

Perceptual Noise Shaping in Dual-tree Complex Wavelet Transform for Image Coding

by

Junwu Zhu, B.Eng.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
March, 2010

©Copyright
Junwu Zhu, 2010



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-68638-6
Our file Notre référence
ISBN: 978-0-494-68638-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■+■
Canada

The undersigned hereby recommends to the
Faculty of Graduate Studies and Research
acceptance of the thesis

**Perceptual Noise Shaping in Dual-tree Complex Wavelet Transform
for Image Coding**

submitted by **Junwu Zhu, B.Eng.**

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Professor Richard M. Dansereau, Thesis Supervisor

Professor Chris Joslin, Thesis Co-supervisor

Professor Howard Schwartz, Chair,
Department of Systems and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
March, 2010

Abstract

In this thesis, we propose a perceptual extension to noise shaping for a dual-tree complex wavelet transform (DCWT) based image coding system. In classical noise shaping, the spatial error information is compensated by adding it back into the whole DCWT domain, which allows the retained coefficients have better capability to approximate the original image. However, the error is only calculated by magnitude distance which may not effectively reflect human perception. The structural similarity (SSIM) index is a novel image quality index that involves more perceptual consideration than classical peak signal-to-noise ratio (PSNR). Our perceptual noise shaping introduces SSIM as a weight to the spatial errors. Moreover, the perceptual weighting process also uses other parameters to control the gain of error so that the system has more capability to balance the overall image quality. Compared to classical noise shaping, the performance of the proposed perceptual noise shaping has better results for visual quality and has higher SSIM index in most of our testing cases. For example, the proposed perceptual noise shaping achieves an overall SSIM of 0.891 for the 8 bit 512×512 “barbara” compared to 0.878 in classical noise shaping when 5000 coefficients are retained.

This is the dedication to my family.

Acknowledgments

I would like to acknowledge and express my deepest gratitude to my supervisor, Professor Richard M. Dansereau, who has been providing abundant help and offering invaluable support and guidance from the very beginning of my study. I also sincerely appreciate his time and patience to point out and correct my numerous mistakes during the research. Without his continuous support, guidance and encouragement, I would not have come this far.

I would also like to acknowledge my gratitude to my other supervisor, Professor Chris Joslin, who gave me valuable suggestions and guidance. His humor always enlivens meetings and are unforgettable.

I would also like to thank all professors who have taught me at Carleton University and the University of Ottawa. The excellent courses are like a buffet to fill my curiosity and enrich my knowledge in DSP.

Special thanks to Khalil S. Yonis and Felix Petngang for many helpful discussions and all the people in the department who gave me support and help.

I would like to thank my wife Hanwen Cai, for her understanding and support, as well as my parents Zheng Liu and Xuming Zhu, for their encouragements.

Table of Contents

Abstract	iii
Acknowledgments	v
Table of Contents	vi
List of Figures	viii
List of Tables	x
List of Acronyms	xi
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem description	3
1.4 Contributions	4
1.5 Image quality measurement metrics and software used in this thesis	4
1.6 Organization of the thesis	5
2 Background	6
2.1 Transform coding	6
2.1.1 Source coding and transform coding	6
2.1.2 Components of the transform coding model	7
2.2 Comparison of discret cosine transform and discrete wavelet transform	9
2.2.1 Discret cosine transform	9
2.2.2 Discrete wavelet transform	11
2.3 The dual-tree complex wavelet transform	15

2.3.1	Complex wavelet and dual tree structure	16
2.3.2	2D dual-tree complex wavelet transform	17
2.3.3	Dual-tree complex wavelet transform based image coding	21
2.4	Human visual system	21
3	Noise Shaping - Balance the Coefficient Expansion	23
3.1	Noise shaping fundamentals	23
3.1.1	Noise shaping by feedback and oversampling	24
3.1.2	Noise shaping in dual-tree complex wavelet transform	25
3.1.3	Transform projections and convergence analysis	27
3.2	Coefficients selection by coefficient magnitude	31
3.2.1	Hard thresholding	31
3.2.2	Thresholding by Wiener denoising function	33
4	Perceptual Noise Shaping Framework	36
4.1	Perceptual image quality measurement	36
4.1.1	Structural similarity index	36
4.1.2	Structural similarity index in dual-tree complex wavelet transform .	40
4.2	Perceptual noise shaping	40
4.2.1	Coefficient selecting by perceptual weighting	42
4.2.2	Feedback error shaping by perceptual weighting	45
5	Conclusions and Discussions	55
5.1	Conclusions	55
5.2	Discussions	56
5.3	Future work	57
List of References		58
Appendix A Additional examples by perceptual noise shaping		62

List of Figures

2.1	Source coding model and transform coding model	8
2.2	Typical image coding system	8
2.3	JPEG and JPEG2000 coding performance comparision	12
2.4	Filter bank structure for multiresolution wavelet analysis	13
2.5	Image in 2D DWT domain	14
2.6	Dual-tree filter bank structure in DCWT.	17
2.7	2D Fourier spectrum comparison between DWT and DCWT	19
2.8	2D impulse response comparison between DWT and DCWT	19
2.9	Image in 2D DCWT domain and real DWT domain.	20
3.1	Diagram of noise shaping in A/D applications.	24
3.2	Diagram of noise shaping in DCWT for image coding	27
3.3	Noise shaping performance example	28
3.4	Projections onto convex sets	30
3.5	PSNR plot with different start number of nonzero coefficients for hard threshold noise shaping.	32
3.6	Wiener threshold curve	35
4.1	Spatial domain SSIM comparison by different types of distortions	37
4.2	Diagram of the SSIM measurement system	38
4.3	Complex wavelet domain SSIM comparison by different types of distortions	41
4.4	Diagram of perceptual threshold for noise shaping	43
4.5	Image results examples by using classical threshold and perceptual threshold	44
4.6	Diagram of feedback error perceptual weighting for noise shaping.	45
4.7	Relationship curve between SSIM and feedback error weighting.	46
4.8	Image SSIM comparison among different wavelet transforms	49
4.9	Histogram of SSIM comparsion between classical noise shaping and error perceptual weighting noise shaping	50
4.10	Unstable examples of error perceptual weighting noise shaping.	50

4.11	Noise shaping performance comparison between classical feedback error and feedback error with reverse weighting.	51
4.12	SSIM comparison between classical noise shaping and perceptual noise shaping by image database	54
A.1	“Boat” image with 3000 non-zero coefficients	63
A.2	“Boat” image with 5000 non-zero coefficients	64
A.3	“Boat” image with 10000 non-zero coefficients	65

List of Tables

2.1	Weighting comparsion between orthogonal and biorthogonal wavelets . . .	15
3.1	A/D quantization table example	25
3.2	A/D quantization result without noise shaping	26
3.3	A/D quantization result with noise shaping	26
3.4	Image PSNR results of noise shaping by hard thresholding	34
3.5	Image PSNR results of noise shaping by Wiener thresholding	35
4.1	Image SSIM comparison between classical and perceptual noise shaping .	52

List of Acronyms

- A/D** analog-to-digital converter
- CSF** contrast sensitivity functions
- DCT** discrete cosine transform
- DCWT** dual-tree complex wavelet transform
- DFT** discrete Fourier transform
- DWT** discrete wavelet transform
- EBCOT** embedded block coding with optimal truncation
- EZW** embedded zerotree wavelet
- HVS** human visual system
- LET** local energy threshold
- MSE** mean square error
- POCS** projections onto convex sets
- PR** perfect reconstruction
- PSD** perceptual sub-band decomposition
- PSNR** peak signal-to-noise ratio
- SPIHT** set partitioning in hierarchical trees
- SSIM** structural similarity

Chapter 1

Introduction

1.1 Introduction

Transforms have been pervasive in many aspects of signal processing for years. One of the most visible successes of transforms is their application in image compression. The transforms used in image compression work as a decorrelator to remove spatial redundancy of the image [1]. Two famous transforms, the discrete cosine transform (DCT) and the discrete wavelet transform (DWT), are used in state-of-the-art image coding standards. The DCT was first adapted in image coding systems as a standard transform of JPEG in the early 1990s. As it has low complexity, it is very good for real time applications such as motion pictures, videos, etc. However, typical block-based DCT image compression fails in exploiting long-range correlations [2], and moreover, breaking the image into many small blocks results in serious blocking artifacts at low bit rates. Using the DWT usually does not require the image to be broken up into small blocks, hence, it allows long-range correlation to be effectively removed and avoids block artifacts as well. Another advantage of DWT based image coding is that inter subband properties and relations can be used to develop subband coding methods. Recently, a new transform, dual-tree complex wavelet transform (DCWT), as a member of the DWT family which has advantages of directional selectivity and nearly shift invariance, is active in research as a potential candidate transform for image coding.

Compared to the critical-sampled DWT, the DCWT is an over-sampled (overcomplete) transform. In other words, the coefficients must expand in the wavelet domain. For example, an image of 1000 pixels will have 2000 coefficients after it is transformed into the DCWT domain in 2D case. Although the directional selectivity of the DCWT is proven

in many aspects to be beneficial in image processing, there is a dilemma that the “coefficient expansion” contradicts the goal of compression. Noise shaping has been introduced in DCWT image coding to ease this dilemma. It was proposed by Reeves and Kingsbury in [3] and [4], and we call it “classical noise shaping” in this thesis. Noise shaping attempts to set small coefficients to zero and modify the large coefficients to compensate as much as possible for the loss of the small ones, while still producing a good approximation to the original image [4]. In other words, it could be regarded as a way to balance the “coefficient expansion” problem for an overcomplete wavelet transform. With noise shaping, it allows the overcomplete DCWT to possibly be adapted to an image compression system as a decorrelator candidate.

The classical noise shaping is considered as an iterative projection system, which is, the wavelet coefficients are first cut off by a threshold and then shaped by the spatial error feedback between the original signal and distorted signals. In terms of peak signal-to-noise ratio (PSNR), classical noise shaping contributes a few dB improvement when it is used in a DCWT image coding system [4]. However, it may not always take the right direction to shape the coefficients for human perception. Since human observers are the final “receiver” and “judge” of the images considered in this thesis, perceptual consideration is definitely a key point in selecting and shaping “important” coefficients. We felt that incorporating properties of the human visual system (HVS) to noise shaping and even to the DCWT based image coding structure would be beneficial in improving visual performance. In perceptual based image coding, a perceptual metric is usually used in controlling its performance. In this thesis, we use the structural similarity (SSIM) index [5] to weight the spatial errors which are feedback to shape the large coefficients in the noise shaping. We call this novel noise shaping with weighting process “perceptual noise shaping”. By testing it on various images with different kept rate of coefficients, we found that the overall SSIM index is improved in most test situations and the visual performance is also improved slightly in many test images. We also note that visual performance is only based on our subjective assessment which may vary according to different assessors and different viewing environments.

1.2 Motivation

Using the wavelet transform in image coding has been an active area of research for years. In order to achieve the compression goal, the critically-sampled DWT is traditionally applied in state-of-the-art image coding systems. However the DWT suffers from the lack of

shift invariance and the poor directional selectivity problems. In order to overcome these problems, a non critically-sampled DCWT was developed recently [6]. It is proven to be able to provide near shift invariance and directional selectivity at the expense of coefficient expansion. Using the DCWT directly in image coding increases the difficulty of subsequent subband coders since a subband coder has to deal with more coefficients than the critically-sampled DWT. Noise shaping is a state-of-the-art and a critical procedure to reduce the number of DCWT coefficients by discarding small magnitude coefficients and shaping the remaining coefficients to have better capability to approximate the original signals. Because of its importance in DCWT based image coding system, we felt that studying and understanding noise shaping and its approach as well as contributing in this procedure will be very beneficial to any later DCWT based image coding systems or similar overcomplete wavelet transforms.

Since the image is finally assessed by human observers, an image result that looks best to a human observers is our ultimate goal for image compression. Therefore, perceptual consideration is another task in our research after noise shaping itself was carefully studied. In order to maintain good visual quality, how to incorporate a perceptual metric into the noise shaping to control the visual performance is hence a primary focus.

1.3 Problem description

Classical noise shaping uses a global threshold to remove small coefficients whose magnitudes are lower than the threshold; the retained large coefficients are shaped by adding the information of the discarded small coefficients through feedback. However, the coefficients with large magnitude may not always be “important” with regards to human perception since the HVS involves complicated physiological and psychophysical processes which can not be simply decided by the signal’s intensity. A typical solution is to apply image quality metrics into the system to control the overall image quality. However, applying traditional mean square error (MSE) into image coding to control the system’s performance may not be a very accurate predictor for HVS [7]. Therefore, how to effectively incorporate a quality metric which accurately reflects the perceptual need in an image coding system is a challenge.

1.4 Contributions

The contributions in this thesis are listed below:

- We propose a perceptual noise shaping structure by perceptually weighting the spatial error which will be used as feedback into the noise shaping iterations.
- We introduce a new perceptual weighting equation by using SSIM and other parameters to weight the spatial error in noise shaping. By introducing SSIM to weight the spatial errors and using other parameters to control its gain in the noise shaping, the performance of perceptual noise shaping achieves better visual quality and higher overall SSIM index in most of our tests.
- We analyze the possibility and the issue of perceptual threshold in classical noise shaping.

1.5 Image quality measurement metrics and software used in this thesis

In order to compare and evaluate the performances of different image coding algorithms, we used PSNR and SSIM in this thesis. Although PSNR has known issues for measuring image quality, due to its low complexity and wide acceptance, we used it also for comparing with the literatures. The PSNR in image processing is calculated as

$$PSNR = 10 \log_{10} \frac{M \cdot N \cdot MAX^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - K(i, j))^2}. \quad (1.1)$$

Here, $I(i, j)$ is the original image and $K(i, j)$ is the reconstructed image; M and N are the width and height of the image; MAX is the maximum possible pixel value of the image. Since all images tested in this thesis are 8 bits per pixel, MAX is 255.

About SSIM, more details could be found in Chapter 4.

Matlab is the main simulation environment for experiments in the thesis. Some Matlab code, such as SSIM function is adapted from [8]; DCWT version SSIM is adapted from Yonis [9]; DCWT function is adapted from [10]. The figures in the thesis that show JPEG and JPEG2000 examples are generated by the VcDemo software [11]. The rest is written by us using Matlab.

1.6 Organization of the thesis

The thesis is organized as follows. Chapter 2 provides the background information of image coding system from a broad perspective. It starts with introducing transform coding and its components, DCT and DWT, and ends with discussing the DCWT transform for image coding. A brief introduction of the HVS is also included in this chapter. Chapter 3 mainly discusses the noise shaping in DCWT. It starts with introducing the noise shaping algorithm, followed by coefficient selection by various thresholds. Chapter 4 discusses the perceptual noise shaping framework, our experiments and our test results. Finally, conclusions and discussion are made in Chapter 5, with suggestions for future work and directions.

Chapter 2

Background

2.1 Transform coding

In state-of-the-art lossy image compression systems, transform coding is the most commonly used. Beyond image compression, transform coding is also widely used in audio compression and video compression systems. By using transform coding, the compression problem is divided into three relatively simply steps: transform, quantization and entropy [12]. This process is kind of like using “Divide and Conquer” strategy, which divides a complicated and big problem into smaller problems so that they can be understood and solved easily [12]. Moreover, by transforming images from the spatial domain into the transform domain, the energy of the image is more likely to concentrate in a specific range so that we can discard less important information. Although it may not be perfect for exploiting dependencies, since it is easy to implement, it is still the primary way to effectively compress an image. Transform coding could be regarded as a suboptimal but low complexity source coding technique [12]. Please note that although transform coding could also be used in lossless image compression, the term of transform coding or image coding in this thesis will only refer to the lossy case.

2.1.1 Source coding and transform coding

Source coding is defined by using a small number of bits to represent the source information as accurately as possible [12]. Figure 2.1(a) depicts the ideal source coding model which is cited in [12]. As shown in the figure, the source coding includes two mappings: encoder and decoder. The input signal x is converted to a string of bits after the encoding, and the bits are reconstructed to \hat{x} by decoding. In general, “the encoder maps any vector

$x \in \mathbb{R}^N$ to a finite string of bits" and the decoder maps it back to an approximation $\hat{x} \in \mathbb{R}^N$ [12]. The input signal x and reconstruction or approximation signal \hat{x} could be identical or different. Usually the identical input and output signal needs more bits to represent than when the output is an approximation of the input. For compression, the former case is called lossless compression while the later is called lossy compression. The mapping process is implemented by two factors $\gamma \circ \alpha$ in encoding and $\beta \circ \gamma^{-1}$ in decoding. The two factors α and γ are quantizer and entropy coder. In ideal source coding, the peak performance is achieved when a large block or the whole data of the input signal is sent to α . Thus, the α will become much more complicated to be handled in the real world.

Transform coding is a practical example of constrained source coding which means it is not optimal but has low complexity, and more importantly, it is practical to implement [12]. Transform coding handles the α as depicted in Figure 2.1(b). It breaks the whole α into many smaller α (from α_1 to α_N) through the transform, which makes it easier to be implemented in practice. For the same reason, β in the source coding is replaced by N inverse scalar quantizers (from β_1 to β_N) and inverse linear transform U , where $U = T^{-1}$. Although it is not best optimized, it performs well in almost all of the modern audio, image and video standards and will probably be the primary model in compression for the near future.

2.1.2 Components of the transform coding model

A typical transform coding system used in image processing is depicted in Figure 2.2. It has three components: transform, quantization and entropy coding. Selecting an appropriate transform is one of the core tasks in transform coding systems. Nowadays, the most commonly used transforms in image coding include the DCT and DWT, more details about these two transforms will be discussed in Section 2.2.

