of $d_2[n]$ (Figure 5.3). Hence, $S(z)$ requires only one polynomial $(s[0] + s[1]z^{-q_1})$ that chooses two consecutive samples of $d[n]$ because every sample of $d[n]$ is repeated q_1 times as explained earlier.

On passing $d_2[n]$ through the update stage filter $S(z)$, we obtain

$$d_3^k[m] = s[0]d\left[(k-1)q_2 + \left\lfloor \frac{m}{q_1} \right\rfloor\right] + s[1]d\left[(k-1)q_2 + \left\lfloor \frac{m-q_1}{q_1} \right\rfloor\right], \tag{5.28}$$

where $m = 0, 1, ..., q_1q_2 - 1$. This signal is downsampled by a factor of q_2 resulting in the k^{th} block of $d_4[n]$ given by

$$d_4^k[m] = s[0]d\left[(k-1)q_2 + \left\lfloor \frac{q_2m}{q_1} \right\rfloor\right] + s[1]d\left[(k-1)q_2 + \left\lfloor \frac{q_2m-q_1}{q_1} \right\rfloor\right], \tag{5.29}$$

where $m = 0, 1, ..., q_1 - 1$. Signal $d_4[n]$ helps with the update of signal in the upper branch.

From (5.29), we note that

(i) The first term of the $a^k[m]$, i.e., $a^k[0] = x[(k-1)M]$ for $m = 0$ is updated with $d[(k-1)q_2] = x[(k-1)M + \ q_1]$ and $d[(k-1)q_2 - 1] = x[(k-1)M - 1]$ that are the elements of the k^{th} and $(k-1)^{th}$ blocks of $d[n]$, respectively. Also, these are the past and future samples of $x[(k-1)M]$.

(ii) The last term of $a^k[m]$, i.e., $a[kq_1 - 1] = x[(k-1)M + q_1 - 1]$ for $m = q_1 - 1$ is updated with $x\left[kM - \left\lfloor \frac{q_2}{q_1} \right\rfloor - 1\right]$ and $x\left[kM - \left\lfloor \frac{q_2}{q_1} \right\rfloor - 2\right]$ that are the elements in the k^{th} block of detail signal.

(iii) From (i) and (ii) above, it is clear that in between elements of the k^{th} block of $a[n]$ will be updated from only the $(k-1)^{th}$ and k^{th} block elements of $d[n]$.

This proves Theorem-3. □

Fig. 5.11 illustrates the sample updates in rational wavelet with sampling ratio $(\frac{2}{3}, \frac{1}{3})$ using 2-tap update stage filter $S(z)$. Predict stage filter $T(z)$ is also assumed to be 2-tap in this case.

Similar to the predict stage polynomial, the above choice of polynomial provides the best possible generic solution for update using nearest neighbors for different values of q_1 and q_2. Thus, although $S(z)$ can be chosen in many ways, we choose to use the polynomial provided in Theorem-3.

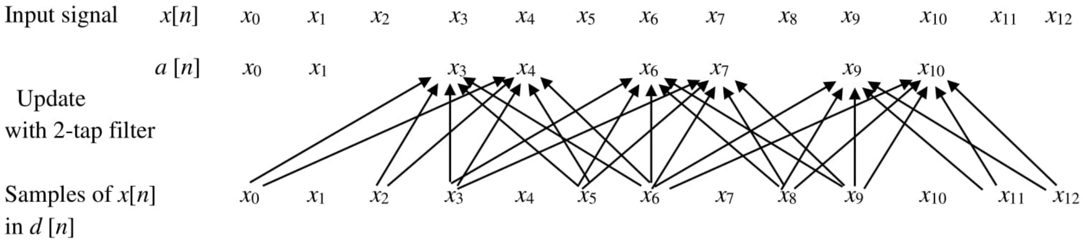

Input signal $x[n]$
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}
$a[n]$
x_0 x_1 x_3 x_4 x_6 x_7 x_9 x_{10}
Update with 2-tap filter
Samples of $x[n]$ in $d[n]$
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}

Figure 5.11: Illustration of sample updates in rational wavelet with sampling ratio $(\frac{2}{3}, \frac{1}{3})$ using 2-tap filter $S(z)$. Predict stage filter is also assumed to be 2-tap in this case.

The above theorem of 2-tap update filter can be easily extended to obtain an even length N_s-tap filter. For example, a 4-tap $S(z)$ can be given by

$$S(z) = s[0]z^{q_1} + s[1] + s[2]z^{-q_1} + s[3]z^{-2q_1}, \tag{5.30}$$

that will choose four consecutive samples of $d[n]$ and are from the immediate neighboring blocks of $a[n]$. In general, an even length N_s-tap filter $S(z)$ can be defined by the following relation:

$$S(z) = z^{(\frac{N_s}{2}-1)q_1} \sum_{k=0}^{N_s-1} s[k]z^{-kq_1}, \tag{5.31}$$

where N_s is the length of the update stage filter, $S(z)$ and is assumed to be even.

Learning of update stage filter $S(z)$ from a given signal

In order to learn an N_s-length update stage filter $S(z)$, we consider the updated signal $a^{new}[n]$ shown in Figure-5.10 as below:

$$\begin{aligned} a^{new}[n] =& a[n] + d_4[n] \\ =& a[n] + \sum_{j=0}^{N_s-1} \sum_{k_1=0}^{q_1-1} s[j]d_1[q_2 n - (j + k_1)] \end{aligned} \tag{5.32}$$

On passing these approximate coefficients through an M-fold upsampler (Figure-5.10), we obtain

$$v[n] = \begin{cases} a^{new}\left(\frac{n}{M}\right) & \text{if } n \text{ is a multiple of } M \\ 0 & \text{otherwise.} \end{cases} \tag{5.33}$$

This signal $v[n]$ is passed through the synthesis lowpass filter $F_l(z)$ followed by a q_1-fold downsampler as shown in Figure-5.10 to obtain

$$\begin{aligned} x_{rl}[n] &= \sum_{k_2=0}^{L_{f_l}-1} f_l[k_2]v[q_1 n - k_2] \\ &= \sum_{k_2=0}^{L_{f_l}-1} f_l[k_2]\{a[\frac{q_1 n - k_2}{M}] + \sum_{j=0}^{N_s-1}\sum_{k_1=0}^{q_1-1} s[j]d_1[\frac{q_1 q_2 n - q_2 k_2 - M(j+k_1)}{M}]\} \end{aligned} \tag{5.34}$$

where $\mathbf{x}_{rl}$ is the reconstructed signal at the lowpass branch of synthesis side with the same length as that of the input signal $\mathbf{x}$. L_{f_l} is the length of the synthesis lowpass filter.

Assuming input signals to be rich in low frequency, most of the energy of the input signal moves to lowpass branch. Hence, signal $x_{rl}[n]$ is assumed to be the close approximation of the input signal $x[n]$. Correspondingly, the following optimization problem is solved to learn the update stage filter $S(z)$:

$$\begin{aligned} \hat{\mathbf{s}} =& \underset{\mathbf{s}}{\text{argmin}} \sum_{n=0}^{N-1} (x[n] - x_{rl}[n])^2 \\ =& \underset{\mathbf{s}}{\text{argmin}} \sum_{n=0}^{N-1} \left\{ x[n] - \sum_{k_2=0}^{L_{f_l}-1} f_l[k_2]\{a[\frac{q_1 n - k_2}{M}] + \sum_{j=0}^{N_s-1}\sum_{k_1=0}^{q_1-1} s[j]d_1[\frac{q_1 q_2 n - q_2 k_2 - M(j+k_1)}{M}]\} \right\}^2 \end{aligned} \tag{5.35}$$

where $\mathbf{s} = [s[0]s[1]\ldots s[N_s-1]]'$ is the column vector of the coefficients of polynomial $S(z)$. The above equation can be converted to the following vectorized form:

$$\tilde{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} ||\mathbf{b} - \mathbf{A}\mathbf{s}||_2^2, \tag{5.36}$$

where we have $b(i+1) = x[i] - \sum_{k_2=0}^{L_{f_l}-1} f_l[k_2] a[\frac{q_1 i - k_2}{M}]$, $A(i+1, l+1) = \sum_{k_2=0}^{L_{f_l}-1} \sum_{k_1=0}^{q_1-1} f_l[k_2] d_1[\frac{q_1 q_2 i - q_2 k_2 - M(l+k_1)}{M}]\}$ $\forall$ $i \in \{0, 1, ..., N-1\}$ and $l \in \{0, 1, ..., N_s-1\}$. The above equation has the following closed-form solution:

$$\tilde{\mathbf{s}} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{b}. \tag{5.37}$$

Update of RFB filters using learned $S(z)$

Similar to the predict stage discussed earlier, we propose equation for the update of analysis lowpass filter using the update stage polynomial $S(z)$ for a rational filterbank in Theorem-4 below.

Theorem 6. *The analysis lowpass filter of a 2-channel rational filterbank can be updated using the update polynomial $S(z)$ used in the 'composite update branch' via the following relation:*

$$G_l^{new}(z) = G_l(z) + \sum_{r=0}^{q_2-1} \frac{1}{q_2} G_h(z^{\frac{q_1}{q_2}} W_{q_2}^{q_1 r}) U(z^{\frac{M}{q_2}} W_{q_2}^{Mr}), \tag{5.38}$$

where $U(z) = R_u(z)S(z)$.

