WAVELET BASED IMAGE CODING AND INTERPOLATION

ENGİN TAMER
B.S., Electronics Engineering, Işık University, 2007

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Electronics Engineering

IŞIK UNIVERSITY
2009

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

WAVELET BASED IMAGE CODING AND INTERPOLATION

ENGİN TAMER

APPROVED BY:

Assist. Prof. Hasan F. Ateş (Thesis Supervisor)	Işık University	____________________
Assist. Prof. Onur Kaya	Işık University	____________________
Assoc. Prof. Ercan Solak	Işık University	____________________

APPROVAL DATE: ..../..../....

“A person who never made a mistake never tried anything new.”

Albert Einstein

WAVELET BASED IMAGE CODING AND INTERPOLATION

Abstract

The need of higher resolution on imaging systems and displays increases everyday. Data rates and bandwidth are still limited to satisfy the demands of enhanced resolutions. So, we should develop intelligent enhancement tools which yield higher resolution images with comparably limited bitrate. In this thesis, we examine the quality enhancement capabilities of two different approaches, i.e. image coding and image interpolation. Better image coding algorithms are capable of producing enhanced higher detail images at the same bitrate. For that purpose, we design an efficient and intelligent wavelet based image coding algorithm that codes the hierarchical description of wavelet coefficients instead of coding themselves. Namely, we introduce the hierarchical quantization index tree which is composed of quantization index classes. These index classes are constructed using combination of similar wavelet coefficients which leads an adaptive structure. Then, this hierarchical tree is optimized by a simple rate-distortion analysis to achieve efficient bit allocation among various different regions of natural images. In the second part of the thesis, we propose a wavelet based interpolation algorithm that exploits the correlation between high resolution(HR) and low resolution(LR) images. Basically, we design linear minimum mean square error filters between HR and LR images to recover lost high frequency information. For modeling the relationship between two resolutions, we use two different approaches, i.e. block based filter design and context adaptive filter design. In block based filter design algorithm, we partition the image into blocks to capture local frequency variations. In context adaptive algorithm, we use a simple context to adapt different image structures. We also briefly mention how these new approaches can be integrated into a novel coding + resolution enhancement joint framework. Simulations show that both our coding and interpolation algorithms perform better than most existing schemes.

DALGACIK BAZLI İMGE KODLAMA VE ARADEĞERLEME

Özet

Görüntüleme sistemleri ve ekranlardaki daha yüksek çözünürlük ihtiyacı gün geçtikçe artmaktadır. Veri gönderim hızları ve band genişliği çözünürlük iyileştirme ihtiyacına göre yetersiz kalmaktadır. Bu nedenle, neredeyse sabit bit hızına sahip yüksek çözünürlüklü imge elde etmeye yarayan akıllı iyileştirme araçları geliştirmeliyiz. Bu tezde, iki farklı yaklaşımın, yani imge kodlama ve imge aradeğerlemenin kalite iyileştirme yeteneklerini inceliyoruz. Daha iyi imge kodlama algoritmaları aynı bit hızı için daha yüksek ayrıntı içeren iyileştirilmiş imge elde etme olanağına sahiptir. Bu amaçla, dalgacık katsayılarının kendilerini kodlamak yerine, onların sıradüzensel tanımını kodlayan verimli ve akıllı bir dalgacık bazlı imge kodlama algoritması tasarlıyoruz. Yani, nicemleme indeks sınıflarının birleşmesinden oluşan sıradüzensel nicemleme indeks ağacını tanıtıyoruz. Bu indeks sınıfları, uyarlanabilir bir yapı sağlayacak biçimde benzer dalgacık katsayılarının birleştirilmesi ile oluşturulmaktadır. Daha sonra, bu sıradüzensel ağaç, doğal imgelerin farklı bölgeleri arasında verimli bit dağıtımını sağlamak amacıyla basit bir hız-bozulum analizi ile eniyilenmektedir. Tezin ikinci bölümünde, yüksek çözünürlüklü(YÇ) imge ile düşük çözünürlüklü(DÇ) imge arasındaki ilintiden yararlanan dalgacık bazlı bir imge aradeğerleme algoritması öneriyoruz. Temel olarak, kayıp yüksek frekans bilgisini toparlamak amacıyla, YÇ ve DÇ imge arasında doğrusal en düşük ortalama karesel hata süzgeçleri tasarlıyoruz. İki çözünürlük arasındaki ilişkiyi modellemek için iki farklı yaklaşım kullanıyoruz, bunlar blok bazlı süzgeç tasarımı ve bağlam uyarlamalı süzgeç tasarımıdır. Her iki algoritma içinde, DÇ imgeyi yeniden boyutlandırmak amacıyla bir başlangıç aradeğerleme yöntemi kullanıyoruz. Blok bazlı süzgeç tasarımı algoritmasında yerel frekans değişimlerini yakalamak amacıyla imgeyi bloklara bölüyoruz. Bağlam uyarlamalı algoritmada farklı imge yapılarına uyarlanabilirlik sağlamak amacıyla basit bir bağlam kullanıyoruz. Ayrıca, bu yaklaşımların nasıl yeni bir kodlama + çözünürlük iyileştirme çatısı altında biraraya getirilebileceğinden kısaca bahsediyoruz. Benzetimler, kodlama ve aradeğerleme algoritmalarımızın varolan yaklaşımlardan daha iyi sonuçlar verdiğini göstermektedir.

Acknowledgements

First, I would like to thank my thesis supervisor, Assist. Prof. Hasan Fehmi Ateş for his valuable guidance and endless support. Also, I would like to express my gratitude to my research buddy, Burak Çizmeci, this thesis would not have been the same without his splendid ideas. And, I should say that, it is an honor for me to work with very talented researchers as Mehmet Güneş and Murat İşleyen. My special thanks must go to my family and my girl for their patience and understanding throughout my thesis.

Dedicated to my family,

İhsan, Sönmez, Derya and Ayhan

Table of Contents

Abstract iii

Özet iv

Acknowledgements v

List of Tables ix

List of Figures x

List of Symbols xii

List of Abbreviations xiv

1 Introduction **1**
- 1.1 Overview of Wavelet Research 1
- 1.2 Image Coding 3
- 1.3 Image Interpolation 4
- 1.4 Overview of the Thesis 5

2 Wavelet Theory and Applications to Image Processing **8**
- 2.1 Wavelets in Perspective 8
- 2.2 Continuous Wavelet Transform 11
- 2.3 Discrete Wavelet Transform 13
  - 2.3.1 Multiresolution Analysis 15
  - 2.3.2 Scaling Functions 17
  - 2.3.3 Construction of Wavelet Bases Using Multiresolution Analysis 20
- 2.4 Subband Decomposition 24
- 2.5 Discrete Wavelet Transform in Image Processing 27
- 2.6 Image Coding Algorithms using Wavelet Transform 30
- 2.7 Image Interpolation Algorithms using Wavelet Transform 34

3 Wavelet Based Image Coding **37**
- 3.1 Hierarchical Quantization Index Classes 38
- 3.2 Class Assignment Functions 41
- 3.3 Subband Coding by Quantization Index Hierarchy 43

3.4 Simulation Results 47

4 Wavelet Based Image Interpolation 56

4.1 Wavelet Based Image Interpolation Algorithms 57

4.2 Our Resolution Enhancement Approach 58

4.3 Block Based Filter Design Algorithm 60

4.4 Context Adaptive Filter Design Algorithm 63

4.5 Simulation Results 69

5 Conclusion 78

References 79

List of Tables

3.1 PSNR results of HIC coder for circular and max assignment functions 48
3.2 PSNR results of HIC coder and state of the art wavelet coders for Lena 48
3.3 PSNR results of HIC coder and state of the art wavelet coders for Barbara 49

4.1 PSNR results of block based filter design algorithm for Lena . . . 69
4.2 PSNR results of block based filter design algorithm for Barbara . 70
4.3 PSNR results of Zerofill algorithm for Lena and Barbara 70
4.4 PSNR results of context adaptive filter design algorithm for Lena 72
4.5 PSNR results of context adaptive filter design algorithm for Barbara 73
4.6 PSNR results of state of the art algorithms in image interpolation for Lena and Barbara 74

List of Figures

2.1 Wave(Sinusoid) 9

2.2 Wavelet(Daubechies ψ_{D8}) 9

2.3 Wavelet(Daubechies ψ_{D20}) 10

2.4 Sampling Grid for Discrete Wavelet Transform 15

2.5 Signal $f(t)$ and its representations in V_6 to V_3 21

2.6 Representation of the spaces V_6 to V_3 and W_5 to W_3 in the frequency domain 22

2.7 Splitting the signal space using ideal low-pass(H_0) and high-pass(H_1) filters 24

2.8 The filter bank for splitting low frequency bands consecutively, as in DWT 25

2.9 Frequency domain representation of the non-uniform subband structure(DWT structure) 25

2.10 The filter bank for splitting both low and high frequency bands consecutively 26

2.11 Frequency domain representation of the uniform subband structure 26

2.12 Analysis part and synthesis part of the subband filtering 27

2.13 2-D separable analysis filter bank 28

2.14 Lena image 29

2.15 1-level DWT of Lena image 29

2.16 2-level DWT of Lena image 30

2.17 Periodic and symmetric extension of image data 30

2.18 Parent-children dependency of subbands 32

3.1 Top levels of the hierarchical index tree 41

3.2 Circular assignment classes 42

3.3 Max assignment classes 43

3.4 Original Lena image 49

3.5 1.00bpp coded Lena image using HIC coder PSNR:40.70 50

3.6 0.50bpp coded Lena image using HIC coder PSNR:37.45 50

3.7 0.50bpp coded Lena image using EBCOT coder PSNR:37.43 . . . 51

3.8 0.25bpp coded Lena image using HIC coder PSNR:34.37 51

3.9 0.25bpp coded Lena image using EBCOT coder PSNR:34.32 . . . 52

3.10 Original Barbara image 53

3.11 1.00bpp coded Barbara image using HIC coder PSNR:37.05 53

3.12 0.50bpp coded Barbara image using HIC coder PSNR:32.14 54
3.13 0.50bpp coded Barbara image using EBCOT coder PSNR:32.50 . 54
3.14 0.25bpp coded Barbara image using HIC coder PSNR:28.34 55
3.15 0.25bpp coded Barbara image using EBCOT coder PSNR:28.53 . 55

4.1 Basic encoder/decoder framework 59
4.2 Our Encoder Assisted Resolution Enhancement Approach 59
4.3 Block scheme of our filter design algorithm with initial interpolation 61
4.4 Block scheme of our filter design algorithm with undecimated wavelet transform 64
4.5 Block scheme of our filter design algorithm with wavelet transform without decimation 65
4.6 Original Lena image(Left), Zerofill reconstruction of Lena image(Right) PSNR:35.26 71
4.7 Block based algorithm using bicubic interpolation for Lena with block size 8 × 8(Left) PSNR:36.43 and 32 × 32(Right) PSNR:35.65 72
4.8 Block based algorithm with wavelet transform without decimation(Left) PSNR:36.38 and undecimated wavelet transform(Right) PSNR:36.32 for Lena 73
4.9 Original Barbara image(Left), Zerofill reconstruction of Barbara image(Right) PSNR:25.85 74
4.10 Block based algorithm using bicubic interpolation for Barbara with block size 8 × 8(Left) PSNR:29.90 and 32 × 32(Right) PSNR:27.90 75
4.11 Block based algorithm with wavelet transform without decimation(Left) PSNR:29.18 and undecimated wavelet transform(Right) PSNR:29.83 for Barbara 75
4.12 Context adaptive algorithm using undecimated wavelet transform for Lena with window size 64(Left) PSNR:36.50 and 1024(Right) PSNR:35.79 76
4.13 Context adaptive algorithm with bicubic interpolation(Left) PSNR:36.34 and wavelet transform without decimation(Right) PSNR:36.45 for Lena 76
4.14 Context adaptive algorithm using undecimated wavelet transform for Barbara with window size 64(Left) PSNR:27.64 and 1024(Right) PSNR:26.16 77
4.15 Context adaptive algorithm with bicubic interpolation(Left) PSNR:27.19 and wavelet transform without decimation(Right) PSNR:27.01 for Barbara 77

List of Symbols

t	time variable
$f(.)$	function or signal
$\psi(.)$	mother wavelet
$\psi_{s\tau}(.)$	daughter wavelet
s	scaling parameter
τ	shift parameter
c	constant
$\Psi(.)$	Fourier transform of mother wavelet
H	analysis filter
G	synthesis filter
H_0	low-pass analysis filter
H_1	high-pass analysis filter
G_0	low-pass synthesis filter
G_1	high-pass synthesis filter
LL	low-pass subband
LH	horizontal subband
HL	vertical subband
HH	diagonal subband
W	subband
$c(.,.)$	wavelet coefficients
$Q[.]$	quantization function
$s(.,.)$	signs of wavelet coefficients
$\tilde{c}(.,.)$	reconstructed wavelet coefficients
$i(.,.)$	quantization index/level of wavelet coefficients
T	dead-zone interval size
q	quantization step size
C_r	index classes

N_r	number of integer pairs in index classes
u, v	level of the hierarchy
$\Gamma_{u,v}(.,.)$	class assignment variables
$L_{u,v}(.,.)$	Lagrangian cost
$I(.,.)$	sign bit cost
f_{HR}	high resolution image
$f_{HR,IE}$	initially estimated high resolution image
f_{LR}	low resolution image
LH_{HR}	horizontal subband of high resolution image
HL_{HR}	vertical subband of high resolution image
$LH_{HR,IE}$	horizontal subband of initially estimated high resolution image
$HL_{HR,IE}$	vertical subband of initially estimated high resolution image
x	block number
$\Upsilon(.,.)$	mapping function

List of Abbreviations

DCT	**D**iscrete **C**osine **T**ransform
DWT	**D**iscrete **W**avelet **T**ransform
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
JPEG2000	**J**oint **P**hotographic **E**xperts **G**roup
FBI	**F**ederal **B**ureau of **I**nvestigations
EZW	**E**mbedded **Z**erotree **W**avelet
SPIHT	**S**et **P**artitioning **I**n **H**ierarchical **T**rees
SFQ	**S**pace **F**requency **Q**uantization
EBCOT	**E**mbedded **B**lock **C**oding with **O**ptimal **T**runcation
SPHE	**SPHE**rical Coder
EQ	**E**stimation **Q**uantization
EZBC	**E**mbedded **Z**ero **B**lock **C**oding
HIC	**H**ierarchical **I**ndex **C**oder
HMT	**H**idden **M**arkov **T**rees
STFT	**S**hort **T**ime **F**ourier **T**ransform
CWT	**C**ontinuous **W**avelet **T**ransform
MRA	**M**ulti**R**esolution **A**nalysis
LP	**L**ow **P**ass
HP	**H**igh **P**ass
PR	**P**erfect **R**econstruction
FIR	**F**inite **I**mpulse **R**esponse
LL	Low Pass Subband
LH	Horizontal Subband
HL	Vertical Subband

HH	Diagonal Subband
POS	**POS**itive Signed Coefficient
NEG	**NEG**ative Signed Coefficient
Z	**Z**ero Coefficient
ZTR	**Z**ero**T**ree **R**oot
IZ	**I**solated **Z**ero
LIS	**L**ist of **I**nsignificant **S**ets
LIP	**L**ist of **I**nsignificant **P**ixels
LSP	**L**ist of **S**ignificant **P**ixels
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
LSE	**L**east **S**quare **E**rror
EM	**E**xpectation **M**aximization
bpp	**b**its **p**er **p**ixel
LR	**L**ow **R**esolution
HR	**H**igh **R**esolution
LMMSE	**L**inear **M**inimum **M**ean **S**quare **E**rror
IE	**I**nitial **E**stimate
NEDI	**N**ew **E**dge **D**irected **I**nterpolation
LAWBII	**L**ocally **A**daptive **W**avelet **B**ased **I**mage **I**nterpolation

Chapter 1

Introduction

1.1 Overview of Wavelet Research

Wavelet theory is a full-grown tree with roots reaching several different fields such as applied mathematics, physics, computer science and engineering. It is grown with the efforts of researchers from many different branches from all over the world. Therefore, finding the real history and development of wavelets can be tough work to do, because there are many different opinions about it. I choose to admit the history mentioned in Ingrid Daubechies' personal point of view about where wavelets come from [1]. Daubechies is an experienced researcher in wavelet theory who has significant contributions to the field.

In the late 1970's, Jean Morlet, a geophysical engineer at the French oil company Elf Acquitane wanted an alternative for the short time Fourier transform to analyze signals which have different features in time and frequency. Morlet could use wide-band short time Fourier transform to gain time resolution for the high frequency transients and narrow-band short time Fourier transform to have good frequency resolution for the low frequency components but not both at the same time. In order to have both features in one transform, he took a windowed cosine wave and compressed it in time to get a higher frequency function or spread it out to obtain a lower frequency function and these functions were shifted in time as well to analyze what happened at different times.

Then, he would take the inner product of the signal to be analyzed with all these transform functions which he called “wavelets with constant shape”. After that, Morlet needed help to discover mathematical background of this transform and found a referred colleague Alex Grossmann who is a theoretical physicist. Grossmann constructed an exact inversion formula for Morlet’s transform and they explored several applications together. In 1985, Yves Meyer, a mathematician, heard about Morlet and Grossmann’s work and he realized that their work is the different interpretation of Alberto Calderon’s work in harmonic analysis in the 1960’s. Then, Meyer worked with this new phenomenon and constructed orthonormal wavelet basis with excellent time-frequency localization properties. Meyer’s student, Pierre Gilles Lemarie, a harmonic analyst and Guy Battle, a mathematical physicist, came up independently and by completely different techniques with constructions of wavelet bases consisting of spline functions, with better exponential decay than Meyer’s wavelets, at the price of some loss of regularity. In 1986, Stephane Mallat, a graduate student with field of specialization, computer vision and image analysis, heard about the new wavelet bases from Meyer’s graduate student. Mallat was very interested in this subject, because the philosophy of wavelet decompositions, where you use narrow functions for fine scale features and much wider functions for coarse scale features was very suitable for multiscale representations used in image analysis. Then, Mallat arranged a meeting with Meyer, and they worked together to built ”multiresolution analysis” framework with all mathematical details which made it very easy to construct other orthonormal wavelet bases. This multiresolution analysis led to a simple and recursive filtering algorithm to compute the wavelet decomposition of a function from its finest scale approximation. In 1987, orthonormal wavelet bases of compactly supported wavelets was constructed by Daubechies. Nonetheless, biorthogonal wavelets were constructed by Daubechies, wavelet packet bases were found by Ronald Coifman and Meyer and best-basis algorithms were developed by Coifman, Meyer and Victor Wickerhauser.

1.2 Image Coding

In image coding (or compression), transform-based coders are one step ahead of other coders with their coding efficiency. A typical transform-based coder transforms the data to another domain to remove redundancy, quantizes the transform coefficients and codes the quantized coefficients with some type of entropy coder. So, in this coding scheme, which transform you use is so important. Because, removing redundancy in data makes it possible to code it in an efficient way. Various types of transforms are used in image coding, but two of them are leading, Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). DCT is the basis of the well-known Joint Photographic Experts Group's JPEG algorithm [2] and also DCT is used in many other significant algorithms. However, with great energy compaction property and correspondence with the human visual system, coding systems using wavelet transform surpassed the DCT-based coders by their superior objective and subjective results. Besides, Federal Bureau of Investigations (FBI) uses its own wavelet coding algorithm to code and store the fingerprint data [3]. And also, Joint Photographic Experts Group's subsequent algorithm JPEG2000 [4] uses wavelet transform as a basis.

Image coding algorithms using wavelet transform became prevalent in last ten years and so many ingenious coding tools were developed. Among these coders, most successful ones share the same philosophy on how they group the wavelet coefficients, e.g. significant and insignificant. They use some kind of data structures to represent the significant and insignificant coefficients to reach efficient bit allocation. Embedded Zerotree Wavelet (EZW) [5], Set Partitioning in Hierarchical Trees (SPIHT) [6], Space-Frequency Quantization (SFQ) [7],Embedded Block Coding with Optimal Truncation (EBCOT) [8], Spherical Coder (SPHE) [9] and Estimation-Quantization (EQ) [10] are the main and the most successful coders in the literature.