After the transform, quantization is the second step in the transform coding system, there are numerous quantization methods that have been developed over the past decades. For example, uniform quantization adapted in the DCT based JPEG standard [13], subband coding style zero-tree quantization embedded zerotree wavelet (EZW) [14] and more recently set partitioning in hierarchical trees (SPIHT) [15], which are widely used in wavelet-based image coding. Quantization is the main contributions for the lossy compression since it performs a many to one mapping strategy. As a result, it also introduces the quantization

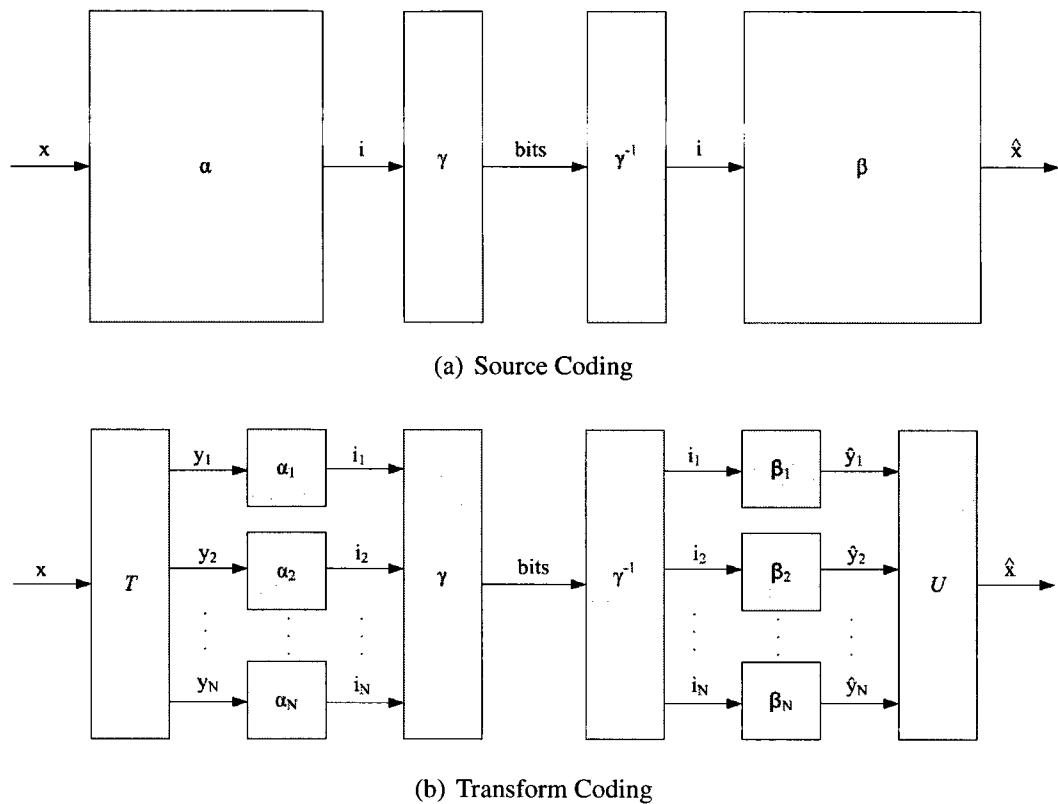


Figure 2.1: Source coding model and transform coding model [12]

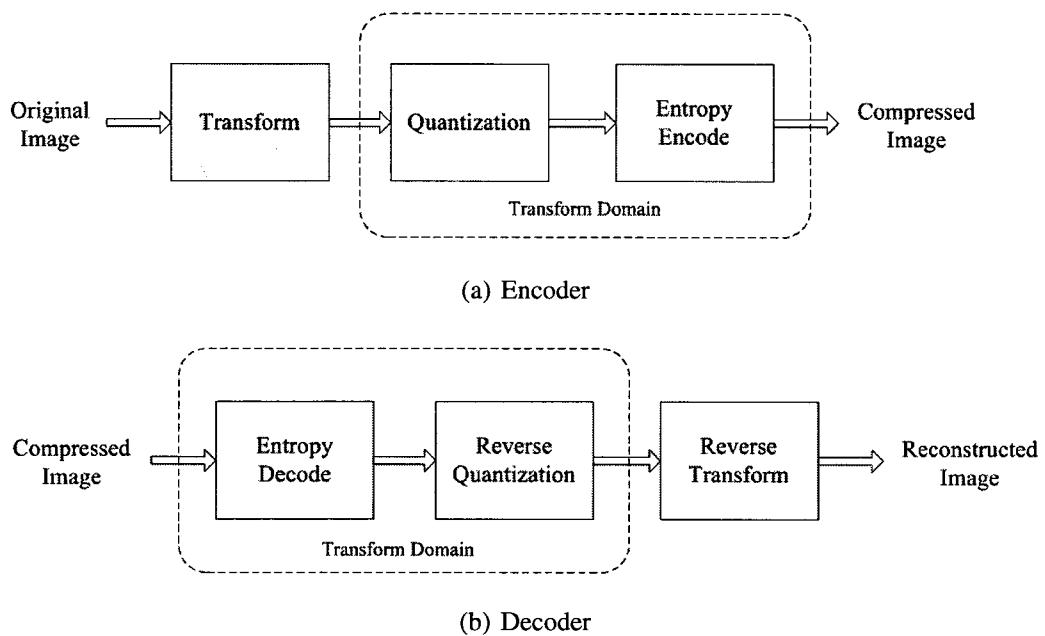


Figure 2.2: Typical image coding system

error as well [16].

Entropy coding is the last step to get additional compression. Based on information theory, entropy coding uses a model to determine the probabilities of the input signals and it then codes the input along those probabilities. As in Figure 2.2(a), given the finite sequence of z_1, z_2, \dots, z_N from quantization, the output of $\gamma(z_1), \gamma(z_2), \dots, \gamma(z_N)$ are the concatenation of input sequence. The goal of entropy coding is to decrease the total number of bits by minimizing the mean number of bits used to represent z [12]. Huffman coding [17] and arithmetic coding [18] are early developed techniques and are commonly used. Besides these methods, a new kind of context-based adaptive entropy coding method has been studied in image and video processing in recent years [19].

2.2 Comparison of discret cosine transform and discrete wavelet transform

A core part of transform coding systems is the transform unit. Transforms simplify the problem and makes the following processes simple as well. The DCT and DWT are the most commonly used and the most investigated transforms in image coding. The Joint Photographic Experts Group adopted DCT based and DWT based image coding standards JPEG and JPEG2000 in 1992 and 2000. The DCT is not new, with its introduction in 1974, and it has made a significant impact in the signal processing field. Due to its energy packing capabilities and low computation load, the DCT was the first used in image and video compression and it is still the primary choice in current video compression systems [20].

2.2.1 Discret cosine transform

In signal processing, time-frequency transforms like the discrete Fourier transform (DFT) have been widely used for more than one century in engineering and research. In some situations, it is much easier to analyze or process by transforming a signal from the time domain into the frequency domain. Similar to the Fourier transform, the DCT can be regarded as “a discrete-time version of Fourier-cosine series” [16]. Equation 2.1 and 2.2 are the DCT and inverse DCT pairs used in the JPEG standard [13]. Unlike the Fourier transform, which involves complex operations, the DCT is a pure real transform which indicates lower complexity. Like the famous fast Fourier transform (FFT), the DCT can also be computed “fast”. For example, in [21] it can be computed by using as little as 54

multiplications to calculate a 2D 8×8 DCT.

$$F(u, v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \right]$$

where $C(u), C(v) = \frac{1}{\sqrt{2}}$ for $u, v = 0$, otherwise $C(u), C(v) = 1$ (2.1)

$$\hat{f}(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v) F(u, v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \right]$$

where $C(u), C(v) = \frac{1}{\sqrt{2}}$ for $u, v = 0$, otherwise $C(u), C(v) = 1$ (2.2)

In order to achieve good performance, a DCT based image compression system usually partitions the whole image into small blocks. The block is typically 8×8 pixels in the JPEG standard, thus there are 64 pixels in a block of input. By applying Equation 2.1, they are transformed into 64 orthogonal basis coefficients $F(u, v)$ which are the two-dimensional spatial frequencies of the input block image. The 64 orthogonal basis coefficients $F(u, v)$ are also called DCT coefficients. In the decoding phase, the reconstruction signal $\hat{f}(x, y)$ is obtained by calculating the DCT coefficients using Equation 2.2. At the zero frequency along the horizontal and vertical dimensions ($x = 0$ and $y = 0$ in Equation 2.1), the DCT coefficient is called the “DC coefficient” while the remaining 63 coefficients are called the “AC coefficients”. Because natural images do not change dramatically in a small region, it gives more similarity among the 64 point inputs which leaves more room for “compression”. For most natural images that are partitioned into small blocks like 8×8 , most of the DCT coefficients $F(u, v)$ are zero or close to zero after the image block is transformed into DCT domain, thus the less significant DCT coefficients or zero DCT coefficients could be removed and encoded as zeros [13].

The appropriate block size is important in the DCT based image coding system. The block has less frequency fluctuation when the size is small, which means there is more frequency similarity inside the block. But if the block size too small, it is difficult to achieve the purpose of compression. In practice, 8×8 and 16×16 pixel blocks are a good compromise between performance and compression and they are the most common for image coding [20]. However, blocking artifacts are always an issue since they can

be perceived at low bit rates. Because natural images are partitioned into many smaller blocks, when the blocks are encoded at low bit rates, the boundary difference among the blocks is perceived which is called blocking artifact. Figure 2.3 shows the DCT based JPEG and DWT based JPEG2000 comparison at the same bit rate of 0.2 bpp and 0.4 bpp. As we can see, blocking artifacts are clearly shown at 0.2 bpp, as the encoded bit rate increases, the blocking artifacts will be alleviated and are more difficult to see. Please note that the result shown in the image may not be fair since JPEG used here does not involve an advanced quantizer like EZW. In order to fairly compare the performance of DCT based and DWT based image coding systems, the authors in [22] tested DCT by applying advanced quantizers. Nevertheless, the result shows that the wavelet transform in image compression still outperforms at around 1 dB [22] in PSNR for the test images.

2.2.2 Discrete wavelet transform

Compared to the DCT, image coding using the DWT usually does not involve breaking the image into small blocks, hence it has the inherent capability to avoid block artifacts. Theoretically, any finite energy signal $x(t)$ can be decomposed by a series of real-valued stretched and shifted bandpass wavelets $\psi(t)$ and lowpass scaling functions $\phi(t)$ as in Equation 2.3

$$x(t) = \sum_{n=-\infty}^{\infty} c(n)\phi(t-n) + \sum_{j=0}^{\infty} \sum_{n=-\infty}^{\infty} d(j, n)2^{j/2}\psi(2^jt-n), \quad (2.3)$$

In Equation 2.3, the scaling coefficients $c(n)$ and wavelet coefficients $d(j, n)$ are calculated by Equation 2.4 and 2.5 [23]

$$c(n) = \int_{-\infty}^{\infty} x(t)\phi(t-n)dt, \quad (2.4)$$

$$d(j, n) = 2^{j/2} \int_{-\infty}^{\infty} x(t)\psi(2^jt-n)dt. \quad (2.5)$$

In practice, the wavelet transform is usually implemented using filter banks, thus the filter bank design plays a vital role in the DWT. In the wavelet transform, the filter bank is a tool to split the signal into different frequency bands (resolution). Figure 2.4 shows a typical three level filter bank by unbalanced decimate decomposition tree. First, level one is done by dividing the signal into one low pass band (forward filter H0) and one



(a) DCT based JPEG at bit rate 0.2 bpp
(PSNR=28.9dB)



(b) DWT based JPEG2000 at bit rate 0.2 bpp
(PSNR=32.9dB)



(c) DCT based JPEG at bit rate 0.4 bpp
(PSNR=33.4dB)



(d) DWT based JPEG2000 at bit rate 0.4 bpp
(PSNR=36.1dB)

Figure 2.3: JPEG and JPEG2000 coding performance comparision

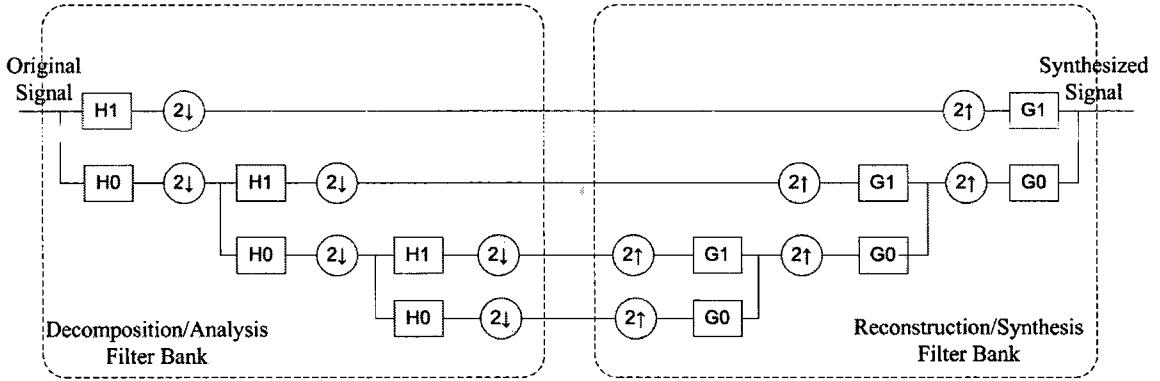


Figure 2.4: Filter bank structure for multiresolution wavelet analysis

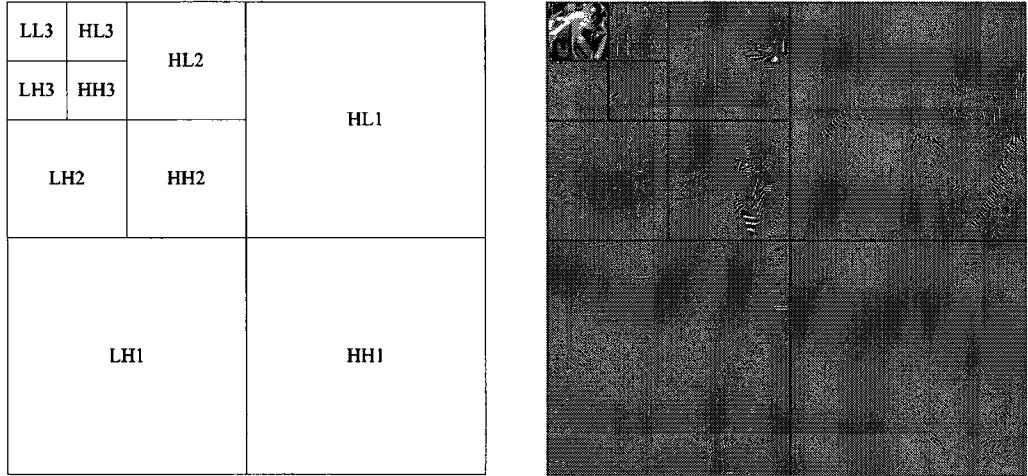
high pass band (forward filter H1) which are followed by down sampling. Then level two and all subsequent levels repeat the same process on the low pass band. On the right side of the figure, the reconstruction signal is computed reversely by backward filtering through G0 and G1. In this way, the image is decomposed as in Figure 2.5 to different multiresolution levels. The advantage of DWT based image coding is that lots of inter subband properties and relations could be utilized to develop subband coding methods. With the help of the subband coding techniques like EZW, SPIHT and embedded block coding with optimal truncation (EBCOT, discussed further in [24]), wavelet based image coding achieves better performance and scalability and it was finally adapted as a newer standard of image compression in JPEG2000 in the beginning of this century.

According to information compression theory, the best compression result occurs when the transform base is orthogonal so that it satisfies the energy conservation property. Since the DCT uses orthogonal bases, the early wavelet filters were also implemented by orthogonal wavelets, with forward filters and backward filters that should satisfy constraints as in Equation 2.6, 2.7, and 2.8 [25]

$$\sum_n h(n - 2i)h(n - 2j) = \delta(i - j), \quad (2.6)$$

$$\sum_n g(n - 2i)g(n - 2j) = \delta(i - j), \quad (2.7)$$

$$\sum_n h(n - 2i)g(n - 2j) = 0. \quad (2.8)$$



(a) The subband Labeling scheme for a three-level, 2D symmetric wavelet transform

(b) Corresponding image in the wavelet domain

Figure 2.5: Image in 2D DWT domain

Similar to the Parseval property in Fourier analysis, orthogonal wavelets also have the energy conservation property for a given input signal $x(n)$ as in Equation 2.9 [25]

$$\sum_{n=0}^{N-1} x^2(n) = \sum_{l=0}^{L-1} w^2(l). \quad (2.9)$$

However, there are two problems which hinder orthogonal wavelets to be used in image compression. First, there is “coefficient expansion” by using standard orthogonal wavelets [25]. As in Equation 2.9, the total number of wavelet coefficients L is greater than the total number of input samples in the maximally decimated wavelet transform. Although there are some techniques like circular convolution that deal with the “coefficient expansion”, it brings new problems which are also hard to solve. Second, there are few linear phase filters that support orthogonal wavelets. Actually for real value wavelets, only the trivial Haar filter supports orthogonal wavelets [25]. These two shortcomings make orthogonal wavelets hard to be the ideal wavelet transform in image coding. In practice, biorthogonal wavelets were used in image coding to overcome these problems. Biorthogonal wavelets have two implications: first, although it is not orthogonal which means it is not fully energy preserving, it is close to full energy preserving which indicates that it could still be regarded as a good and practical transform in image coding; second, there are lots of linear phase filters that could be used in biorthogonal wavelets transform which

Table 2.1: Weighting comparsion between orthogonal and biorthogonal wavelets

Weight	Orthogonal	9/7
w_{10}	1.00000	0.98295
w_{11}	1.00000	1.04043
w_{20}	1.00000	1.03060
w_{21}	1.00000	0.96721
w_{30}	1.00000	1.05209
w_{31}	1.00000	1.03963
w_{40}	1.00000	1.03963
w_{41}	1.00000	1.07512

means more design freedom in wavelet-based image coding. The following is an example to show the near orthogonal property of the famous biorthogonal 9/7 filter set developed by Cohen, Daubechies and Feauveau. In the wavelet transform, the total output variance should be the sum of weights times subband variances in different subband. The weights of orthogonal wavelet are always equal to 1 which indicates that energy is fully preserved in the transform. As shown in the Table 2.1, weights of the 9/7 filter set are only a few percent from the orthogonal weighting [25]. Due to its good properties, the 9/7 set was successfully adopted by the Joint Photographic Experts Group as the core transform in JPEG2000.

2.3 The dual-tree complex wavelet transform

The dual-tree complex wavelet transform (DCWT) was developed by Nick Kingsbury in order to overcome the lack of shift invariance and the poor directional selectivity of the DWT [6]. Due to its near shift invariance and directional selectivity in two or higher dimensions, it has a wide success and has shown to have a good prospect in image coding [26] [27], video coding [28] [29], image denoising [30], digital watermarking [31] and image/video quality assessment [9].