Proof. Refer to Figure-5.12. Following the similar procedure as in Theorem 2, the update structure of Figure 5.12a can be equivalently converted to Figure 5.12c with the filter $\tilde{H}_u(z)$ given by:

$$\tilde{H}_u(z) = \frac{1}{q_2} \sum_{r=0}^{q_2-1} G_h(z^{\frac{q_1}{q_2}} W_{q_2}^{q_1 r}) U(z^{\frac{M}{q_2}} W_{q_2}^{Mr}). \tag{5.39}$$

Considering the structure in Figure 5.12c, signals $d_4[n]$, $a[n]$, and $a^{new}[n]$ can be written in Z-domain as:

$$D_4(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{\frac{q_1}{M}} W_M^{q_1 k}) \tilde{H}_u(z^{\frac{1}{M}} W_M^k), \tag{5.40}$$

$$A(Z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{\frac{q_1}{M}} W_M^{q_1 k}) G_l(z^{\frac{1}{M}} W_M^k), \tag{5.41}$$

$$\begin{aligned} A^{new}(z) &= D_4(z) + A(z) \\ &= \frac{1}{M} \sum_{k=0}^{M-1} X(z^{\frac{q_1}{M}} W_M^{q_1 k}) \big[\tilde{H}_u(z^{\frac{1}{M}} W_M^k) + G_l(z^{\frac{1}{M}} W_M^k)\big]. \end{aligned} \tag{5.42}$$

The above signal is equivalent to passing the signal $X(z)$ through an equivalent low pass filter $G_l^{new}(z)$ of the RFB (Figure 5.12d) and is given by:

$$\begin{aligned} G_l^{new}(z) =& G_l(z) + \tilde{H}_u(z) \\ =& G_l(z) + \frac{1}{q_2} \sum_{r=0}^{q_2-1} G_h(z^{\frac{q_1}{q_2}} W_{q_2}^{q_1 r}) U(z^{\frac{M}{q_2}} W_{q_2}^{Mr}) \end{aligned} \tag{5.43}$$

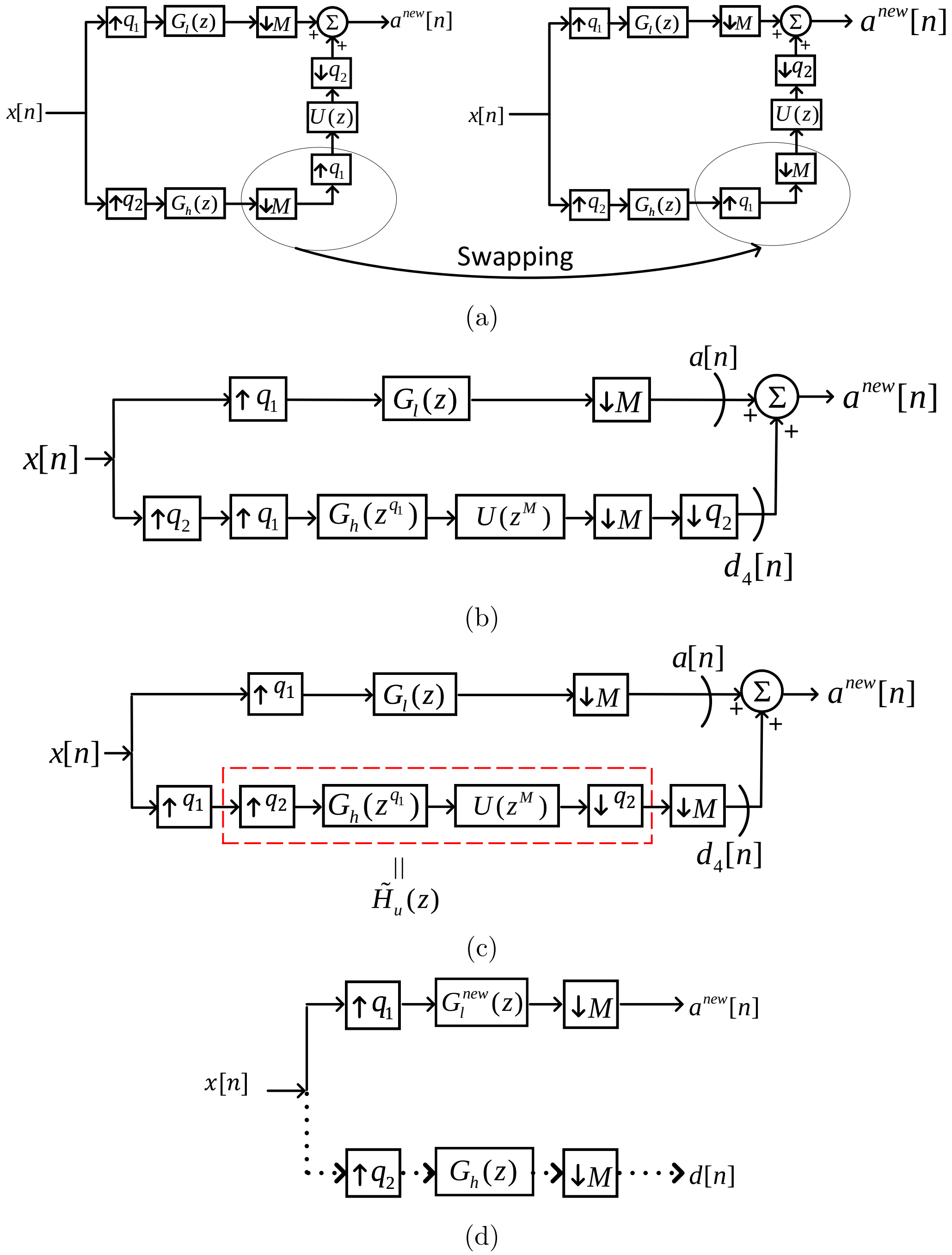
This proves the above theorem. □

Similar to the predict stage, the synthesis highpass filter $F_h^{new}(z)$ is learned using the updated analysis lowpass filter $G_l^{new}(z)$, polyphase matrices, and equations (2.6), (2.8), (2.9), (2.10)-(2.14).

Note that the resultant filters of the rational filterbank are learned from the given signal because these are updated based on the predict and update stage filters $T(z)$ and $S(z)$ learned from the signal using (5.14) and (5.35), respectively.

Table 5.1: Illustration of RWTL learned based on the theory developed in section 5.1.1

Sampling rate in two branches	$(\frac{2}{3}, \frac{1}{3})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{3}{4}, \frac{1}{4})$	$(\frac{2}{5}, \frac{3}{5})$
(q_1, q_2, M)	$(2,1,3)$	$(1,1,2)$	$(3,1,4)$	$(2,3,5)$
Lazy $G_l(z)$	$1+z^{-1}$	1	$1+z^{-1}+z^{-2}$	$1+z^{-3}$
Lazy $G_h(z)$	z^2	z	z^3	$z^2+z^4+z^6$
Lazy $F_l(z)$	$1+z$	1	$1+z+z^2$	$1+z^3$
Lazy $F_h(z)$	z^{-2}	z^{-1}	z^{-3}	$z^{-2}+z^{-4}+z^{-6}$
$R_p(z)$	1	1	1	$1+z^{-1}+z^{-2}$
$T(z)$	$z^2(t[0]z^{-1}+t[1])$	$z(t[0]z^{-1}+t[1])$	$z^3(t[0]z^{-1}+t[1])$	$z^6(t[0]z^{-3}+t[1])$
$R_u(z)$	$1+z^{-1}$	1	$1+z^{-1}+z^{-2}$	$1+z^{-1}$
$S(z)$	$s[0]+s[1]z^{-2}$	$s[0]+s[1]z^{-1}$	$s[0]+s[1]z^{-3}$	$s[0]+s[1]z^{-2}$



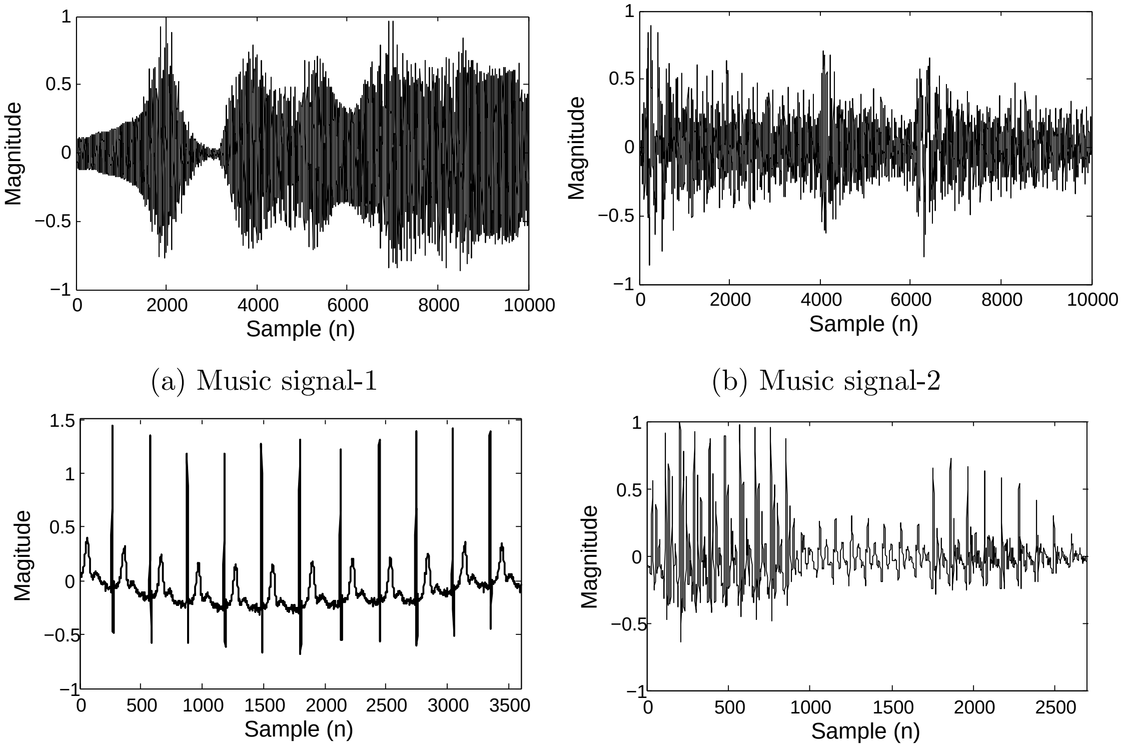
(a)

(b)

(c)

(d)

Figure 5.12: Illustration for the proof of Theorem-4. Structures in (a)-(e) are equivalent of each other



(a) Music signal-1

(b) Music signal-2

(c) ECG signal

(d) Speech signal

Figure 5.13: Signals used in the experiments

5.1.1.4 Design Examples

We present some examples of RWTL and the corresponding rational filterbank learned with the proposed method. Table-5.1 presents parameters used for learning RWTL with the following sampling rates in two branches: $\left(\frac{2}{3}, \frac{1}{3}\right)$, $\left(\frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{3}{4}, \frac{1}{4}\right)$, $\left(\frac{2}{5}, \frac{3}{5}\right)$. The first row in table provides values of q_1, q_2, and M. Second to fifth row presents *Lazy* wavelet of the corresponding rational filterbank structure from which the learning is initialized. Sixth and eighth row represents polynomial $R_p(z)$ and $R_u(z)$ respectively. Seventh and ninth row represents the structure of 2-tap predict and update polynomial $T(z)$ and $S(z)$ respectively.