1.3 Image Interpolation

Interpolation is a process of creating new data points in the range of known data points. When it comes to image interpolation, it is a method of resampling the image to a different grid and then finding the unknown points from known ones. There are several mathematical interpolation algorithms in the literature, and their two-dimensional versions are used in image interpolation, mainly they are bilinear interpolation, bicubic interpolation and spline interpolation, for further reading on these topics [11]. However, these algorithms suffer from significant problems, such as blurred edges, edge halos and aliasing. Algorithms like unsharp masking were developed to fix blurred edge problem of interpolation algorithms, but it is not enough to have superior quality images. And also, there are adaptive methods and edge-directed interpolation algorithms which have results slightly better than bilinear and bicubic algorithms.

In image coding, wavelets led to a discovery of new coding tools and these tools have paramount coding results in subjective and objective quality. Likewise, success of wavelets in image coding motivates new approaches to the image interpolation problem. Using wavelets in image interpolation is a recent subject compared to image coding and it is a very challenging problem with no advanced solution. In wavelet image interpolation (or resolution enhancement), generally we assume that the image we have in low resolution is the low pass subband of a wavelet transformed high resolution image that we want to obtain. Therefore, the problem is to estimate the detail coefficients of this wavelet transformed high resolution image, i.e. vertical, horizontal and diagonal subband coefficients. Algorithms worth mentioning in wavelet image interpolation are resolution enhancement using wavelet extrema extrapolation [12], wavelet interpolation with hidden Markov trees (HMT) [13], wavelet interpolation via statistical estimation [14], regularity-preserving image interpolation [15], locally adaptive image interpolation [16] and image interpolation using wavelet-based contour estimation [17].

1.4 Overview of the Thesis

In recent years, evolution of technology gradually gets faster and faster. We begin to use technology in almost every area of life and it is becoming part of our lives by the day. This progress urges all fields of technological development to grow rapidly and in parallel. So many areas are connected to each other and synchronized progress between these connected areas is a significant problem. For instance, data rates are growing fast in communication field, but not as much as the resolution of imaging systems and displays get bigger. Here, advances in coding technology helps to reduce the discrepancy between the progress speeds of these fields. Developing an efficient and advanced image coding system makes it possible to accommodate higher data rates with comparably limited bandwidth. On the other hand, since the resolution of imaging displays increases very fast, better coding techniques may not be sufficient alone to get desired objective and subjective results. Thus, image interpolation is an alternative and complementary solution to achieve higher resolution without increasing the bitrates.

In this thesis, we look at these two problems of image processing together, namely wavelet based image coding and image interpolation. We propose an "Encoder Assisted Resolution Enhancement" framework to combine the efficiency of intelligent coding tools and the resolution enhancement capabilities of interpolation algorithms together for generating better quality and resolution enhanced images with no or little extra bitrate. In other words, image coding and image interpolation are treated under the same framework where the ultimate goal is producing higher quality images at the decoder side.

Thus, we propose a wavelet based image coding algorithm which hierarchically classifies wavelet coefficients by using their quantization indices. First, we quantize wavelet coefficients using scalar quantizer. After that, we locally group wavelet coefficients of each subband individually based on their quantization indices to form index classes. At the first level of this hierarchical index tree, we combine neighboring wavelet coefficients to form index classes that represent some statistics of the quantization indices of these coefficients. Upper levels of tree is constructed using the same method, by pairing neighboring coefficients to have larger subsets. At each level of the tree, the class assignment of a given subset describes some local statistics of the quantization levels of corresponding coefficients. This hierarchical quantization index tree provides an inherently adaptive bit allocation. Coding this index tree means that coding class membership information of each level of the tree. The bitrate needed to code this class membership information of a group of coefficients is related with the average quantization levels of these coefficients. Hence, majority of the bitrate will be spent to code regions of subbands which have more significant coefficients than others. This scheme allocates the bitrate by considering the local statistics of regions of subbands which leads accurate and efficient bit allocation. Nonetheless, we propose to use rate-distortion cost analysis to optimize the coding efficiency of the hierarchical quantization index tree for better bit allocation.

Also, we propose a wavelet based algorithm for image interpolation problem. We construct encoder assisted resolution enhancement framework using this interpolation algorithm and also using our wavelet based coding algorithm. In this framework, we assume that we have the high resolution (HR) image available on the encoder side, however, instead of coding and sending HR image, we can code and send low resolution (LR) image with some extra side information which can be used by the decoder for interpolation. Basically, we can extract some information that is very valuable for interpolation from high resolution image on the encoder side and we can send this extraction as side information with coded sequence. Then, we can use this extra information on the decoder side to improve the interpolation of the coded image. We propose to use two different algorithms to extract this information, i.e. block based filter design and context adaptive filter design algorithms. In both algorithms, we design linear minimum mean square error filters to model the correlation between HR and LR image. In block based algorithm, we use blocks to catch local high frequency variations. Moreover, in context adaptive algorithm, we try to find a suitable adaptive context to model the relationship between HR and LR image.

In Chapter 2, we give brief mathematical information about wavelets, wavelet transform and its applications to image coding and interpolation. Then, in Chapter 3, we explain detailed information about our wavelet-based image coding algorithm. Chapter 4 introduces our wavelet based image interpolation algorithm and describes the encoder assisted resolution enhancement framework. In Chapter 5, we have conclusion of the thesis.

Chapter 2

Wavelet Theory and Applications to Image Processing

There are two common representations for a one-dimensional signal, i.e. temporal representation and spectral representation. These are the dual representations and they appear in two opposite corners of the time-frequency resolution plane. Therefore, higher accuracy in one domain means complete uncertainty in the other. Furthermore, different choices of the time-frequency resolution will result in different signal representations. So, there are other representations which lie down in the middle of the time-frequency plane and handle time-frequency trade-off better than the duals, e.g. Short Time Fourier Transform (STFT), Gabor Transform and Wavelet Transform.

We are going to analyze wavelet transform which has superior time-frequency localization, outstanding match with human visual system and excellent energy compaction property.

2.1 Wavelets in Perspective

A wave is defined as an oscillating function of time, such as sinusoid. Fourier Transform is wave analysis. It expands signals in terms of sinusoids, i.e. complex exponentials, which has proven to be extremely valuable for analyzing periodic, time-invariant, or stationary signals. A wavelet is a “small wave” which has its energy concentrated in time to give a tool for the analysis of transient, nonstationary or time-varying signals. Wavelet still has the oscillating wave-like characteristic but also has the ability to allow simultaneous time and frequency analysis with a flexible mathematical foundation. This is illustrated in Figure 2.1, the wave (sinusoid) oscillating with equal amplitude over $-\infty \leq t \leq \infty$ with infinite energy and in Figure 2.2, 2.3 wavelets having their finite energy concentrated around a point. We will take wavelets and use them in a series expansion of signals much the same way a Fourier transform uses the wave, i.e. sinusoid to represent a signal.

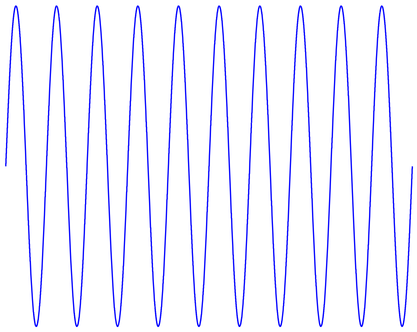

Figure 2.1: Wave(Sinusoid)

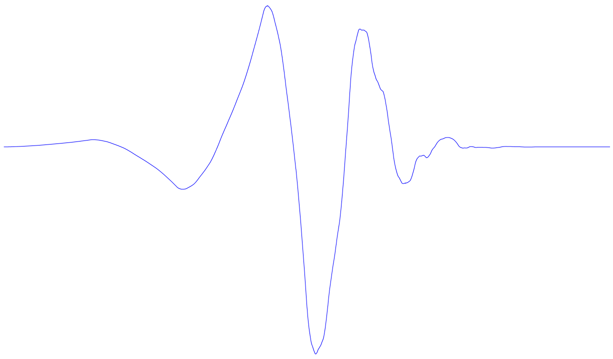

Figure 2.2: Wavelet(Daubechies ψ_{D8})

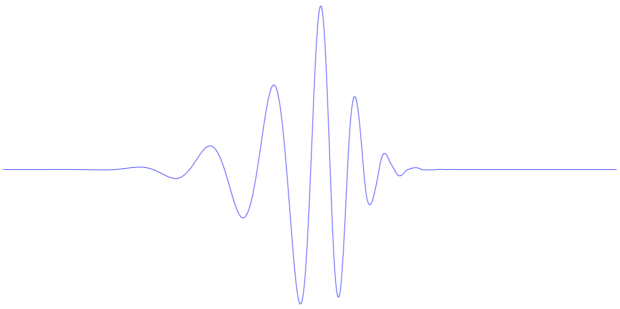

Figure 2.3: Wavelet(Daubechies ψ_{D20})

A signal or function $f(t)$ can often be better analyzed, described, or processed if expressed as a linear decomposition by

$$f(t) \;=\; \sum_{\ell} a_\ell \; \psi_\ell(t) \tag{2.1}$$

where ℓ is an integer index for finite or infinite sum, a_ℓ are the real-valued expansion coefficients, and $\psi_\ell(t)$ are a set of real-valued functions of t called the expansion set. If the expansion is unique, the set is called a basis for the class of functions that can be so expressed. If the basis is orthogonal, meaning

$$\langle \psi_k(t), \psi_\ell(t) \rangle \;=\; \int \psi_k(t)\; \psi_\ell(t)\; dt \;=\; 0 \qquad\qquad k \neq \ell \tag{2.2}$$

then the coefficients can be calculated by the inner product

$$a_k = \langle f(t), \psi_k(t) \rangle \;=\; \int f(t)\; \psi_k(t)\; dt \tag{2.3}$$

For a Fourier series, the orthogonal basis functions $\psi_k(t)$ are $\sin(k\omega_0 t)$ and $\cos(k\omega_0 t)$ with frequencies of $k\omega_0$. For the wavelet expansion, a two-parameter system is constructed such that Equation 2.1 becomes

$$f(t) \;=\; \sum_{j} \sum_{k} a_{j,k}\; \psi_{j,k}(t) \tag{2.4}$$

where both j and k are integer indices and the $\psi_{j,k}(t)$ are the wavelet expansion functions that usually form an orthogonal basis [18].

2.2 Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) can be considered the most generalized representation of the wavelet transform. So, by being general, CWT is highly redundant transform. By using sampling theorem in time and frequency or adding some constraints on basis functions used in transform, we can reduce redundancy and reach the Discrete Wavelet Transform (DWT) and the subband transform concepts at the cost of reduced versatility.

The wavelet transform maps a signal or function, $f(t)$, into a two dimensional domain, i.e. time-scale plane, and is denoted by $W_f(s,\tau)$ given by

$$\begin{aligned} W_f(s,\tau) \;&=\; \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t)\; \psi^{*}\left(\frac{t-\tau}{s}\right)\; dt \\ &=\; \int_{-\infty}^{+\infty} f(t)\; \psi^{*}_{s\tau}(t)\; dt \end{aligned} \tag{2.5}$$

where $\psi(t)$ is in general called the mother wavelet, and the basis functions of the transform, called daughter wavelets, are given by

$$\psi_{s\tau}(t) \;=\; \frac{1}{\sqrt{s}}\, \psi\left(\frac{t-\tau}{s}\right) \tag{2.6}$$

Equation 2.5 is also known as the expansion formula, the forward transform or analysis; $\psi_{s\tau}(t)$ is a set of basis functions obtained from the mother wavelet $\psi(t)$

by compression or dilation using scaling parameter **s** and the temporal translation using shift parameter τ. And, there is a normalization factor $1/\sqrt{s}$ in Equation 2.6 to keep the energy of the daughter wavelets constant. Without this factor, for different s values, the wavelets dilate or compress and their total energy changes. It is also to be noted that

$$\psi(t) \;=\; \psi_{10}(t) \tag{2.7}$$

The scaling parameter **s** is positive and varies from 0 to ∞. For $s < 1$, the transform performs compression of the signal, and for $s > 1$, the transform performs dilation of the signal. The signal $f(t)$ can be recovered from the wavelet coefficients $W_f(s, \tau)$ by the inverse wavelet transform given by

$$f(t) \;=\; \frac{1}{c} \int_{-\infty}^{+\infty} \int_{0}^{+\infty} W_f(s,\tau)\, \psi\left(\frac{t-\tau}{s}\right) \frac{ds}{s^2}\, d\tau \tag{2.8}$$

provided that the constant c is

$$c \;=\; \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{\omega}\, d\omega \quad < \quad +\infty \tag{2.9}$$

Equation 2.8 is also referred to as the reconstruction formula, inverse transform, or synthesis, and Equation 2.9 is generally known as the admissibility condition. It can be shown that functions $\psi(t)$ satisfying the admissibility condition can be used to first analyze and then reconstruct a signal without loss of information. The admissibility condition implies that the Fourier transform of $\psi(t)$ vanishes at the zero frequency, i.e.

$$\left.|\Psi(\omega)|^2\right|_{\omega=0} \;=\; 0 \tag{2.10}$$

This means that wavelets must have a band-pass like spectrum. A zero at the zero frequency also means that the average value of the wavelet in the time domain must be zero [19]

$$\int_{-\infty}^{+\infty} \psi(t)\, dt \;=\; 0 \tag{2.11}$$

2.3 Discrete Wavelet Transform

In the continuous wavelet transform, the scaling parameter s and the shift parameter τ are assumed to be continuous in value ($s \in \mathbb{R}^+$ and $\tau \in \mathbb{R}$), so that the CWT is defined in the $(\mathbb{R}^+)^2$ plane. Since no new information can be created by this transform, the same information contained in the signal $f(t)$ with $t \in \mathbb{R}$ is available with the CWT. The increase in complexity from $t \in \mathbb{R}$ to $(s, \tau) \in (\mathbb{R}^+)^2$ results only in a redundant representation of the signal. This redundancy can be reduced by discretizing the transform parameters (s, τ). It should be noted that, after making parameters discrete, we can still achieve reconstruction without any loss of information.

The (s, τ) plane is sampled by first taking the scaling coefficient s sampled as

$$s \;\rightarrow\; s_0^j\,, \quad j \in \mathbb{Z}\,, \quad s_0 \;\neq\; 1 \tag{2.12}$$

For $j = 0$, we sample the shift parameter τ by taking integer multiples of a constant $\tau_0 > 0$. The step τ_0 should be chosen in such a way that $\psi(t - k\tau_0)$ covers the whole time axis. Since the basis functions are rescaled for $j \neq 0$, also the step size τ should be a function of the scaling parameter. If we define the width of the wavelet $\psi(t)$ as ΔT

$$\Delta T^2(\psi) \; = \; \frac{\displaystyle\int_{-\infty}^{+\infty} t^2 \, |\psi(t)|^2 \, dt}{\displaystyle\int_{-\infty}^{+\infty} |\psi(t)|^2 \, dt} \tag{2.13}$$

and the width of $\psi_{s,j}(t)$, with $s = s_0^j$, times the width of $\psi(t)$

$$\Delta T \left(\psi_{s_0^j,0}(t)\right) \; = \; s_0^j \, \Delta T \left(\psi(t)\right) \tag{2.14}$$

then it is obvious that to cover the entire axis we need a step on the order of $\tau_0 s_0^j$. Hence, the sampling of the wavelet transform is commonly defined as

$$\begin{gathered} s \; = \; s_0^j \; , \qquad \tau \; = \; k \; \tau_0 \; s_0^j \; , \qquad j, k \in \mathbb{Z} \\ with \quad s_0 \; \neq \; 1 \; , \quad \tau_0 \; \neq \; 0 \end{gathered} \tag{2.15}$$

Large basis functions (s_0^j is large) are thus shifted in large steps ($\tau_0 s_0^j$), whereas small basis functions are shifted in small steps. This grid is shown in Figure 2.4. In order for the sampling of the (s, τ) plane to be sufficiently fine resolution, s_0 has to be chosen sufficiently close to 1 and τ_0 close to 0. These discretized version of continuous transforms are examples of wavelet frames.

Using the discretization of Equation 2.15, we obtain the following expressions for the daughter wavelets

$$\begin{aligned} \psi(s,\tau)\Big|_{s=s_0^j, \tau=k\tau_0 s_0^j} \; &= \; \psi(s_0^j, k\tau_0 s_0^j) \\ &= \; \frac{1}{\sqrt{s_0^j}} \; \psi \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right) \\ &= \; s_0^{-j/2} \; \psi \left(s_0^{-j} t \; - \; k\tau_0 \right) \; = \; \psi_{j,k}(t) \end{aligned} \tag{2.16}$$

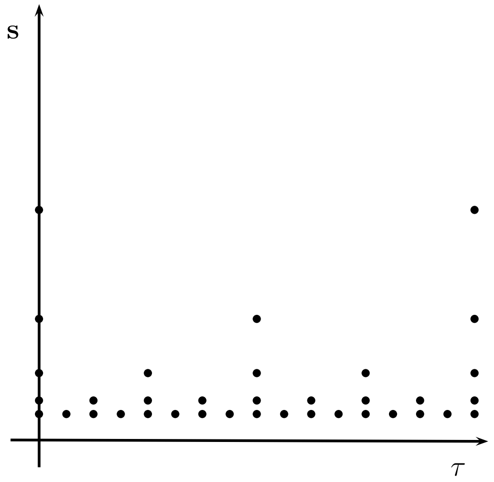

Figure 2.4: Sampling Grid for Discrete Wavelet Transform

From this follows the expression for the Discrete Wavelet Transform

$$\begin{aligned} W_f(j,k) &= \int_{-\infty}^{+\infty} f(t)\ \psi_{j,k}(t)\ dt \\ W_f(j,k) &= s_0^{-j/2} \int_{-\infty}^{+\infty} f(t)\ \psi\left(s_0^{-j}t - k\tau_o\right)\ dt \end{aligned} \tag{2.17}$$

Here, we still need an infinite number of scalings and translations to calculate the discrete wavelet transform. So, for understanding the realization and practical aspects of the transform, we are going to define some concepts like orthogonality, orthonormality of basis functions and use the link between wavelets, multiresolution analysis and filters.

2.3.1 Multiresolution Analysis

Multiresolution analysis (MRA), formulated in 1986 by Mallat and Meyer, provided the natural framework for understanding the logic behind the wavelet basis and subsequently, became the tool for constructing new ones.

The Multiresolution Theorem states that, given a sequence $\{V_j\}$ where j is an element of a closed subspace of $L^2(\mathbb{R})$, the sequence defines a multiresolution approximation of a function $f(t)$ if the following six properties are satisfied.

Property 1

$$\forall j, k \in \mathbb{Z}, \qquad f_j \in V_j \;\Leftrightarrow f_j(t - 2^j n) \in V_j \tag{2.18}$$

This means that if $f_j(t)$ is an element of V_j, also its translated version $f_j(t - 2^j n)$ belongs to V_j. Note that not all possible translations are acceptable, only those multiples of 2^j.

Property 2

$$\forall j \in \mathbb{Z}, \qquad V_{j+1} \;\subset\; V_j \tag{2.19}$$

This means that the signal approximation at scale j can also be considered a signal approximation at a scale $j+1$. This is a causality property.

Property 3

$$\forall j \in \mathbb{Z}, \qquad f(t) \in V_j \;\Leftrightarrow f(t/2) \in V_{j+1} \tag{2.20}$$

Dilating the function in V_j by a factor of 2 enlarges the details and again can be considered an approximation at a scale $j+1$

Property 4

$$\lim_{j\to\infty} V_j \;=\; \bigcap_{k=-\infty}^{+\infty} V_k \;=\; 0 \tag{2.21}$$

If j tends to infinity, then the resolution 2^{-j} goes to zero and thus the projection of $f(t)$ on V_j as j tends to infinity is zero since we have lost all the details; in other words

$$\lim_{j \to \infty} \|P_{V_j} f\| \;=\; 0 \tag{2.22}$$

where $P_{V_j} f$ represents the projection of $f(t)$ onto space V_j.

Property 5

$$\lim_{2^{-j} \to \infty} V_j \;=\; \bigcup_{k=-\infty}^{+\infty} V_k \;=\; L^2(\mathbb{R}) \tag{2.23}$$

When the resolution 2^{-j} goes to infinity, we can perfectly represent the signal, hence, the signal approximation is equivalent to $f(t)$.

Property 6

Given a subspace V_0, there exists a function $\phi(t)$ so that $\phi(t-k)_{k \in \mathbb{Z}}$ is an orthonormal basis for V_0. Furthermore, we will impose that the function $\phi(t)$ is such that

$$\phi_{j,k}(t) \;=\; 2^{-j/2} \phi(2^{-j} t - k) \tag{2.24}$$

is an orthonormal basis of V_j.