2.3.1 Complex wavelet and dual tree structure

A typical real DWT uses a decimated filter bank to lighten the redundancy and subsequent processing computation, it also brings shift variance between different levels of decomposition. In the Fourier spectrum, the ideal situation is that the energy in each frequency band does not have any shift in time or space. However, in the decimate structure of the real DWT, subsampling at different wavelet levels causes aliasing, which leads to major variations in the distribution of energy [32]. The shift variance makes some signal processing techniques based on position of the key features hard to handle, for example motion estimation in wavelet domain. Moreover, it happens frequently around the image edges so that it increases the computational requirement for subsequent processing [33]. Since the shift variance is caused by an unbalanced decimate tree structure during wavelet decomposition, the straightforward way is to use an undecimated form of the dyadic filter tree. However, the redundancy is increased significantly by doing this. One alternative solution for this problem is to replace the real wavelet transform by a complex wavelet transform at the cost of limited redundancy ($2 : 1$ for 1-D, $2^m : 1$ for m-D). The phase based complex wavelet is inspired by the perfect shift invariant Fourier transform. Although the complex wavelets could provide approximate shift invariance, it may bring out the problem of perfect reconstruction (PR). To achieve shift invariance as well as perfect reconstruction, Kingsbury proposed the dual-tree complex wavelet transform (DCWT) to ensure PR (by using orthogonal or biorthogonal filters) [32].

Figure 2.6 shows the three level dual-tree filter bank structure. Compared to the typical filter bank of DWT (Figure 2.4), it replaces one tree by two so that it is called dual-tree. In the dual-tree framework, Tree A and Tree B are both real DWT, they work together as a complex wavelet as Tree A gives the real part of transform while Tree B gives the imaginary part [32]. In other words, the whole complex dual-tree wavelet transform is only made by two pure real DWT that work in parallel. Moreover, the two real wavelet transforms both satisfy the perfect reconstruction conditions and they should also satisfy Hilbert transform. In the dual-tree, the imaginary part is derived from the real tree by the Hilbert transform. The Q-shift dual-tree filters are designed for the dual-tree complex wavelet by Kingsbury. Since all filters beyond level one are designed to have a group delay of one quarter sample, it is so called Q-shift dual-tree filter. Besides, because the non symmetric property of the filter coefficients, it makes designing perfect-reconstruction filter sets possible [32]. In summary, the properties of PR, the approximate shift invariance and good directional selectivity give the DCWT inherent capacity for image coding. More details about Q-shift

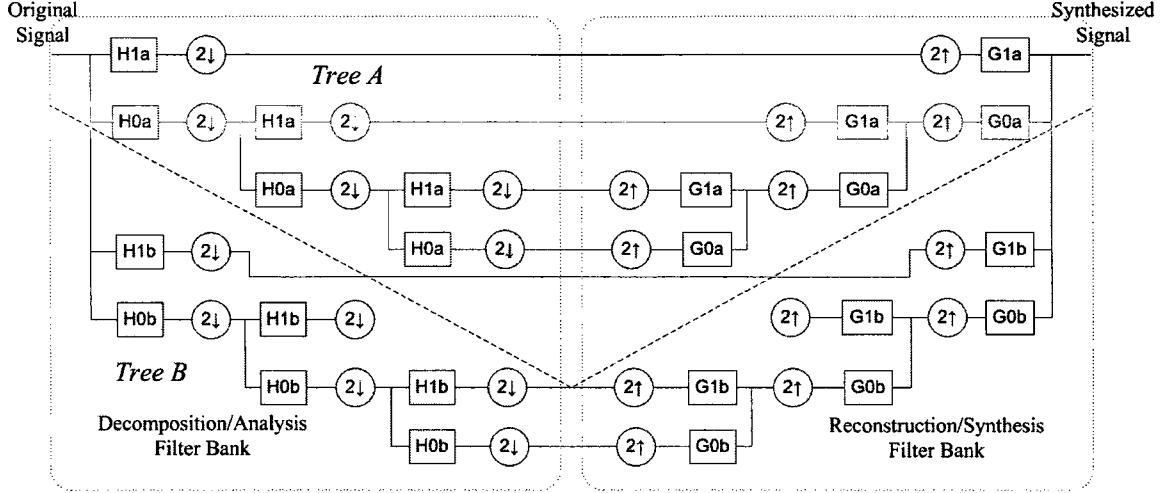


Figure 2.6: Dual-tree filter bank structure in DCWT.

filter design, mathematical derivation and verification of the DCWT's properties can be found in [32] [6] [23].

2.3.2 2D dual-tree complex wavelet transform

When the dual-tree complex wavelet is extended to the two dimensional case, it not only retains all of the properties in 1D, but also provides more orientations which benefit image processing. As the edges in the image can be regarded as oriented singularities, theoretically, the more orientations we have, the better analyzing and processing result we get. In order to show the dual-tree complex structure more clearly, the real 2D DWT is discussed first. The real 2D wavelet transform is usually implemented by performing 1D DWT two times along the rows and columns separately. It is characterized by three level wavelets:

$$\psi_1(x, y) = \phi(x)\psi(y) \text{ (DWT LH orientation)}, \quad (2.10)$$

$$\psi_2(x, y) = \psi(x)\phi(y) \text{ (DWT HL orientation)}, \quad (2.11)$$

$$\psi_3(x, y) = \psi(x)\psi(y) \text{ (DWT HH orientation)}. \quad (2.12)$$

In the above Equations (2.10, 2.11, 2.12), $\psi()$ and $\phi()$ denote the lowpass filter and high pass filters (usually band pass instead of high pass in practice). The LH orientation is the product of $\psi()$ along the first dimension and $\phi()$ along the second dimension [23]. In the

vertical or horizontal orientations like LH and HH, the real DWT performs well. However, it suffers from checkerboard artifacts which are showing in Figure 2.8(c) in diagonal orientations. Given a real wavelet $\psi(x, y)$ which is composed of $\psi(x)\psi(y)$ as Equation 2.12, the Fourier spectrum of $\psi(x, y)$ is illustrated by Figure 2.7(a). Because $\psi(x)$ is a real function, its spectrum must be two-sided so that the spectrum is contained in four corners after $\psi(x)$ multiplied by $\psi(y)$. This phenomenon occurs both at 45° and -45° orientations and makes separable DWT fail to isolate the orientations in the spatial domain [23].

This situation is improved with the dual-tree complex wavelet. First, it doubles the orientations from 3 to 6 so that it removes the checkerboard in diagonal orientations. Equation 2.13, 2.14 and 2.15 define a dual tree structure of separable 2D wavelet bases.

$$\psi_{1,1}(x, y) = \phi_{Ha}(x)\psi_{Ha}(y), \quad \psi_{2,1}(x, y) = \phi_{Hb}(x)\psi_{Hb}(y) \text{ (DCWT LH)} \quad (2.13)$$

$$\psi_{1,2}(x, y) = \psi_{Ha}(x)\phi_{Ha}(y), \quad \psi_{2,2}(x, y) = \psi_{Hb}(x)\phi_{Hb}(y) \text{ (DCWT HL)} \quad (2.14)$$

$$\psi_{1,3}(x, y) = \psi_{Ha}(x)\psi_{Ha}(y), \quad \psi_{2,3}(x, y) = \psi_{Hb}(x)\psi_{Hb}(y) \text{ (DCWT HH)} \quad (2.15)$$

In the frequency domain, the dual-tree complex wavelet only has a signal-sided spectrum as shown in Figure 2.7(b), therefore the spectrum of complex 2D DWT covers one corner instead four corners in the real 2D DWT. Hence, the checkerboard artifacts will not occur in the diagonal orientations. As shown in Figure 2.8, the DCWT provides 6 directionally selective orientations which are 75° , 45° , 15° , -15° , -45° and -75° . Compared with the DCWT, the DWT only provides 3 orientations at 90° , 45° and 0° , but 45° is not a good direction because of the checkerboard artifacts.

For a better illustration of the advantage of the DCWT in directional selectivity, we plot Figure 2.9. By using a test image with a star made of numerous lines with different degrees, when it goes into real wavelet transform domain, the most significant coefficients only occur in the vertical or horizontal directions. However in the DCWT, the most significant coefficients are shown along many directions in different subbands. For signal processing, the DCWT gives us more flexibility and possibility to deal with different information in different subbands.

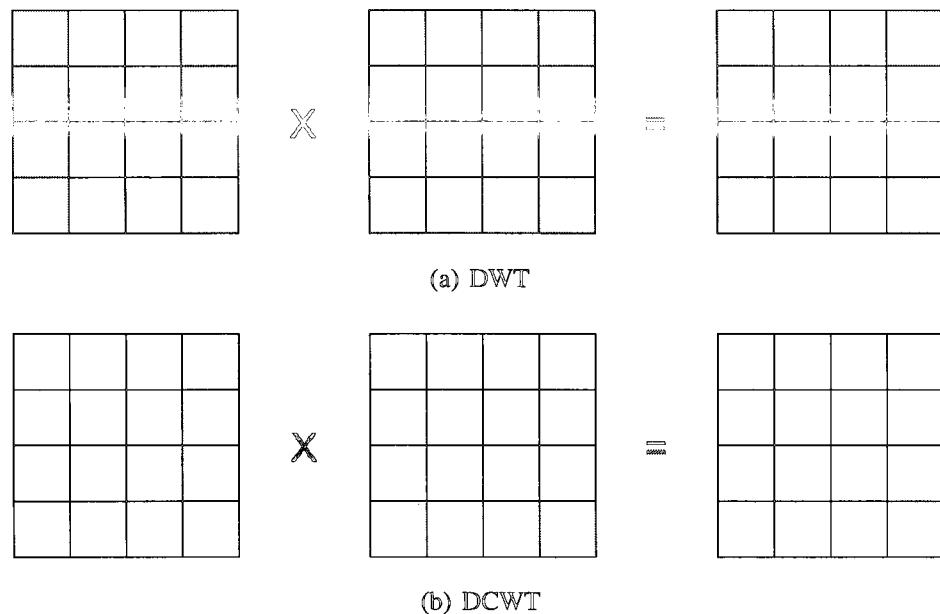


Figure 2.7: 2D Fourier spectrum for DWT and DCWT [23].

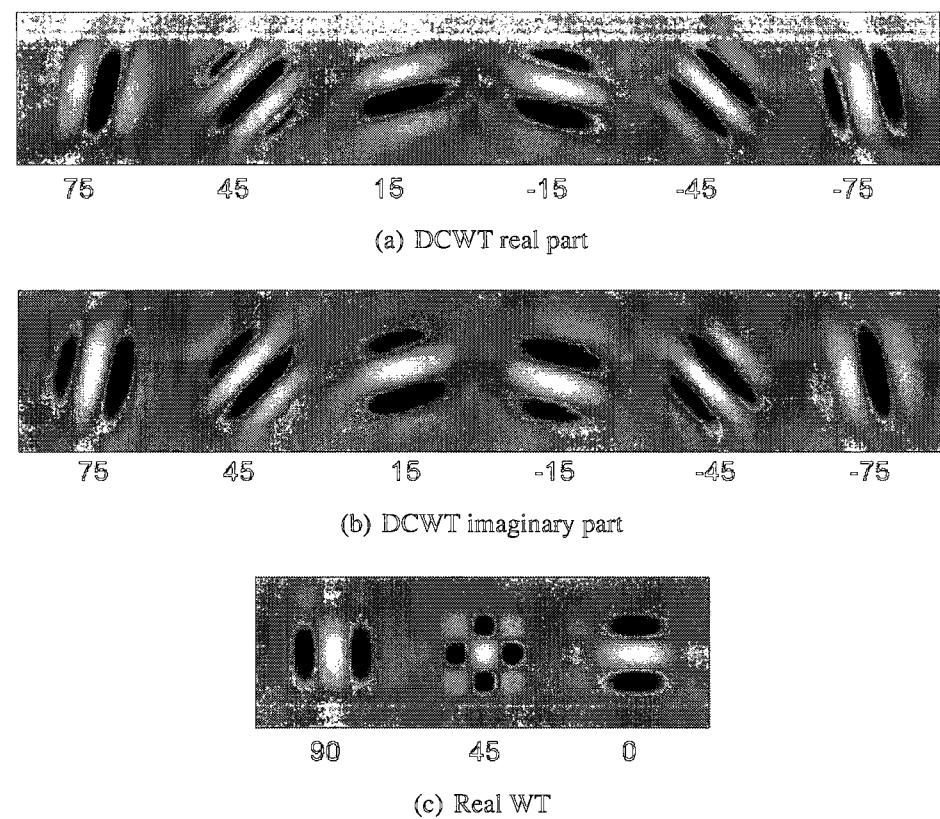
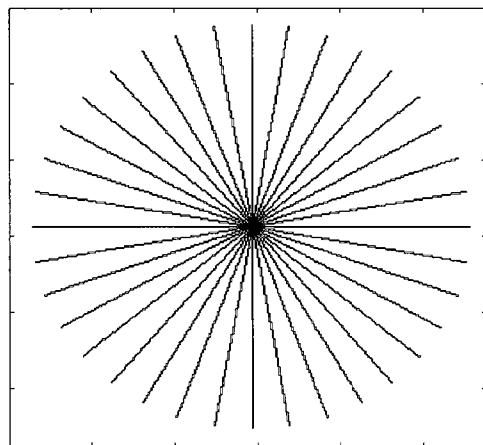
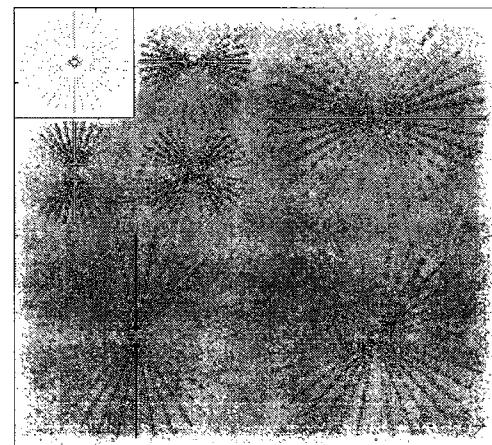


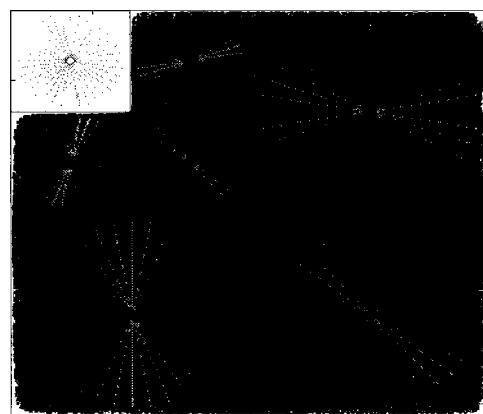
Figure 2.8: 2D impulse response of DWT and DCWT [6].



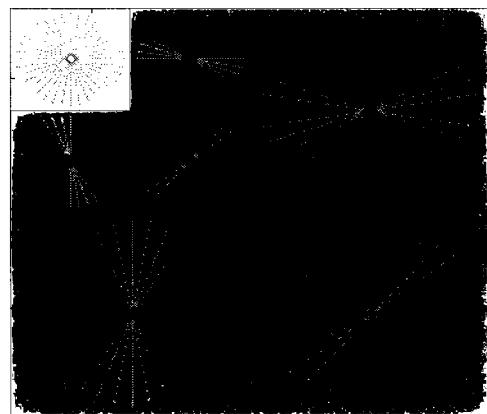
(a) Original Image



(b) Image in real DWT domain



(c) Image in DCWT domain (Tree A)



(d) Image in DCWT domain (Tree B)

Figure 2.9: Image in 2D DCWT domain and real DWT domain.

2.3.3 Dual-tree complex wavelet transform based image coding

With its good directional selectivity, near shift invariance and guarantee in PR, the DCWT could be a good transform candidate in image coding systems. From 1998 to 2008, many works were done to investigate the possibility of DCWT or its derivative on image coding [34] [35] [27] [26]. Generally, the DCWT works as a transformer and the subband coder works as quantizer in the DCWT based image coding system. Since there are numerous subband coding methods that are ready for wavelet based transform image coding system, investigations are performed to determine the most suitable for DCWT. The investigation on DCWT involves subband coding and the DCWT itself. For example in [26], SPIHT, EBCOT [36] and TCE [37] are tested with DCWT to evaluate which one has the best result. They also tried to use anisotropic DCWT instead of isotropic DCWT in image coding system. In [27], the authors tested the performance of BISK [38] in DCWT based image coding systems. Before all works mentioned above are done, one of the most significant work is Kingsbury's noise shaping algorithm, which shapes the coefficients in DCWT domain [3] [4]. Noise shaping helps the DCWT to achieve sparsity by applying iteration projection theories. The noise shaping algorithm is developed based on the overcomplete structure of DCWT and iteration feedback process. Since it has good performance, it makes the DCWT a powerful candidate to apply in image coding system.

2.4 Human visual system

“The HVS is a system by which a human observer views, interprets and responds to visual stimuli.” [7] Since the image is finally assessed by human observers, it is important to understand how the HVS works and what properties of the HVS would be helpful in image coding. The study of the HVS involves complicated physiological and psychophysical experiments [39] [40], which are beyond the scope of this thesis. But we still felt that it is necessary to briefly discuss some important properties: the multi-channel structure, the contrast sensitivity and the masking effect of the HVS.

The human brain processes visual information by decomposing it into different channels, which are known as “perceptual channels”. Thus, the HVS has a multi-channel structure. This conclusion is made by electro-physiological and psychophysical experiments [41] [42], which indicate that the neurons respond to stimulus frequencies in a band-pass way. In order to simulate the multi-channel structure of HVS, several transforms were developed such as the cortex transform and perceptual sub-band decomposition (PSD),

with details found in [43] [44]. No matter the complexity of the multi-channel structure of the HVS, using the wavelet transform to process images may be considered to be the most natural way to simulate the property since the wavelet transform is inherently implemented by multi-channel filter banks.

The HVS has nonlinear contrast sensitivity, which means the response of the eye to changes of intensity of illumination is nonlinear. In [45], it states that the sensitivity of human eyes to discriminate between difference in intensity depends on the differences itself and the level of intensity. Based on that, many of contrast sensitivity functions (CSF) have been developed to incorporate into image codecs [46] [47].

Besides its multi-channel structure and contrast sensitivity, the HVS also has a masking effect. For an image, when a region has a high activity content, it is less sensitive to errors than a region with flat content. However, it is not so easy to quantify the amount of masking due to variations in different viewing distances, sampling frequencies and picture content. Some masking functions developed for image codecs can be found in [48] [49].