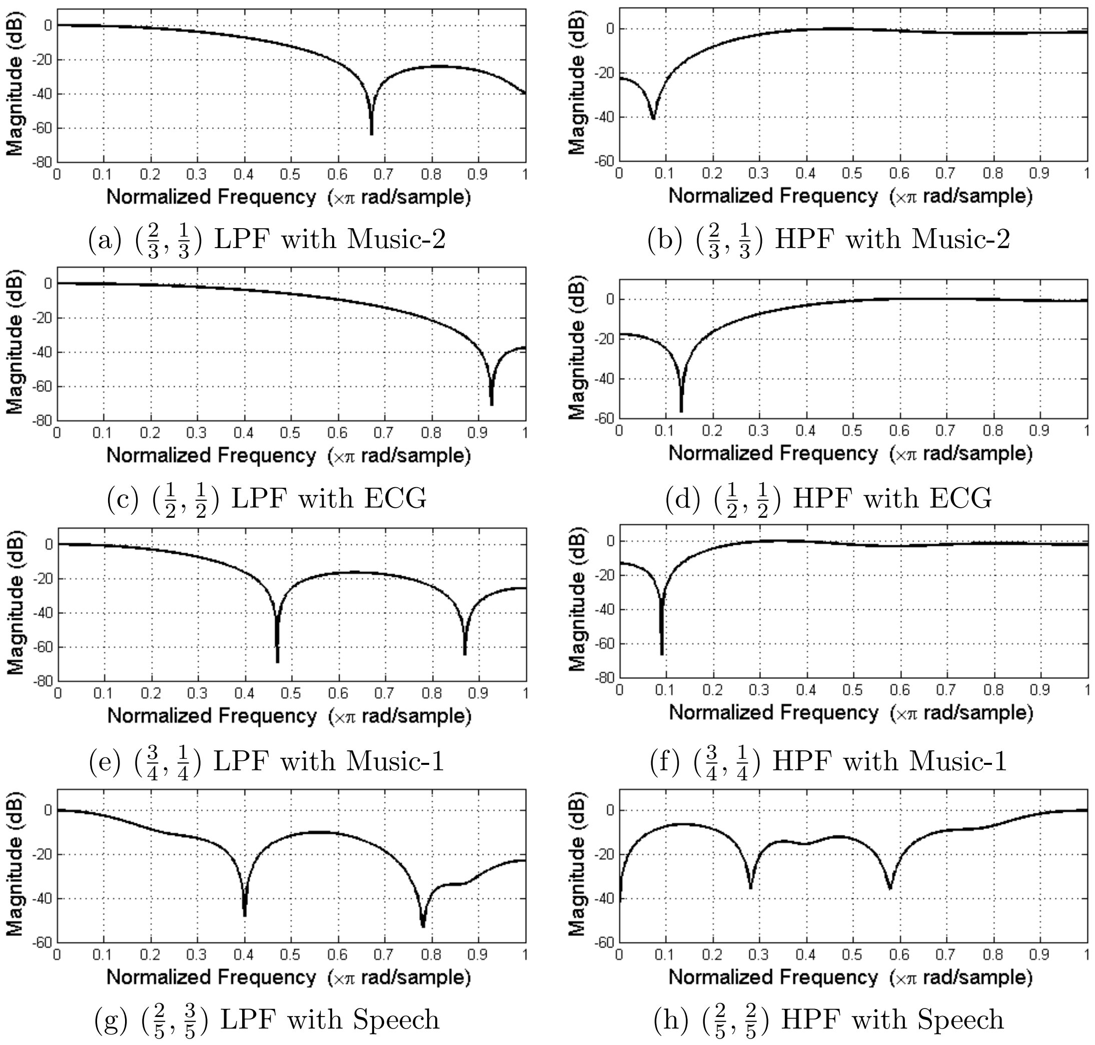
With these parameters and structure, we learn the RWTL matched to given signals of interest. We consider four signals of different types: 1) ECG signal, 2) speech signal, and 3) two music signals (named as music-1 and music-2). These signals are shown in Figure-5.13. We learn RWTL with sampling rates $\left(\frac{2}{3}, \frac{1}{3}\right)$,

Table 5.2: Coefficients of predict polynomial, update polynomial, and synthesis filters of RWTL learned with different sampling rates. f_s is the sampling frequency of the signal in kHz and N is the number of samples of the signal used in experiments.

Sampling rate	Signal	Predict/Update polynomial
in two branches	(f_s, N)	and Filter coefficients
$\left(\frac{2}{3}, \frac{1}{3}\right)$	Music-2 (11.025, 10000)	$t[n]$ =[0.4777 0.5101] $s[n]$ =[0.1751 0.1893] $f_l[n]$ =[0.1707 0.3347 0.3347 0.1599] $f_h[n]$ =[-0.0748 -0.1466 -0.1466 0.6867 -0.1586 -0.1586 -0.0758]
$\left(\frac{1}{2}, \frac{1}{2}\right)$	ECG (0.36,3600)	$t[n]$ =[0.5119 0.5143] $s[n]$ =[0.2818 0.2834] $f_l[n]$ =[0.2538 0.4935 0.2526] $f_h[n]$ =[-0.1449 -0.2818 0.7100 -0.2834 -0.1451]
$\left(\frac{3}{4}, \frac{1}{4}\right)$	Music-1 (11.025, 10000)	$t[n]$ =[0.6100 0.6109] $s[n]$ =[0.1496 0.1499] $f_l[n]$ =[0.1447 0.2369 0.2369 0.2369 0.1445] $f_h[n]$ =[-0.0755 -0.1236 -0.1236 -0.1236 0.6751 -0.1239 -0.1239 -0.1239 -0.0756]
$\left(\frac{2}{5}, \frac{3}{5}\right)$	Speech (11.025,2700)	$t[n]$ =[0.7048 0.3336] $s[n]$ =[0.1430 0.4474] $f_l[n]$ =[0.0652 0.0652 0 0.0652 0.1955 0.1378 0.1378 0.1955 0.1378] $f_h[n]$ =[-0.0137 -0.0137 0 -0.0137 0 -0.0427 -0.0715 -0.0409 -0.0427 -0.0409 -0.0288 0.1959 -0.1280 0.2572 -0.1280 0.1959 0 0 -0.0902]

$\left(\frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{3}{4}, \frac{1}{4}\right)$, $\left(\frac{2}{5}, \frac{3}{5}\right)$ in the two branches matched to music-1, music-2, ECG, and speech signals, respectively. Since we learn RWTL in the lifting framework that always satisfies PR condition, our learned wavelet system achieves PR with NMSE (normalized mean square error) of the order of 10^{-19}. Table-5.2 presents coefficients of predict polynomial, update polynomial, and synthesis filters learned with the proposed method. Figure-5.14 shows the frequency response of filters associated with the learned RWTL.

Although we presented RWTL method from original signals above, the method can be extended to learn RWT in inverse problems and from a class of signals similar to what we have proposed in section 3.1.2 and 3.1.3, respectively, for dyadic wavelet transform.



(a) $(\frac{2}{3}, \frac{1}{3})$ LPF with Music-2

(b) $(\frac{2}{3}, \frac{1}{3})$ HPF with Music-2

(c) $(\frac{1}{2}, \frac{1}{2})$ LPF with ECG

(d) $(\frac{1}{2}, \frac{1}{2})$ HPF with ECG

(e) $(\frac{3}{4}, \frac{1}{4})$ LPF with Music-1

(f) $(\frac{3}{4}, \frac{1}{4})$ HPF with Music-1

(g) $(\frac{2}{5}, \frac{3}{5})$ LPF with Speech

(h) $(\frac{2}{5}, \frac{2}{5})$ HPF with Speech

Figure 5.14: Frequency response of synthesis filters presented in Table-5.2.

5.1.2 Application

Application of the proposed method of rational wavelet transform is presented in this section.

5.1.2.1 Application in Compressed Sensing based reconstruction

In this section, we explore the performance of the learned RWTL in the applications of Compressed Sensing (CS)-based reconstruction of 1-D signals.

As discussed before, CS problem aims to recover a full signal from a small number of its linear measurement [87, 46], $\mathbf{y}$ given below:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{5.44}$$

where $\mathbf{x}$ is the original signal of size $N \times 1$, compressively measured as $\mathbf{y}$ of size $M \times 1$ by the measurement matrix $\mathbf{\Phi}$ of size $M \times N$.

Full signal $\mathbf{x}$ is reconstructed from compressive measurements by solving the following optimization problem:

$$\tilde{\mathbf{r}} = \min_{\mathbf{r}} \; ||\mathbf{r}||_1 \quad \text{subject to } \mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{r}, \tag{5.45}$$

where $\mathbf{x} = \mathbf{\Psi}\mathbf{r}$. As wavelets are extensively used as the sparsifying transforms [46], we consider $\mathbf{\Psi}$ as the wavelet transform above and $\mathbf{r}$ is wavelet transform coefficients of $\mathbf{x}$. The above problem is known as basis pursuit (BP) [95] and we used SPGL1 solver to solve the above problem. Full signal is reconstructed as $\tilde{\mathbf{x}} = \mathbf{\Psi}\tilde{\mathbf{r}}$.

Table-5.3 presents CS-based reconstruction performance of RWTL with different sampling rates on the four signals shown in Figure-5.13. Reconstruction performance is measured via PSNR (peak signal to noise ratio) presented in 2.5. Measurement matrix $\mathbf{\Phi}$ is Gaussian and sampling ratio (SR) is varied from 10% to 90% with a difference of 10%, where sampling ratio was defined in 2.5. Three-level wavelet transform decomposition has been used for all the experiments. Results are averaged over 50 independent trials.