2.3.2 Scaling Functions

Multiresolution Analysis ensures that if the properties are satisfied , there exists an orthonormal basis $\varphi_{j,k}(t)$ such that the signal detail on the subspace V_j is given by

$$d_j(t) \;=\; \sum_{k=-\infty}^{+\infty} \langle f, \varphi_{j,k} \rangle \varphi_{j,k} \tag{2.25}$$

and the projection of the signal onto the space V_j is linked to the projection on the space V_{j+1} by

$$P_j f \;=\; P_{j+1} f \;+\; d_{j+1} \tag{2.26}$$

The approximation of a signal f at the resolution 2^{-j} is defined as the orthogonal projection on V_j, and it is represented as $P_{\mathrm{V}_j} f = P_j f$. To compute this projection, we must find an orthonormal basis for V_j. It is interesting to find out that we can construct an orthogonal basis for each space V_j by dilating and scaling a single function $\phi(t)$ similar to the wavelet function. The function $\phi(t)$ is called the scaling function.

We can define the scaling function $\phi(t)$ using its Fourier transform $\Phi(\omega)$ and the following relationship

$$\Phi(\omega) \;=\; \frac{\Theta(\omega)}{\left(\sum_{n=-\infty}^{+\infty} |\Theta(\omega - 2\pi n)|^2 \right)^{1/2}} \tag{2.27}$$

and the family of functions $\{\phi_{j,k}\}_{k \in \mathbb{Z}}$ calculated as

$$\phi_{j,k}(t) \;=\; \frac{1}{\sqrt{2^j}} \, \phi\left(\frac{t-k}{2^j} \right) \tag{2.28}$$

is then an orthonormal basis for V_j for all $j \in \mathbb{Z}$.

The orthogonal projection of f over V_j is thus obtained with an expansion in the scaling orthogonal basis

$$P_{V_j} f \;=\; \sum_{k=-\infty}^{+\infty} \langle f, \phi_{j,k} \rangle \phi_{j,k}(t) \tag{2.29}$$

The inner products $a_{j,k} = \langle f, \phi_{j,k} \rangle$ provide a discrete approximation at the scale 2^j. The inner product can also be written as a convolution integral

$$a_{j,k} \;=\; \int_{-\infty}^{+\infty} f(t)\, \frac{1}{\sqrt{2^j}}\, \phi\left(\frac{t - 2^j k}{2^j}\right) dt \;=\; f * \phi_{j,k} \tag{2.30}$$

where

$$\phi_j(t) \;=\; \sqrt{2^{-j}}\, \phi\left(2^{-j} t\right) \tag{2.31}$$

Typically, the energy of the $\Phi(\omega)$ is limited to $|\omega| < \omega_1$ hence, the energy of the function $\Phi_j(\omega)$ is concentrated in $|\omega| < \omega_1/2^j$. The discrete approximation $a_{j,k}$ is therefore a low-pass filtering of the signal f sampled at intervals 2^j.

The multiresolution analysis of the signal $f(t)$ is completely defined by the scaling function $\phi(t)$ that generates the orthogonal basis for the subspaces V_j. If $\phi(t)$ is an orthonormal basis in V_0, then $1/\sqrt{2}\phi(t/2)$ is an orthonormal basis in V_1. But any function that belongs in V_1 can be represented by a linear combination of the basis of V_2, hence

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) \;=\; \sum_{k=-\infty}^{+\infty} h_k\, \phi(t - k) \tag{2.32}$$

where

$$h_k \;=\; \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-k) \right\rangle \tag{2.33}$$

The discrete sequence h_k represents the projection of the function $1/\sqrt{2}\phi(t/2)$ onto the basis $\phi(t-k)$. In signal processing, h_k can be seen as the taps or coefficients of a discrete filter which is called a quadrature mirror filter.

2.3.3 Construction of Wavelet Bases Using Multiresolution Analysis

The projection of the signal $f(t)$ on the space V_j represents the approximation of this signal at the level j, whereas the wavelet transform represents the detail information that is added to create an approximation at level $(j-1)$. Hence, the wavelet transform of level j can also be considered as the projection of the signal $f(t)$ on a space W_j which is the orthogonal complement of V_j

$$V_{j-1} \;=\; V_j \oplus W_j \tag{2.34}$$

and the Equation 2.26 can be rewritten as

$$P_j^V(f) \;=\; P_{j+1}^V(f) \;+\; P_{j+1}^W(f) \tag{2.35}$$

where $P_j^V(f)$ and $P_j^W(f)$ are the projection of the signal f onto the V_j and W_j spaces, respectively

$$P_j^W(f) \;=\; \sum_{k=-\infty}^{+\infty} \langle f, \psi_{j,k}\rangle \; \psi_{j,k}(t) \tag{2.36}$$

As an example, we assume that we have $f(t)$ function illustrated in Figure 2.5, let us consider that this function can be made discrete by 2^6 sampled values using the Nyquist criterion. We can say that $f(t)$ is completely represented in the space V_6. In the next graph of Figure 2.5, we show the same function in subspace V_5 or when there are only 2^5 sampled values. In this space, we are sampling every $2T_0$, whereas we sampled every T_0 in V_6. This means that $f(t)$ in V_5 is contained in V_6. And this is true for all going to V_3.

$$V_3 \subset V_4 \subset V_5 \subset V_6 \tag{2.37}$$

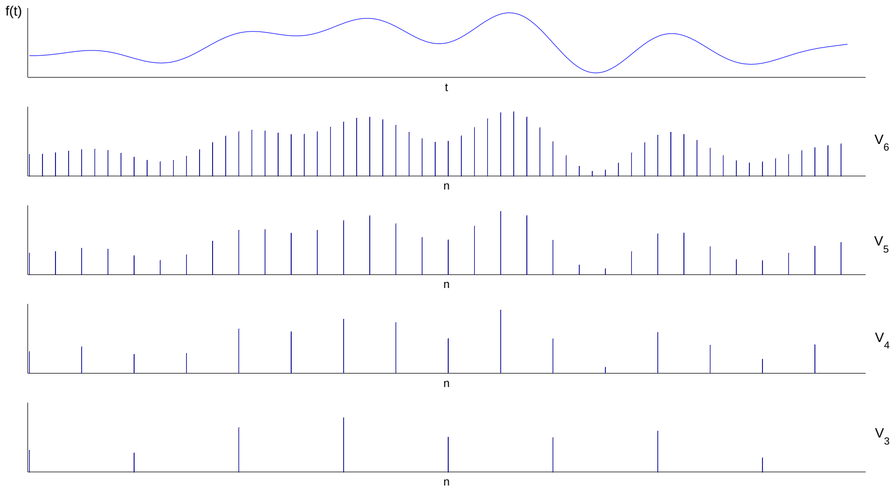

Figure 2.5: Signal $f(t)$ and its representations in V_6 to V_3

In the subspaces with index j lower than 6, it is obvious that the function is represented by a coarser approximation, that is, some detail information about the function is lost. We can define a set of detail subspaces, W_j, which will contain this lost information. For this particular example, we can write

$$\begin{aligned} V_6 &= (V_5 + W_5) \\ V_6 &= (V_4 + W_4) + W_5 \\ V_6 &= (V_3 + W_3) + W_4 + W_5 \end{aligned} \tag{2.38}$$

The meaning of detail subspace becomes clearer if we consider the spaces in the frequency domain as shown in the Figure 2.6. The top graph shows the V_6 subspace which contains the full bandwidth of the signal. We split the bandwidth into half, the lower half represents V_5 and the higher one represents W_5. We split

V_5 again into two halves, V_4 and W_4, and so on. This also shows that the subband decomposition which is another manifestation of multiresolution [19].

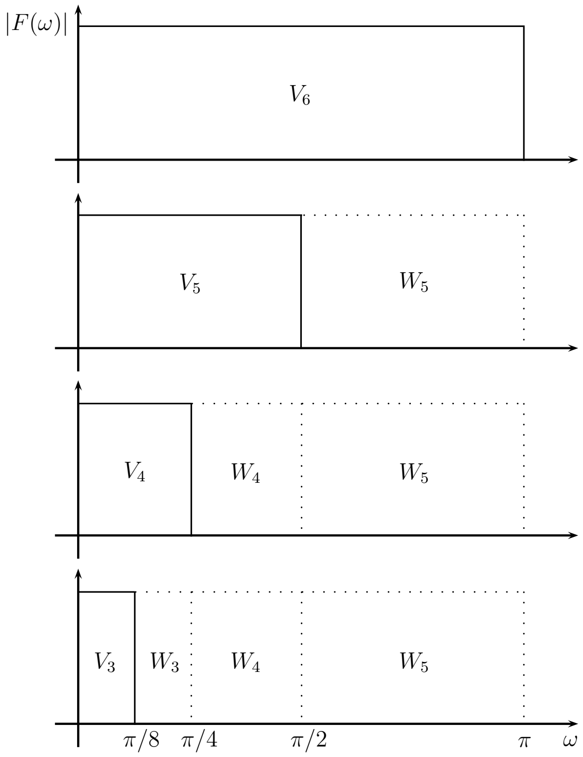

Figure 2.6: Representation of the spaces V_6 to V_3 and W_5 to W_3 in the frequency domain

Moreover, Mallat and Meyer proved that if $\phi(t)$ is the scaling function constructed from the filter h_k, then the function $\psi(t)$ has a Fourier transform $\Psi(\omega)$ given by

$$\Psi(\omega) \;=\; \frac{1}{2}G(\omega)\Phi(\omega) \tag{2.39}$$

with

$$G(\omega) \;=\; e^{-j\omega}H^*(\omega+\pi) \tag{2.40}$$

The function $\psi(t)$ can be used to construct an orthonormal basis for W_j given that

$$\psi_{j,k}(t) \;=\; \frac{1}{\sqrt{2^j}}\,\psi\left(\frac{t-2^j k}{2^j}\right) \tag{2.41}$$

This theorem is extremely important since it links the two functions, wavelet and scaling, by means of two filters, h_k and g_k, which are mirror conjugate of each other as defined in Equation 2.40. Calculating the inverse Fourier transform of this equation, we have

$$g_k \;=\; (-1)^{1-k}\,h_{1-k} \tag{2.42}$$

Mallat and Meyer also defined the necessary and sufficient conditions on $G(\omega)$ for designing orthogonal wavelets. The family $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of W_k if and only if

$$|G(\omega)|^2 \;+\; |G(\omega+\pi)|^2 \;=\; 2 \tag{2.43}$$

and

$$G(\omega)H^*(\omega) \;+\; G(\omega+\pi)H^*(\omega+\pi) \;=\; 0 \tag{2.44}$$

As a result, the filter coefficients g_k can be expressed as

$$g_k \;=\; \left\langle \frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right), \phi(t-k)\right\rangle \tag{2.45}$$

which means that $\psi(t/2)$ can be expressed as

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{k=-\infty}^{+\infty} g_k \, \phi(t-k) \tag{2.46}$$

2.4 Subband Decomposition

Let us consider the classic problem of designing two filters, one low-pass and the other one high-pass, such that we split the signal space exactly in half shown in Figure 2.7. The low-pass region extends from 0 to $\pi/2$ and the high-pass region extends from $\pi/2$ to π.

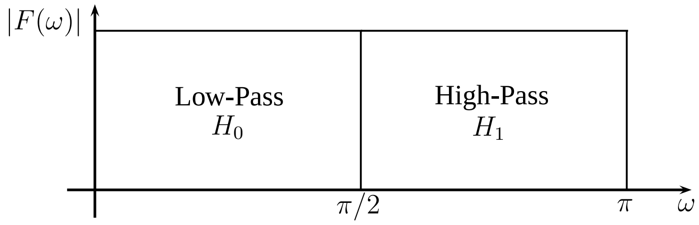

Figure 2.7: Splitting the signal space using ideal low-pass(H_0) and high-pass(H_1) filters

Once we design the filters to split a frequency band, we can split it again and again, forming subbands. Thus, in subband theory, we design a set of two prototype filters, such that repeated application of these filters divides the signal frequency band into equal parts. This is also referred as a filter bank. There are different ways to split up the frequency space, some examples are shown in Figure 2.8, 2.9 and Figure 2.10, 2.11.

In a dyadic wavelet transform, we design a mother wavelet from which we generate a set of daughter wavelets which forms a complete set. In other words, remembering the connection between wavelet transform and the filter bank, we see that for this case we have designed one filter, and then using the scaling property, we have constructed a filter bank. All the filters in the filter bank are identical in properties except for their scale. In a subband implementation, we

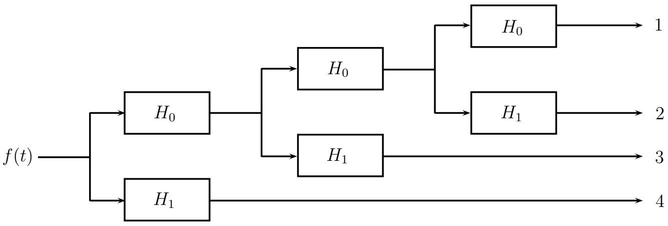

Figure 2.8: The filter bank for splitting low frequency bands consecutively, as in DWT

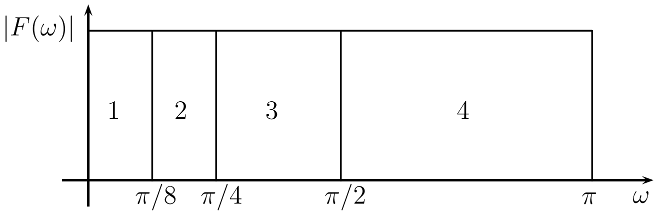

Figure 2.9: Frequency domain representation of the non-uniform subband structure(DWT structure)

initially design two filters which are then scaled to form a filter bank. For most applications, it is convenient to construct wavelet transforms using two filters. Their corresponding impulse responses are then used to obtain a scaling function (low-pass filter) and a wavelet function (high-pass filter). The difference at this point between subband and wavelet transform becomes very small as, in general, subbands refer to the discrete-time signal analysis application, whereas wavelet transforms refer to the continuous-time case.

Let us assume that we have the discrete time signal $f[n], n = 0, 1, 2, \dots,$ and we want to filter it by a low-pass filter H_0 ,of bandwidth $\pi/2$ (i.e. $0 \leq \omega \leq \pi/2$) and by a high-pass filter H_1, of bandwidth $\pi/2$ (i.e. $\pi/2 \leq \omega \leq \pi$). After filtering the signal $f[n]$, we have the low-pass filtered signal f_{LP} and the high-pass filtered signal f_{HP} of $f[n]$ with the same size of $f[n]$ and the bandwidth reduced by a factor of two. So, we can decimate these signals by a factor of two without any aliasing or distortion. After downsampling process, we get f'_{LP} and f'_{HP} having

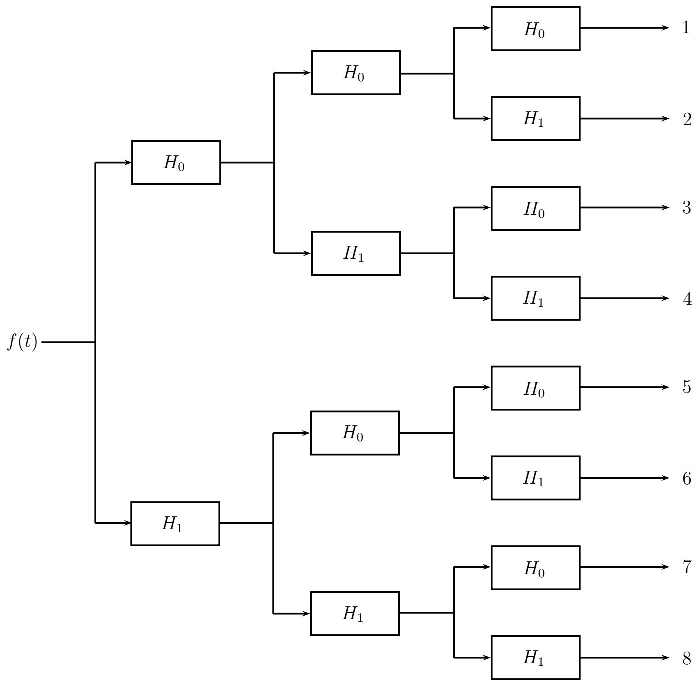

Figure 2.10: The filter bank for splitting both low and high frequency bands consecutively

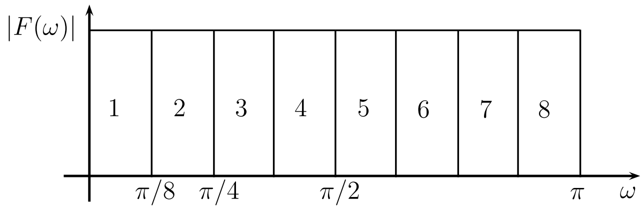

Figure 2.11: Frequency domain representation of the uniform subband structure

size by the half of the size of $f[n]$. These filtering and decimation processes of input $f[n]$ are called the analysis part. We have also the synthesis part, which is a mirror operation of the analysis part. We upsample the inputs, filter them by a synthesis filter pair, G_0, G_1, and obtain output signal $\tilde{f}[n]$. This is illustrated in Figure 2.12.

The problem is how to reconstruct the signal $f[n]$ back again without any aliasing or distortion meaning that $\tilde{f}[n] = f[n]$. As we mentioned before, using the

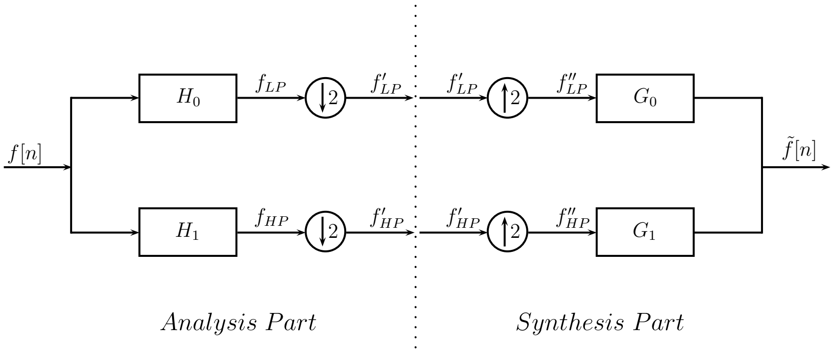

Figure 2.12: Analysis part and synthesis part of the subband filtering

relationship between low-pass filter and high-pass filter(scaling and wavelet filters,respectively) in Equation 2.42, 2.44, we can choose proper filters for Perfect Reconstruction (PR). If we consider using finite impulse response (FIR) filters extending to N, then the PR condition is satisfied provided that

$$\begin{aligned} h_1[n] &= (-1)^{1+n}\, h_0[N-1-n] \\ g_0[n] &= h_0[N-1-n] \\ g_1[n] &= (-1)^n\, h_0[n] \end{aligned} \tag{2.47}$$

Note that the above equations fully describe all the filters h_1, g_0 and g_1 once the low-pass filter h_0 defined.

2.5 Discrete Wavelet Transform in Image Processing

We mentioned about DWT and subbands in 1-D signals. But, what if we are dealing with 2-D signals such as images, we should look at how these processes are going to change. We assume that, we have $f[m,n]$, 2-D signal, since 2-D filtering process is separable, first we can apply filters and decimation along rows and then along columns. More specifically, assume that $f[m,n]$ has M rows and

N columns, we first apply low-pass and high-pass filters along the rows. Then, we decimate the rows and get low-pass filtered and high-pass filtered images with $M/2$ rows and N columns. We again apply low-pass and high-pass filters to these two images along the columns. After filtering, we decimate these 4 images along the columns and obtain 4 subbands with $M/2$ rows and $N/2$ columns. This is illustrated in Figure 2.13.