Chapter 3

Noise Shaping - Balance the Coefficient Expansion

One of the most important reasons for the success of the DWT in image coding is that it is optimally sparse for singular signals. For example, the edges and ridges of an image in the wavelet domain often have large coefficients while flat regions have small coefficients. That is because a wavelet oscillates locally so that the singularity signal overlaps, which leads to large wavelet coefficients. Moreover, the sparsity of the DWT also simplifies subsequent processes such as compression and denoising by simple thresholding [23]. As one number of wavelet family, keeping the sparsity of DCWT is a primary task. After successfully developing the DCWT in 2002, Kingsbury and Reeves proposed noise shaping to achieve the sparsity in DCWT which is based on: 1) feedback structure that is similar to any noise shaping coder and 2) overcomplete transform projections. Moreover, noise shaping attempts to set the small coefficients to zero and modify the large coefficients to compensate as much as possible for the loss of the small ones, while still producing a good approximation to the original image [4]. It also could be regarded as a way to balance the coefficient expansion introduced by the overcomplete DCWT. Since the importance of noise shaping in DCWT based image coding system and video coding systems has previously been shown [4] [27] [28], a lot of investigations and experiments conducted in this thesis focus on it. In this chapter, the theory and analysis of noise shaping is discussed first.

3.1 Noise shaping fundamentals

Noise is a type of unwanted signal which exists widely in nature and in electronics in acoustic or visual forms. In the development of DSP, the first occurred “manmade” noise

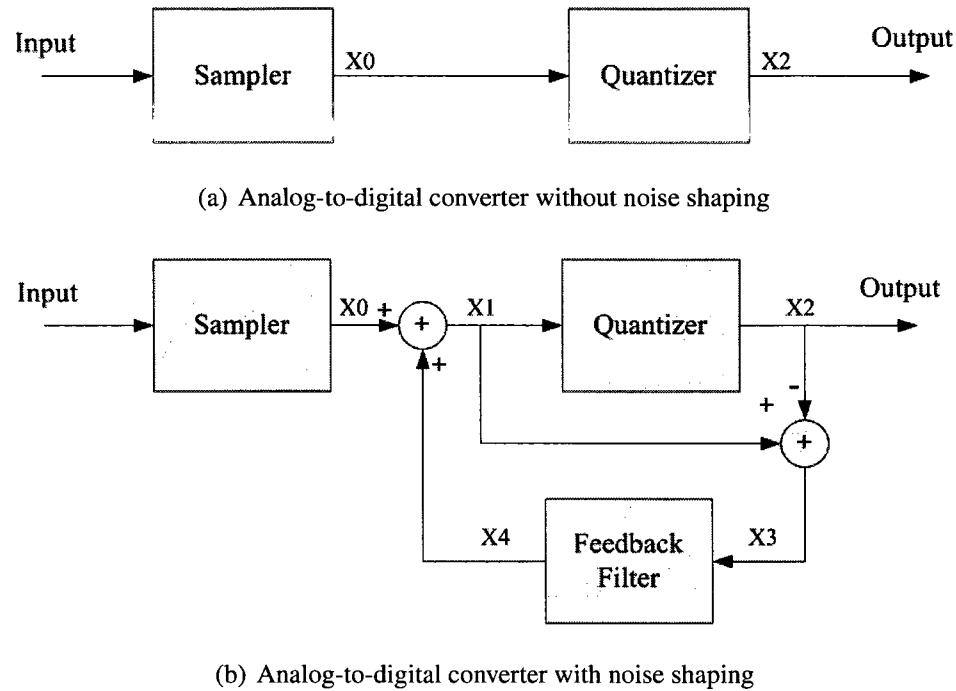


Figure 3.1: Diagram of noise shaping in A/D applications.

was quantizing noise by the analog-to-digital converter (A/D). In the 1960s or earlier, researchers found that the spectrum of noise could be shifted by a simple feedback. Based on that fact, unwanted quantization noise could be moved to another range of the spectrum that is different from the range of the wanted signals. Thus, the noise could be attenuated by a simple filtering processing in the noise spectrum. This type of method using feedback to shift the noise could generally be called the noise shaping algorithm.

3.1.1 Noise shaping by feedback and oversampling

A core part of noise shaping is the feedback between input and output of the quantizer which is shown in Figure 3.1(b) and was first used in A/D system [50]. Compared with Figure 3.1(a), the input before the quantizer is calculated as original input plus a value from the feedback as Equation 3.1:

$$X_1 = X_0 + X_4 \quad (3.1)$$

A simple example described in the following is adopted from [50]. It demonstrates that

Table 3.1: A/D quantization table example

Before quantization	After quantization
[0, 1)	0
[1, 2)	1
[2, 3)	2

the noise spectrum can be shifted in the frequency domain. Suppose we have a quantization table as shown in Table 3.1, a constant input of 1.5 and the sample rate $f_s = 1$. In Figure 3.1(b), with a simple feedback filter whose output only has one sample delay as input ($x_4(n) = x_3(n - 1)$), then $x_1(n)$ will be $x_1(n) = x_0(n) + x_4(n) = x_0(n) + x_3(n - 1) = x_0(n) + x_1(n - 1) - x_2(n - 1)$. The derivation of the output sample is included in Table 3.3. Compared to Table 3.2, which applies simple quantization without feedback as Figure 3.1(a), although the magnitude of the quanatization error are the same at every sample instant, the sign of the error alternates by using feedback. Therefore, the quantization error (noise) has been shifted in the frequency domain from DC to $2 \times f_s$, which suggests that the noise could be eliminated by an averaging operation or low-pass filter. Besides shifting the noise spectrum, the noise power can also be reduced by working in over sampling(eg. 95% noise error power decreases by increasing samping rate by 25%) [50]. Similar derivation that supports this conclusion can also be found in [51]. The quantization noise reduction by feedback and oversampling technique is also successfully applied in high order noise shaping coders [52] and oversample filter banks [53].

3.1.2 Noise shaping in dual-tree complex wavelet transform

Like the feedback structure of noise shaping in oversampled filter banks, Kingsbury and Reeves proposed the iterative projection-based noise shaping [4] in overcomplete DCWT to shape the large coefficients for better sparsity in the DCWT domain. Instead of oversampling in the 1D case, they used an overcomplete dual-tree structure as well as projection feedback. Figure 3.2 shows the structure of noise shaping used in DCWT. Please note that although the original purpose of noise shaping is to suppress the noise, the goal of noise shaping used here is to modify the retained coefficients so that they can compensate for the

Table 3.2: A/D quantization result without noise shaping

n	x_0 (Input)	x_2 (Output)	error
0	1.5	1	0.5
1	1.5	1	0.5
2	1.5	1	0.5
3	1.5	1	0.5
4	1.5	1	0.5
5	1.5	1	0.5
6	1.5	1	0.5
7	1.5	1	0.5
8	1.5	1	0.5
9	1.5	1	0.5
10	1.5	1	0.5

Table 3.3: A/D quantization result with noise shaping

n	x_0 (Input)	x_1	x_2 (Output)	error
0	1.5	1.5	1	0.5
1	1.5	2	2	-0.5
2	1.5	1.5	1	0.5
3	1.5	2	2	-0.5
4	1.5	1.5	1	0.5
5	1.5	2	2	-0.5
6	1.5	1.5	1	0.5
7	1.5	2	2	-0.5
8	1.5	1.5	1	0.5
9	1.5	2	2	-0.5
10	1.5	1.5	1	0.5

loss of the discarded coefficients while still producing a good approximation to the original image [4]. The process of noise shaping is described as follows: first, the original image X is transform to DCWT domain as Y_0 ; after the thresholding, the retained coefficients \hat{Y}_i are projected back to the image domain and are subtracted from the original signal to obtain the error signal e_i ; then, the error is multiplied by an energy gain factor k and is transformed again into the DCWT domain as w_i ; at last, w_i is added to \hat{Y}_i to form Y_i and feeds back to the thresholding unit. By repeating the same process, the retained coefficients will become larger which indicates that a better sparsity of DCWT is achieved. Experimental results shows that the DCWT with noise shaping usually gets more than 4 dB higher in average PSNR at different number of retained coefficients than the system without noise shaping. Figure 3.3 shows the difference of keeping the 10000 largest coefficients in DWT, DCWT without noise shaping and DCWT with noise shaping. Figure 3.3(a) shows the DWT with the PSNR of 27.21dB; Figure 3.3(c) shows the DCWT without noise shaping with the PSNR of 25.66 dB, and Figure 3.3(e) shows the DCWT using noise shaping with the PSNR of 31.27 dB. Figure 3.3(b), (d) and (f) show histograms of (a), (c) and (e). In these figures, the x axis denotes the range of magnitude of the coefficients and the y axis

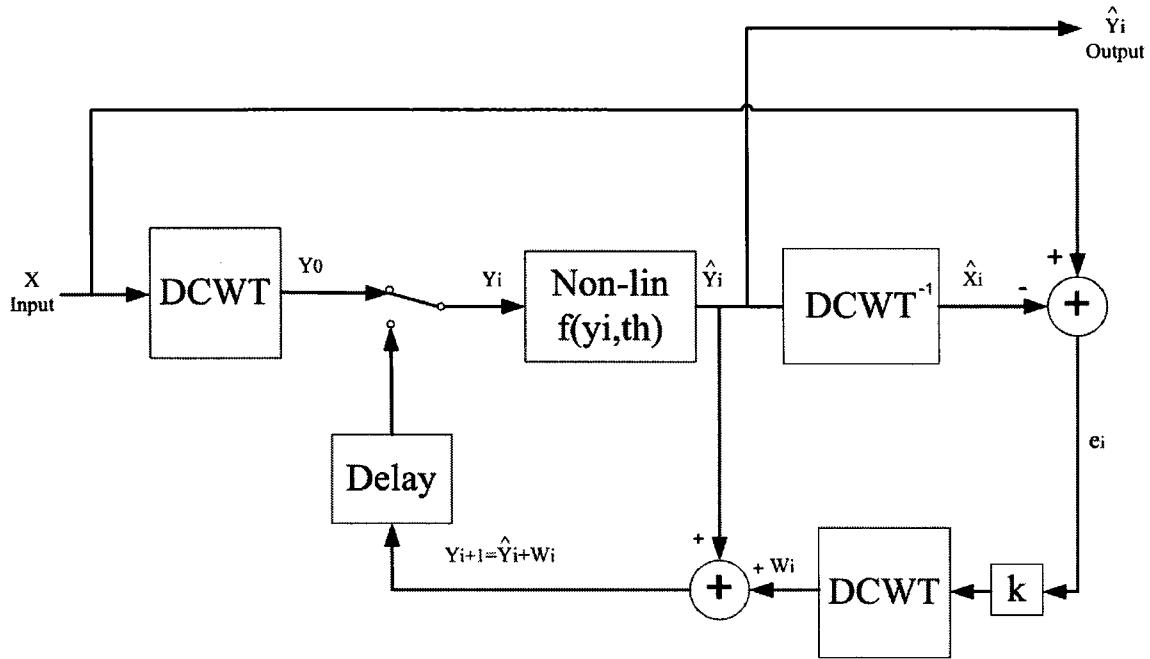


Figure 3.2: Diagram of noise shaping in DCWT for image coding [4].

shows the number of coefficients that fall in that range. If we compare (d) and (f) in Figure 3.3, we can easily find that there are more coefficients located in a higher magnitude range after noise shaping. This phenomenon could be a factor of noise shaping's good performance with the DCWT.

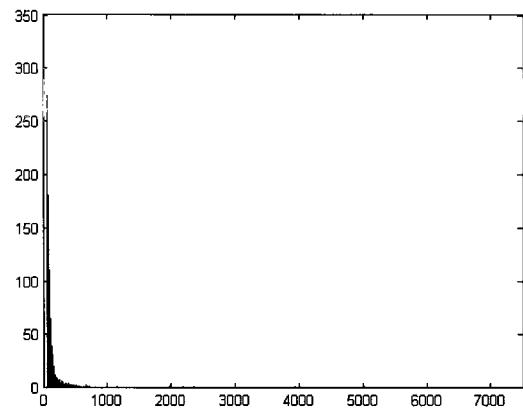
3.1.3 Transform projections and convergence analysis

Following [4], the projection is from a higher dimensional transform space into lower dimensional transform space for overcomplete transforms. For example, the reverse transform of the DCWT, which is from $4N$ coefficients to N images pixels in the 2D case, is the projection process. Inside the $4N$ redundancy wavelet space, the authors classified an N -dimensional range space and an orthogonal $3N$ -dimensional null space. According to the transform projection theory, the movement within the $3N$ null space may not produce change in the wavelet space. So, there is a possibility to shape the coefficients in the null space without changing the projection output.

Assume that A is a transform matrix, R is a reconstruction matrix and \hat{R} is the pseudo-inverse of A , under the perfect reconstruction condition $R = \hat{R}$. In Figure 3.2, y_{i+1} could be



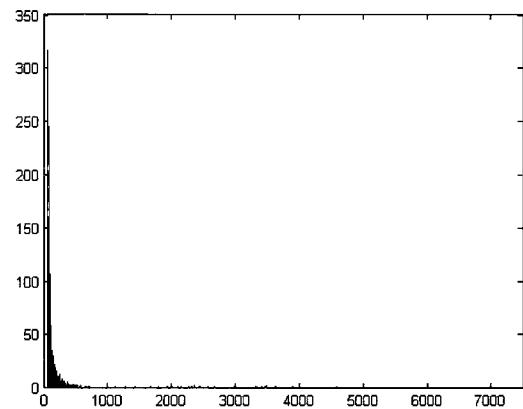
(a) DWT (PSNR=27.21dB)



(b) Histogram plot of the non-zero coefficients



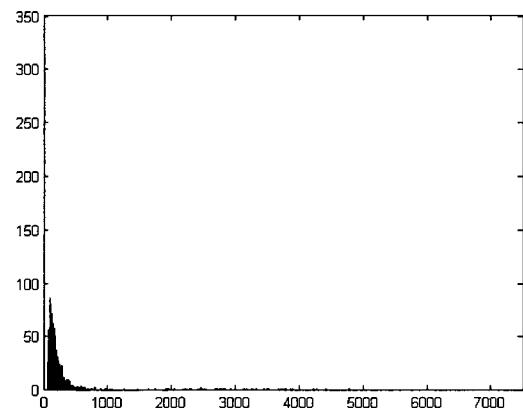
(c) DCWT Without Noise Shaping (PSNR=25.66dB)



(d) Histogram plot of the non-zero coefficients



(e) DCWT With Noise Shaping (PSNR=31.27dB)



(f) Histogram plot of the non-zero coefficients

Figure 3.3: Noise shaping performance with 10000 largest coefficients kept

calculated as

$$y_{i+1} = \hat{y}_i + w_i = \hat{y}_i + kA(x - \hat{R}\hat{y}_i). \quad (3.2)$$

Let S be the range of space of A , then the projection operation onto S is $P^S = AR$, P^\perp is the null space of A . Also, because $Ax = y_0$, Equation 3.2 can be rewritten as

$$\begin{aligned} y_{i+1} &= \hat{y}_i + ky_0 - kP^S\hat{y}_i = ky_0 + (I - kP^S)\hat{y}_i \\ &= ky_0 + P^\perp\hat{y}_i \quad \text{if } k=1. \end{aligned} \quad (3.3)$$

From the above equation, every new y equals the original y_0 plus null space components. Compared with Equation 3.1, x_1 is the sum of x_0 , which is from the input and x_4 , which is from the feedback. The expression of Equation 3.3 could also be regarded as an item y_0 from the input, and an item $P^\perp\hat{y}_i$ from the feedback. Therefore, even the structure of noise shaping in the DCWT is not identical to general noise shaping format as in Figure 3.1. The structure shown in Figure 3.2 still satisfies the general noise shaping format after the deviation of transform projection from Equation 3.2 to Equation 3.3.

Other than the error spectrum analysis in A/D [50], the error analysis of noise shaping in DCWT is much more complicated in the spatial overcomplete transform domain. The analysis tools in [4] use projections onto convex sets (POCS) as analysis tools. Although it may not be the best way to interpret the noise shaping in the spatial overcomplete transform domain, it is still necessary to describe POCS here as the only analysis tool used in the current study.

The POCS is mainly used in image processing techniques such as super-resolution image synthesis and compression artifacts removal. POCS can be done in the time domain or transform domain. Loosely speaking, POCS could be described as a computational approach to find the intersection of elements by a number of convex sets from an arbitrary point via a number of projections or the closest pair of points in the two convex sets. Figure 3.4, which is adapted from [54], shows the POCS process. In image processing, POCS is usually used to recover some unknown image from every piece of available knowledge. Since the intersection of all sets satisfies all available knowledge, it makes recovering unknown images from the intersection possible. Because we start from an arbitrary point, using iteration projections to find the intersection is an obvious and a common choice. The

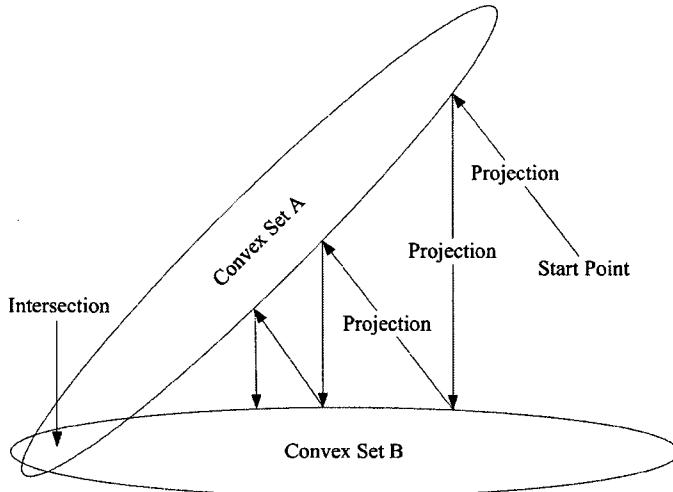


Figure 3.4: Projections onto convex sets [54].

POCS involves two steps in general: the first step is to define the convex sets and the second step is to derive the projection [55].