RWTL with the following sampling rate are considered: $\left(\frac{2}{3}, \frac{1}{3}\right)$, $\left(\frac{1}{3}, \frac{2}{3}\right)$ and $\left(\frac{3}{4}, \frac{1}{4}\right)$ and are represented as RWTL$\left(\frac{2}{3}, \frac{1}{3}\right)$, RWTL$\left(\frac{1}{3}, \frac{2}{3}\right)$ and RWTL$\left(\frac{3}{4}, \frac{1}{4}\right)$, respectively. The original signal $\mathbf{x}$ is not available in the compressed sensing application, while the proposed method requires the signal for learning matched rational wavelet. Thus, we propose to sample one-third of the data fully (at 100% sampling ratio) to learn signal-matched RWT. Next, we apply the learned RWT for the reconstruction of the rest of the data sampled at lower compressive sensing ratio. One may also use an approach, similar to the one presented in section 3.1.2 for RWTL in inverse problems.

Table 5.3: CS based signal reconstruction performance of RWTL on different signals. Rational wavelet is learned from one-third of the data samples. Results are averaged over 50 independent trails.

Signal	Sampling ratio (in %)	90	80	70	60	50	40	30
Music-1	db2	28.3	23.9	21.0	18.5	16.6	15.1	13.8
	db4	31.2	26.2	22.5	19.8	17.6	15.8	14.2
	Bi 5/3	28.7	23.6	20.1	17.4	15.5	14.0	12.9
	Bi 9/7	33.2	28.0	24.1	20.8	18.2	16.1	14.4
	S$\left(\frac{2}{3},1\right)$	29.6	24.4	20.6	17.6	15.4	13.8	12.5
	S$\left(\frac{1}{3},1\right)$	24.8	20.8	18.2	16.3	14.8	13.7	12.8
	S$\left(\frac{3}{4},1\right)$	29.2	24.0	20.6	17.9	15.8	14.2	12.9
	RWTL$\left(\frac{2}{3},\frac{1}{3}\right)$	**33.3**	**28.7**	**25.4**	**22.6**	**19.9**	**17.7**	**15.6**
	RWTL$\left(\frac{1}{3},\frac{2}{3}\right)$	27.6	23.6	20.8	18.6	16.7	15.0	13.6
	RWTL$\left(\frac{3}{4},\frac{1}{4}\right)$	32.7	27.8	24.3	21.3	18.9	16.8	15.1
Music-2	db2	29.7	26.1	23.9	22.0	20.4	18.9	17.5
	db4	30.0	26.4	24.1	22.3	20.7	19.1	17.6
	Bi 5/3	29.4	25.9	23.5	21.6	19.9	18.4	16.9
	Bi 9/7	30.3	26.8	24.5	22.6	21.0	19.4	17.8
	S$\left(\frac{2}{3},1\right)$	29.4	26.0	23.7	21.8	20.2	18.6	17.1
	S$\left(\frac{1}{3},1\right)$	29.2	25.7	23.4	21.6	20.0	18.5	17.2
	S$\left(\frac{3}{4},1\right)$	29.4	25.9	23.5	21.6	19.9	18.2	16.8
	RWTL$\left(\frac{2}{3},\frac{1}{3}\right)$	**30.6**	**27.1**	**24.8**	23.1	21.5	20.0	18.5
	RWTL$\left(\frac{1}{3},\frac{2}{3}\right)$	30.2	26.9	**24.8**	**23.3**	**22.0**	**20.8**	**19.6**
	RWTL$\left(\frac{3}{4},\frac{1}{4}\right)$	30.1	26.5	24.1	22.2	20.5	18.9	17.5
ECG	db2	49.7	45.9	42.9	39.5	34.6	27.0	21.2
	db4	50.1	46.4	43.6	40.6	36.1	27.4	21.1
	Bi 5/3	49.8	45.9	42.8	39.4	34.1	25.9	20.7
	Bi 9/7	**50.9**	**47.1**	**44.2**	**41.2**	37.3	29.4	21.8
	S$\left(\frac{2}{3},1\right)$	46.8	42.9	39.6	35.0	28.2	23.3	20.4
	S$\left(\frac{1}{3},1\right)$	49.0	45.6	43.3	41.1	**38.6**	35.3	30.7

	S$\left(\frac{3}{4},1\right)$	47.5	42.1	36.2	29.5	24.4	21.5	19.6
	RWTL$\left(\frac{2}{3},\frac{1}{3}\right)$	50.1	45.9	42.6	38.6	33.1	26.7	22.9
	RWTL$\left(\frac{1}{3},\frac{2}{3}\right)$	47.9	44.5	42.4	40.5	38.3	**35.6**	**32.1**
	RWTL$\left(\frac{3}{4},\frac{1}{4}\right)$	48.3	43.1	37.2	30.1	25.5	22.7	20.7
Speech	db2	41.7	36.2	31.9	28.5	25.4	22.4	20.1
	db4	44.7	39.2	34.3	30.1	26.4	23.2	20.6
	Bi 5/3	42.2	36.6	32.1	28.1	24.6	21.5	19.2
	Bi 9/7	**46.4**	**41.0**	**36.4**	32.1	28.0	24.0	21.2
	S$\left(\frac{2}{3},1\right)$	43.4	37.6	32.5	28.4	25.1	22.1	19.9
	S$\left(\frac{1}{3},1\right)$	38.3	33.5	30.2	27.4	25.2	22.8	20.6
	S$\left(\frac{3}{4},1\right)$	43.6	38.0	32.8	28.4	25.2	22.4	20.2
	RWTL$\left(\frac{2}{3},\frac{1}{3}\right)$	43.6	39.1	35.7	**32.4**	**29.1**	**25.7**	**22.9**
	RWTL$\left(\frac{1}{3},\frac{2}{3}\right)$	39	35.1	32.2	29.5	27.2	25	22.7
	RWTL$\left(\frac{3}{4},\frac{1}{4}\right)$	42.2	37.1	32.8	29.1	25.9	23.3	20.9

The reconstruction performance is compared with standard orthogonal Daubechies wavelets db2 and db4, and standard bi-orthogonal wavelets, bior5/3 and bior9/7 (labeled as Bi 5/3 and Bi 9/7, respectively). The performance is also compared with overcomplete rational wavelets designed in [79]. For fair comparison, we consider the same sampling rate in the low frequency branch for these overcomplete rational wavelets as used in proposed RWTL in Table-III, i.e., we use overcomplete rational wavelets [79] with sampling rates: $\left(\frac{2}{3},1\right)$, $\left(\frac{1}{3},1\right)$ and $\left(\frac{3}{4},1\right)$, represented as S$\left(\frac{2}{3},1\right)$, S$\left(\frac{1}{3},1\right)$ and S$\left(\frac{3}{4},1\right)$.

From Table 5.3, it is observed that Bi 9/7 performs best among the existing wavelets used. Also, the overcomplete rational wavelet $\left(\frac{1}{3},1\right)$ performs better than the existing wavelets on ECG signal at all sampling ratios less than 60%, while these perform comparable or inferior in performance to the existing wavelets on music and speech signals.

On the other hand, the proposed RWTL perform better, with an improvement of upto 1.8 dB of PSNR, in comparison to existing wavelets for music signals. Particularly, RWTL with $\left(\frac{2}{3},\frac{1}{3}\right)$ performs better on music-1 signal for all the sampling ratios. RWTL with $\left(\frac{2}{3},\frac{1}{3}\right)$ performs better on music-2 signal from 90% to

70% sampling ratio beyond which $\left(\frac{1}{3}, \frac{2}{3}\right)$ performs better. On ECG signal, existing wavelet Bi 9/7 performs better than all other wavelets at higher sampling ratios from 90% to 60%. At 50% sampling ratio, overcomplete rational wavelet $\left(\frac{1}{3}, 1\right)$ outperforms all existing and proposed rational wavelets. Below 50% sampling ratio, RWTL with $\left(\frac{1}{3}, \frac{2}{3}\right)$ performs better than all existing as well as overcomplete rational wavelets with an improvement of upto 10 dB than existing wavelets and upto 1.4 dB than overcomplete rational wavelets. Similarly, in case of speech signal, Bi 9/7 performs better than all other wavelets from sampling ratio 90% to 70%. Below 70% sampling ratio, RWTL with $\left(\frac{2}{3}, \frac{1}{3}\right)$ outperforms all existing as well as overcomplete rational with an improvement of upto 1.7 dB.

Further, this is to note that at higher sampling ratios, PSNR of the reconstructed ECG and speech signals is high at around 40 dB and 30 dB, respectively, with different wavelets. Hence, the reconstructed signal appears almost similar to the original signal. The quality of the reconstructed signal deteriorates with decreasing sampling ratios, where the proposed RWTL performs best with as much as 10 dB improvement. Overall, the performance of rational matched wavelets is superior at lower sampling ratios in compressive sensing application. In this work, we have not explored the problem of choosing the optimal sampling rate of learned wavelet for a particular signal or in a particular application. This remains an open problem and can be explored in the future.

5.2 RWTL for images

In this section, we extend the concept of RWTL proposed in section5.1.1 from 1-D signal to 2-D images. Specifically, we present method for separable RWTL and apply the learned separable RWT as the sparsifying transform in CS based reconstruction of natural images.

5.2.1 Proposed method of RWTL for images

We learn separable RWT for images using the method similar to the one proposed in section 4.1.1. First, we convert the given image to two 1-D signals using the serpentine scanning discussed in section 4.1.1.1. We obtain one column-wise scanned and another row-wise scanned signals. We learn RWT corresponding to these column-wise and row-wise scanned signals using the method proposed in the previous section. Thus, we obtain separable RWT for the given image which can be used as the sparsifying transform in the applications.