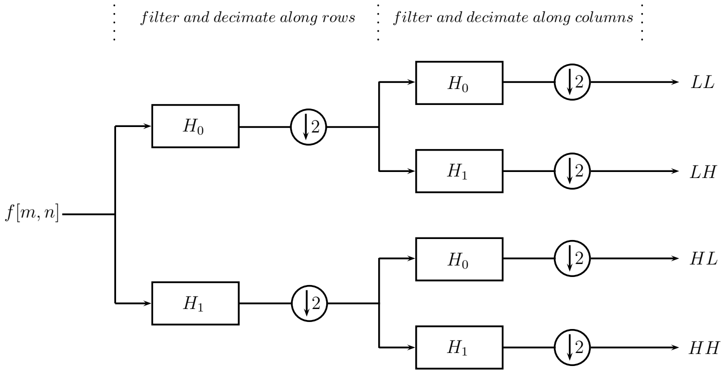
filter and decimate along rows
filter and decimate along columns
$f[m,n]$
H_0
H_1
H_0
H_1
H_0
H_1
2
LL
LH
HL
HH

Figure 2.13: 2-D separable analysis filter bank

If $f[m, n]$ is an image with size 512×512, then four subbands LL,LH,HL and HH have the size 256×256. LL subband is generally called the approximation or the low-pass subband, and the other three subbands, LH,HL and HH are called as the detail subbands and more specifically horizontal, vertical and diagonal subband, respectively. There is a famous Lena image in Figure 2.14, and its 1-level DWT and 2-level DWT images in Figure 2.15 and 2.16, as an example. In Figure 2.15, top-left image is the LL (approximation) subband, top-right is the LH (horizontal) subband and bottom-left and bottom-right are the HL (vertical) and HH (diagonal) subbands respectively.

There is one major issue when taking DWT of image, and it is well-known boundary problem. In the filtering stage, since the image has fixed size of rows and columns, we encounter problems when filtering boundary pixels. We should extend the image to be able to filter boundary pixels properly. However, we should be careful when extending image, because we don't want to corrupt or distort

Figure 2.14: Lena image

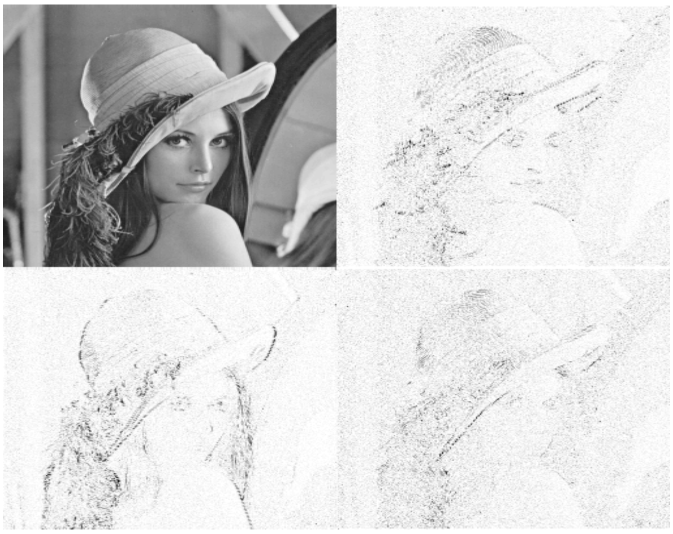

Figure 2.15: 1-level DWT of Lena image

image structures. There are two common solutions to this problem, i.e. periodic extension and symmetric extension [20]. In periodic extension, we copy entire image row or column to the end of that row or column. So, we get periodically repeated image rows or columns. In symmetric extension, we mirror the image row or column by the end of that row or column. So, we have a symmetrical structure relative to the end of rows or columns. These solutions are illustrated

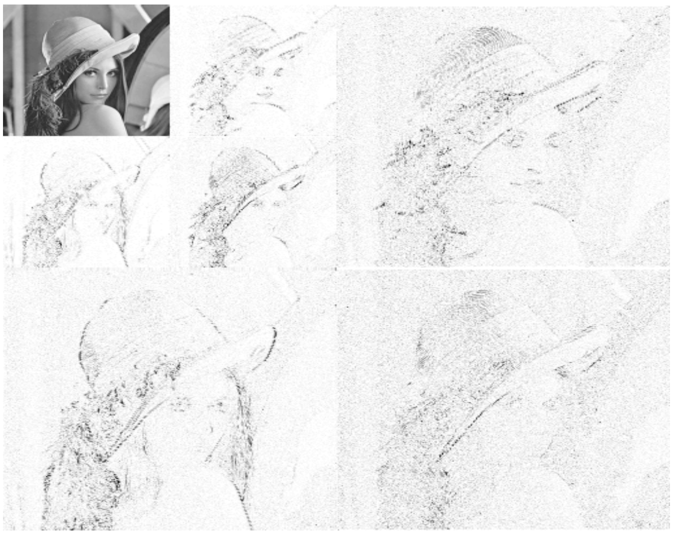

Figure 2.16: 2-level DWT of Lena image

in Figure 2.17.

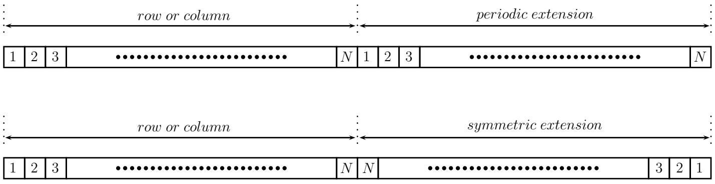

Figure 2.17: Periodic and symmetric extension of image data

2.6 Image Coding Algorithms using Wavelet Transform

Image coding algorithms using wavelet transform have very successful coding results in terms of subjective and objective quality. Wavelet transform compresses signal energy into a few coefficients and makes it possible to have sparse representation of signals. It is a very valuable property for coding purposes. Moreover, the multiresolution nature of the wavelet transform lets us to design flexible coding algorithms like embedded coding. Also, taking the discrete wavelet transform of the image does not have to be done by partitioning the image into blocks which avoids annoying blocking artifacts at the reconstructed image. Despite these great features of wavelet transform, superior results can not be obtained without designing efficient and intelligent algorithms which exploit this effectiveness.

One of the most significant coding algorithms using wavelet transform is Shapiro's EZW coder [5]. It is basically an algorithm that codes the significance maps of subbands hierarchically instead of coding wavelet coefficients themselves. As we mentioned before, wavelet transform makes the signal representation sparse. So, there are few large magnitude coefficients and numerous small magnitude coefficients that can be quantized to zero. EZW coder exploits this distribution of wavelet coefficients by using zerotrees. This data structure clusters the wavelet coefficients into two groups, i.e. significant and insignificant.

Moreover, EZW coder uses the inter-subband relationships of wavelet coefficients. It is called parent-children dependency of subbands, see Figure 2.18. A coefficient from coarser scale subband is called the parent and the coefficients having same spatial location at the finer scale are called children. All parent coefficients have four children except low pass subband coefficients which has three children. And needless to say that, finest scale coefficients have no children. Number of all descendants of a parent coefficient is growing exponentially depending on the number of stages of wavelet transform which makes zerotrees very efficient representation.

Furthermore, Shapiro proposed that, it is statistically reasonable to expect that the magnitude of a children should be smaller than its parent. So, this information is very useful in classification of coefficients. EZW coder uses zerotrees to represent sparse structure of wavelet coefficients. It basically relies on the correlation of parent and children's magnitude. If any parent's magnitude is lower than chosen threshold, and also its all descendants, i.e. its children and its children's children, and so on, have magnitude lower than threshold, we can use only one symbol to represent them all. This can go hierarchically to cover all coefficients. This data structure is actually very appropriate and efficient for coefficient

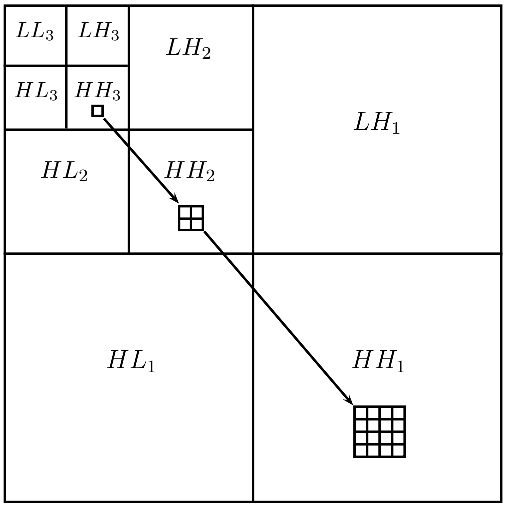

Figure 2.18: Parent-children dependency of subbands

distribution and the subband structure of wavelet transform.

In more detail, EZW coder classifies coefficients into two groups, significants and insignificants for some threshold T. If coefficient is significant according to the threshold, then it outputs POS for positive signed coefficient and NEG for negative signed. If coefficient is insignificant, there are 3 options: if it is a finest scale coefficient with no children, it outputs Z for a zero, if it has all descendants with insignificant value, it outputs ZTR for zerotree root, and at last if it has a descendant with significant value, it outputs IZ for isolated zero. This keeps going with hierarchical manner to process all coefficients. This is called dominant pass. There is an also subordinate pass which refines the output magnitude's precision. And this dominant and subordinate passes continue with threshold half of the former one, i.e. $T/2$, $T/4$ and $T/8$ ..., until coder achieves to a given bitrate or distortion value. Thus, this successive passes yield an embedded bitstream, meaning that whenever bitstream is ceased, decoder can reconstruct image with desired bitrate or distortion value without any artifacts. Embedded bitstream has coarse approximation of image and also successive refinement bits.

EZW coder has profound effect on wavelet based coding algorithms. Using data structures like zerotrees on representing wavelet coefficients leads to various different algorithms. One of the most successful algorithms that share similar philosophy with EZW is Said and Pearlman's set partitioning in hierarchical trees (SPIHT) algorithm [6]. It uses a similar zerotree structure to represent distribution of wavelet coefficients. SPIHT handles keeping track of significance states of sets by maintaining lists. There are three lists, list of insignificant sets (LIS), list of insignificant pixels (LIP) and list of significant pixels (LSP). And an important difference between EZW and SPIHT is SPIHT has 3 different sets of coefficients, i.e. children, all descendants and descendants which are not children. Also, it uses 2 different passes like EZW, sorting pass and refinement pass. SPIHT provides embedded coding and precise rate control and offers 2 different output options, binary uncoded bitstream and arithmetic coded bitstream. At the time when SPIHT was proposed, it has the best PSNR results among all image coding algorithms and majority of these algorithms has very high complexity.

Another important coding algorithm that uses zerotree structure is Xiong et al.'s space-frequency quantization (SFQ) algorithm [7]. It provides a solution to the joint optimization problem of using spatial quantization modes and basic scalar quantization. We mean zeroing out wavelet coefficients in the tree structure (zerotree quantization) by spatial quantization modes. And, scalar quantization stands for using a simple uniform scalar quantization for all significant coefficients. It jointly optimizes the decision of zeroing the tree and the selection of quantizer step size for significant coefficients in a rate-distortion sense. As a result, SFQ has better PSNR values than both EZW and SPIHT for several test images.

Moreover, there are algorithms that differs from zerotree coders, e.g. LoPresto et al.'s coder that uses Estimation-Quantization (EQ) framework [10]. It models the wavelet coefficients of each subband by using generalized Gaussian distribution with zero mean and slowly spatially-varying variances. Using this model, they proposed a framework with two steps, Estimation step and Quantization step. First, they estimate the optimal pixel variances using a Maximum Likelihood estimation. These variance estimates are obtained from causal quantized spatial neighbors using the slowly spatially-varying variances assumption. Then, they optimally quantize and entropy code estimated coefficients using a quantizer/entropy coder-variance lookup-table which is generated using rate-distortion optimization. EQ coder has brilliant PSNR results and it outperforms vast majority of the image coders.

Furthermore, another successful algorithm that differs from zerotree coders is Ates and Orchard's coding algorithm that uses spherical representation [9]. They share a similar philosophy with EQ by using local energy or local variance as a direct measure of information content. Ates and Orchard claim that local energy can be used to represent local image structure and also coding decisions such as how to spend bitrate can be done by using it. Spherical coder constructs a hierarchical tree by combining local energies for each subband starting from the coefficient level going to one total energy variable. Information of how to split one upper energy variable to two sub energy variables is conveyed by phase variables. So, phase variables represent the difference in local image structure between two regions. They proposed that coding the total energy variable and the phase variables of each subband by using the hierarchical energy tree are much more efficient than coding the wavelet coefficients themselves. Also, they use rate-distortion optimization to make decision of coding or zeroing out the branches of the hierarchical tree. The PSNR performance of spherical coder is better than EZW, very close to SPIHT and slightly worse than SFQ and EQ coders.

2.7 Image Interpolation Algorithms using Wavelet Transform

Wavelet based image interpolation algorithms mainly share the same philosophy, that is taking existing image as low-pass subband and estimating high-frequency subbands to reconstruct higher resolution image. They differ from each other by how they estimate these subbands. Various different algorithms are introduced to accomplish this objective, but merely a few of them has promising results. One of the promising wavelet based interpolation algorithms is Chang et al.'s interpolation algorithm that uses wavelet extrema extrapolation [12]. They basically extrapolate the high frequency subbands from existing image by modeling the propagation of wavelet transform extrema across scales. Chang et al. introduced 3 constraints that all estimated subbands should obey. From these constraints, they find how to upsample the existing image and how to extrapolate extremas across scales. Furthermore, they have 2 parameter equation to model the the propagation of extrema values, and these 2 parameters are estimated using the least square error(LSE) criterion. Wavelet based extrema extrapolation algorithm has better PSNR results than conventional algorithms bilinear and bicubic interpolation. Also, this algorithm can be further improved by using more realistic constraints and more complex equation to model the propagation of extremas to compete with the state of the art interpolation algorithms.

Another promising algorithm that uses wavelet transform for interpolation is Carey et al.'s regularity preserving interpolation scheme [15]. They introduce that making assumptions about continuity of natural images leads to oversmoothed edges and textures. Their algorithm has no continuity constraint on image and also they use Hölder regularity criteria to preserve edge regularity. In detail, they use undecimated wavelet transform to obtain high frequency subbands and they model the decay of edge structures along rows and columns separately across scales using Hölder regularity. Algorithm first locates the edge structure along rows or columns, then extrapolates its position and its magnitude decay to higher scale subband. Performance of this algorithm in PSNR metric is higher than bilinear and bicubic interpolation and also has better results than extrema extrapolation algorithm [21].

Moreover, there is an algorithm that uses hidden Markov trees to model the correlation of wavelet coefficients across scales for interpolation purposes proposed by Kinebuchi et al. [13]. They use pre-trained hidden Markov tree with mixture Gaussian distribution for state transitions for estimating 3 high frequency subbands from lower scale high frequency subbands. Algorithm first takes relatively similar images to train hidden Markov tree. Then, it tries to extract sign change probabilities from each subband independently. And, it iterates Expectation-Maximization (EM) algorithm for the pre-trained Markov tree and finds the states of all coefficients. Using Gaussian distribution with obtained parameters, it randomly generates a magnitude value for each coefficient. Then, using sign change probabilities, it changes the signs of coefficients when it is needed. Also, algorithm has post-processing step which is a filtering with Gaussian low pass filter and performing unsharp masking. The algorithm has PSNR values better than conventional algorithms and also extrema extrapolation algorithm. Furthermore, performance of this algorithm and regularity preserving image interpolation has nearly the same PSNR results [21].

Chapter 3

Wavelet Based Image Coding

Wavelet based image coders have better results in PSNR metric and also in visual comparison than other transform based counterparts, e.g. DCT based coders. Wavelet coders have various advantages over DCT coders, notably in their coding efficiency. As we mentioned before, wavelet transform compresses signal energy into a few coefficients and makes the signal representation sparse. So, we can design coding tools which classify the coefficients into significant and insignificant sets to exploit this skewed distribution of wavelet coefficients. These coding tools should code insignificant coefficients with little bitrate and the majority of bitrate should be spent on significant coefficients. Thus, efficient bitrate allocation is an essential step for successful image coding. But, before that, proper handling of classification of wavelet coefficients is crucial for efficient bit allocation. Hence, we should develop some kind of data structures to represent significant/insignificant coefficient sets which are suitable to wavelet coefficient distribution. In EZW coder, Shapiro developed zerotrees structure to represent these sets efficiently and also SPIHT coder uses hierarchical trees for handling these sets. After the success of EZW and SPIHT coders in image coding, it is obvious that the hierarchical description of wavelet coefficients is an efficient and appropriate way to represent wavelet coefficient structure.

In this thesis, we propose a wavelet based image coding algorithm which hierarchically classifies wavelet coefficients by using their quantization indices. First, we quantize wavelet coefficients using scalar quantizer. After that, we locally group wavelet coefficients of each subband individually based on their quantization indices to form index classes which hierarchically classify quantized data. At the first level of this hierarchical index tree, we combine neighboring wavelet coefficients to form index classes that represent some statistics of the quantization indices of these coefficients. Upper levels of tree is constructed using the same method, by pairing neighboring coefficients to have larger subsets. At each level of the tree, the class assignment of a given subset describes some local statistics of the quantization levels of corresponding coefficients.

This hierarchical quantization index tree provides an inherently adaptive bit allocation. Coding this index tree means coding class membership information of each level of the tree. The bitrate needed to code this class membership information of a group of coefficients is related with the average quantization levels of these coefficients. Hence, majority of bitrate will be spent to code regions of subbands which have more significant coefficients than others. This scheme allocates the bitrate by considering the local statistics of regions of subbands which leads accurate and efficient bit allocation. Nonetheless, we propose to use rate-distortion cost analysis to optimize the coding efficiency of the hierarchical quantization index tree for better bit allocation.

3.1 Hierarchical Quantization Index Classes

In this section, we analyze concepts like how to construct hierarchical index tree, how to obtain index classes and how to group coefficients in detail. Let us define $c(m,n)$ as wavelet coefficients of the subband W of size $2^j \times 2^j$ $(0 \leq m,n < 2^j)$ in k-level wavelet transform of an image. Then, the absolute value of each coefficient, $|c(m,n)|$ is scalar quantized and assigned to a non-negative quantization index, $i(m,n)$

$$i(m,n) \;=\; Q\left[|c(m,n)|\right] \tag{3.1}$$

also sign of each coefficient is kept in sign function $s(m,n)$

$$s(m,n) \;=\; sign\,(c(m,n)) \tag{3.2}$$

reconstructed coefficients $\tilde{c}(m,n)$ can be found by inverse quantization

$$\tilde{c}(m,n) \;=\; s(m,n)\,Q^{-1}\,[i(m,n)] \tag{3.3}$$

The quantization function $Q[.]$ could be selected as any scalar quantizer. Here, we choose to use dead-zone uniform quantizer. Because, we can easily adjust the set of coefficients to be quantized to zero by adjusting dead-zone interval.

$$Q[c] \;=\; \begin{cases} 0 & if \quad 0 \le c < T \\ \left\lfloor \frac{c-T}{q} + 1 \right\rfloor & if \quad T \le c \end{cases} \tag{3.4}$$

here T represents the dead-zone interval size and q is the quantization step size. The inverse quantization is defined as

$$Q^{-1}[i] \;=\; \begin{cases} 0 & if \quad i = 0 \\ iq + T - \frac{q}{2} & else \end{cases} \tag{3.5}$$

After defining $i(m,n)$, i.e. quantization indices, we combine them to form quantization index classes, C_r

$$C_r \;=\; \left\{(i_1,i_2)\,|f\,(i_1,i_2) = r,\,\forall\, i_1,\, i_2,\, r \,\epsilon\, \mathbb{Z}^+\right\} \tag{3.6}$$

where i_1,i_2 and r are non-negative integers and $f(.,.)$ represents class assignment function. The number of integer pairs in each class is defined as $N_r \;=\; |C_r|$. The purpose of the class assignment function is to combine similar index pairs into the same class. We have two different class assignment functions to classify coefficients, and also there can be various choices of assignment functions which defines the similarity in such a different way. Two different class assignment functions will be analyzed in next section.

Index classes, C_r, are used to construct a hierarchical description of the quantization indices of wavelet coefficients. Now, let us define how the hierarchical index tree are constructed, the first level of tree is formed by combining neighboring wavelet coefficients by using class assignment function

$$\begin{aligned} \Gamma_{0,0}(s,t) &= i(s,t) \\ \Gamma_{1,0}(s,t) &= f\left(\Gamma_{0,0}(2s,t), \Gamma_{0,0}(2s+1,t)\right) \end{aligned} \tag{3.7}$$

Upper levels of hierarchy are defined analogously,

$$\begin{aligned} \Gamma_{u,u}(s,t) &= f\left(\Gamma_{u,u-1}(s,2t), \Gamma_{u,u-1}(s,2t+1)\right) \\ \Gamma_{u+1,u}(s,t) &= f\left(\Gamma_{u,u}(2s,t), \Gamma_{u,u}(2s+1,t)\right) \end{aligned} \tag{3.8}$$

here, we pair horizontal and vertical neighbors in an alternating fashion from bottom to top of the index tree. In Figure 3.1, we can see an example of how upper levels of tree are combined in an alternating fashion. In $(J-2, J-2)$ level, we have 16 indices and they are combined in horizontal direction to form $(J-1, J-2)$ level's 8 indices. Then $(J-1, J-2)$ level indices are paired in vertical direction to form $(J-1, J-1)$ level's 4 indices and so on.