Since there are two projections in Figure 3.2, the two convex sets C_1 and C_2 are defined naturally on the two projection processes by [4]. The first convex set C_1 is defined as follows: first, generating a coefficient retained mask vector m_i of zeros and ones in the center-clipper unit of Figure 3.2; second, all y_i are multiplied by m_i (element by element) to get \hat{y}_i . During these two steps there is a projection P_1 from the center-clipper of Figure 3.2, which is from y_i onto the convex set C_1 of all non-zero elements which are selected by mask m_i . As a result, $\hat{y}_i = P_1 y_i$. The second convex set C_2 is defined in the remaining parts of the loop in Figure 3.2, where the projection P_2 occurs from \hat{y}_i to $\hat{y}_{i+1} = P_2 \hat{y}_i$ if $k = 1$. In this projection process, convex set C_2 is defined by the set of all range-space components of $y_0 = Ax$. Until now, the first step of POCS is done by defining two sets C_1 and C_2 . From the POCS point of view, the noise shaping here is to find the closest or the intersected part of these two sets by iteration projections. As we understand, the first convex set C_1 here represents the capability of the remaining coefficients, which could approximate all the coefficients. The second convex set C_2 represents the $1N$ “actual” information which is hidden in the $4N$ redundant overcomplete 2D DCWT space. Hence, if the two sets are closer, the “shaped” \hat{y}_i will approximate the original image better after going back to the time domain. Please also note that in [4], the k used in the actual experiments is larger than 1 in order to compensate the energy loss around the loop due to the projection process. From Equation 3.3, the range space components of \hat{y}_i have a gain of $(1 - k)$ and it should

be inside the unit circle for stability so that ($0 < k < 2$). In Kingsbury's experiments, k is set equal to 1.8 which results in an average 0.3 dB increase in PSNR. We tested different values of k in our experiments and the obtained similar results as Kingsbury. Around $k = 2$, the PSNR achieve a maximum . In cases where $k > 2.1$, the unstable stitution occured in our tests and the PSNR drops dramatically. Therefore, we follow Kingsbury and keep $k = 1.8$ in all our following experiments. As Figure 3.4 illustrates, the POCS is a process to find the intersection or closest distance by iteration projection. It could be understood as an convergence process. In the theory of POCS, the convergence speed reaches the maximum when the sets C_1 and C_2 are fixed numbers. If C_1 and C_2 are fixed, the mask vector m_i should also be fixed in every iteration. In Kingsbury and Reeves' experiments [4], they found that the maximum convergence speed is achieved by a variable size mask m_i by using hard threshold technique. In our experiments, we got similar results as Kingsburys which is shown in Figure 3.5 of Section 3.2.1.

3.2 Coefficients selection by coefficient magnitude

To our knowledge, Kingsbury's experiments [4] may be the first time that the concept of noise shaping is used in overcomplete transform domain. Hence the analysis tool POCS adopted here to explain the noise shaping process is also quite new and may need more precise study and derivation. A lot of experiments presented here are attempting to verify if this analysis tool is suitable for the transform domain noise shaping. First, we will describe hard thresholding and Wiener soft thresholding experiments, which was originally mentioned in [4], with some extensions by ourselves.

3.2.1 Hard thresholding

Hard thresholding is the simplest useful way to suppress the coefficients below the threshold. The threshold is given by ¹

$$\hat{y} = \begin{cases} 0 & \text{if } |y| < th \\ y & \text{otherwise.} \end{cases}$$

In the theory of POCS, the fixed number of retained coefficients at every iteration

¹Please note that the y here are complex numbers which is contributed by Tree A and Tree B, therefore, the condition in the equation is to compare the magnitudes of the y and threshold.

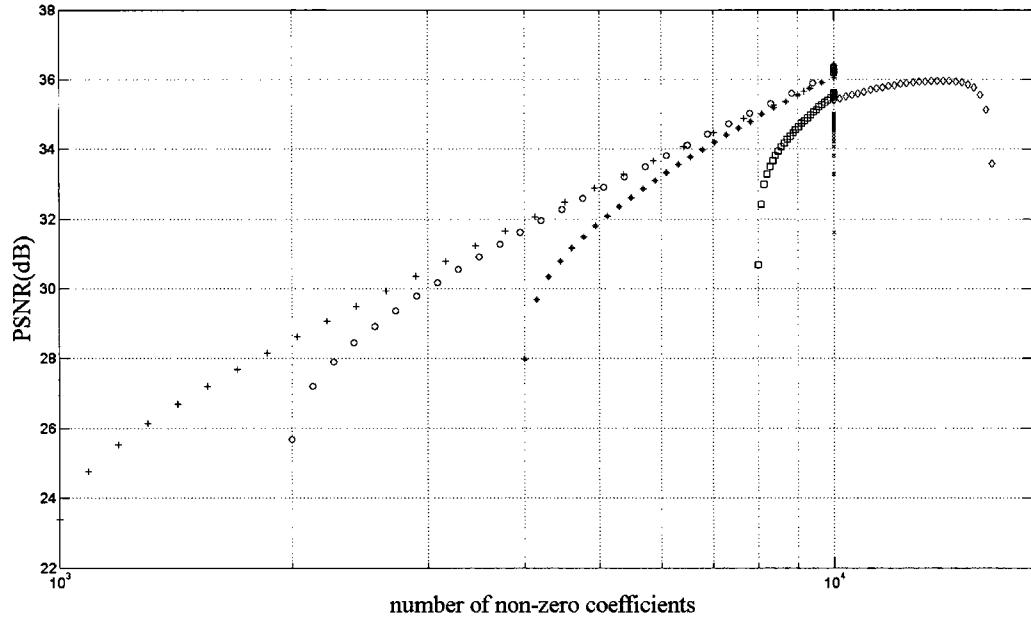


Figure 3.5: PSNR plot with different start number of nonzero coefficients for hard threshold noise shaping.

(means fixed mask m_i for all iterations) will result in fast converge. However, the experiment show a different result. We used the 512×512 “lena” as the test image and the DCWT decomposition level is set to 5. The iteration number is fixed to 30 in all experiments in this thesis for a moderate computation load. Figure 3.5 is the resulting plot from the experiment. There were 5 groups of experiments in the figure, which all kept 10000 non-zero coefficients in the output. One group of experiments was to fix the number of retained coefficients (10000) in every iterations, shown as a “cross” sign in the figure. While others started at a initial retained coefficient of 1000 (“+” sign), 2000 (“circle” sign), 4000 (“*” sign), 8000 (“square” sign) and 16000 (“diamond” sign). The retained coefficients were increased or decreased over the first 26 iterations towards the finial 10000 coefficients in a geometric series. Then the following four iterations were set to 10000 non-zero coefficients for final convergence. As the actual results show, the fixed number of non-zero coefficients obtained the worst PSNR results in all test groups. The maximum result was from the test with initial retained coefficients of 2000, and there was more than 1 dB gap between the maximum and the minimum.

Why are the actual test results different from the POCS theory? Kingsbury gave the

following argument: If the mask m_i is fixed in every iteration and always equals m_0 at convergence, $y_{i+1} \simeq \hat{y}_i$, then the transform domain loop error is $y_i - \hat{y}_i \simeq y_{i+1} - \hat{y}_i = w_i = Ae_i$ which is fully located in the range space. If the mask is chosen with a better value $m_L \neq m_0$ from the y_L , it will result in a smaller image domain error e_i because: 1) $e_i = R(y_i - \hat{y}_i)$, and low initial value leads to less difference of $y_i - \hat{y}_i$ so that it also decreases e_i ; 2) the new $y_i - \hat{y}_i$ is likely moving out of the range space so that the error will be small. Therefore, modifying the mask at every iteration will reduce the error in the loop and achieve fast convergence [4].

As the experimental result from Figure 3.5 shows, when keeping 10000 coefficients of the “lena” image, the best initial retained coefficients is 2000. And then in every iteration the coefficients are added gradually until it reaches 10000. Now the question is: for a fixed 30 iterations, is the initial retained coefficients of 2000 suitable for other final retained coefficients other than 10000? And is it also good for other images? So we tested different images with different non-zero retained coefficients in different initial values. The images we tested are the 512×512 “lena”, “barbara” and “boat” images; we chose 5000, 10000, 20000 and 40000 as the retained coefficients and 1000, 2000, 4000, 8000 as initial coefficients. From Table 3.4, the results suggest that the initial value should be adjusted along with the number of kept coefficients to obtain a better PSNR.

3.2.2 Thresholding by Wiener denoising function

Since various soft thresholding techniques have success in image denoising and compression [56] [57], a natural thought is if the soft thresholding technique could also benefit the noise shaping thresholding process? Following Kingsbury’s experiment, we also tested the performance of Wiener denoising function comparing to hard thresholding in noise shaping.

$$\hat{y} = \begin{cases} 0 & \text{if } |y| < th \\ y \frac{|y|^2 - th^2}{|y|^2} & \text{otherwise} \end{cases}$$

The above equation shows the Wiener thresholding function. It adjusts the coefficients around the threshold value to make them much smaller than the original value. Therefore, if y is large but is close to the threshold, then \hat{y} will become small; if $y \gg th$ then $\hat{y} \simeq y$. The solid line of Figure 3.6 shows the Wiener thresholding plot. Compared to hard thresholding,

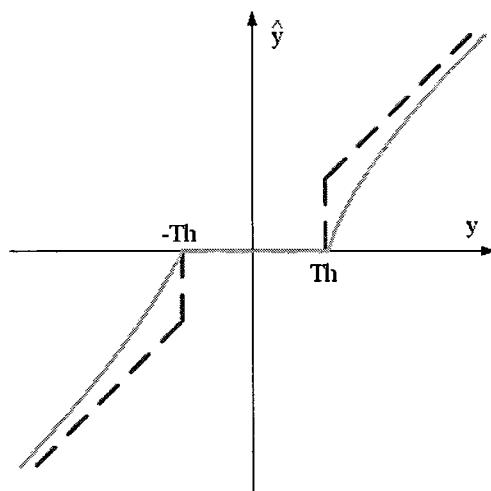
Table 3.4: Image PSNR results of noise shaping by hard thresholding

Non-zero Coefficients		5000				10000			
Initial Coefficients		1000	2000	4000	6000	1000	2000	4000	6000
barbara		27.45	27.20	26.67	26.56	31.12	31.11	30.93	30.78
boat		29.42	29.24	28.70	28.46	32.32	32.39	32.13	31.85
goldhill		29.40	29.25	28.78	28.53	31.84	31.75	31.57	31.31
lena		33.36	33.25	32.62	32.25	36.34	36.38	36.24	35.98

Non-zero Coefficients		15000				20000			
Initial Coefficients		1000	2000	4000	6000	1000	2000	4000	6000
barbara		33.61	33.68	33.66	33.65	35.48	35.57	35.66	35.68
boat		34.13	34.17	34.13	33.99	35.45	35.53	35.54	35.46
goldhill		33.46	33.44	33.34	33.16	34.76	34.77	34.70	34.61
lena		38.05	38.10	38.08	37.96	39.28	39.37	39.37	39.30

which is plotted as the dashed line in the figure, the Wiener threshold smoothes the curve near the threshold to make the output \hat{y} more “continuous” than hard thresholding.

In [4], Kingsbury demonstrated good performance by using mixed hard and soft thresholding. In their experiment, the first 15 iterations used Wiener thresholding and the other 15 used hard thresholding, all with fixed 2400 coefficients retained as the first step of iteration. Then in the following iterations, the number of retained coefficients are increased each time by 2%. The disadvantage of this scenario is obvious: if we need to keep 10000 non-zero coefficients as the input, we shall need more than 100 iterations to do it. So in our experiment, we used a fixed number of retained coefficients in every iteration with the first 25 iterations by Wiener threshold, followed by 5 hard thresholding iterations to stabilize the final converge. The results are shown in Table 3.5 by using same test images. Unfortunately, we did not always obtain an outperformance for Wiener denoising function in terms of PSNR when comparing Table 3.4 and 3.5. However, since 1) Wiener thresholding fixed mask m_i in every iteration to avoid initial retained coefficients setting and 2) the retained coefficients are beneficial to subsequent SPIHT coding than hard thresholding, we decide to adopt Wiener thresholding as a standard threshold unit of classical noise shaping for

**Figure 3.6:** Wiener threshold curve**Table 3.5:** Image PSNR results of noise shaping by Wiener thresholding

Number of non-zero coefficients	5000	10000	15000	20000	25000
barbara	27.44	31.27	33.87	35.03	37.23
boat	29.36	32.28	34.15	35.49	36.56
goldhill	29.35	31.73	33.35	34.64	35.74
lena	33.36	36.30	37.98	39.18	40.17

further studying.

Chapter 4

Perceptual Noise Shaping Framework

In developing a perceptual noise shaping framework, we should first include a perceptual quality metric into classical noise shaping to control the overall image quality. After we decided which perceptual quality metric to use with noise shaping, we also need to find the most effective place where the perceptual quality metric can be introduced.

4.1 Perceptual image quality measurement

A good quality perceptual measure is the first step to control image quality, but pixel-by-pixel based quality metrics, like PSNR, have questionable performance for evaluating image quality. Our approach is to look at SSIM [5] in each iteration of noise shaping and then use it to weight and shape the wavelet coefficients. Compared to PSNR, SSIM is more likely to be considered as a perceptual image quality metric. Due to its low complexity in calculation and perceptual meanings, we decided to adopt it in our perceptual noise shaping framework over other, more computational expensive, perceptual measures.

4.1.1 Structural similarity index

The structural similarity (SSIM) index is a new image quality algorithm developed by Wang et al. [5]. Compared to other techniques which are trying to quantify the early steps used by the HVS in processing a perceived scene, it considers the overall perceptual system especially for structural information since it is believed that HVS is highly adapted to extract structural information from the visual field [9].

Traditional image quality assessment algorithms such as MSE use an absolute error difference as the criterion to assess the distorted image. MSE is good to “find” some types of

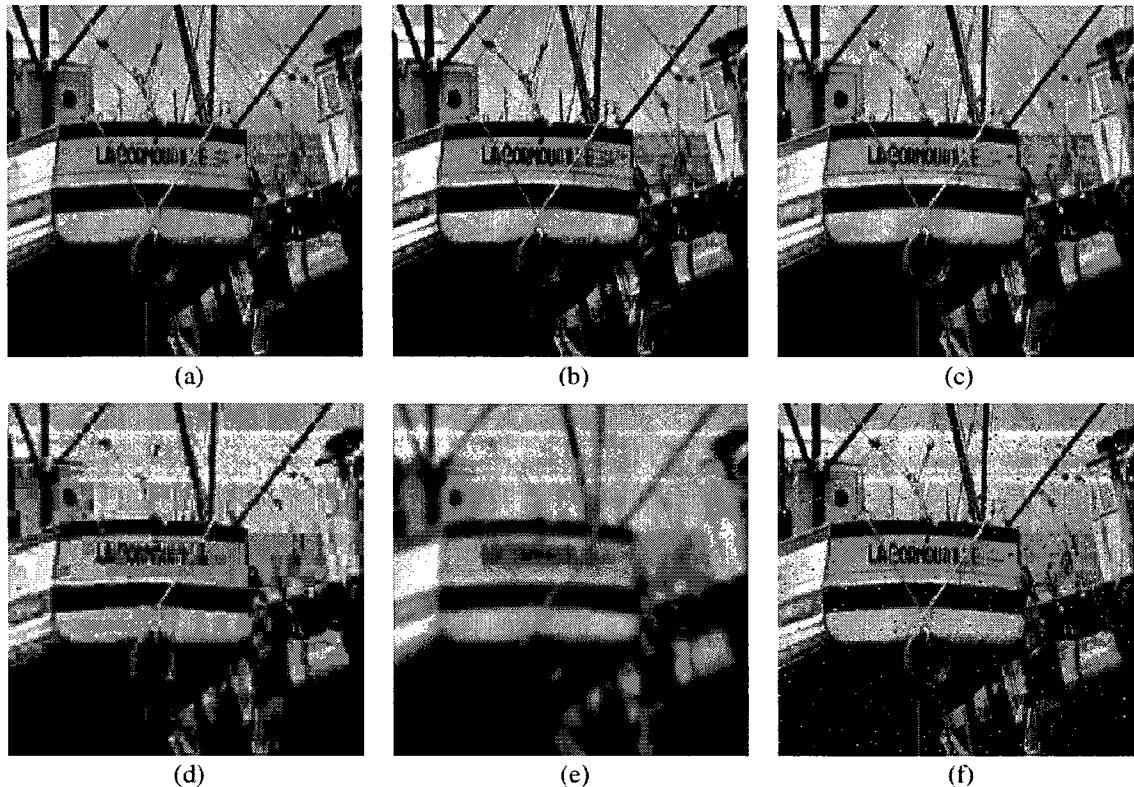


Figure 4.1: Comparison of boat images with different types of distortions, all with MSE = 210. (a) Original image (8 bits/pixel; cropped from 512 by 512 for visibility). (b) Contrast-stretched image, mean SSIM (MSSIM) = 0.9168. (c) Mean-shifted image, MSSIM = 0.9900. (d) JPEG compressed image, MSSIM = 0.6949. (e) Blurred image, MSSIM = 0.7052. (f) Salt-and-pepper impulsive noise contaminated image, MSSIM = 0.7748 (taken from [5]).

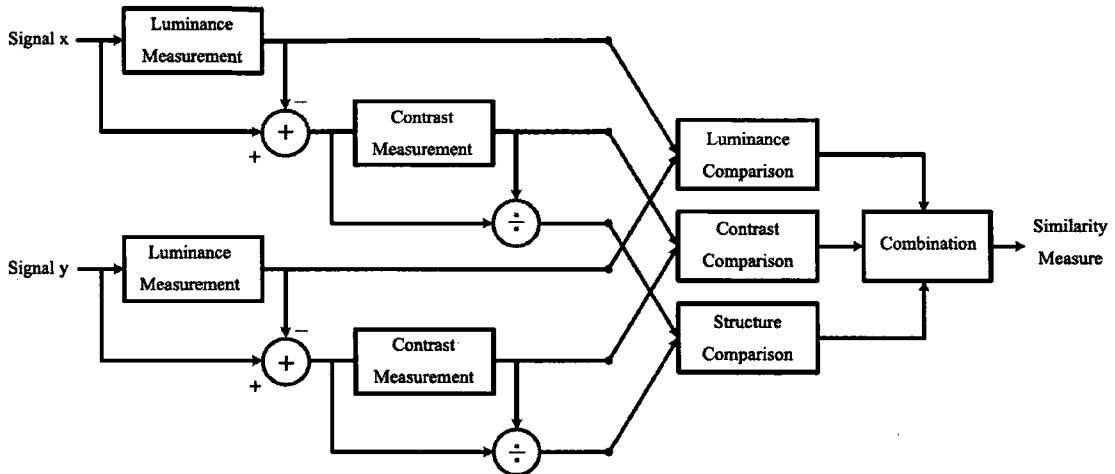


Figure 4.2: Diagram of the SSIM measurement system [5]

image distortions, like blurring and added noise, however, it is over-sensitive to some other types of image distortions like luminance shift artifacts, which do not affect the perceived quality of the overall scene. Figure 4.1 shows examples of quality results of various types of distorted images by using the pixel error based approach MSE and the structure based approach SSIM. The range of SSIM index is between 0 and 1, with 0 being the lowest quality and 1 being the highest quality. From the figure, we can see that for the same value of MSE, it is generally hard to see distortions in image (b) and (c) that are mainly introduced by a luminance shift. As the SSIM index is less sensitive in these images, it provides us a higher image quality value than others, which indicates that utilizing the structure information to assess a scene's quality can reduce or eliminate some of the classical quality assessment problems.