5.2.2 Applications

5.2.2.1 CS based Image Reconstruction using RWTL via Non-Separable Regularization

In this section, we learn RWT in the application of CS based reconstruction of the images. The method proposed in the previous subsection for RWTL for images can not be applied in CS application directly as that method requires the full image, which is not available in CS applications. Hence, we use a method similar to the one proposed in section 4.1.2 for learning separable RWT in inverse problems. This is described in the rest part of this section. Please note that, here we learn separable RWT with downsampling factor of $\left(\frac{3}{2}, \frac{3}{1}\right)$ in two branches. The similar method can be applied for learning rational wavelet transform with other downsampling factor as well with the help of the theory presented in section 5.1.1.

Similar to the method proposed in section 4.1.2, the proposed method is divided into three stages. In the first stage, we obtain the coarse estimate of the image. We use this coarse estimate of the image to learn separable RWT in the second stage. The learned separable RWT is used as the sparsifying transform in the third stage for the reconstruction of full image. All these three stages are described below. Also, we use recently proposed non-separable non-convex regularization (discussed in section 2.3) for CS based reconstruction in stage-1 and stage-3.

Stage-1: Coarse Estimation of Image

In this stage, we obtain a coarse estimate of the image from compressive measurements $\mathbf{y}$ using one of the existing wavelets (we chose 9/7 biorthogonal wavelet) in equation (2.28).

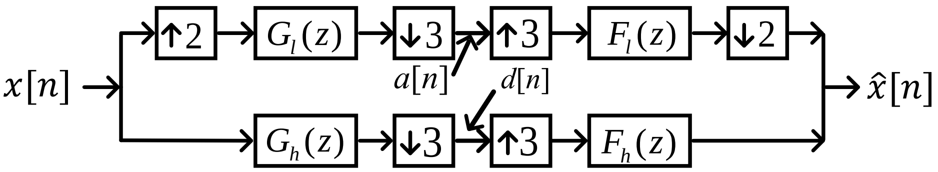


Figure 5.15: Rational wavelet with downsampling factor of $\left(\frac{3}{2}, \frac{3}{1}\right)$ in two branches

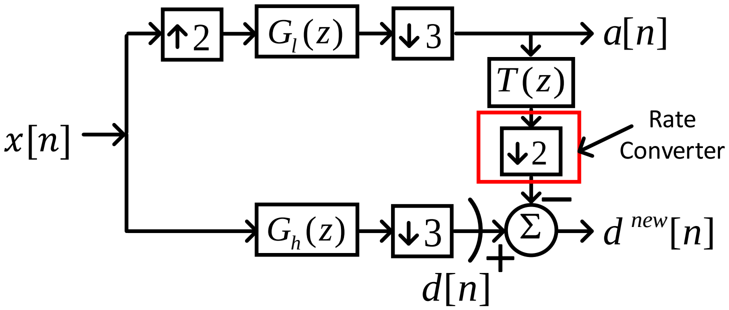


Figure 5.16: Predict stage for learning rational wavelet with downsampling factor of $\left(\frac{3}{2}, \frac{3}{1}\right)$ in two branches

Stage 2: Learning of separable RWT

Next, we learn the separable RWT for row and column spaces of a given image separately. The method is identical for row and column space and hence, we present the formulation for only one of the spaces (here column space is used) for RWT with downsampling factor of $\left(\frac{3}{2}, \frac{3}{1}\right)$ in two branches (see Figure 5.15).

Rational Lazy wavelet system For learning rational wavelet transform with downsampling factor of $\left(\frac{3}{2}, \frac{3}{1}\right)$ in two branches, we start with rational Lazy wavelet with $G_l(z) = 1 + z^{-1}$, $G_h(z) = z^2$, $F_l(z) = 1 + z$, and $F_h(z) = z^2$, respectively, in Figure 5.15. The output of Lazy wavelet provides two sets of disjoint samples, where $\frac{2}{3}$ of the samples constitute $a[n]$ and $\frac{1}{3}$ samples constitute $d[n]$.

Learning predict polynomial We feed the column space of the coarse estimated image $\tilde{\mathbf{X}}$ as input to the rational Lazy wavelet in Figure 5.16 and obtain $a[n]$ and $d[n]$ as output. Signal $a[n]$ is passed through the predict polynomial $T(z) = t_1 z + t_2 z^2$ followed by downsampler to obtain the predicted signal.

This predicted signal is subtracted from $d[n]$ to obtain prediction error $d^{new}[n]$ as below:

$$d^{new}[n] = d[n] - a[2n] * t[2n], \tag{5.46}$$

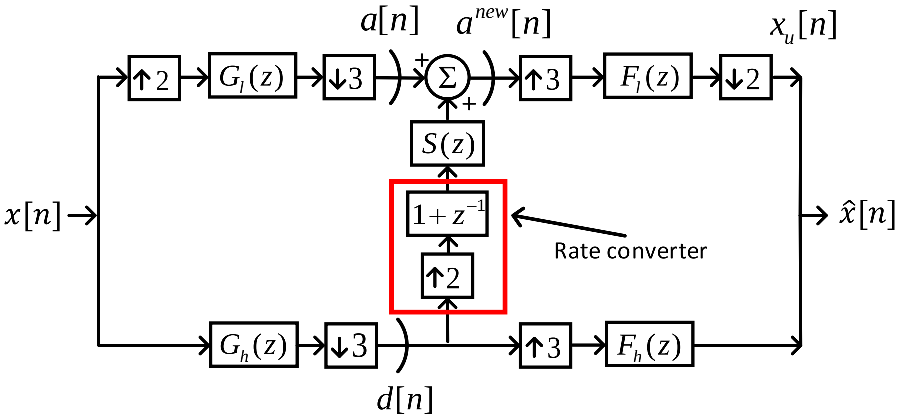


Figure 5.17: System Diagram for Update stage

where $t[n]$ denotes predict polynomial $T(z)$ in the time domain. We consider this prediction error/new detail coefficients as the noisy version of the actual detail coefficients that could be obtained if one has access to the original signal because we obtained these from the coarse estimate of the input image. Mathematically, it can be written as:

$$\begin{aligned}\hat{d}[n] =& d^{new}[n] + \eta[n], \\ =& d[n] - a[2n] * t[2n] + \eta[n], \end{aligned} \tag{5.47}$$

where η is assumed to follow Gaussian distribution. Writing the above relation in matrix form, we obtain:

$$\hat{\mathbf{d}} - \mathbf{d} = \mathbf{At} + \boldsymbol{\eta}, \tag{5.48}$$

The above equation has closed form solution and is solved using least squares to learn $T(z)$. The analysis highpass filter is updated using $T(z)$ (using (5.19)):

$$G_h^{new}(z) = G_h(z) - \sum_{k=0}^{1} G_l(z^{\frac{1}{2}} W_2^{2k}) T(z^{\frac{3}{2} W_2^{3k}}), \tag{5.49}$$

where $W_r = e^{-j\frac{2\pi}{r}}$. Synthesis lowpass filter is updated using polyphase decomposition matrices with the condition of perfect reconstruction filterbank as discussed in section 5.1.1.2.

Learning update polynomial The updated detail coefficients $d^{new}[n]$ are passed through the rate converter, which is a combination of upsampler and polynomial $1 + z^{-1}$ as shown in Figure 5.17. The resultant signal is passed through $S(z) = s_1 + s_2 z^{-2}$ and added to $a[n]$ in the upper branch to obtain updated approximate coefficients, $a^{new}[n]$. The updated approximate coefficients are passed through synthesis lowpass branch to obtain the signal $x_u[n]$ (Figure 5.17). As highpass (lower) branch captures small energy of signal (only prediction error), most of the energy of the input signal lies in the lowpass branch. Hence, signal $x_u[n]$ should be in close approximation to the input signal $x[n]$. Thus, we minimize energy of the difference of these two signals to learn $S(z)$ as:

$$\tilde{\mathbf{s}} = \min_{\mathbf{s}} \sum_{n=0}^{N-1} (x[n] - x_u[n])^2, \tag{5.50}$$

The learned $S(z)$ is used to update analysis lowpass filter as (using (5.38)):

$$G_l^{new}(z) = G_l(z) + G_h(z^2)S(z^3). \tag{5.51}$$

Synthesis highpass filter is updated using the perfect reconstruction condition as discussed in section 5.1.1.3. This completes all steps of learning RWT for the column space of a given image. Likewise, this step is repeated for the row space and hence, RWT is learned for a given image that can be implemented as a separable transform.

Stage 3: Image Reconstruction using the learned RWT

The learned RWT obtained in the previous stage is used as the sparsifying transform $\mathbf{\Psi}$ in non-separable non-convex optimization of (2.28) to reconstruct full image.

Experiments and Results

We applied the proposed methodology of CS based image reconstruction using RWTL and non-separable non-convex regularization on 36 natural images of dataset [126]. All these images were of equal size 321×481. We used the partial canonical identity matrix (PCI), discussed in section 4.2.1.1, as the sensing matrix $\mathbf{\Phi}$.

Table 5.4: Mean PSNR (in dB) performance on a dataset of 36 natural images of CS based reconstruction

Transform used		Sampling ratio (in %)								
		90	80	70	60	50	40	30	20	10
db4	W	32.6	30.6	29.0	27.5	26.1	24.6	23.0	20.8	11.5
	W+D	32.1	30.1	28.3	26.8	25.4	24.0	22.7	21.3	19.8
db10	W	32.7	30.8	29.2	27.8	26.4	24.9	23.4	21.3	12.0
	W+D	32.3	30.3	28.7	27.2	25.8	24.4	23.0	21.5	19.8
sym10	W	32.8	30.9	29.3	27.8	26.4	24.9	23.3	21.1	11.5
	W+D	32.3	30.3	28.6	27.0	25.6	24.2	22.8	21.4	19.8
Bi 9/7	W	33.0	31.1	29.5	28.1	26.7	25.2	23.6	21.3	10.6
	W+D	32.5	30.5	28.8	27.3	26.0	24.6	23.3	21.8	20.0
DCT		31.8	30.4	29.0	27.8	26.6	25.4	24.1	22.8	21.1
MB [125]		35.1	33.2	31.5	30.0	28.5	27.0	25.3	23.2	14.6
RWTL+D proposed		33.0	31.5	30.2	29.0	27.8	26.6	25.5	**24.2**	**22.2**
MB with RWTL+D proposed		**35.7**	**33.8**	**32.2**	**30.6**	**29.1**	**27.6**	**25.9**	23.9	15.9

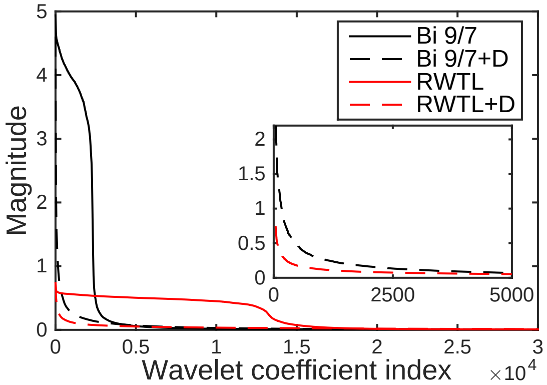


Figure 5.18: Plot for sorted magnitude of wavelet coefficients

We used $\gamma = 0.8$, as suggested by authors in [101], and $\lambda = 0.05$ in all of our experiments.