In our algorithm, we propose to code this hierarchical index tree, instead of coding wavelet coefficients themselves. In other words, we code the class assignment values $\Gamma_{u,v}(s,t)(0 \leq u \leq J, v \epsilon \{u, u-1\}, 0 \leq s,t < 2^{J-u}, 2^{J-v})$ starting from top going to bottom of the tree.

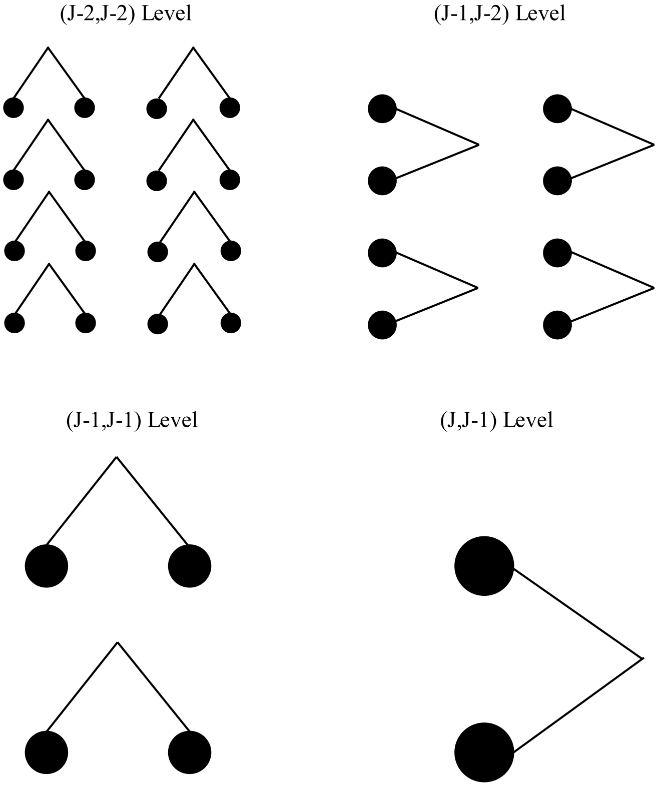

Figure 3.1: Top levels of the hierarchical index tree

3.2 Class Assignment Functions

We should design effective and efficient class assignment functions which group similar coefficients together. How to define this similarity results in different class assignment functions. But, all of the class assignment functions share the same philosophy, combine coefficients or groups of coefficients together based on their statistical characteristics. That is to say, coefficients or groups of coefficients having similar information content should be assigned to the same class. Furthermore, we designed two different class assignment functions, i.e. "circular" and "max" assignment functions which provide good coding results.

In circular assignment function, it assigns index pairs (i_1, i_2) to the closest circle with integer radius, see Figure 3.2

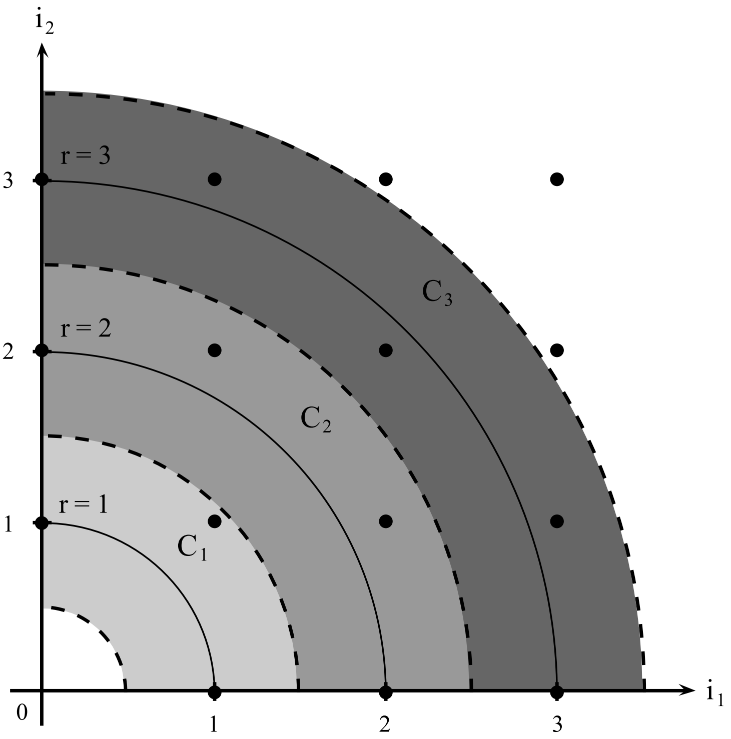

Figure 3.2: Circular assignment classes

$$f(i_1, i_2) \;=\; \left\lfloor \sqrt{i_1^2 + i_2^2} + 0.5 \right\rfloor \tag{3.9}$$

In Figure 3.2, light gray area shows the class C_1 and black dots in this area show the corresponding index pairs, i.e. $(0, 1)$, $(1, 0)$, $(1, 1)$ and $N_1 = 3$ for circular assignment function. And, the gray area represents the class C_2, black dots in this region is the integer index pairs, i.e. $(0, 2)$, $(2, 0)$, $(1, 2)$, $(2, 1)$ and $N_2 = 4$. Analogously, dark gray region shows the class C_3 and dots in this area correspond to index pairs, $(0, 3)$, $(3, 0)$, $(1, 3)$, $(3, 1)$, $(2, 2)$ and $N_3 = 5$.

Another reasonable choice for class assignment function is the maximum of two indices (i_1, i_2)

$$f(i_1, i_2) \;=\; max(i_1, i_2) \tag{3.10}$$

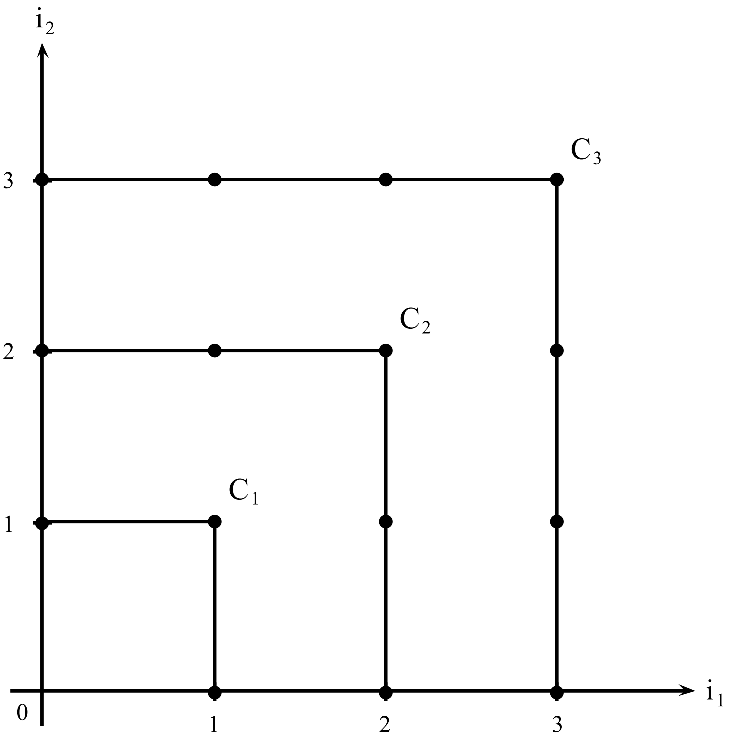

Figure 3.3: Max assignment classes

In Figure 3.3, 3 classes, C_1, C_2 and C_3 can be seen, and the corresponding index pairs are the black dots. For C_1, index pairs are the same as circular $(0,1)$, $(1,0)$, $(1,1)$ and $N_1 = 3$. But, for C_2 max and circular functions have different index pairs, max has $(0,2)$, $(2,0)$, $(1,2)$, $(2,1)$, $(2,2)$ and $N_2 = 5$. Also, C_3 has index pairs $(0,3)$, $(3,0)$, $(1,3)$, $(3,1)$, $(2,3)$, $(3,2)$, $(3,3)$ and $N_3 = 7$ for max assignment function. Number of index pairs for max assignment function can be calculated as $N_r = 2r+1$. It should be noted that $f(0,0) = 0$ and $N_0 = 1$ for both assignment functions.

3.3 Subband Coding by Quantization Index Hierarchy

Our coding algorithm is applied to each wavelet subband W of size $2^j \times 2^j$ independently. The coded data are the class assignment variables, $\Gamma_{u,v}(s,t)$ and sign bits of each significant wavelet coefficient. Encoding and also decoding are performed hierarchically, starting from the top of the tree, $\Gamma_{J,J}(0,0)$, going to the bottom, i.e. coefficient level, $\Gamma_{0,0}(m,n)$.

During encoding of level (u,u) of the hierarchical index tree, given $\Gamma_{u,u}(s,t)$ should be one of the $N_{\Gamma_{u,u}(s,t)}$ index pairs in class $C_{\Gamma_{u,u}(s,t)}$. Assuming all index pairs in class $C_{\Gamma_{u,u}(s,t)}$ are equiprobable, entropy coding this class assignment requires $\log_2(N_{\Gamma_{u,u}(s,t)})$ bits on average. This estimation of bitrate required to code class assignment variables is essential for rate-distortion cost analysis and will be used in Lagrangian cost calculations.

Moreover, it is necessary to handle coefficients that will be assigned to zero-class C_0 carefully to improve coding efficiency of our algorithm. Because, most of the bitrate gain comes from subtrees which are assigned to zero-class. Whenever a subtree of index hierarchy is assigned to C_0, then all subsequent class assignments, and so all wavelet coefficients belonging to this subtree are zeroed out and no additional bitrate is spent to code the remaining class indices of this subtree.

Here, there is one major problem, constructing the hierarchical index tree using original index values and coding this tree may not achieve optimal coding results. Because, one coefficient can be individually considered as significant and quantized to a non-zero level. But, when considering part of a subtree, assigning zero-class to this subtree can be better according to the rate-distortion optimization, so it may be more efficient to zero-quantize this coefficient although it is individually significant. Therefore, rate-distortion optimization is an essential step in our coding algorithm to decide either to code the original subtree or zero-quantize it for optimal performance.

We propose a simple rate-distortion cost analysis that is similar to the one described in [24] for optimization of index tree. Going from coefficient level to the top of tree, we compare the Lagrangian cost of coding a given subtree to zero-quantizing all coefficients belonging to this subtree. Former one is equal to the cost of coding the class assignment of the current subtree plus the minimum costs of the two children subtrees. We are going to explain this in detail in algorithm flow.

Algorithm is given as,

1. Quantize wavelet coefficients using a dead-zone uniform quantizer

$$\begin{aligned}\Gamma_{0,0}(m,n) &= Q\left[|c(m,n)|\right] \\ s(m,n) &= sign\left(c(m,n)\right) \\ \tilde{c}(m,n) &= s(m,n)\, Q^{-1}\left[\Gamma_{0,0}(m,n)\right]\end{aligned}$$

2. **Optimizing the index tree:** Compare the Lagrangian cost of coding and sending class variables of the wavelet coefficients to the cost of zero-quantizing them all. If the latter cost is smaller, assign that subtree to C_0. Assume that $L_{u,v}(s,t)$ represents the Lagrangian cost,

$$L_{0,0}(m,n) = \left(c(m,n) - \tilde{c}(m,n)\right)^2 + \lambda I(m,n)$$

   here $L_{0,0}(m,n)$ shows the Lagrangian costs of the coefficient level of the index tree, $I(m,n)$ represents the sign bit cost for coefficient $c(m,n)$, and $\left(c(m,n) - \tilde{c}(m,n)\right)^2$ represents the distortion for coefficient level. $I(m,n)$ can be found as

$$I(m,n) = \begin{cases} 0 & if\ \Gamma_{0,0}(m,n) = 0 \\ 1 & else \end{cases}$$

   Set $u = 1$ and while $u < J$ do

   - For $0 \leq s < 2^{(J-u)}$, $0 \leq t < 2^{(J-u+1)}$, split $\Gamma_{u,u-1}(s,t)$ to $\Gamma_{u-1,u-1}(2s,t)$ and $\Gamma_{u-1,u-1}(2s+1,t)$ by using Equation 3.8, and define corresponding Lagrangian costs as

$$L_{u,u-1}(s,t) = L_{u-1,u-1}(2s,t) + L_{u-1,u-1}(2s+1,t) + \lambda \log_2(N_{\Gamma_{u,u-1}(s,t)})$$

here total cost of class assignments for the current subtree is found by adding total cost of the two children subtrees to Lagrangian cost for coding class information, $\lambda \log_2(N_{\Gamma_{u,u-1}(s,t)})$. Then, we compare this cost to the total distortion caused by zero-quantization

$$L_{u,u-1}(s,t) \;>\; \sum_{m=2^u s}^{2^u(s+1)-1} \; \sum_{n=2^{u-1}t}^{2^{u-1}(t+1)-1} c(m,n)^2 \;\Rightarrow\; \Gamma_{u,u-1}(s,t) \;=\; 0$$

- Repeat the same procedure for $\Gamma_{u,u}(s,t)$ and $L_{u,u}(s,t)$ for $0 \leq s,t < 2^{J-u}$
- Set $u = u+1$ and repeat step 2

3. **Encoding/Decoding:** Start encoding/decoding from top of tree and going down to the coefficient level hierarchically. Code $\Gamma_{J,J}(0,0)$. Set $u = J$ and while $u > 0$ do,
   - Encode/Decode subtree assignments, $\Gamma_{u,u-1}(s,2t)$ and $\Gamma_{u,u-1}(s,2t+1)$ for $0 \leq s,t < 2^{J-u}$
   - Encode/Decode subtree assignments, $\Gamma_{u-1,u-1}(2s,t)$ and $\Gamma_{u-1,u-1}(2s+1,t)$ for $0 \leq s < 2^{J-u}$, $0 \leq t < 2^{J-u+1}$
   - Set $u = u-1$ and repeat step 3
4. Code the sign information if $\Gamma_{0,0}(m,n) > 0$. At the end of encoding/decoding, we reconstruct the decoded wavelet coefficients

$$\tilde{c}(m,n) \;=\; s(m,n)\, Q^{-1}\left[\Gamma_{0,0}(m,n)\right]$$

In the coding algorithm, optimal q and T, i.e. optimal quantization step size and dead-zone interval size are chosen for best rate-distortion performance for a given Lagrangian multiplier λ. For a given bitrate, optimal λ is found using the convex bisection algorithm mentioned in [25].

The class assignment variables, $\Gamma_{u,v}(s,t)$, are coded using arithmetic coder. The coding model of each class $C_{\Gamma_{u,v}(s,t)}$ is adapted based on number of index pairs, $N_{\Gamma_{u,v}(s,t)}$ and the level of tree, (u,v). Furthermore, the output bitrate of the arithmetic coder is slightly better than our equiprobable index pair bitrate estimate. So, accuracy of our bitrate estimate used in cost analysis is proved to be right.

While decoding the final index tree, when algorithm reaches to a subtree that is assigned to zero-class, all the coefficients belong to this subtree are set to zero and no bitrate is spent for coding the remaining class indices.

3.4 Simulation Results

In simulations, we use biorthogonal quadrature mirror filter pairs [26] in a 6-level dyadic decomposition to implement hierarchical index coder(HIC). Optimal step size q and dead-zone interval size T are chosen among the set $\{t : t = 0.1k,\ k = 50, 51, 52 \ldots 400\}$. Same quantization step size q and dead-zone interval size T are used for all subbands. Low pass subband is first DCT transformed(8×8) and quantized using optimal scalar quantizer for a given λ, then arithmetic coded. Standard test images Lena and Barbara of size 512×512 are used in simulations. Furthermore, simulations results are obtained for bitrates 1.00, 0.50 and 0.25 bits per pixel (bpp).

First, we compare the results of HIC coder using circular and max assignment functions for Lena and Barbara, see Table 3.1. PSNR results are nearly the same for max and circular functions for Lena and max function is slightly better for Barbara image.

We compare the PSNR results of HIC coder using circular assignment function to the state of the art coders in literature, i.e. SPIHT [6], SFQ [7], EBCOT [8], EZBC [27], SPHE [9] and EQ [10] in Table 3.2 and 3.3. EQ is implemented using 10/18 biorthogonal filters, so for fair comparison, EQ and HIC coder which is implemented using 10/18 biorthogonal filters are kept separately. HIC coder has

Table 3.1: PSNR results of HIC coder for circular and max assignment functions

HIC	PSNR(dB)			
	Lena		Barbara	
Rate(bpp)	Max	Circular	Max	Circular
1.00	40.69	40.70	37.08	37.05
0.50	37.45	37.45	32.17	32.14
0.25	34.37	34.37	28.35	28.34

the best results for Lena in 1.00 bpp and 0.25 bpp. And, in 0.50 bpp, it has slightly worse results than EZBC coder and better results than all of the rest. Furthermore, EQ has 0.1 dB better results over HIC coder for Lena. For Barbara, EBCOT has the best results. Note that, EBCOT uses complex models which can adapt local frequency variation of textured images, like Barbara. However, especially in lower bitrates, HIC coder has very successful results which can compete with complex algorithms.

Table 3.2: PSNR results of HIC coder and state of the art wavelet coders for Lena

Lena		PSNR(dB)		
Coder	Filter	0.25bpp	0.50bpp	1.00bpp
HIC	9/7	**34.37**	37.45	**40.70**
SPIHT[6]	9/7	34.11	37.21	40.46
SFQ[7]	9/7	34.33	37.36	40.52
EBCOT[8]	9/7	34.32	37.43	40.55
EZBC[27]	9/7	34.35	**37.47**	40.62
SPHE[9]	9/7	34.28	37.40	40.67
HIC	10/18	34.44	37.54	40.73
EQ[10]	10/18	**34.57**	**37.69**	**40.88**

For subjective evaluation, Figure 3.4 and 3.5 show the original Lena image and 1.00bpp coded version using circular assignment. Also, Figure 3.6 and 3.7 represent the visual difference between 0.50bpp coded images using HIC and EBCOT coders. Analogously, Figure 3.8 and 3.9 show the visual difference between two coders at 0.25bpp.