Figure 4.2 illustrates the SSIM structure which is operated in spatial domain. In Wang et al's approach [5], the SSIM measurement system includes three independent comparison units, which measure the scene's luminance attributes, contrast attributes and structure attributes respectively. In natural images, since the luminance and contrast values are not constant, and vary throughout the scene, local luminance and the contrast are used in the SSIM algorithm to better reflect the local change [9].

In the SSIM measurement system [5], the local luminance is calculated first as the local mean of the scene using

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i. \quad (4.1)$$

After the local luminance is moved from the scene, the local contrast is calculated by the standard deviation function

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2}. \quad (4.2)$$

After the local luminance, contrast are defined, the luminance comparison function $l(x, y)$, contrast comparison function $c(x, y)$ and structure comparison function $s(x, y)$ are defined as

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (4.3)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (4.4)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x^2 + \sigma_y^2 + C_3}. \quad (4.5)$$

In the above equations, C_1 , C_2 and C_3 are small stability constants. These equations are known as having incorporated several HVS properties. For example, Weber's law, which is widely used for light adaptation, is incorporated into $l(x, y)$; $c(x, y)$ is less sensitive to the same amount of contrast change $\delta\sigma = \sigma_y - \sigma_x$ in high base contrast than low base contrast which simulates the masking behavior of the HVS ; and the structure similarity is simply and effectively incorporated into $s(x, y)$ by estimating the correlation of two images [9].

Finally, the SSIM index is calculated by combining the above three functions into one as

$$SSIM(x, y) = l(x, y)^\alpha c(x, y)^\beta s(x, y)^\gamma \quad (4.6)$$

where α , β and γ are positive constant weighting exponents. Since SSIM is applied for local image windows, the overall SSIM quality index for the whole image is simply calculated

as the mean of all local SSIM values.

4.1.2 Structural similarity index in dual-tree complex wavelet transform

Spatial domain SSIM is successful in detecting a wide range of image distortions. In [58], the authors mentioned that the HVS is not very sensitive to the image's geometrical distortions (translation, rotation and scaling) if there is only a small amount in the image like image (h)-(l) in Figure 4.3. However, spatial domain SSIM is too sensitive for this type of distortion and may not accurately reflect HVS' properties. By moving from the spatial domain SSIM into complex wavelet domain, the new SSIM index is more robust to such distortions. Examples and comparisons could also be found in Figure 4.3.

In [58], the SSIM in the complex wavelet domain is defined as

$$CW-SSIM(c_x, c_y) = \frac{2 \sum_{i=1}^N |c_{x,i}| |c_{y,i}| + K}{\sum_{i=1}^N |c_{x,i}|^2 + \sum_{i=1}^N |c_{y,i}|^2 + K} \cdot \frac{2 |\sum_{i=1}^N c_{x,i} c_{y,i}^*| + K}{2 \sum_{i=1}^N |c_{x,i} c_{y,i}^*| + K} \quad (4.7)$$

where c^* denotes the complex conjugate of c and $|\cdot|$ denotes the magnitude operator. Similar to the spatial domain SSIM, the index is 1 if the two images are identical. In the equation above, the first part is used to detect luminance and contrast changes and the last part is used to detect phase changes which reflect structural similarity since the image's structure is not changed by constant phase shifts. Since any image changes in the spatial domain can be represented as a magnitude and/or phase change in the complex frequency domain, the SSIM in the complex domain ideally should have the same or even better capabilities to detect image distortions than in the spatial domain.

4.2 Perceptual noise shaping

After deciding to use SSIM as the perceptual image quality metric, where and how to incorporate SSIM into noise shaping is another challenge. We could add SSIM into the thresholding unit of noise shaping to weight the wavelet coefficients, which is similar to the idea in [59]. Another option is to add SSIM in the feedback path of noise shaping to weight the spatial feedback errors. The following sections discuss our analysis of the two



Figure 4.3: Comparison of images with different types of distortions by complex wavelet domain SSIM (a) reference image (8 bits/pixel, assumed to have perfect quality); (b) contrast stretch; (c) mean luminance shift; (d) Gaussian noise contamination; (e) impulsive noise contamination; (f) JPEG compression; (g) blurring; (h) spatial scaling (zooming out); (i) spatial translation (to the right); (j) spatial translation (to the left); (k) rotation (counterclockwise); (l) rotation (clockwise). Note that images (b)-(g) have approximately the same MSE values but drastically different visual quality, which is better predicted by both SSIM and CW-SSIM. Also note that MSE and SSIM are both sensitive to geometrical distortions (images (h)-(l)), which CW-SSIM is robust to (taken from [58]).

options and our experiments with them:

4.2.1 Coefficient selecting by perceptual weighting

Other than the sparsity property in the wavelet domain, there is also a clustering property which is defined as: given a particular wavelet coefficient A within a level, if A is large, the coefficients surrounding A tend to be large as well; if A is small, the coefficients surrounding A tend to be small [60]. If the simple hard threshold that is mentioned in the previous section is used to select coefficients, the retained coefficients will be from the region that contains high-contrast features, and the ones from medium and low contrast features will be discarded. It may cause the visual quality to decrease. Based on it, a new threshold called local energy threshold (LET) was proposed to replace the hard threshold in noise shaping algorithm [59]. That paper inspired us that we may incorporate SSIM into the thresholding process so that the coefficient can be selected in a somewhat perceptual way.

In [9], Yonis went further to evaluate the performance of the SSIM in DCWT domain (CW-SSIM) and found it maintains the performance of the SSIM algorithm while increasing its tolerance to geometric distortion, which is not sensitive to HVS. Based on their previous work, our work to incorporate the perceptual weighting of DCWT coefficients became much more straightforward. Since the CW-SSIM has been done in DCWT domain, we only need to add this CW-SSIM metric into the noise shaping to help select coefficients. Our approach estimates the perceptual threshold function by the CW-SSIM index and coefficient magnitude. As the CW-SSIM needs to compare reference and test signals, in Figure 4.4, we used Y_0 as the reference signal and Y_i as the test signal. In this way, every coefficient has a CW-SSIM index which indicates its “importance” related to perception. Moreover, since the magnitude of coefficients is also important, we decided to keep it as a part of the selection criteria. Hence, our new local threshold function is calculated as

$$Th = M_{(x,y)}^\alpha \cdot CW-SSIM_{(x,y)}^\beta \quad (4.8)$$

where $M_{(x,y)}$ is the magnitude of coefficients; $CW-SSIM_{(x,y)}$ is the dual tree complex wavelet domain SSIM index and α and β is used to control the weights between magnitude and SSIM.

To verify the performance, we compared the results with classical noise shaping using

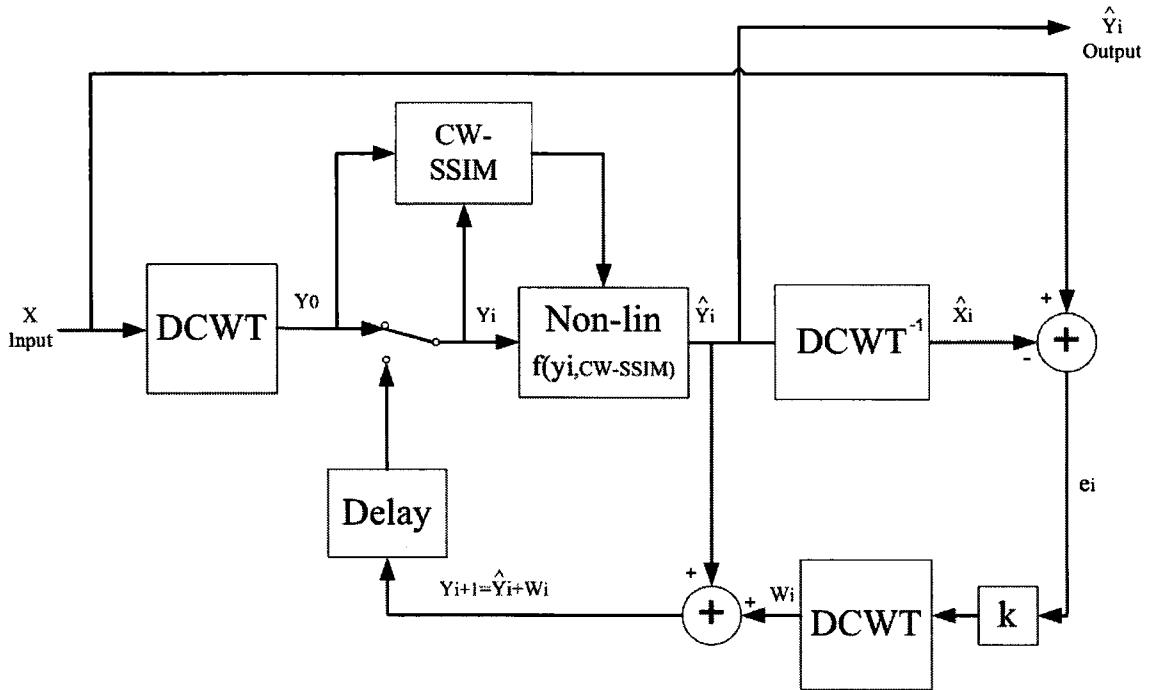


Figure 4.4: Diagram of perceptual threshold for noise shaping

popular test images such as “lena”, “boat”, and “barbara”. The test images are all 8 bit gray scale with size of 512×512 . We adopted Wiener thresholding in our test as standard noise shaping. The parameters are set to be identical in order to fairly compare the two types of noise shaping algorithms. Figure 4.5 shows the comparison of “lena” at 3000 nonzero coefficients: (a) is the classical noise shaping with 3000 coefficients kept while (b) is perceptually weighted noise shaping. If we zoom on the left eye region of the image ((c) and (d)), we find (d) has sharper edge near the iris. However, we also found that the overall SSIM of selecting coefficients by this perceptual weighting method is lower in most other testings than in classical noise shaping. The reason either might be that the subsequent error feedback may neutralize the effect of selecting the coefficient, or it might be that the perceptual measurement in the DCWT domain may still need further tuning. However, since the complicated loop structure and projection theories of noise shaping, we still can not find a good way to support our assumptions at this moment.

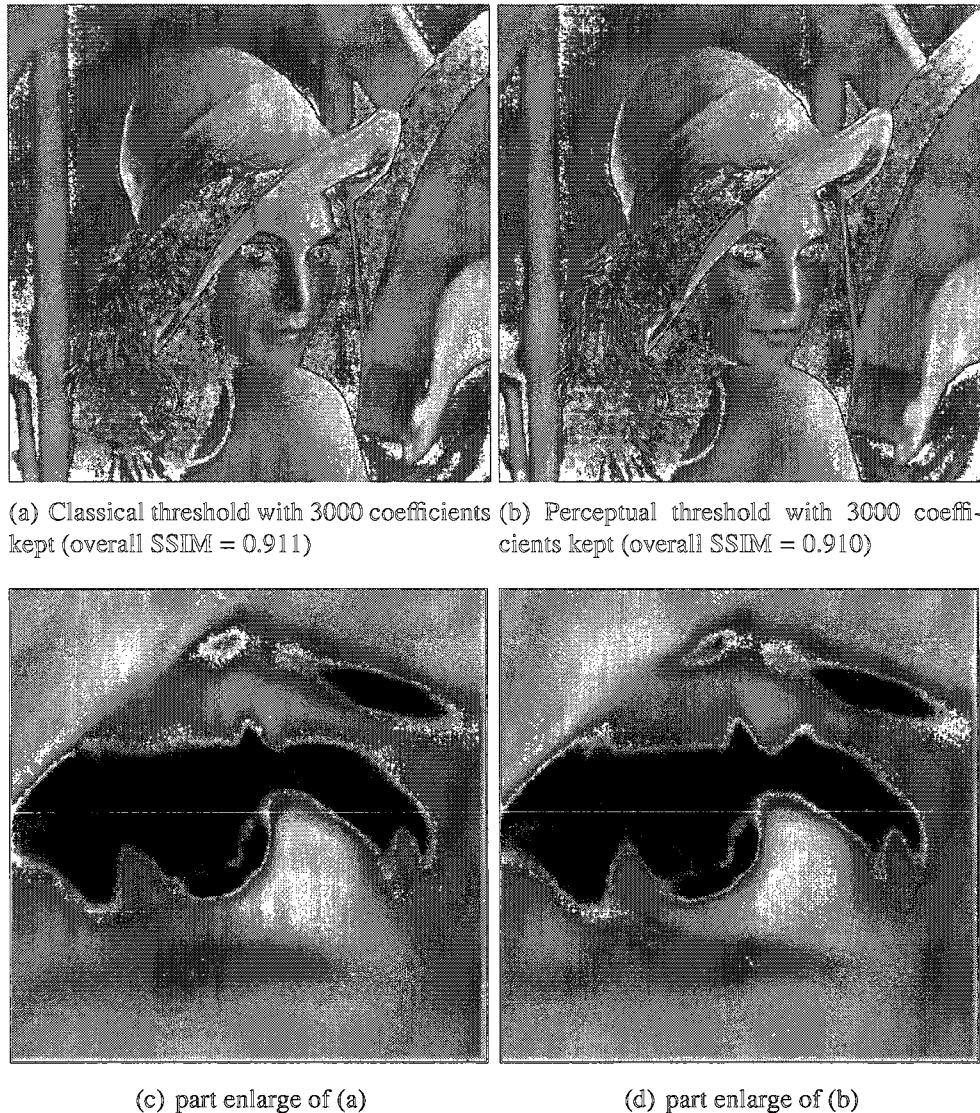


Figure 4.5: Image results examples by using classical threshold and perceptual threshold

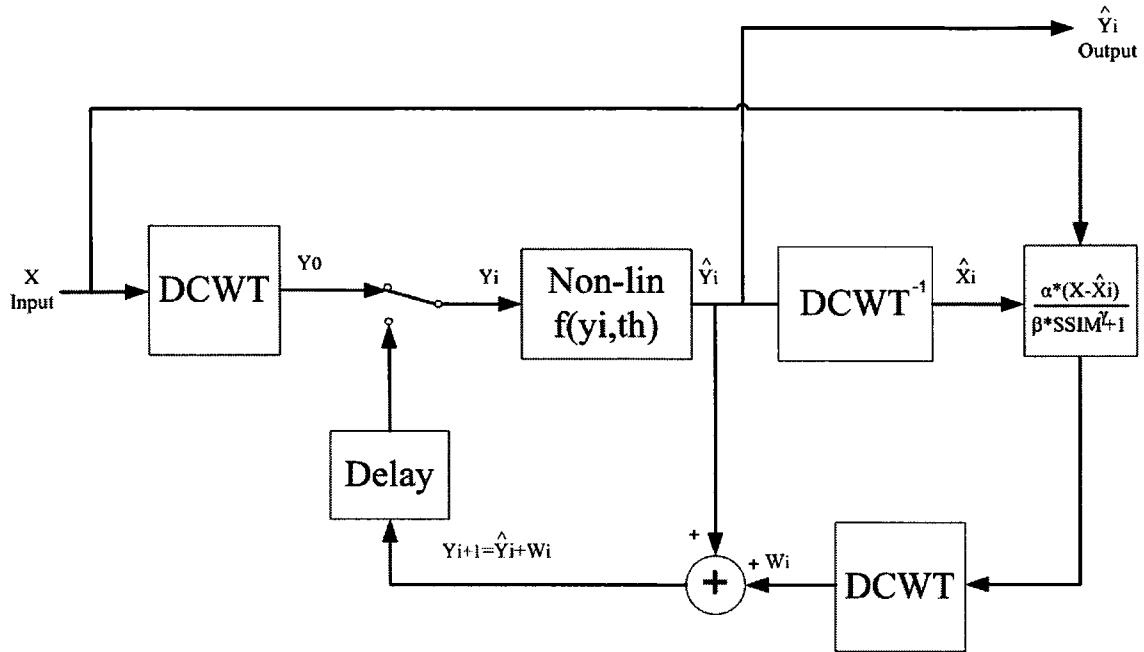


Figure 4.6: Diagram of feedback error perceptual weighting for noise shaping.

4.2.2 Feedback error shaping by perceptual weighting

In classical noise shaping, the error between thresholded coefficients and the original signals are multiplied by a gain in the spatial domain and sent back to the next iteration. The error is calculated by subtracting the original signal and the thresholded wavelet coefficient mapped in spatial domain. It is pixel by pixel operation and the result reflects an error similar to that embedded in MSE or PSNR, which is also a pixel by pixel operation. Since there are some known issues and limits of using MSE to evaluate the image quality (see examples in Figure 4.1 and 4.3), we felt that if we could shape these errors in a perceptual way, the performance of noise shaping would be improved. Another advantage of incorporating perceptual quality measurement in the spatial domain is that it is more straightforward and has been better studied than in the transform domain. For the above reasons, we decided to incorporate SSIM into spatial error feedback as our perceptual noise shaping framework. Figure 4.6 shows our approach, where we combined the error calculate unit and the gain unit in the classical noise shaping into one and incorporated SSIM in it. Based on the above discussion, we also proposed an equation to weight the feedback error as Equation 4.9. To our knowledge, Figure 4.6 and Equation 4.9 may be the first kind of approach to improve classical noise shaping in DCWT in a perceptual way.