We compared the reconstruction performance of RWTL with those of orthogonal Daubechies db4, db10, bi-orthogonal Bi 9/7, Sym10 wavelets, and discrete cosine transform (DCT). We considered 3-level L-Pyramid wavelet transform decomposition, discussed in section 4.1.3, with all the wavelets. We plot the sorted magnitude of coefficients with dyadic Bi 9/7 wavelet and learned RWT in Figure

5.18. We observe that the decay of wavelet coefficients alone is not that steep. Hence, we apply DCT on the wavelet approximate coefficients, shown as Bi9/7+D and RWTL+D in Figure 5.18. This step enhances the sparsity of the coefficients and RWTL+D coefficients are observed to decay much faster than those of Bi 9/7+D.

Table-5.4 shows the CS based reconstruction performance of different transforms considered in terms of peak signal to noise ratio (PSNR, in dB) at sampling ratios from 90% to 10%. Because of the enhanced sparsity of DCT applied to approximate wavelet coefficients, we consider existing wavelet alone (W) as well as DCT applied to wavelet approximate coefficients (W+D) for image reconstruction. We also compare the performance with the recently proposed method of multi-basis [125], which uses multiple sparsifying basis for image reconstruction. We consider all the existing transform basis used in this paper for multi-basis method, shown as MB [125] in Table. We also use RWTL+D along with other transforms in multi-basis method, shown as MB with RWTL+D in Table.

It is observed from Table-1 that: 1) W+D outperforms W at lower sampling ratios, 2) RWTL+D outperforms all existing transforms and the MB method at lower sampling ratios, 3) MB with RWTL+D outperforms other existing transforms and MB method [125] at higher sampling ratios. Figure 5.19 shows the visual quality of one of the reconstructed images from the dataset, compressively sensed at 10% sampling ratio. The figure indicates superior performance of RWTL+D in preserving sharp details of the image compared to other transforms.

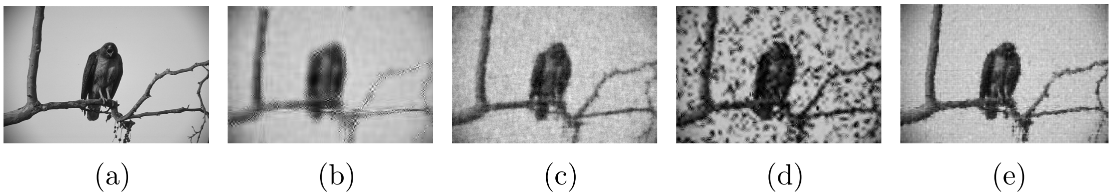

(a) (b) (c) (d) (e)

Figure 5.19: Qualitative results on one image; a) Original image; Image reconstructed from 10% compressive measurements using b) Bi9/7+DCT c) DCT d) Multibases e) RWTL+DCT

5.3 Summary

Methods for rational wavelet transform learning for 1-D and 2-D signals (images) are presented using the lifting framework in this chapter. Lifting theory is extended from dyadic to rational wavelet and the extended lifting theory is used to learn RWT from a given signal, where rational wavelets with any decimation ratio can be learned. Output sample rates in the two branches of analysis filterbank of the rational wavelet transform is different that makes the lifting methodology difficult to be used directly for rational wavelets. Hence, we introduced the concept of rate converter, which equals the rate in the two branches. Learned rational filters are FIR that can be easily implemented in hardware. The proposed method of RWTL has closed-form solution and the transform can be learned from a short duration of signal. The learned RWT for 1-D signal has been applied in the application of CS based reconstruction and is observed to perform better than the existing dyadic as well as rational wavelet transforms. Further, the concept of RWTL is extended from 1-D signal to images to learn separable RWT for the images. The learned separable rational wavelet transform is used in the application of CS based reconstruction of natural images and is observed to perform better than the existing dyadic wavelet transform.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Methods to learn signal-matched dyadic and rational wavelet transforms from original as well as degraded signals are presented in this work. Lifting framework is used in the proposed methods, where both predict and update stage filters are learned from the given signal itself. The learned signal-matched wavelet transform inherits all the advantages of lifting framework, i.e., the learned wavelet transform is always invertible, the method is modular, and the corresponding wavelet system can also incorporate nonlinear filters, if required. The proposed methods can be used to learn wavelet transform from a small duration of the signal. Closed form solution is derived for learning the wavelet transform with the proposed methods and hence, no greedy solution is required. Thus the proposed methods are computationally efficient.

Three Methods are presented to learn DWT. The first method requires full original signal to design/learn wavelet. Since this method cannot be applied in inverse problems, where only the degraded signal is available instead of the original signal, we propose the second method for learning DWT in inverse problems. In this method, first, we obtain a coarse estimate of the signal using an existing wavelet transform and then use it to learn DWT matched to the signal. The learned DWT can be used as the sparsifying transform for sparse signal recovery of the signals. The proposed method is applied in CS based reconstruction of music, speech, and ECG signals, and is observed to outperform the existing wavelets.

Also, it is observed that the reconstruction performance of the wavelet transform learned from the degraded signals is only slightly inferior to the performance of the wavelet transform learned from the original signals. This proves that the proposed method of learning DWT from degraded signal is as good as the learning the same from the original signal.

We also present methods to learn wavelet transform for a class of signals where signals of the class show similarity or low variability from one to another signal. ECG and brain MRI are two examples of such classes of signals. The method is proposed to learn a class-specific matched wavelet off-line from an ensemble of signals and can be used for any signal of the class. Thus, the learned wavelet is not required to be learned again for any single signal or from degraded signals in inverse problems, reducing the computational needs of learning the transform. The method is explored for the class of ECG signals in CS-based reconstruction. Detailed experiments are provided using the learned wavelet transform and the proposed weighted non-convex minimization for CS-based ECG signal recovery with various sensing matrices. The learned wavelet transform along with the proposed weighted non-convex minimization method is observed to provide better ECG signal reconstruction as compared to the existing wavelet transforms as well as the existing methods.

The proposed method of DWTL is extended to images to learn separable DWT from 1) original image and 2) degraded images in inverse problems. The proposed method is applied in CS based recovery of natural images. In CS application, partial canonical identity matrix is used as the sensing matrix for images that is observed to perform much faster compared to the existing sensing matrices and hence, is suited for time-bound real-time reconstruction based applications. Although there is a slight degradation in performance with the proposed sensing matrix but that is easily covered up by the learned separable DWT. A new multi-level L-Pyramid wavelet decomposition strategy is proposed that is observed to outperform the existing multi-level wavelet decomposition strategy. Overall, the proposed work with different sensing matrices, new wavelet decomposition strategy, and the learned DWT provides much better reconstruction results with ease of hardware implementation in CS application of images compared to the existing methodology.

The proposed method of separable wavelet transform learning is also applied in impulse denoising of images. First, corrupted pixels are identified from images using impulse noise detection algorithm. Second, uncorrupted pixels are utilized to learn wavelet transform matched to the given image. The learned transform is used as the sparsifying transform for the sparse recovery based impulse denoising of images. The learned transform is observed to perform better than the existing wavelet transforms and as well as other existing methods, both quantitatively and qualitatively.

The method of learning rational wavelet transform using the lifting framework has also been presented. The existing theory of lifting framework is extended from dyadic to rational wavelets. The extended lifting theory is used to learn critically sampled signal-matched rational wavelet transform for any general decimation ratio. The concept of rate converters is introduced to handle variable sample rates of subbands. Similar to the dyadic wavelet transform, the learned signal-matched rational wavelet inherits all the advantages of lifting framework. The learned analysis and synthesis filters are FIR and are easily implementable on hardware, thus making rational wavelet transform easily usable in applications. As a proof of concept, the learned rational wavelet transform is applied in the CS-based reconstruction of 1-D signals and is observed to perform better compared to the existing dyadic as well as rational wavelets.

The method of rational wavelet transform learning has been extended from 1-D signals to images to learn separable rational wavelet transform. The learned rational wavelet transform is used as the sparsifying transform for CS based reconstruction of images. Recently, non-separable non-convex regularization is proposed in the literature, which has been observed to yield better performance than convex l_1 and other non-convex regularized optimization problems in the case of signal denoising. We used this regularization for CS based reconstruction of images. The proposed methodology is applied on natural images and is shown to perform better than existing transforms and methods.

6.2 Future work

Some of the future directions of the work are as follows.

- Optimal sampling ratio in two branches of rational wavelet transform: Although the learned rational wavelet transform is observed to perform better in the CS based reconstruction, it is not known apriori as to what sampling ratios in the two branches of the rational wavelet transform are optimal for a given signal in a particular application. This can be explored in future.

- Non-separable wavelet transform learning: Discrete wavelet transform is largely applied on images as separable wavelet transform. In that case, the wavelet transform is able to provide efficient representation for the structure present in the horizontal and vertical directions. However, separable wavelet transform are not capable to efficiently represent the structures present in any random directions, which can be compactly represented by non-separable wavelet transform. Hence, non-separable dyadic as well as rational wavelet transform learning can be explored using the lifting framework.