Table 3.3: PSNR results of HIC coder and state of the art wavelet coders for Barbara

Barbara		PSNR(dB)		
Coder	Filter	0.25bpp	0.50bpp	1.00bpp
HIC	9/7	28.34	32.14	37.05
SPIHT[6]	9/7	27.58	31.40	36.41
SFQ[7]	9/7	28.29	32.15	37.03
EBCOT[8]	9/7	**28.53**	**32.50**	**37.38**
EZBC[27]	9/7	28.25	32.15	37.28
SPHE[9]	9/7	28.22	32.06	37.00
HIC	10/18	**28.62**	32.50	37.49
EQ[10]	10/18	28.48	**32.87**	**37.65**

Figure 3.4: Original Lena image

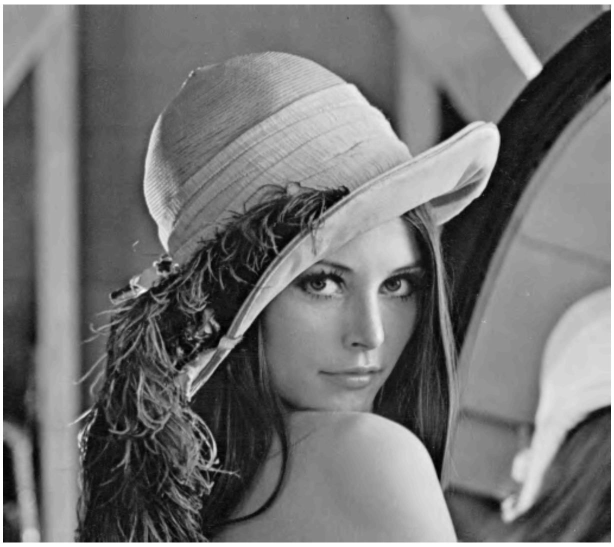

Figure 3.5: 1.00bpp coded Lena image using HIC coder PSNR:40.70

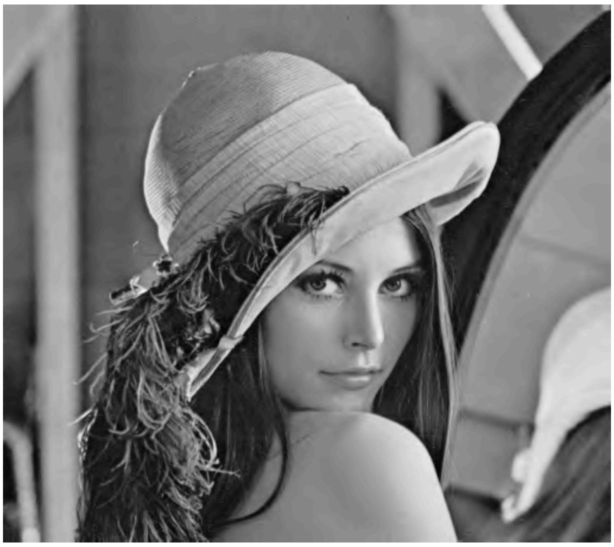

Figure 3.6: 0.50bpp coded Lena image using HIC coder PSNR:37.45

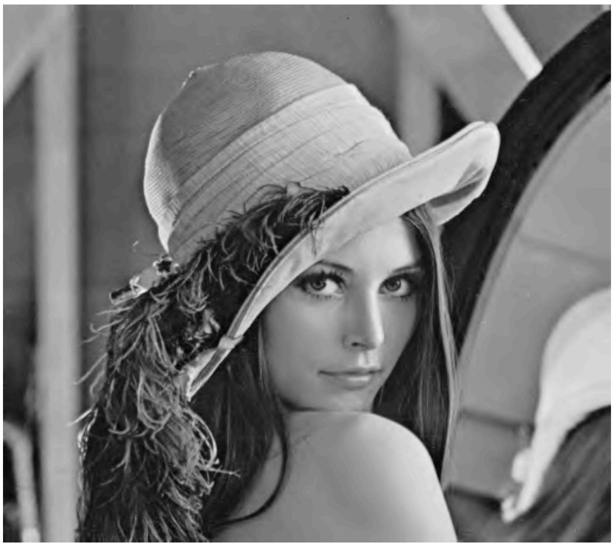

Figure 3.7: 0.50bpp coded Lena image using EBCOT coder PSNR:37.43

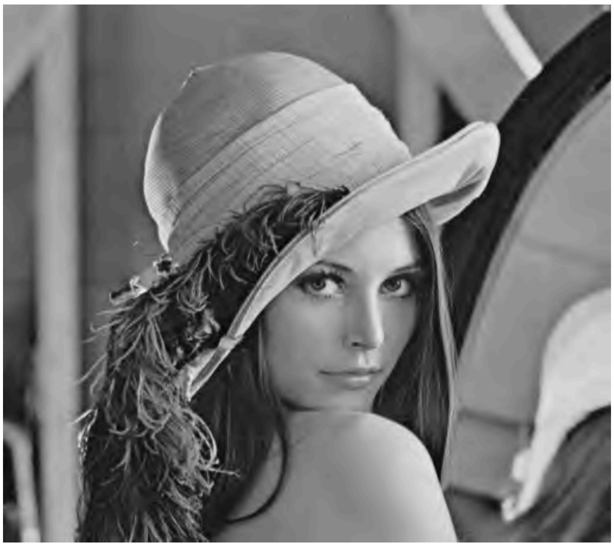

Figure 3.8: 0.25bpp coded Lena image using HIC coder PSNR:34.37

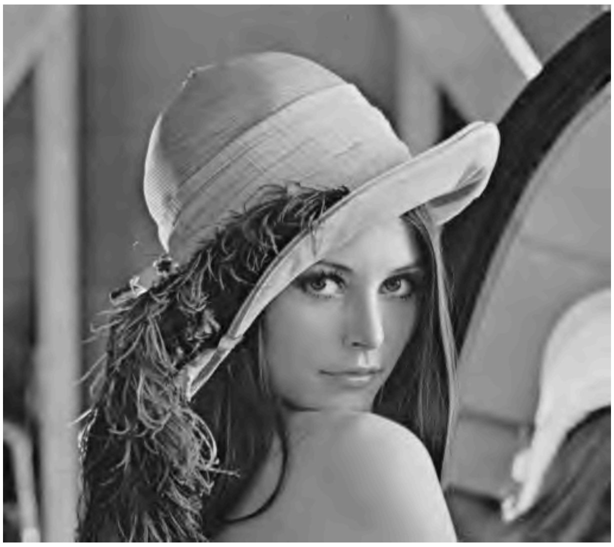

Figure 3.9: 0.25bpp coded Lena image using EBCOT coder PSNR:34.32

Furthermore, Figure 3.10 and 3.11 represent the original and 1.00bpp coded Barbara image, respectively. Figure 3.12 and 3.13 show the visual difference between 0.50bpp coded Barbara images using HIC and EBCOT coders, respectively. Similarly, Figure 3.14 and 3.15 represent the visual difference between two coders at 0.25bpp.

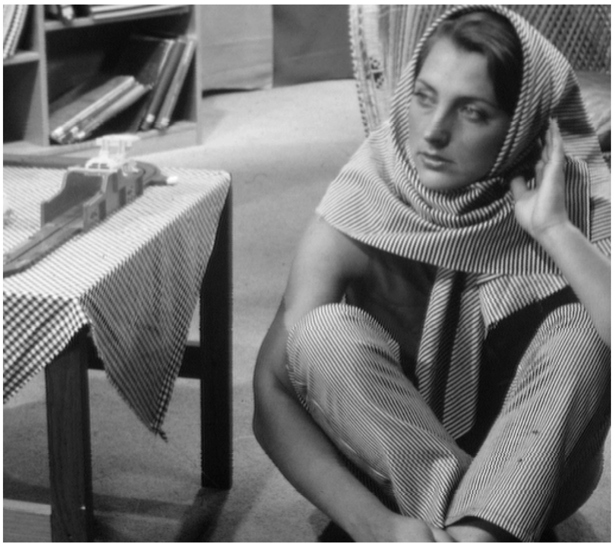

Figure 3.10: Original Barbara image

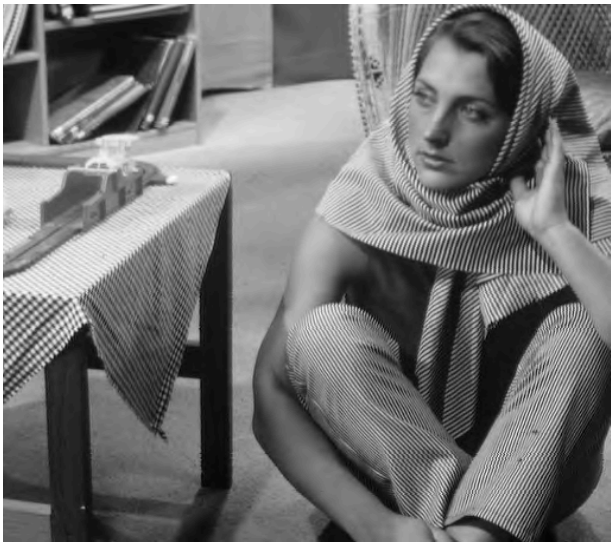

Figure 3.11: 1.00bpp coded Barbara image using HIC coder PSNR:37.05

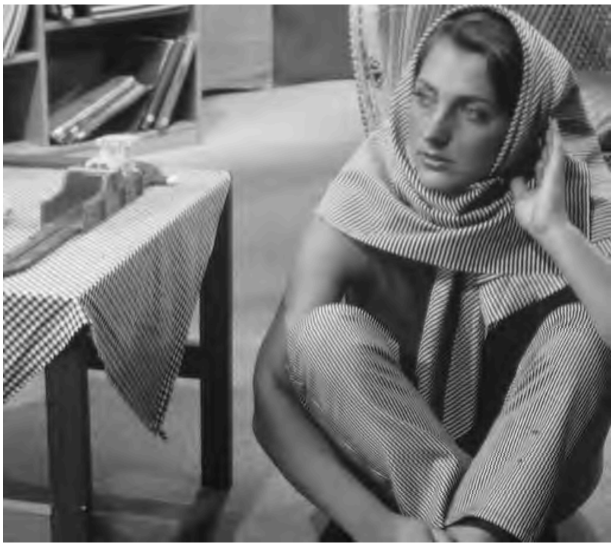

Figure 3.12: 0.50bpp coded Barbara image using HIC coder PSNR:32.14

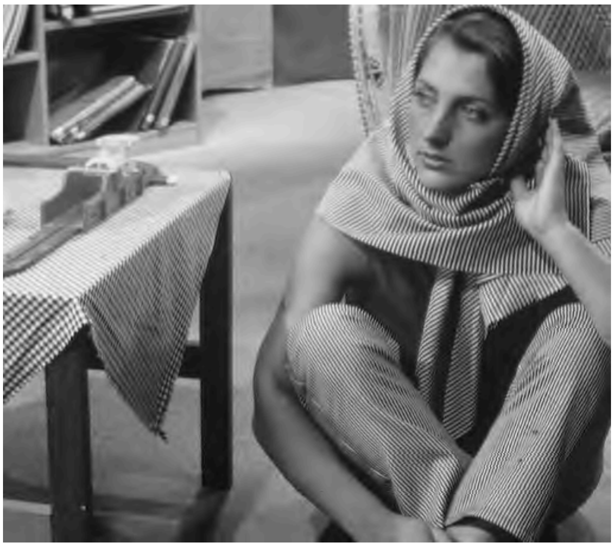

Figure 3.13: 0.50bpp coded Barbara image using EBCOT coder PSNR:32.50

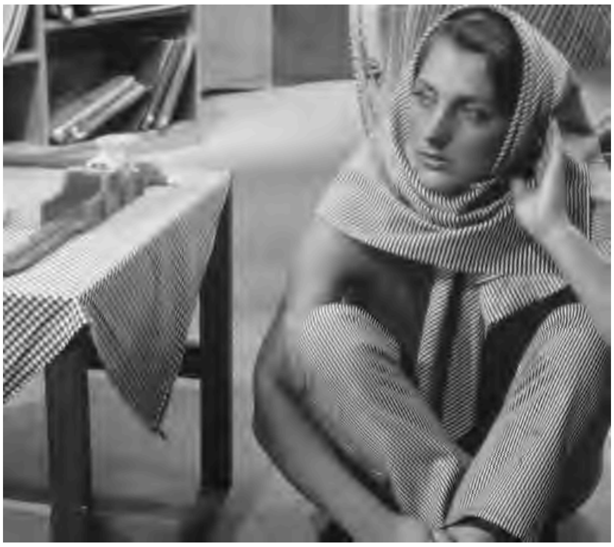

Figure 3.14: 0.25bpp coded Barbara image using HIC coder PSNR:28.34

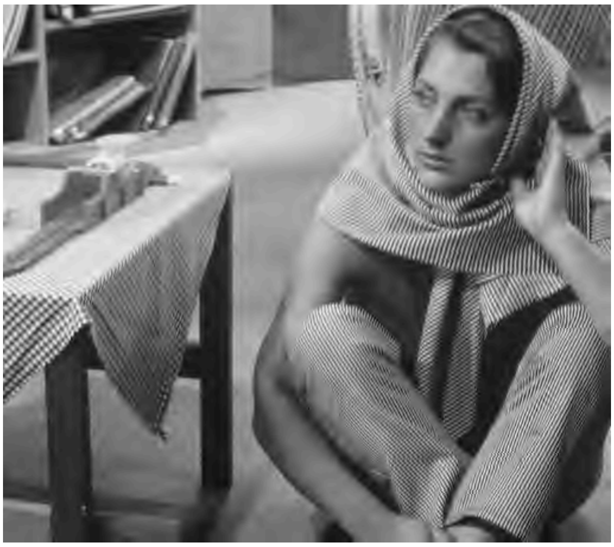

Figure 3.15: 0.25bpp coded Barbara image using EBCOT coder PSNR:28.53

Chapter 4

Wavelet Based Image Interpolation

Interpolation is a classical and grueling problem of several fields such as, applied mathematics, statistics, economics and signal processing. So many researchers have been studying this problem for a long time. However, there is no clear-cut solution for this phenomenon. As a matter of fact, the ultimate goal of the interpolation is the extraction of the underlying continuous data from existing undersampled discrete data. When stated that way, it seems like a problem that has no prospective solution, but that should also be pursued. Besides the ultimate goal, interpolation can be interpreted as the estimation of new data points in the range of known ones. So, we can consider it as a process that changes the sampling rate of the discrete data. This process especially has overriding importance in image processing since resolution enhancement (zooming), error concealment (image inpainting) and geometric transform of images are performed using it.

Image interpolation, here after used for a synonym for resolution enhancement of image, is a very challenging issue, because natural images have various regions with different statistics e.g. smooth regions,edges and textures which should be carefully handled when estimating new data points. Therefore, the performance of the image interpolation algorithm is related to how accurate it handles these different regions. Conventional algorithms, i.e. bilinear and bicubic interpolation, are spatial domain interpolators and they suffer from blurring and ringing artifacts. Because, they mostly smooth out edges and textured regions due to the continuity constraint they have on entire image. There are complementary algorithms to bilinear and bicubic like unsharp masking to boost high frequencies, but they slightly increase the performance of these algorithms. Moreover, there are wavelet based interpolation algorithms which are transform domain interpolators, handling edges and texture regions better than conventional algorithms, because they inherently have the high frequency information at low scales which can be used to preserve edges and textures at higher scales.

4.1 Wavelet Based Image Interpolation Algorithms

Most of the wavelet interpolation algorithms make the assumption that the image we have on hand is the low-pass subband of the higher resolution image, and interpolation algorithm is trying to do the estimation of the detail coefficients of this higher resolution image. This assumption is reasonable when we consider the acquisition of natural images, because image acquisition system makes image lose its high frequency contents by low-pass filtering and downsampling. Therefore, what we try to do by interpolation is to recover the lost high frequency information to have sharp, high quality image with higher resolution without any artifacts. Furthermore, wavelet interpolation algorithms differ from each other by how they estimate these detail coefficients. Most basic method is to fill all the detail subbands with zeros and use existing image as the low-pass subband, then take the inverse wavelet transform to reconstruct higher resolution image. This reconstructed image is a blurred version of existing image due to the fact that, we didn't estimate any high frequency information. This method is called "Zerofill", and used as a basis in performance comparisons.

For having successful interpolation results in a subjective and objective manner, we should have very good estimation of detail subbands. Because, substantial visual information is conveyed by edges and textured regions, i.e. high frequency contents, so we should handle these regions very carefully while performing interpolation. Fortunately, we know that, coefficients representing edges decays across scales in wavelet transform, therefore we can use lower scale subbands to estimate higher scale coefficients. There are algorithms using neural networks and hidden Markov trees [13] to model the relationship between higher scale and lower scale detail coefficients with extensive training. These algorithms suffer from the variation of regions in natural images that can't be modeled with general distribution and also the need of heavy training makes these algorithms impracticable. There is another algorithm which models the extremas between lower scale and higher scale [12]. It uses the evolution of the wavelet transforms extrema across scales to extrapolate extremas for estimating detail coefficients. It is useful for finding and extrapolating isolated extremas, so it has relatively good results on non-textured images. But, in textured images, it has mediocre results. Because, assumptions that they made on parameters are not true for images with complex edge structure, also location and magnitude information of extremas in textured images can not be found without complex modeling.

4.2 Our Resolution Enhancement Approach

As opposed to the mentioned wavelet interpolation algorithms, we propose here a different framework or scenario for interpolating images. First, we setup a basic encoder/decoder structure with a communication channel between these two nodes. This basic structure is illustrated in Figure 4.1. In this framework, we assume that we have the high resolution (HR) image available on the encoder side, however, instead of coding and sending HR image, we can code and send low resolution (LR) image with some extra side information which can be used by the decoder for interpolation, see Figure 4.2. This problem definition provides us a different perspective for solution of interpolation problem. Basically, we can extract some information that is very valuable for interpolation from high resolution image on the encoder side and we can send this extraction as side information with coded sequence. Then, we can use this extra information on the decoder side to improve the interpolation of the coded image. In this thesis, we work on how to extract these information from HR image for better interpolation results and we leave the coding of this extra information as future work.

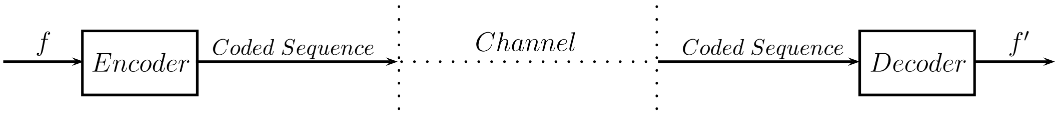

Figure 4.1: Basic encoder/decoder framework

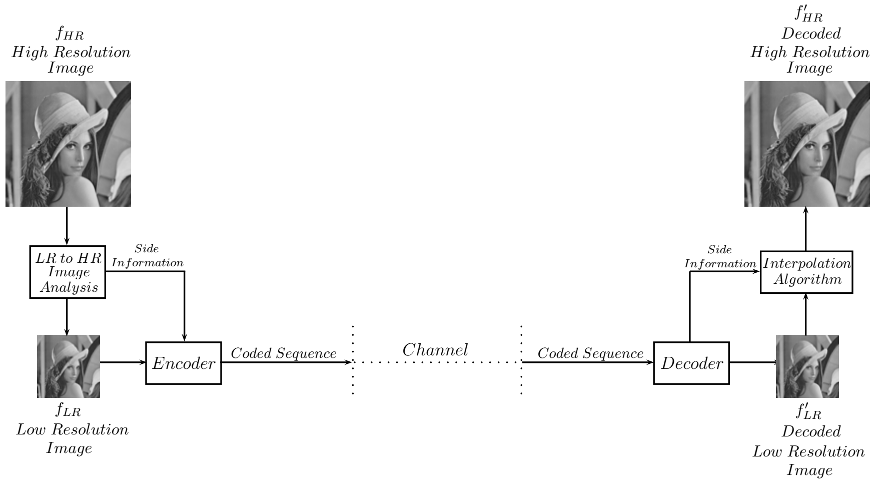

Figure 4.2: Our Encoder Assisted Resolution Enhancement Approach

Finally, we design linear filters between subbands of high resolution and low resolution image (in encoding/decoding scheme, of course we use encoded/decoded HR and LR images) for modeling the correlation between these two. Then, we send coefficients of these filters to the decoder part as side information. This whole process is our “Encoder Assisted Resolution Enhancement Approach”. In this approach, how to design linear filters is the main and the most significant issue. As we mentioned before, successful image interpolation requires proper handling of edges and textured regions which have major effect on visual content. So, we try to design our filters to handle high frequency context appropriately by two different algorithms, i.e. “Block Based Filter Design” and “Context Adaptive Filter Design”.

4.3 Block Based Filter Design Algorithm

In image interpolation, most significant and fundamental task to do is to recover the lost high frequency components to have sharp and high quality image. In our approach, we had our high resolution image in the encoder side which contains high frequency content. So, we should design linear filters to extract this information from high resolution image and use these filters to recover the high frequency content of low resolution image. Here, we propose to use block based filter design algorithm, which tends to catch local variations and try to model the neighbor pixel relations. It is useful for both preserving the edge structure and also smoothness of the image.

In more detail, we design “Linear Minimum Mean Square Error (LMMSE)” filters for blocks of subbands of low resolution and high resolution image. First, we use a standard interpolation algorithm to make an initial estimate of the HR image. This initial estimate does not have to provide an accurate description of the high frequency details, because after resizing low resolution image, we design high frequency compensation filters to recover the lost high frequency content. So, we should use any of the conventional algorithms, i.e. bilinear or bicubic interpolation. We will see that, type of initial interpolation has a minor effect on results in the simulations section. In our algorithm, we prefer to use bicubic interpolation as initial interpolation algorithm. After initial interpolation, we take wavelet transform of original high resolution image and initially interpolated high resolution image, to have detail subbands, i.e. horizontal (LH), vertical

(HL) and diagonal (HH) subbands. Estimating the accurate diagonal subband coefficients has less effect on the result than the other two subbands and gain/cost tradeoff makes us to skip the filter design process for diagonal subband. So, we fill whole diagonal subband with zeros in reconstruction. Block scheme of the algorithm can be seen in Figure 4.3.