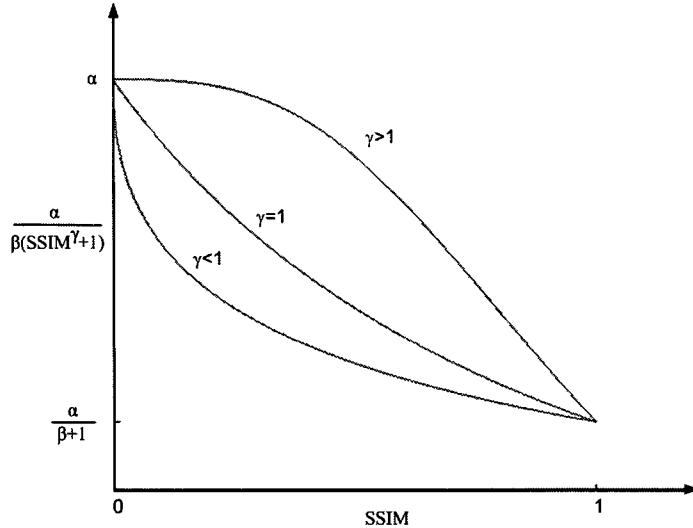


Figure 4.7: Relationship curve between SSIM and feedback error weighting.

$$\text{Error} = \frac{\alpha}{(\beta \cdot SSIM^\gamma + 1)}(X - \hat{X}_i) \quad (4.9)$$

In the above equation, X and \hat{X}_i denote the signal values of the original image and the thresholded image; $SSIM^\gamma$ denotes the SSIM index with γ exponent control. Since $X - \hat{X}_i$ is the spatial error in classical noise shaping, the first part $\frac{\alpha}{(\beta \cdot SSIM^\gamma + 1)}$ of the equation could be regarded as a perceptual weight incorporating k in the classical noise shaping with SSIM. The α and β are used for controlling the range of the weights, and the exponent γ control in the equation is a nonlinear operation to modify the contribution of SSIM. To better illustrate the error weighting process, we plot the error weighting curve in Figure 4.7, which shows the relationship of SSIM and the error weights. In the figure, the x axis denotes the value of SSIM which ranges between 0 and 1; the y axis denotes the value of $\frac{\alpha}{(\beta \cdot SSIM^\gamma + 1)}$, which is the weight of the error. As we can see, the maximum weight is α at $SSIM = 0$ and the minimum weight is $\frac{\alpha}{\beta+1}$ at $SSIM = 1$. The γ changes the shape of curve. If $\gamma > 1$, the weight tends to stay high in a low SSIM region but drops dramatically in a higher SSIM region. If $\gamma < 1$, then the weight goes to the opposite direction.

To improve overall image quality, we made the following assumption: the overall quality of an image could be improved by balancing the quality of all regions, that is, the overall quality could be improved when we improve the quality in lower perceptual quality regions (SSIM is close to 0) by “moving” some quality from higher quality regions into it. The weight curve we proposed in Figure 4.7 reflects our assumption: we give a large

error compensation in a lower quality region (SSIM is close to 0) and we give a small error compensation in a higher quality region (SSIM is close to 1) since in that region the image quality may already be too good to need any compensation. In that scenario, the condition $\gamma > 1$ is likely better because it compensates much more in a lower quality region than the condition with $\gamma < 1$.

Our experiment results support this assumption, example shown in Figure 4.8 and 4.9. We tested various parameters, but the ones in Figure 4.8 are set as $\alpha = 2.2$, $\beta = 1.6$ and $\gamma = 6$ for the best SSIM result in 8 bit 512×512 “barbara” image with 5000 non-zero coefficients kept. For classical noise shaping in our experiment, we used Wienier thresholding and let $k = 1.8$. Moreover, we also include the result from standard DWT with Cohen-Daubechies-Feauveau 9/7 wavelet for the purpose of comparison, which is shown in Figure 4.8(a) and its SSIM map in (b). Although the result from Cohen-Daubechies-Feauveau 9/7 wavelet also contains 5000 coefficients, the coefficients are all real numbers. It may not be fair to compare them with complex coefficients of DCWT with real and imaginary parts. Therefore, we mainly focus on comparing classical noise shaping and perceptual noise shaping that both use DCWT in this thesis.

In the experiment, if there is a region whose SSIM is really low (close to 0), according to Equation 4.9, multiplying the maximum weight ($\alpha = 2.2$) by the feedback error means the system compensates the low SSIM region information as much as possible during the loop. When there is a region in which SSIM is really high (close to 1), multiplying the minimum weight ($\frac{\alpha}{\beta+1} = 0.85$) by the feedback error means the system compensates less high SSIM region information during the loop. From Figure 4.8, we see that there are more details retained by error weighting than by classical noise shaping. For example, in the table cloth region of the images, image (e) shows much more texture information than image (c).

We also compared and plotted the SSIM map as (b),(d) and (f) of Figure 4.8 (a), (c) and (e) respectively. Dark region in the SSIM map means low quality region of the image. The brighter it is, the better the image quality in the SSIM map. The SSIM map in figure (d) and (f) clearly shows that the SSIM is greatly improved in the region where it is dark in classical noise shaping. For example the table cloth region in (d) and (f), (f) is much brighter than (d) in that region. In our test, the SSIM range is narrowed from (0.197 - 0.998) in classical noise shaping to (0.424 - 0.997) in our perceptual noise shaping.

In addition, we plotted the histograms of the SSIM in the Figure 4.9, where Figure 4.9(a) is the histogram from Figure 4.8(d) and Figure 4.9(b) is the histogram from Figure

4.8(f). By comparing the histograms of Figure 4.9 (a) and (b), it shows that more SSIM values are concentrated in the range of 0.7-1.0 with perceptual error weighting approach, while more SSIM values are outside this range in classical noise shaping. Finally, not only the SSIM range is narrowed and the visual performance is improved, but the overall SSIM of the test image was also improved from 0.878 in classical noise shaping to 0.891 in our perceptual noise shaping.

Since the α and β are adjustable in Equation 4.9, we also tried to test their limits in perceptual noise shaping. From the discussion in the Chapter 3, the feedback error cannot be multiplied by a number larger than 2 for stability reasons. Our tests also support this statement. Figure 4.10 shows some unstable examples for different number of retained nonzero coefficients when we set the α larger than 2.3. When an unstable situation happens, the magnitude of image either becomes extremely large or small so that the unstable region in the image becomes totally dark or white. For example, Figure 4.10(a) shows an unstable situation on the table; Figure 4.10(b) shows an unstable situation on the right side of the scarf of the woman.

As β is mainly used to control the lower bound, we found it could be lower than one in our tests. The maximum SSIM usually occurs when the lower bound equals 0.7. Since the curve only hits the lower bound in higher quality region, it suggests that the information in higher quality regions may be less important than in lower quality regions in terms of improving the overall SSIM index value.

Since our approach is based on an assumption that the overall image quality could be improved by balancing its quality in all regions of an image, we also verified the possibility to improve the overall image quality by the reverse scenario which makes lower quality regions worse to compensate higher quality regions so that the higher quality regions have even better quality. To test this scenario, a simple approach is to modify the weighting equation as $Error = (SSIM + 1)(X - \hat{X}_i)$ so that the image quality may be intensified in a higher quality region and weakened in a lower quality region. We call this scenario reverse weighting. Figure 4.11 is an example using 8 bits 512×512 “goldhill” image. Compared with classical noise shaping with overall SSIM 0.864, the overall SSIM of reverse weighting is down to 0.853. As in the SSIM map, the range of classical noise shaping is from 0.375 to 0.998 while the reverse weightings range is from 0.267 to 0.998. Therefore, the reverse weights may not be a good scenario to perceptually improve image quality.

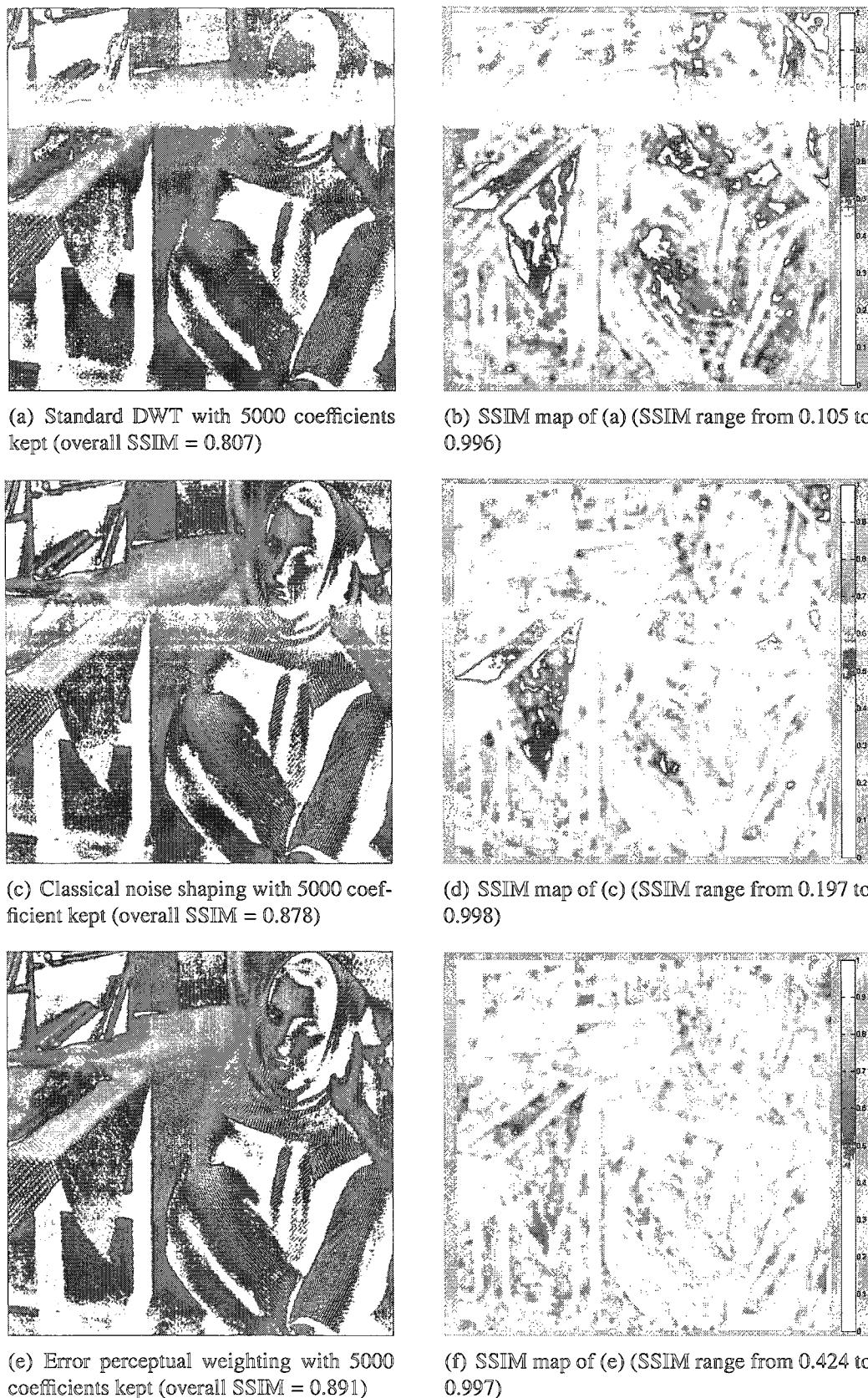


Figure 4.8: Image SSIM comparison among different wavelet transforms

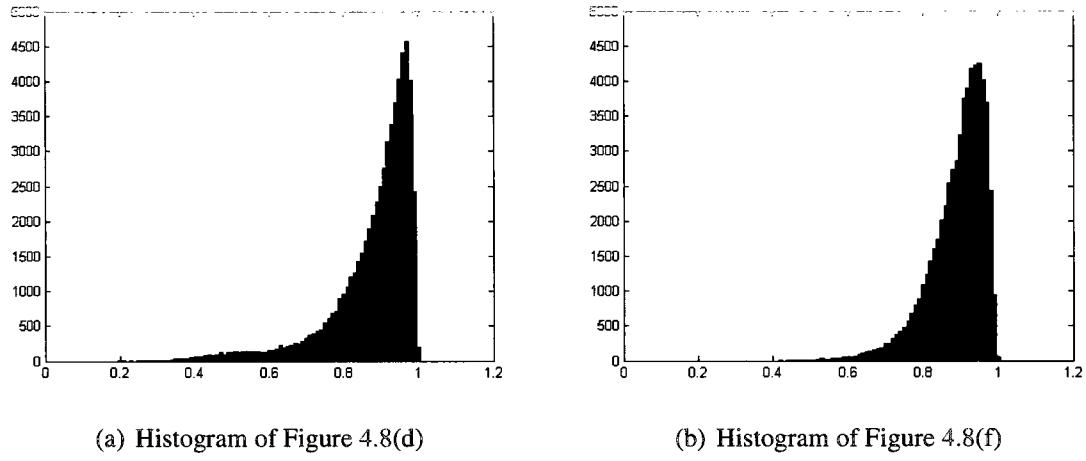


Figure 4.9: Histogram of SSIM comparsion between classical noise shaping and error perceptual weighting noise shaping

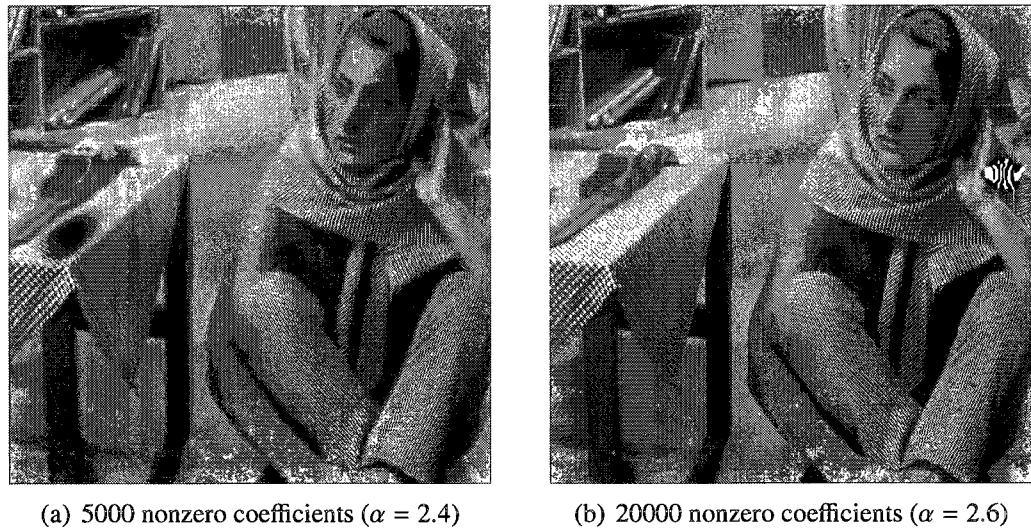


Figure 4.10: Unstable examples of error perceptual weighting noise shaping.

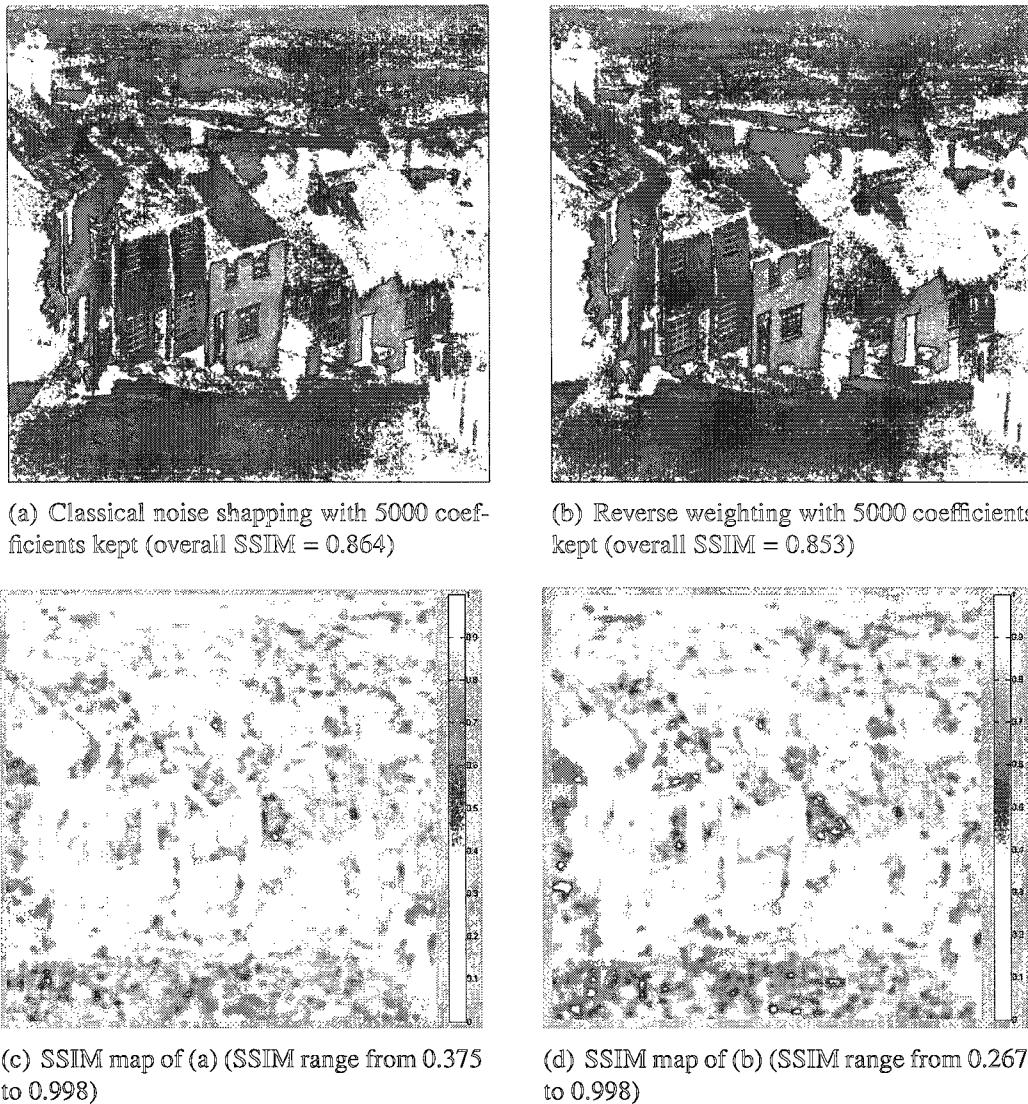


Figure 4.11: Noise shaping performance comparison between classical feedback error and feedback error with reverse weighting.