- Extension to higher dimensional signals: Wavelet transform is widely used as the sparsifying transform in three dimensional signals like video, MRI or hyper-spectral images. The proposed method of wavelet transform learning can be extended to three dimensional signals as 3-D wavelet transform learning and can be used in various applications. Also, the proposed L-Pyramid wavelet decomposition on images can also be extended from 2-D to 3-D L-Pyramid wavelet decomposition and can be explored in different applications.

- The optimal length of the predict and update stage filters for dyadic and rational wavelet transforms remains an open problem. We would like to address this problem in future. We would also like to explore statistical-based approaches to find the optimal value of p for CS based ECG signal recovery.

Publications

Journal Articles

- **N. Ansari** and A. Gupta, "Image Reconstruction Using Matched Wavelet Estimated From Data Sensed Compressively Using Partial Canonical Identity Matrix," in *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3680-3695, Aug. 2017.

- **N. Ansari** and A. Gupta, "M-RWTL: Learning Signal-Matched Rational Wavelet Transform in Lifting Framework,", *IEEE Access*, December 2017.

- **N. Ansari** and A. Gupta, "WNC-ECGlet: Weighted Non-Convex Minimization based Reconstruction of Compressively Transmitted ECG using ECGlet," *Submitted*, *Biomedical Signal Processing and Control*, 2017.

Conference Articles

- **N. Ansari** and A. Gupta, "Joint framework for signal reconstruction using matched wavelet estimated from compressively sensed data," *Data Compression Conference (DCC)*, Snowbird, UT, pp. 580-580, 2016. (Core A* conference)

- **N. Ansari** and A. Gupta, "Signal-matched wavelet design via lifting using optimization techniques," *IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, pp. 863-867, 2015.

- **N. Ansari** and A. Gupta, "Lifting-based rational wavelet design from a given signal," *IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, pp. 853-857, 2015.

- **N. Ansari** and A. Gupta, "WTL-I: Wavelet transform learning based impulse denoising," *Submitted*, *IEEE Statistical Signal Processing Workshop (SSP)*, 2018.

Bibliography

[1] H. K. Aggarwal and A. Majumdar, "Generalized synthesis and analysis prior algorithms with application to impulse denoising," in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing.* ACM, 2014, p. 10.

[2] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, 2013.

[3] B. Wen, S. Ravishankar, and Y. Bresler, "Video denoising by online 3d sparsifying transform learning," in *Image Processing (ICIP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 118–122.

[4] S. Ravishankar and Y. Bresler, "Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2519–2557, 2015.

[5] S. Mallat, *A wavelet tour of signal processing.* Academic press, 1999.

[6] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on image processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

[7] T. Gagie, G. Navarro, and S. J. Puglisi, "New algorithms on wavelet trees and applications to information retrieval," *Theoretical Computer Science*, vol. 426, pp. 25–41, 2012.

[8] R. C. Guido, J. F. W. Slaets, R. Köberle, L. O. B. Almeida, and J. C. Pereira, "A new technique to construct a wavelet transform matching a specified signal with applications to digital, real time, spike, and overlap pattern recognition," *Digital Signal Processing*, vol. 16, no. 1, pp. 24–44, 2006.

[9] K. Najarian and R. Splinter, *Biomedical signal and image processing.* CRC press, 2005.

[10] A. Depeursinge, A. Foncubierta-Rodriguez, D. Van de Ville, and H. Müller, "Rotation–covariant texture learning using steerable riesz wavelets," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 898–908, 2014.

[11] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[12] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 2091–2110, 1989.

[13] ——, "Multiresolution approximations and wavelet orthonormal bases of ^{2} ()," *Transactions of the American mathematical society*, vol. 315, no. 1, pp. 69–87, 1989.

[14] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 45, no. 5, pp. 485–560, 1992.

[15] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE transactions on signal processing*, vol. 40, no. 9, pp. 2207–2232, 1992.

[16] C. K. Chui, *An introduction to wavelets.* Academic press, 2014, vol. 1.

[17] C. K. Chui and J.-z. Wang, "A cardinal spline approach to wavelets," *Proceedings of the American Mathematical Society*, vol. 113, no. 3, pp. 785–793, 1991.

[18] C. K. Chui and J.-Z. Wang, "A general framework of compactly supported splines and wavelets," *Journal of approximation theory*, vol. 71, no. 3, pp. 263–304, 1992.

[19] M. Unser, A. Aldroubi, and M. Eden, "A family of polynomial spline wavelet transforms," *Signal processing*, vol. 30, no. 2, pp. 141–162, 1993.

[20] A. Aldroubi and M. Unser, "Families of multiresolution and wavelet spaces with optimal properties," *Numerical Functional Analysis and Optimization*, vol. 14, no. 5-6, pp. 417–446, 1993.

[21] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[22] H. Krim, D. Tucker, S. Mallat, and D. Donoho, "On denoising and best signal representation," *IEEE transactions on information theory*, vol. 45, no. 7, pp. 2225–2238, 1999.

[23] A. H. Tewfik, D. Sinha, and P. Jorgensen, "On the optimal choice of a wavelet for signal representation," *IEEE Transactions on information theory*, vol. 38, no. 2, pp. 747–765, 1992.

[24] R. Gopinath, J. Odegard, and C. Burrus, "Optimal wavelet representation of signals and the wavelet sampling theorem," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 4, pp. 262–277, 1994.

[25] J. O. Chapa and R. M. Rao, "Algorithms for designing wavelets to match a specified signal," *IEEE transactions on signal processing*, vol. 48, no. 12, pp. 3395–3406, 2000.

[26] W.-S. Lu and A. Antoniou, "Design of signal-adapted biorthogonal filter banks," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 1, pp. 90–102, 2001.

[27] M. K. Tsatsanis and G. B. Giannakis, "Principal component filter banks for optimal multiresolution analysis," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 1766–1777, 1995.

[28] A. Gupta, S. D. Joshi, and S. Prasad, "A new method of estimating wavelet with desired features from a given signal," *Signal Processing*, vol. 85, no. 1, pp. 147–161, 2005.

[29] ——, "A new approach for estimation of statistically matched wavelet," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1778–1793, 2005.

[30] A. Gupta and S. Joshi, "On the concept of intrinsic wavelet functions," in *2014 International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2014, pp. 1–5.

[31] L. K. Jiwani, S. Joshi, and G. Visweswaran, "Modulated wavelet basis," in *Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on*. IEEE, 2010, pp. 324–328.

[32] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and computational harmonic analysis*, vol. 3, no. 2, pp. 186–200, 1996.

[33] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier analysis and applications*, vol. 4, no. 3, pp. 247–269, 1998.

[34] W. Dong, G. Shi, and J. Xu, "Signal-adapted directional lifting scheme for image compression," in *2008 IEEE International Symposium on Circuits and Systems*. IEEE, 2008, pp. 1392–1395.

[35] H. J. Heijmans, B. Pesquet-Popescu, and G. Piella, "Building nonredundant adaptive wavelets by update lifting," *Applied and Computational Harmonic Analysis*, vol. 18, no. 3, pp. 252–281, 2005.

[36] G. Piella, B. Pesquet-Popescu, and H. J. Heijmans, "Gradient-driven update lifting for adaptive wavelets," *Signal Processing: Image Communication*, vol. 20, no. 9, pp. 813–831, 2005.

[37] G. Piella, B. Pesquet-Popescu, and H. Heijmans, “Adaptive update lifting with a decision rule based on derivative filters,” *IEEE Signal Processing Letters*, vol. 9, no. 10, pp. 329–332, 2002.

[38] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, “Adaptive nonseparable wavelet transform via lifting and its application to content-based image retrieval,” *IEEE transactions on image processing*, vol. 19, no. 1, pp. 25–35, 2010.

[39] Y. Liu and K. N. Ngan, “Weighted adaptive lifting-basedwavelet transform,” in *2007 IEEE International Conference on Image Processing*, vol. 3. IEEE, 2007, pp. III–189.

[40] M. C. Kale and Ö. N. Gerek, “Lifting wavelet design by block wavelet transform inversion,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2014, pp. 2619–2623.

[41] X. Zhang, W. Wang, T. Yoshikawa, and Y. Takei, “Design of iir orthogonal wavelet filter banks using lifting scheme,” *IEEE transactions on signal processing*, vol. 54, no. 7, pp. 2616–2624, 2006.

[42] J. Blackburn and M. N. Do, “Two-dimensional geometric lifting,” in *2009 16th IEEE International Conference on Image Processing (ICIP).* IEEE, 2009, pp. 3817–3820.

[43] M. Vrankic, D. Sersic, and V. Sucic, “Adaptive 2-d wavelet transform based on the lifting scheme with preserved vanishing moments,” *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 1987–2004, 2010.

[44] W. H. Organization., “Cardiovascular diseases,” Reviewed September 2016. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs317/en/

[45] H. Cao, V. Leung, C. Chow, and H. Chan, “Enabling technologies for wireless body area networks: A survey and outlook,” *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84–93, 2009.

[46] D. L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[47] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, “Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[48] A. Ravelomanantsoa, H. Rabah, and A. Rouane, “Compressed sensing: a simple deterministic measurement matrix and a fast recovery algorithm,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405–3413, 2015.

[49] J. Zhang, Z. Gu, Z. L. Yu, and Y. Li, “Energy-efficient ecg compression on wireless biosensors via minimal coherence sensing and weighted ell_1 minimization reconstruction,” *IEEE journal of biomedical and health informatics*, vol. 19, no. 2, pp. 520–528, 2015.

[50] S. Li, L. Da Xu, and X. Wang, “Compressed sensing signal and data acquisition in wireless sensor networks and internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, 2013.

[51] A. M. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, “Compressed sensing system considerations for ecg and emg wireless biosensors,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[52] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, “Design and exploration of low-power analog to information conversion based on compressed sensing,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 493–501, 2012.

[53] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, “Exploiting prior knowledge in compressed sensing wireless ecg systems,” *IEEE journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 508–519, 2015.

[54] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, “Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ecg via block

sparse bayesian learning," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 300–309, 2013.

[55] C. Soussen, J. Idier, D. Brie, and J. Duan, "From bernoulli–gaussian deconvolution to sparse signal restoration," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4572–4584, 2011.