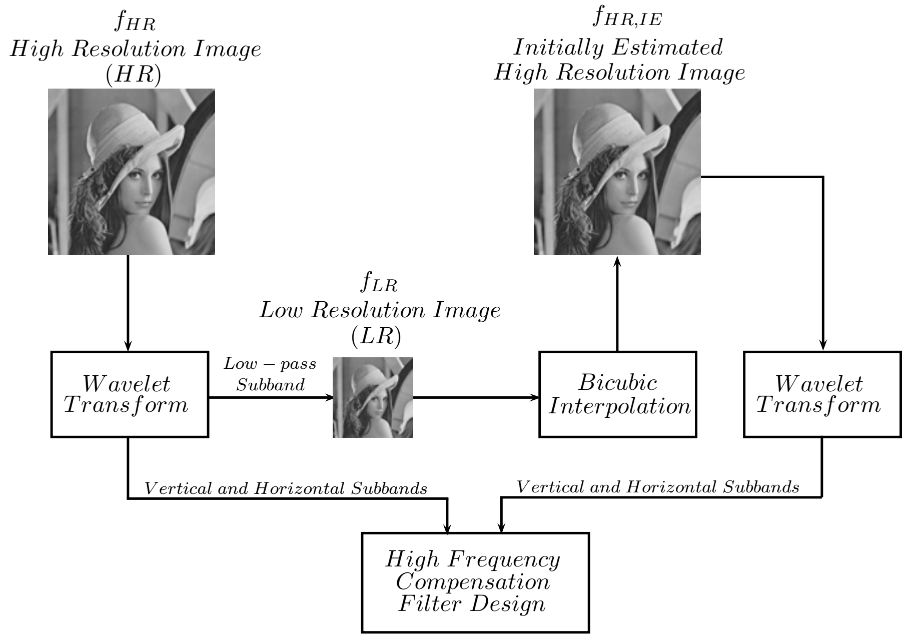

Figure 4.3: Block scheme of our filter design algorithm with initial interpolation

Now, we study how we design high frequency compensation filters in depth. After obtaining initially interpolated image and original high resolution image subbands, we have only horizontal and vertical subbands to process them. We take blocks from two corresponding subband and find a linear minimum mean square filter for that block. The problem here in matrix notation is to find the filter f^x

$$LH^x_{HR} = LH^x_{HR,IE}\, f^x \tag{4.1}$$

here, x is the block number which is related with block size, LH^x_{HR} is the x^{th} block of LH subband of original high resolution image and $LH^x_{HR,IE}$ is the x^{th} block of LH subband of initially interpolated high resolution image. We can obtain LMMSE filter f^x from Equation 4.1:

$$f^x \; = \; [(LH^x_{HR,IE})^T \, LH^x_{HR,IE}]^{-1} \, (LH^x_{HR,IE})^T \, LH^x_{HR} \tag{4.2}$$

We find filters f^x for all blocks of two corresponding subbands. Here, we derive equations for only LH subband, but these equations are all valid for HL subband.

$$f^x_{HL} \; = \; [(HL^x_{HR,IE})^T \, HL^x_{HR,IE}]^{-1} \, (HL^x_{HR,IE})^T \, HL^x_{HR} \tag{4.3}$$

There is only one difference between LH and HL subbands, that is the application direction of filters. In both subbands, we extract and apply the filters by the high pass direction of this subband, meaning that, we apply the filters by the column direction in LH subband, and by the row direction in HL subband.

In Equation 4.2 and 4.3, number of blocks, namely block size and filter size are chosen adaptively to catch local frequency variations better. Thus, block and neighborhood structure of original high resolution image are mainly protected in the filtering process, so we have very similar structures in the interpolated image. Using a localized block based prediction scheme has the advantage of adapting to the local signal structure, whether it consists of sharp changes or smooth transitions. Therefore, our interpolation algorithm preserves edges and textured regions and also keep the continuity of smooth areas as much as possible when estimating detail coefficients. This prevents the occurrence of undesirable artifacts on the reconstructed image.

In reconstruction, we assume that, we have only low resolution image and filter coefficients. First, we use initial interpolation on low resolution image and obtain initially interpolated image $f_{HR,IE}$, then take wavelet transform of $f_{HR,IE}$ to have $LH_{HR,IE}$ and $HL_{HR,IE}$. Using Equation 4.1, we get $LH_{HR'}$ and $HL_{HR'}$. After filling zeros to $HH_{HR'}$ subband, we take inverse wavelet transform using low resolution image,$LH_{HR'}$,$HL_{HR'}$ and $HH_{HR'}$ subbands to obtain interpolated high resolution image $f_{HR'}$.

In our block based algorithm, we use initial interpolation algorithm as a basis to design filters on it. Furthermore, we used one of the conventional algorithms, bicubic interpolation to resize our low resolution image. When bicubic interpolation doubles the size of LR image, it smoothes out the high frequency components, but it does not change the block and neighborhood structure. So, we design our filters to compensate the lost high frequency information. One question should be asked here, is there any better way to estimate initial subbands? There are alternative methods to generate these initial estimates. For instance, as in Figure 4.4, we can use undecimated wavelet transform which uses a specially designed set of analysis and synthesis filters that satisfy perfect reconstruction conditions [28] without the downsampling and upsampling steps of the critically sampled wavelet transform. Also, as in Figure 4.5, we can use wavelet transform without decimation which is implemented by using original wavelet transform filters and skipping the downsampling and upsampling steps.

But, simulation results show us that using bicubic interpolation as a basis is the best choice to estimate initial subbands. Because, undecimated wavelet transform or wavelet transform without decimation cause phase shift and they can change the block structure of subbands, so it does not match with the block structure of the subbands of original high resolution image. Thus, when using block based algorithm, basic initial interpolation algorithms have better performance in subjective and objective quality.

4.4 Context Adaptive Filter Design Algorithm

Besides block based algorithm, we can catch the local high frequency content in subbands by context adaptive structures. For example, magnitudes of the

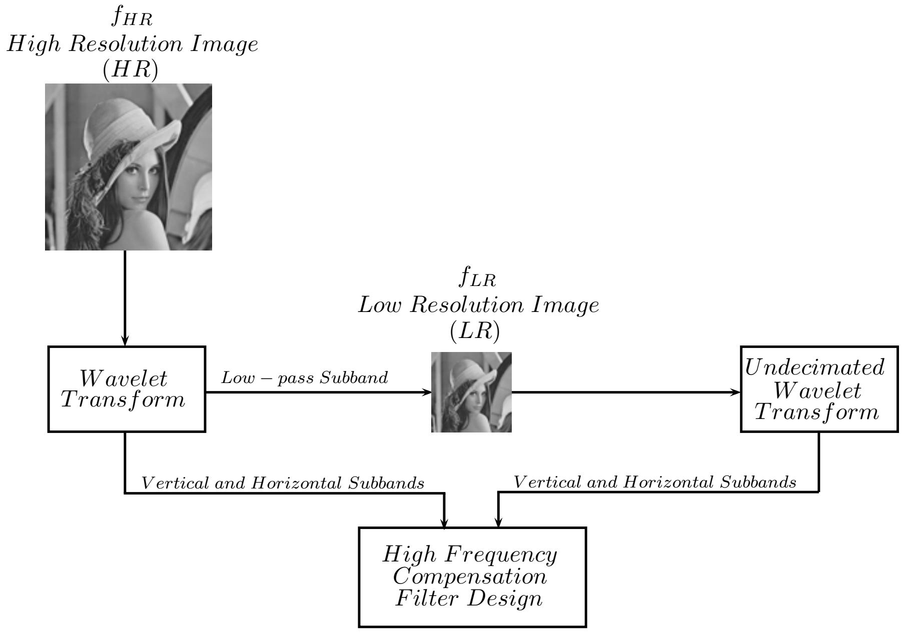
f_{HR}
High Resolution Image
(HR)
f_{LR}
Low Resolution Image
(LR)
Wavelet Transform
Low – pass Subband
Undecimated Wavelet Transform
Vertical and Horizontal Subbands
Vertical and Horizontal Subbands
High Frequency Compensation Filter Design

Figure 4.4: Block scheme of our filter design algorithm with undecimated wavelet transform

wavelet coefficients give an important information about the content of regions. So, we can group coefficients with similar magnitudes in subbands. This grouped coefficients and their relationships with their neighbors give us a good context to design adaptive filters. These context adaptive filters can model the correlation between LR and HR images.

In more detail, as in block based algorithm, we use the same scheme, Figure 4.3, up to filter design. We use initial interpolation algorithm to have $f_{HR,IE}$. Then, take wavelet transform of it to have $LH_{HR,IE}$ and $HL_{HR,IE}$. After that, we take wavelet transform of original high resolution image to have LH_{HR} and HL_{HR}. For grouping the coefficients with similar magnitudes in subbands, we should sort the coefficients of subbands by their magnitudes. So we sort the initial subbands,

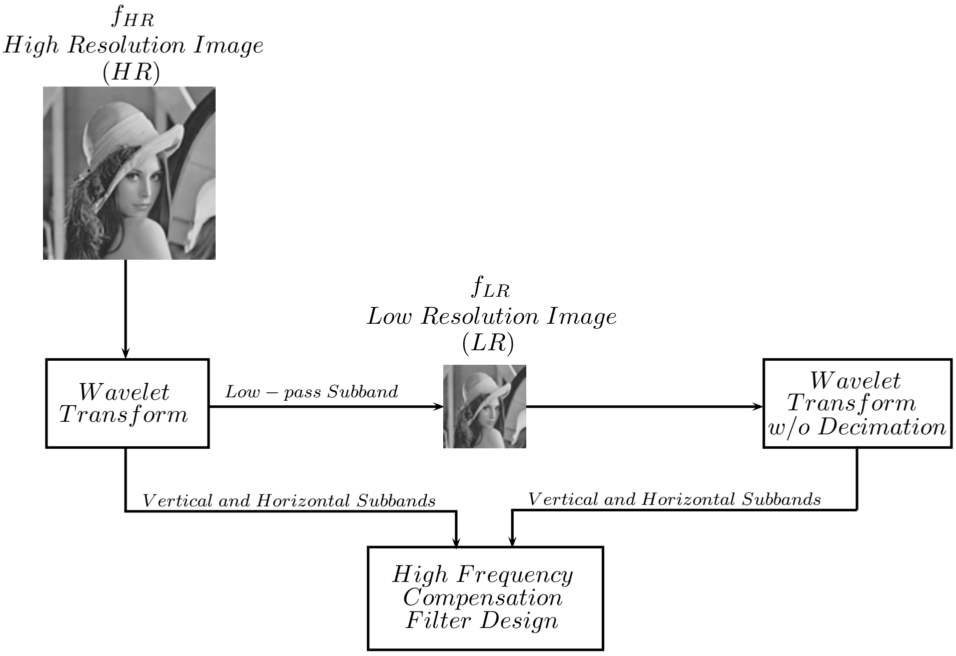
f_{HR}
High Resolution Image
(HR)
f_{LR}
Low Resolution Image
(LR)
Wavelet Transform
Low − pass Subband
Wavelet Transform w/o Decimation
Vertical and Horizontal Subbands
Vertical and Horizontal Subbands
High Frequency Compensation Filter Design

Figure 4.5: Block scheme of our filter design algorithm with wavelet transform without decimation

$$\begin{aligned} LH^s_{HR,IE} &= sort(LH_{HR,IE}) \\ \Upsilon^{LH}_{HR,IE}(j,k) &= (m) \end{aligned} \tag{4.4}$$

and

$$\begin{aligned} HL^s_{HR,IE} &= sort(HL_{HR,IE}) \\ \Upsilon^{HL}_{HR,IE}(j,k) &= (m) \end{aligned} \tag{4.5}$$

here, $LH^s_{HR,IE}(m)$ and $HL^s_{HR,IE}(m)$ stand for sorted versions of $LH_{HR,IE}(j,k)$ and $HL_{HR,IE}(j,k)$ respectively. m is the new location variable for sorted versions of subbands where $m = 1$ represents the highest magnitude coefficient. $\Upsilon^{LH}_{HR,IE}$ represents the index mapping between $LH_{HR,IE}$ and its sorted version $LH^s_{HR,IE}$; that is, $LH^s_{HR,IE}(m) = LH_{HR,IE}(j,k)$. Also $\Upsilon^{HL}_{HR,IE}$ is the index mapping between $HL_{HR,IE}$ and its sorted version $HL^s_{HR,IE}$.

We want to construct $LH^x_{HR,IE}$ matrix taking x^{th} block of elements by size $windowsize$ from $LH^s_{HR,IE}$. So, we get elements having m value from $(((x-1).windowsize)+1)$ to $(x.windowsize)$. Then, we find the corresponding (j,k) pairs for these elements by using $\Upsilon^{LH}_{HR,IE}$. Assume that, $m = ((x-1).windowsize)+1$ corresponds to (j_1,k_1) pair and $m = ((x-1).windowsize)+2$ corresponds to (j_2,k_2) pair and so on. $LH^x_{HR,IE}$ matrix is constructed by

$$LH^x_{HR,IE} = \begin{bmatrix} \cdots & LH_{HR,IE}(j_1-1,k_1) & LH_{HR,IE}(j_1,k_1) & LH_{HR,IE}(j_1+1,k_1) & \cdots \\ \cdots & LH_{HR,IE}(j_2-1,k_2) & LH_{HR,IE}(j_2,k_2) & LH_{HR,IE}(j_2+1,k_2) & \cdots \\ \cdots & LH_{HR,IE}(j_3-1,k_3) & LH_{HR,IE}(j_3,k_3) & LH_{HR,IE}(j_3+1,k_3) & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.6}$$

and analogously, we have the corresponding (j,k) pairs for $HL_{HR,IE}$ by using $\Upsilon^{HL}_{HR,IE}$, and construct $HL^x_{HR,IE}$ by,

$$HL^x_{HR,IE} = \begin{bmatrix} \cdots & HL_{HR,IE}(j_1,k_1-1) & HL_{HR,IE}(j_1,k_1) & HL_{HR,IE}(j_1,k_1+1) & \cdots \\ \cdots & HL_{HR,IE}(j_2,k_2-1) & HL_{HR,IE}(j_2,k_2) & HL_{HR,IE}(j_2,k_2+1) & \cdots \\ \cdots & HL_{HR,IE}(j_3,k_3-1) & HL_{HR,IE}(j_3,k_3) & HL_{HR,IE}(j_3,k_3+1) & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \tag{4.7}$$

here, $LH^x_{HR,IE}$ and $HL^x_{HR,IE}$ matrices have the size $(windowsize \times filtersize)$. Therefore, we take $filtersize - 1$ vertical and horizontal neighbors for $LH^x_{HR,IE}$ and $HL^x_{HR,IE}$, respectively.

For constructing LH^x_{HR}, again we use $\Upsilon^{LH}_{HR,IE}$ to find corresponding (j,k) pairs (j_1,k_1),(j_2,k_2) and so on. Then, we construct LH^x_{HR} by

$$LH^x_{HR} = \begin{bmatrix} LH_{HR}(j_1,k_1) \\ LH_{HR}(j_2,k_2) \\ LH_{HR}(j_3,k_3) \\ \vdots \end{bmatrix} \tag{4.8}$$

and HL^x_{HR} by

$$HL^x_{HR} = \begin{bmatrix} HL_{HR}(j_1,k_1) \\ HL_{HR}(j_2,k_2) \\ HL_{HR}(j_3,k_3) \\ \vdots \end{bmatrix} \tag{4.9}$$

here, LH^x_{HR} and HL^x_{HR} matrices have the size $(windowsize \times 1)$. After obtaining these matrices, we find LMMSE filters f^x for LH subband by using an equation below as in Equation 4.2

$$f^x = [(LH^x_{HR,IE})^T \, LH^x_{HR,IE}]^{-1} \, (LH^x_{HR,IE})^T \, LH^x_{HR} \tag{4.10}$$

here x is the window number which is related with window size, f^x is our LMMSE filter for x^{th} window. We get LMMSE filters for HL subband by

$$f^x_{HL} = [(HL^x_{HR,IE})^T \, HL^x_{HR,IE}]^{-1} \, (HL^x_{HR,IE})^T \, HL^x_{HR} \tag{4.11}$$

In reconstruction, we assume that, we have only low resolution image and filter coefficients. First, we use initial interpolation on low resolution image and obtain initially interpolated image $f_{HR,IE}$, then take wavelet transform of $f_{HR,IE}$ to have $LH_{HR,IE}$ and $HL_{HR,IE}$. We sort both subbands to have $LH^s_{HR,IE}$ and $HL^s_{HR,IE}$ and get $\Upsilon^{LH}_{HR,IE}$ and $\Upsilon^{HL}_{HR,IE}$. Then, using Equation 4.6 and 4.7, we obtain $LH^x_{HR,IE}$ and $HL^x_{HR,IE}$. From here, LH^x_{HR} and HL^x_{HR} can be found by

$$LH^x_{HR} = LH^x_{HR,IE}\, f^x \tag{4.12}$$

and

$$HL^x_{HR} = HL^x_{HR,IE}\, f^x_{HL} \tag{4.13}$$

here, we can write LH^x_{HR} and HL^x_{HR} values to their original locations, i.e. (j,k) by using mapping functions $\Upsilon^{LH}_{HR,IE}$ and $\Upsilon^{HL}_{HR,IE}$. So, we obtain $LH_{HR'}$ and $HL_{HR'}$ subbands. After that, we fill zeros to $HH_{HR'}$ and take inverse wavelet transform using low resolution image, $LH_{HR'}$,$HL_{HR'}$ and $HH_{HR'}$ to reconstruct high resolution image $f_{HR'}$.

In our context adaptive filter design algorithm, we use initial interpolation algorithm for doubling the size of low resolution image. As we mentioned, instead of initial interpolation, we can use undecimated wavelet transform and wavelet transform without decimation to obtain initial subbands. In context adaptive case, these two initial transforms do not have the same disadvantages in contrast to block based structure. Because, in this algorithm, we do not use block based structure to model the relationships of coefficients, we exploit the correlation between coefficients with similar magnitude using a context adaptive framework. So, using these two transforms provide a parallel context structure with original high resolution image which yields better interpolation performance.

4.5 Simulation Results

In simulations, we use 9/7 biorthogonal quadrature mirror filter pairs [26] for wavelet transform and wavelet transform without decimation. For undecimated wavelet transform, we used H, G, K and L filters that correspond to the quadratic spline wavelet by the size of 4,2,6 and 7, respectively, mentioned in [28]. Standard test images Lena and Barbara of size 512×512 are used for simulations. And, all results of these images are for magnifying by $2\times$, i.e. 256×256 to 512×512. Furthermore, block sizes are adaptively chosen from the set, $\{8 \times 8, 16 \times 16, 32 \times 32\}$. Analogously, window sizes are chosen among the set, $\{64, 256, 1024\}$. Filters are designed by the size of 3 *tap*, 5 *tap* and 7 *tap*.

PSNR results of our block based filter design algorithm are tabulated in Table 4.1 for Lena and Table 4.2 for Barbara. In Table 4.1 and 4.2, Bicubic means that, we use bicubic interpolation as initial interpolation algorithm. Similarly, w/o Decimation and Undecimated mean that, we use wavelet transform without decimation and undecimated wavelet transform to obtain initial subbands. And also, Table 4.3 shows the results of Zerofill algorithm for Lena and Barbara.

Table 4.1: PSNR results of block based filter design algorithm for Lena

Lena				
Block Size	Filter Size	Bicubic	w/o Decimation	Undecimated
8×8	3	36.08	35.92	35.81
	5	36.27	36.16	36.10
	7	**36.43**	36.38	36.32
16×16	3	35.76	35.63	35.54
	5	35.83	35.75	35.71
	7	**35.87**	35.82	35.81
32×32	3	35.65	35.54	35.47
	5	35.68	35.61	35.59
	7	**35.69**	35.66	35.66

From these results, we can say that our block based filter design approach works better in textured images like Barbara. It has 2.3dB-4dB better PSNR results than Zerofill algorithm. It justifies that, our algorithm works great on capturing

Table 4.2: PSNR results of block based filter design algorithm for Barbara

Barbara				
Block Size	Filter Size	Bicubic	w/o Decimation	Undecimated
8×8	3	29.06	28.48	28.47
	5	29.57	28.93	29.43
	7	**29.90**	29.18	29.83
16×16	3	28.41	27.90	27.67
	5	28.68	28.15	28.51
	7	**28.84**	28.27	28.73
32×32	3	27.90	27.38	27.16
	5	28.07	27.61	27.87
	7	**28.17**	27.71	28.04

Table 4.3: PSNR results of Zerofill algorithm for Lena and Barbara

Zerofill	
Lena	Barbara
35.26	25.85

local high frequency variations. Especially, our algorithm avoids aliasing in Barbara's scarf and trousers unlike Zerofill, see Figure 4.9, 4.10. So, we can say that, block based structure preserves edges and textured regions well. In a relatively smoother image, Lena, our algorithm has 0.4dB-1.2dB better PSNR values than Zerofill. In our algorithm, by increasing filter size and/or decreasing block size, performance of the algorithm is getting better as expected.

Furthermore, we should compare our initial interpolation algorithms. Using bicubic interpolation as initial interpolation is better than the other two methods. As we mentioned, using wavelet transform without decimation and undecimated wavelet transform cause a phase shift, and they change the neighborhood and block structure. But, it should be noted that, with increasing filter size, these two transforms, especially undecimated wavelet transform, have better results than bicubic algorithm in terms of improvement in performance. It is because, a longer neighborhood could capture better the underlying high frequency structure and therefore eliminate the effects of phase mismatch.

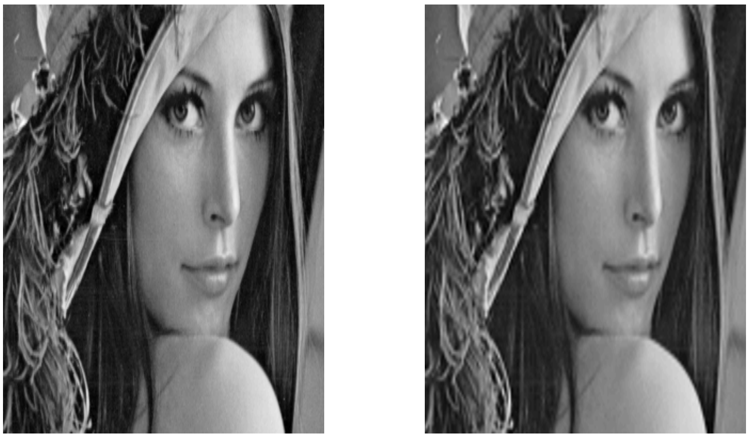

Figure 4.6: Original Lena image(Left), Zerofill reconstruction of Lena image(Right) PSNR:35.26

Figure 4.6 shows subregions from original Lena image and Zerofill reconstruction result. Figure 4.7 presents the difference between the best case(block size 8 × 8 and filter size 7) and the worst case(block size 32 × 32 and filter size 3) of our algorithm using bicubic algorithm as initial interpolation for subregions from Lena image. Next figure, Figure 4.8 shows the performances of wavelet transform without decimation and undecimated wavelet transform as initial interpolation with block size 8×8 and filter size 7 for subregions from Lena image. Figure 4.9, 4.10 and 4.11 presents exact same algorithm results for Barbara image.

As for context adaptive filter design algorithm, PSNR results are tabulated in Table 4.4 for Lena image and in Table 4.5 for Barbara image.

PSNR results of context adaptive filter design algorithm show us that, our context is appropriate for smooth images and it is not sufficiently complex to adapt sharp variations in textured images. Thus, algorithm can be improved by using more advanced context structures which models the coefficient correlation better than ours. However, our context adaptive approach has slightly better PSNR

Figure 4.7: Block based algorithm using bicubic interpolation for Lena with block size 8 × 8(Left) PSNR:36.43 and 32 × 32(Right) PSNR:35.65

Table 4.4: PSNR results of context adaptive filter design algorithm for Lena

Lena				
Window Size	Filter Size	Bicubic	w/o Decimation	Undecimated
64	3	35.91	35.89	35.94
	5	36.14	36.24	36.26
	7	36.34	36.45	**36.50**
256	3	35.66	35.61	35.69
	5	35.74	35.80	35.85
	7	35.80	35.87	**35.96**
1024	3	35.59	35.53	35.60
	5	35.64	35.67	35.71
	7	35.65	35.71	**35.80**

performance than block based structure in relatively smoother Lena image. Furthermore, context adaptive algorithm using undecimated wavelet transform as initial interpolation has better performance as we expected, because unlike block based case, undecimated wavelet transform shows very similar context structure with original high resolution image.

Figure 4.12 shows the difference between the best case(window size 64 and filter size 7) and the worst case(window size 1024 and filter size 3) of context adaptive

Figure 4.8: Block based algorithm with wavelet transform without decimation(Left) PSNR:36.38 and undecimated wavelet transform(Right) PSNR:36.32 for Lena

Table 4.5: PSNR results of context adaptive filter design algorithm for Barbara

Barbara				
Window Size	Filter Size	Bicubic	w/o Decimation	Undecimated
64	3	26.66	26.37	26.63
	5	26.98	26.75	27.24
	7	27.19	27.01	**27.64**
256	3	26.44	26.16	26.29
	5	26.62	26.39	26.70
	7	26.71	26.52	**26.95**
1024	3	26.37	26.10	26.16
	5	26.52	26.29	26.46
	7	26.57	26.39	**26.66**

algorithm using undecimated wavelet transform for initial interpolation of subbands for subregions from Lena image. Figure 4.13 shows the performances of bicubic interpolation and wavelet transform without decimation as initial interpolation with window size 64 and filter size 7 for subregions from Lena image. Figure 4.14 and 4.15 presents exact same algorithm results for Barbara image.

Table 4.6 shows the PSNR results of state of the art algorithms in image interpolation for Lena and Barbara. All results are for 2x magnification, i.e. size of

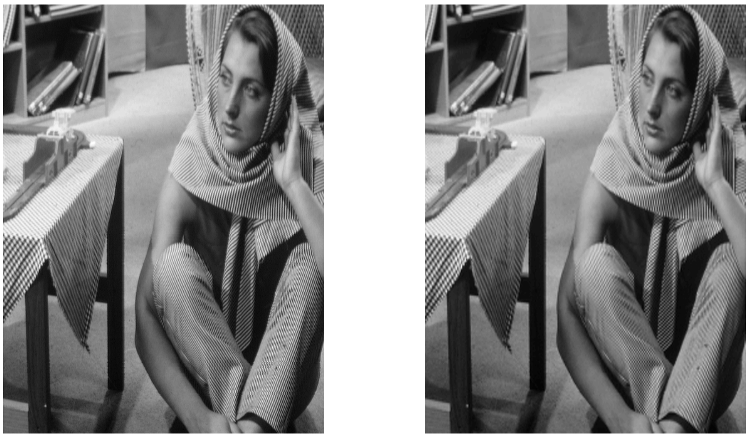

Figure 4.9: Original Barbara image(Left), Zerofill reconstruction of Barbara image(Right) PSNR:25.85

256 × 256 to 512 × 512. Also, wavelet filters are used in filtering and downsampling process to obtain low resolution image from high resolution image for all results. Bilinear and bicubic interpolation are the conventional algorithms and aforementioned zerofill algorithm is useful for comparison. NEDI is the algorithm proposed in “New Edge-Directed Interpolation” [29]. LAWBII is presented in “Locally Adaptive Wavelet-Based Image Interpolation” [16].

Table 4.6: PSNR results of state of the art algorithms in image interpolation for Lena and Barbara

Algorithm	Lena	Barbara
Bilinear	33.62	25.63
Bicubic	34.74	25.72
Zerofill	35.26	25.85
NEDI[29]	34.02	23.58
LAWBII[16]	32.46	26.94
Proposed(Block based)	36.43	**29.90**
Proposed(Context Adaptive)	**36.50**	27.64

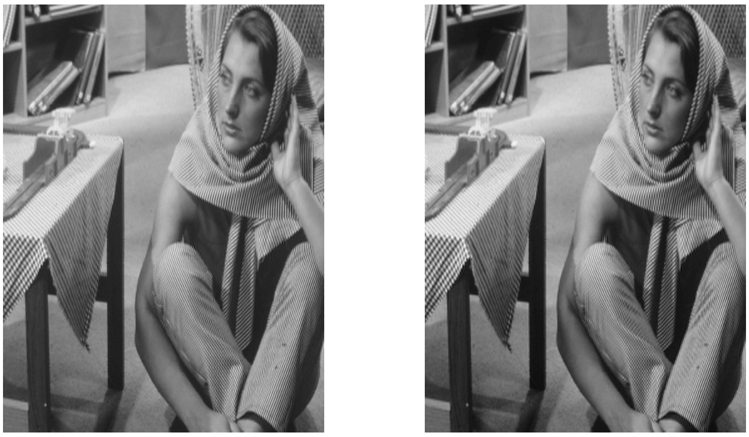

Figure 4.10: Block based algorithm using bicubic interpolation for Barbara with block size 8 × 8(Left) PSNR:29.90 and 32 × 32(Right) PSNR:27.90

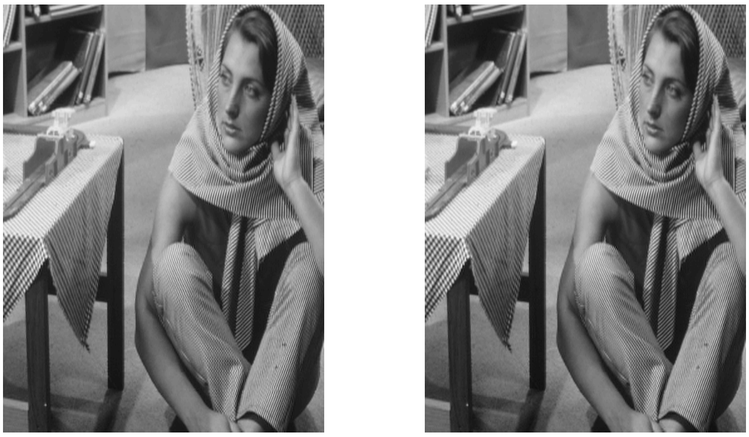

Figure 4.11: Block based algorithm with wavelet transform without decimation(Left) PSNR:29.18 and undecimated wavelet transform(Right) PSNR:29.83 for Barbara

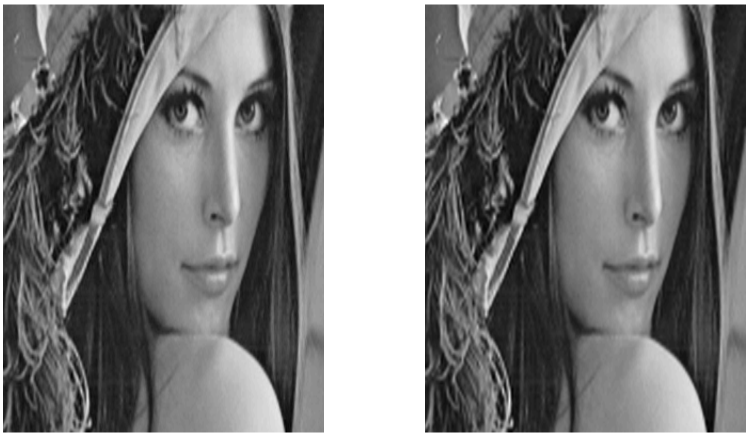

Figure 4.12: Context adaptive algorithm using undecimated wavelet transform for Lena with window size 64(Left) PSNR:36.50 and 1024(Right) PSNR:35.79

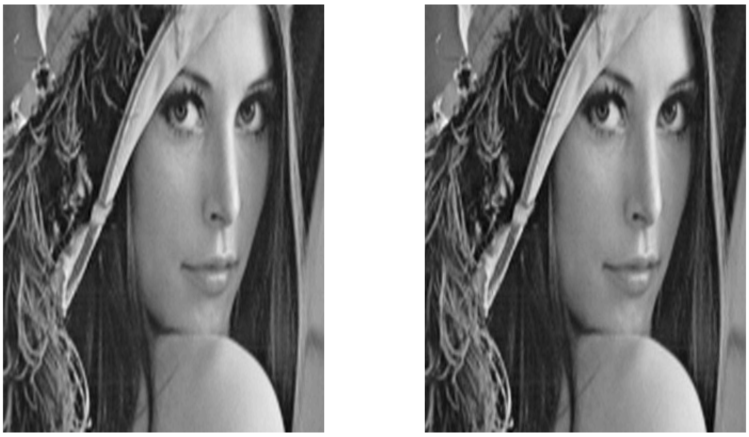

Figure 4.13: Context adaptive algorithm with bicubic interpolation(Left) PSNR:36.34 and wavelet transform without decimation(Right) PSNR:36.45 for Lena

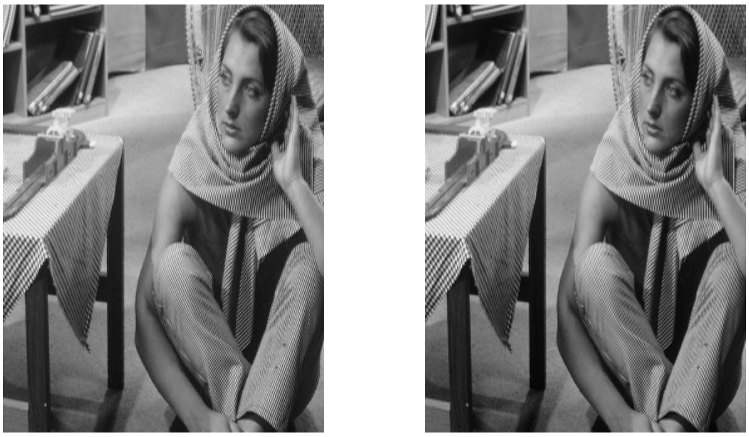

Figure 4.14: Context adaptive algorithm using undecimated wavelet transform for Barbara with window size 64(Left) PSNR:27.64 and 1024(Right) PSNR:26.16

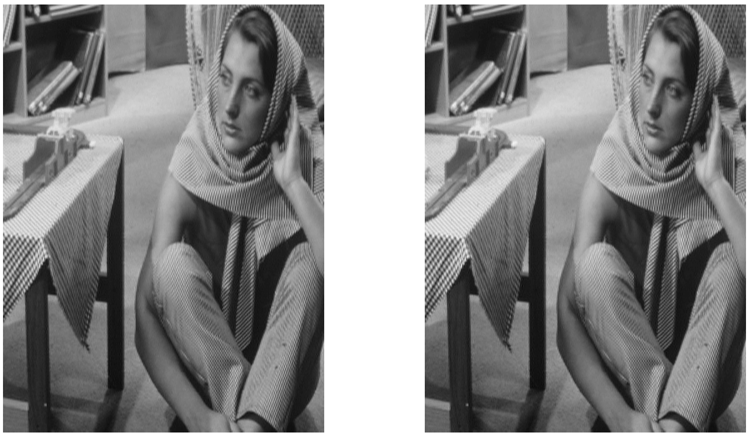

Figure 4.15: Context adaptive algorithm with bicubic interpolation(Left) PSNR:27.19 and wavelet transform without decimation(Right) PSNR:27.01 for Barbara

Chapter 5

Conclusion

In this thesis, we tried to utilize the resolution enhancement capabilities of both coding and interpolation algorithms. Generally, coding algorithms are considered independent from resolution enhancement. However, coding is another way of obtaining higher resolution images. Therefore, we aimed to construct a framework that consists of efficient coding algorithm and interpolation algorithm with superior results.

We designed an efficient wavelet based image coding algorithm that codes hierarchical description of wavelet coefficients instead of coding themselves. This hierarchical representation was the quantization index tree which was constructed using quantization indices of similar wavelet coefficients. This similarity was defined by using two different concepts. Also, this hierarchical index tree was optimized using a simple rate-distortion analysis. Despite its simplicity, our coding algorithm has paramount coding efficiency and objective results and surpassed majority of the state of the art coding algorithms.

Furthermore, we proposed a wavelet based image interpolation algorithm that models the correlation between high resolution and low resolution image. This correlation was captured using linear filters. And, we developed two different approaches on how to design these linear filters, i.e. block based and context adaptive approaches. Both approaches were successful on adapting different regions of natural images, e.g. sharp variations, textured regions. Especially, block based approach has very promising results which surpassed all of the algorithms in the literature.

In future work, we can develop different class assignment functions for our image coder which should have better match to the wavelet coefficient distribution of natural images. Also, wavelet packets can be used in coding algorithm which yields better results especially in textured images. For interpolation algorithm, we can improve our adaptive filter design approach by using more complex and sophisticated context that suits the characteristics of wavelet coefficients well.

References

[1] I. Daubechies. Where Do Wavelets Come From? - A Personal Point of View. In *Proceedings of the IEEE*, volume 84, pages 510–513, April 1996.

[2] The International Telegraph and Telephone Consultative Committee (CCITT) . Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines. Technical Report T.81, International Telecommunication Union (ITU), September 1992.

[3] T. Hopper, C. Brislawn and J. Bradley. WSQ gray-scale fingerprint image compression specification. Technical Report IAFIS IC-0110v2, Federal Bureau of Investigation (FBI) , February 1993.

[4] ISO/IEC JTC 1/SC 29/WG 1. Information technology – JPEG 2000 image coding system: Core coding system. Technical Report FCD15444-1, ISO/IEC, March 2000.

[5] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[6] A. Said and W. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuit Systems and Video Technology*, 6:243–250, June 1996.

[7] Z. Xiong, K. Ramchandran and M. Orchard. Space-frequency quantization for wavelet image coding. *IEEE Transactions on Image Processing*, 6(5): 677–693, May 1997.

[8] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1219–1235, July 2000.

[9] H. Ates and M. Orchard . Wavelet image coding using the spherical representation. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 89–92, September 2005.

[10] S.M. LoPresto, K. Ramchandran and M.T. Orchard . Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework. In *Proceedings of Data Compression Conference*, pages 221–230, March 1997.

[11] H. Spath. *Two Dimensional Spline Interpolation Algorithms.* AK Peters, 1995.

[12] G. Chang, Z. Cvetkovic and M. Vetterli. Resolution Enhancement Of Images Using Wavelet Transform Extrema Extrapolation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2379–2382, May 1995.

[13] K. Kinebuchi, D.D. Muresan and T.W. Parks . Image Interpolation Using Wavelet-based Hidden Markov Trees. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1957–1960, May 2001.

[14] Y. Zhu, S.C. Schwartz and M.T. Orchard. Wavelet domain image interpolation via statistical estimation. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 840–843, October 2001.

[15] W.K Carey, D.B. Chuang and S.S. Hemami. Regularity-Preserving Image Interpolation. *IEEE Transactions on Image Processing*, 8(9):1293–1297, September 1999.

[16] S.G. Chang, Z. Cvetkovic and M. Vetterli. Locally Adaptive Wavelet-Based Image Interpolation. *IEEE Transactions on Image Processing*, 15(6):1471–1485, June 2006.

[17] H.F. Ates and M.T. Orchard . Image Interpolation Using Wavelet-Based Contour Estimation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 109–112, April 2003.

[18] C.S. Burrus, R.A. Gopinath and H. Guo. *Introduction to Wavelets and Wavelet Transforms : A Primer*. Prentice Hall, 1998.

[19] A. Abbate, C.M. DeCusatis and P.K. Das. *Wavelets and Subbands : Fundamentals and Applications*. Birkhäuser, 2001.

[20] S. Lawson and J. Zhu. Image compression using wavelets and JPEG2000: a tutorial. *IEEE Electronics and Communications Engineering Journal*, 14 (3):112–121, 2002.

[21] A. Temizel. Image Resolution Enhancement using Wavelet Domain Hidden Markov Tree and Coefficient Sign Estimation. In *Proceedings of the IEEE International Conference on Image Processing,ICIP 2007*, volume 5, pages 381–384, September 2007.

[22] K.R. Rao and P.C. Yip. *The Transform and Data Compression Handbook*. CRC Press, 2000.

[23] T.K. Sarkar, C. Su, R. Adve, M. Salazar-Palma, L. Garcia-Castillo, and R.R. Boix. A Tutorial on Wavelets from an Electrical Engineering Perspective. Part 1: Discrete Wavelet Techniques. *IEEE Antennas and Propagation Magazine*, 40(5):49–68, October 1998.

[24] H.F. Ates and M.T. Orchard . Spherical coding algorithm for wavelet image compression. *IEEE Transactions on Image Processing*, 18(5):1015–1024, May 2009.

[25] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(9):1445–1453, Sep 1988.

[26] A. Cohen, I. Daubechies and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45 (5):485–560, 1992.

[27] S.T. Hsiang. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. In *Proceedings of the Data Compression Conference, DCC '01*, pages 83–92, March 2001.

[28] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. 14(7):710–732, July 1992.

[29] X. Li and M.T. Orchard. New Edge-Directed Interpolation. *IEEE Transactions on Image Processing*, 10(10):1521–1527, October 2001.