[56] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and dc programming," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, 2009.

[57] I. W. Selesnick and I. Bayram, "Enhanced sparsity by non-separable regularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2298–2313, 2016.

[58] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.

[59] J. Woodworth and R. Chartrand, "Compressed sensing recovery via nonconvex shrinkage penalties," *Inverse Problems*, vol. 32, no. 7, pp. 75 004–75 028, 2016.

[60] Z. Chen, Y. Fu, Y. Xiang, and R. Rong, "A novel iterative shrinkage algorithm for cs-mri via adaptive regularization," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1443–1447, 2017.

[61] L. Niu, R. Zhou, Y. Tian, Z. Qi, and P. Zhang, "Nonsmooth penalized clustering via l_p regularized sparse regression," *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1423–1433, 2017.

[62] J. Scarlett, J. S. Evans, and S. Dey, "Compressed sensing with prior information: Information-theoretic limits and practical decoders," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 427–439, 2013.

[63] A. A. Saleh, F. Alajaji, and W.-Y. Chan, "Compressed sensing with non-gaussian noise and partial support information," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1703–1707, 2015.

[64] Q. Li, J. Ma, and G. Erlebacher, “A new reweighted algorithm with support detection for compressed sensing,” *IEEE Signal Processing Letters*, vol. 19, no. 7, pp. 419–422, 2012.

[65] C. Herzet, C. Soussen, J. Idier, and R. Gribonval, “Exact recovery conditions for sparse representations with partial support information,” *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7509–7524, 2013.

[66] A. Ramakrishnan and S. Saha, “Ecg coding by wavelet-based linear prediction,” *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 12, pp. 1253–1261, 1997.

[67] M. L. Hilton, “Wavelet and wavelet packet compression of electrocardiograms,” *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 5, pp. 394–402, 1997.

[68] T. Blu, “A new design algorithm for two-band orthonormal rational filter banks and orthonormal rational wavelets,” *IEEE Transactions on Signal Processing*, vol. 46, no. 6, pp. 1494–1504, 1998.

[69] G. F. Choueiter and J. R. Glass, “An implementation of rational wavelets and filter design for phonetic classification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 939–948, 2007.

[70] R. Bregovic and T. Saramaki, “Design of two-channels fir filterbanks with rational sampling factors using the frm technique,” in *2005 IEEE International Symposium on Circuits and Systems.* IEEE, 2005, pp. 1098–1101.

[71] J. Kovačević and M. Vetterli, “Perfect reconstruction filter banks with rational sampling factors,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 6, pp. 2047–2066, 1993.

[72] P. Blanc, T. Blu, T. Ranchin, L. Wald, and R. Aloisi, “Using iterated rational filter banks within the arsis concept for producing 10m landsat multispectral images,” *International journal of remote sensing*, vol. 19, no. 12, pp. 2331–2343, 1998.

[73] A. Baussard, F. Nicolier, and F. Truchetet, "Rational multiresolution analysis and fast wavelet transform: application to wavelet shrinkage denoising," *Signal Processing*, vol. 84, no. 10, pp. 1735–1747, 2004.

[74] P. Auscher, "Wavelet bases for l2 (r) with rational dilation factor," *Wavelets and their applications*, pp. 439–451, 1992.

[75] T.-T. Le, M. Ziebarth, T. Greiner, and M. Heizmann, "Optimized size-adaptive feature extraction based on content-matched rational wavelet filters," in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European.* IEEE, 2014, pp. 1672–1676.

[76] O. Chertov, V. Malchykov, and D. Pavlov, "Non-dyadic wavelets for detection of some click-fraud attacks," in *Signals and Electronic Systems (ICSES), 2010 International Conference on.* IEEE, 2010, pp. 401–404.

[77] L. Yu and L. B. White, "Optimum receiver design for broadband doppler compensation in multipath/doppler channels with rational orthogonal wavelet signaling," *Signal Processing, IEEE Transactions on*, vol. 55, no. 8, pp. 4091–4103, 2007.

[78] ——, "Broadband passive sonar detection using rational orthogonal wavelet filter banks," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2011 Seventh International Conference on.* IEEE, 2011, pp. 461–466.

[79] I. Bayram and I. W. Selesnick, "Frequency-domain design of overcomplete rational-dilation wavelet transforms," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 2957–2972, 2009.

[80] ——, "Design of orthonormal and overcomplete wavelet transforms based on rational sampling factors," in *Optics East 2007.* International Society for Optics and Photonics, 2007, pp. 67 630H–67 630H.

[81] ——, "Overcomplete discrete wavelet transforms with rational dilation factors," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 131–145, 2009.

[82] L. Yu and L. B. White, “Complex rational orthogonal wavelet and its application in communications,” *IEEE Signal Processing Letters*, vol. 13, no. 8, pp. 477–480, 2006.

[83] S. T. N. Nguyen and B. W.-H. Ng, “Design of two-band critically sampled rational rate filter banks with multiple regularity orders and associated discrete wavelet transforms,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3863–3868, 2012.

[84] J. Kovačević and W. Sweldens, “Wavelet families of increasing order in arbitrary dimensions,” *Image Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 480–496, 2000.

[85] P. P. Vaidyanathan, *Multirate systems and filter banks.* Pearson Education India, 1993.

[86] R. Baraniuk, M. A. Davenport, M. F. Duarte, C. Hegde *et al.*, “An introduction to compressive sensing,” *Connexions e-textbook*, 2011.

[87] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[88] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[89] E. J. Candès, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[90] Z. Chen and J. J. Dongarra, “Condition numbers of gaussian random matrices,” *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 3, pp. 603–620, 2005.

[91] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, “Toeplitz-structured compressed sensing matrices,” in *Statistical Signal Pro-*

cessing, 2007. SSP'07. IEEE/SP 14th Workshop on. IEEE, 2007, pp. 294–298.

[92] H. Rauhut, "Compressive sensing and structured random matrices," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, pp. 1–92, 2010.

[93] H. Rauhut, J. Romberg, and J. A. Tropp, "Restricted isometries for partial random circulant matrices," *Applied and Computational Harmonic Analysis*, vol. 32, no. 2, pp. 242–254, 2012.

[94] Y. Sharon, J. Wright, and Y. Ma, "Computation and relaxation of conditions for equivalence between l1 and l0 minimization," *submitted to IEEE Transactions on Information Theory*, vol. 5, 2007.

[95] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[96] R. Chartrand, "Shrinkage mappings and their induced penalty functions," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 1026–1029.

[97] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted 1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5, pp. 877–905, 2008.

[98] I. Selesnick and M. Farshchian, "Sparse signal approximation via nonseparable regularization," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2561–2575, 2017.

[99] M. Nikolova, M. K. Ng, and C.-P. Tam, "Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3073–3088, 2010.

[100] A. Lanza, S. Morigi, and F. Sgallari, "Convex image denoising via non-convex regularization with parameter selection," *Journal of Mathematical Imaging and Vision*, vol. 56, no. 2, pp. 195–220, 2016.

[101] I. Selesnick, “Sparsity amplified,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4356–4360.

[102] B. Shen, I. K. Sethi, and V. Bhaskaran, “Dct convolution and its application in compressed domain,” *IEEE transactions on circuits and systems for video technology*, vol. 8, no. 8, pp. 947–952, 1998.

[103] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The jpeg 2000 still image compression standard,” *IEEE Signal processing magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[104] S. Becker, J. Bobin, and E. J. Candès, “Nesta: a fast and accurate first-order method for sparse recovery,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[105] D. L. Donoho and J. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[106] P.-E. Ng and K.-K. Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images,” *IEEE Transactions on image processing*, vol. 15, no. 6, pp. 1506–1516, 2006.

[107] E. van den Berg and M. P. Friedlander, “Probing the pareto frontier for basis pursuit solutions,” *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008. [Online]. Available: http://link.aip.org/link/?SCE/31/890

[108] ——, “SPGL1: A solver for large-scale sparse reconstruction,” June 2007, http://www.cs.ubc.ca/labs/scl/spgl1.

[109] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, physiotoolkit, and physionet,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[110] http://www.dave40.co.uk/1/VwSlAr.php?id=15.

[111] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[112] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (wdd) measure for ecg signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1422–1430, 2000.

[113] A. Achim, B. Buxton, G. Tzagkarakis, and P. Tsakalides, "Compressive sensing for ultrasound rf echoes using a-stable distributions," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE.* IEEE, 2010, pp. 4304–4307.

[114] G. Tzagkarakis, "Bayesian compressed sensing using alpha-stable distributions," Ph.D. dissertation, Ph. D. dissertation, Dept. Comput. Sci., Univ. Crete, Crete, Greece, 2009.

[115] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," in *Electronic Imaging 2006.* International Society for Optics and Photonics, 2006, pp. 606 509–606 509.

[116] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[117] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[118] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, "Compressed sensing based method for ecg compression," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on.* IEEE, 2011, pp. 761–764.

[119] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *Signal Processing Letters, IEEE*, vol. 16, no. 6, pp. 449–452, 2009.

[120] Z. He, T. Ogawa, and M. Haseyama, “The simplest measurement matrix for compressed sensing of natural images,” in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 4301–4304.

[121] L. Gan, “Block compressed sensing of natural images,” in *2007 15th International conference on digital signal processing*. IEEE, 2007, pp. 403–406.

[122] B. E. Usevitch, “A tutorial on modern lossy wavelet image compression: foundations of jpeg 2000,” *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 22–35, 2001.

[123] M. Lightstone, E. Majani, and S. K. Mitra, “Low bit-rate design considerations for wavelet-based image coding,” *Multidimensional systems and signal processing*, vol. 8, no. 1-2, pp. 111–128, 1997.

[124] W. K. Pratt, “Median filtering,” *Image Process. Inst., Univ. Southern California, Los Angeles*, 1975.

[125] U. L. Wijewardhana, E. Belyaev, M. Codreanu, and M. Latva-Aho, “Signal recovery in compressive sensing via multiple sparsifying bases,” in *Data Compression Conference (DCC), 2017*. IEEE, 2017, pp. 141–150.

[126] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.