Table 4.1: SSIM comparison between classical and perceptual noise shaping (PSNR result shows inside brackets)

non-zero coefficients		5000	10000	15000	20000
classical noise shaping	barbara	0.878(27.44)	0.947(31.27)	0.969(33.87)	0.979(35.75)
	boat	0.892(29.37)	0.948(32.28)	0.969(34.15)	0.978(35.49)
	goldhill	0.864(29.36)	0.928(31.73)	0.954(33.35)	0.969(34.64)
	lena	0.944(33.36)	0.971(36.30)	0.980(37.98)	0.985(39.18)
perceptual noise shaping	barbara	0.891(27.15)	0.953(31.01)	0.971(33.38)	0.980(35.13)
	boat	0.909(29.04)	0.956(31.98)	0.973(33.74)	0.980(35.07)
	goldhill	0.885(29.12)	0.937(31.40)	0.963(33.02)	0.974(34.26)
	lena	0.951(33.02)	0.973(35.91)	0.982(37.58)	0.986(38.75)

In summary, the method in our approach shapes the feedback error in a perceptual way. Since most of our results by shaping the feedback error in a perceptual way are better than modifying threshold perceptually, we propose this kind of perceptual feedback error shaping as our perceptual noise shaping in this thesis. Table 4.1¹ shows the test result comparison by using some popular test images. More images and their SSIM maps, histogram comparisons from these test images are added in Appendix A.

To further verify our results, we also tested images from an image database. The image database used in the thesis is the University of Southern California's Signal and Image Processing Institute's image database, which was downloaded from [61]. The database categorizes digitized images based on their basic character: textures, high altitude aerial images, image sequences, and classic image processing images such as lena, mandrill, etc. We tested all grayscale images with size of 512 by 512 from the database. (Color images were converted into grayscale for our tests.) Figure 4.12 shows the SSIM comparisons between classical noise shaping and perceptual noise shaping with different retained non-zero coefficients in various categories, which are photos of scenes and people under miscellaneous category, aerial image category and textures image category; each plot shows the SSIM results. The left bar in each plot denotes the average SSIM of the classical noise shaping and the right bar denotes the average SSIM of the perceptual noise shaping. Figure 4.12(d) shows the overall average SSIM comparison across all categories. We tested in

¹Please note the PSNR (inside brackets of the table) does not improve by perceptual noise shaping

total 76 images from the database, with 25 in the miscellaneous category, 12 in the aerial category and 39 in the textures category. For parameter setting, we set $\alpha = 2.2$, $\beta = 1.6$ and $\gamma = 6$ as the same as in previous example in Figure 4.8. Although the parameters may not be optimized for each image in the database, the results still show that perceptual error weighting can improve the overall SSIM in most test cases except for 20,000 retained non-zero coefficients texture images. We also found that the PSNR of perceptual noise shaping did not improve which was in line with our previous results. The reason is that the spatial error in every location is given a constant maximum gain in classical noise shaping, while the spatial error in perceptual noise shaping is given a variable gain where in some locations the gain is much less than the maximum. Since the PSNR calculation is directly related to the spatial error, it is not expected to improve PSNR for perceptual noise shaping.

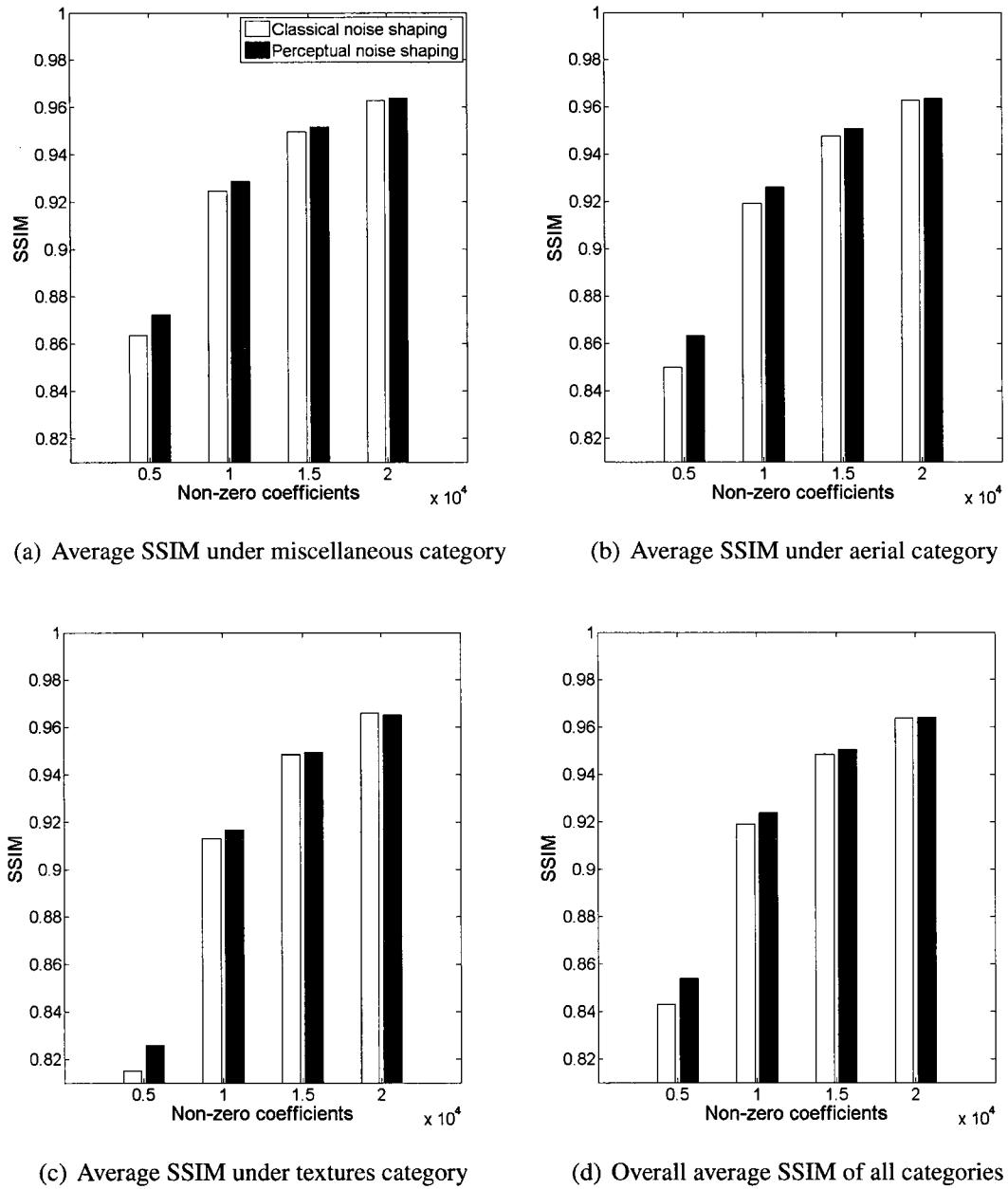


Figure 4.12: SSIM comparison between classical noise shaping and perceptual noise shaping by image database

Chapter 5

Conclusions and Discussions

Generally, the goal of image coding for image compression is to provide the best possible picture quality at a certain bit rate. When using the overcomplete DCWT in image coding, noise shaping is a simple and essential way to balance the coefficient expansion. In order to provide better possible image quality after noise shaping, we tried to incorporate perceptual measures in it to help weighting “important” information in noise shaping which may please human’s perceptual feelings. Although our perceptual Noise Shaping framework may not be most optimized to utilize HVS into classical noise shaping, the performance of our approach still has better result for the visual quality and higher overall SSIM index in most of our testings, moreover, the point is that we do believe that the perceptual consideration is the right and ultimate direction for image coding and we are progressing along the right direction.

5.1 Conclusions

Compared with critically-sampled DWT, DCWT provides more directional orientations and near shift invariance at the expense of redundancy in the wavelet domain. For example, a 1000 pixel image will have 2000 coefficients after it is transformed into the DCWT domain. In order to alleviate the disadvantages of redundancy while keeping the good properties of DCWT, Kingsbury and Reeves proposed the noise shaping algorithm in DCWT domain. The goal of noise shaping is to produce a good approximation to the original image by using as few coefficients as possible. It turns out that noise shaping performs very well in this respect both in our experiments and in other references. Our work is focused on finding a way to shape the coefficients which may lead to better perceptual performance by human assessment. Based on previous works of the development of SSIM, we moved

SSIM into error feedback path of noise shaping as a perceptual quality metric to control the performance. This kind of approach brings new perceptual meaning to the classical noise shaping. We call it perceptual noise shaping. Based on our test results, the perceptual noise shaping provides better visual result and SSIM value comparing to the classical one. The images' overall SSIM can be improved by balancing the quality of the higher quality regions and lower quality regions. On the contrary, we could also adjust the image result in perceptual noise shaping by making higher quality regions' quality better and lower quality regions' quality worse. However, we can not find any overall SSIM improvement by this scenario, nor would we expect it since SSIM is a mean of all local measures.

5.2 Discussions

In our perceptual noise shaping, the SSIM measurement is incorporated into noise shaping loops to help shape the “important” coefficients. The reason to choose SSIM as a quality metric is as follows. First, the SSIM is a perceptual quality assessment that is based on some characteristics of the HVS. Since a human observer is the final assessor, we believe the quality of an image can be improved subjectively by estimating its perceptual quality such as we tried in this thesis. Second, the SSIM has been intensively studied and demonstrated to overcome some limitations of traditional quality measurement metric such as MSE. Third, SSIM is a simple approach compared to other HVS models. In our experiments, the proposed perceptual noise shaping shows both visual and overall SSIM index improvement, especially at low number of retained DCWT coefficient. However, the visual difference is hard to perceive by human's eyes when the number of retained coefficient increases. The potential causes are discussed in the following paragraphs.

First, along the increase of retained coefficient, the overall image quality increases as well. It is hard to perceive differences between high quality images.

Second, the SSIM itself may still need tuning for perceptual measurement. Although the SSIM incorporates the well known HVS properties such as luminance and contrast, it is developed to be more tolerant to structure distortions which is generally not the main contribution of distortion in noise shaping. A modified SSIM which has more considerations in coding artifacts is preferable in the future work.

Third, current HVS model may need to be further studied. We believe that an advanced HVS model may give us more capability to develop better perceptual encoders.

5.3 Future work

Although we got positive results in our approach, we still felt that we just opened a door to add perceptual considerations into overcomplete DCWT based image coding systems. For the perceptual noise shaping framework, we attempted to give our answer to what, where and how to incorporate perceptual consideration into the noise shaping. However, we still need in-depth study to find the most effective solution. There are still many questions that need to be studied in the future. For example, is the feedback error path the best place to incorporate a quality measurement metric?

Since our approach is adjusting the SSIM map by weighting and shaping the feedback error in the noise shaping, the ideal goal is that there is a way that we can control this process. For instance, there is an SSIM map from 0.20 to 0.99 in classical noise shaping, we could get SSIM map result from 0.40 to 0.98 in our perceptual noise shaping. However, we still can not manually set the SSIM upper and lower bound in our current approach. Developing a quality controllable perceptual noise shaping is also a future work.

In the current phase of image coding by DCWT, we also feel that developing a good subband coder is another challenge. Although the number of non-zero coefficients are set to zero by noise shaping, the zeros are still located sparsely in the full DCWT domain, thus the subband coder may need more bits to code the location information which affects the overall compression performance of DCWT based image coding system. When a subband coder that is specialized for overcomplete DCWT is developed, more perceptual techniques will be able to be studied and applied in subband coding as well. In this way, the perceptual performance of DCWT based image coding system should be able to improve. Therefore, to develop a better subband coder for DCWT should definitely be included in the future work as well. Moreover, the comprehensive performance comparsion between the DCWT based image compression system and the DWT based image compression system can be done after the subband coder and entropy coder have been added into the image compression system.

List of References

- [1] M. Barni, *Document and Image Compression*. Boca Raton, FL: CRC, 2006.
- [2] C. Lastri, B. Aiazzi, L. Alparone, and S. Baronti, “Virtually lossless compression of astrophysical images,” *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2521–2535, 2005.
- [3] T. H. Reeves and N. G. Kingsbury, “Overcomplete image coding using iterative projection-based noise shaping,” in *Proc. IEEE Conf. on Image Processing*, pp. 597–600, 2002.
- [4] N. G. Kingsbury and T. H. Reeves, “Redundant representation with complex wavelets: how to achieve sparsity,” in *Proc. IEEE Conf. on Image Processing*, pp. I–45–8 vol.1, 2003.
- [5] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. on Image Processing*, vol. 13, pp. 600–612, 2004.
- [6] N. Kingsbury, “The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters,” in *Proc. IEEE DSP Workshop*, 1998.
- [7] H.R.Wu and K.R.Rao, *Digital Video Image Quality and Perceptual Coding*. Boca Raton, FL: CRC, 2006.
- [8] “SSIM index matlab code,” Available <http://www.ece.uwaterloo.ca/~z70wang/research/ssim/>.
- [9] K. S. Yonis, *Master Thesis: Objective Video Quality Assessment using the 3D Dual-Tree Complex Wavelet Transform*. Ottawa: Carleton University, 2009.
- [10] “Matlab implementation of 2-D dual-tree wavelet transform,” Available <http://taco.poly.edu/WaveletSoftware/dt2D.html>.
- [11] “VcDemo: Image and video compression learning tool,” in *Information and Communication Theory Group*, Delft University of Technology. Available <http://www-ict.its.tudelft.nl/inald/vcdemo>.
- [12] V. K. Goyal, “Theoretical foundations of transform coding,” *IEEE Signal Processing Magazine*, vol. 18, pp. 9–21, 2001.
- [13] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Trans. on Consumer Electronics*, vol. 38, pp. xviii–xxxiv, 1992.

- [14] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, 1993.
- [15] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, 1996.
- [16] S. Saha, “Image compression - from DCT to wavelets: a review,” *ACM Crossroads*, vol. 6, pp. 12–21, 2000.
- [17] D. Huffman, “A method for the construction of minimum redundancy codess,” *In Proceedings IRE*, vol. 40, pp. 1098–1101, 1962.
- [18] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, and R. B. Arps, “An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder,” *IBM Journal of Research and Development*, vol. 32, pp. 717–726, 1988.
- [19] J. Vaisey and T. Jin, “An iterative algorithm for context selection in adaptive entropy coders,” in *Proc. IEEE Conf. on Image Processing*, pp. 93–96, 2002.
- [20] K.R.Rao and P. Yip, *Discrete Cosine Transform*. San Diego: Academic Press, 1990.
- [21] E. Feig, “A fast scaled DCT algorithm,” *Proc. SPIE Image Processing Algorithms and Techniques*, vol. 1244, pp. 2–13, 1990.
- [22] Z. Xiong, K. Ramchandran, M. Orchard, and Y.-Q. Zhang, “A comparative study of DCT and wavelet-based image coding,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, pp. 692–695, 1999.
- [23] I. Selesnick, R. Baraniuk, and N. Kingsbury, “The dual-tree complex wavelet transform,” *IEEE Signal Processing Magazine*, vol. 22, pp. 123–151, 2005.
- [24] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG2000 still image compression standard,” *IEEE Signal Processing Magazine*, vol. 18, pp. 36–58, 2001.
- [25] B. E. Usevitch, “A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000,” *IEEE Signal Processing Magazine*, vol. 18, pp. 22–35, 2001.
- [26] J. Yang, Y. Wang, W. Xu, and Q. Dai, “Image coding using dual-tree discrete wavelet transform,” *IEEE Trans. on Image Processing*, vol. 17, pp. 1555–1569, 2008.
- [27] J. E. Fowler, J. B. Boettcher, and B. Pesquet-Popescu, “Image coding using a complex dual-tree wavelet transform,” in *Proc. the European Signal Processing Conference*, 2007.
- [28] B. Wang, Y. Wang, I. Selesnick, and A. Vetro, “Video coding using 3D dual-tree wavelet transform,” *EURASIP Journal on Image and Video Processing*, vol. 2007.
- [29] J. B. Boettcher and J. E. Fowler, “Video coding using a complex wavelet transform and set partitioning,” *IEEE Signal Processing Letters*, vol. 14, pp. 633–636, 2007.
- [30] M. A. Miller and N. G. Kingsbury, “Image denoising using derotated complex wavelet coefficients,” *IEEE Trans. on Image Processing*, vol. 17, pp. 1500–1511, 2008.

- [31] P. Loo and N. G. Kingsbury, "Digital watermarking using complex wavelets," in *Proc. IEEE Conf. on Image Processing*, pp. 29–32, 2000.
- [32] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Journal of Applied and Computational Harmonic Analysis*, vol. 10, pp. 234–253, 2001.
- [33] H.-W. Park and H.-S. Kim, "Motion estimation using low-band-shift method for wavelet-based moving-picture coding," *IEEE Trans. Image Process.*, vol. 9, pp. 577–587, 2000.
- [34] N. Kingsbury, "Image processing with complex wavelet," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 357, pp. 2543–2560, 1999.
- [35] J. Yang, W. Xu, Q. Dai, and Y. Wang, "Image compression using 2D dual-tree discrete wavelet transform," in *Proc. IEEE International Symposium on Circuits and Systems*, 2007.
- [36] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158–1170, 2000.
- [37] C. Tian and S. S. Hemami, "An embedded image coding system based on tarp filter with classification," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 49–52, 2004.
- [38] J. E. Fowler, "Shape-adaptive coding using binary set splitting with k-d trees," in *Proc. of the International Conference on Image Processing*, pp. 1301–1304, 2004.
- [39] F. W. Campbell, J. J. Kulikowski, and J. Levinson, "The effect of orientation on the visual resolution of gratings," *The Journal of Physiology*, vol. 187, pp. 427–436, 1966.
- [40] C. Blakemore and F. W. Campbell, "On the existence of neurons in the human visual system selectively responsive to the orientation and size of retinal images," *Journal of Physiology*, vol. 203, pp. 237–260, 1969.
- [41] R. L. DeValois and K. K. DeValois, *Spatial Vision*. Oxford, UK: Oxford University Press Inc, 1988.
- [42] J. Daugman, "Spatial visual channels in the Fourier plane," *Vision Research*, vol. 24, pp. 891–910, 1984.
- [43] A. B. Watson, "The cortex transform: rapid computation of simulated neural images," *Comput. Vision Graph. Image Process.*, vol. 39, pp. 311–327, 1987.
- [44] A. S. H. Senane and D. Barba, "The computation of visual bandwidths and their impact in image decomposition and coding," in *Proc. of Intern. Conf. on Signal Processing Applications and Technology*, pp. 766–770, 1993.
- [45] C.H.Chou, "Adaptive transform coding of images based on removing just noticeable distortion," in *Proc. of SPIE Visual Communications and Image Processing*, pp. 607–618, 1995.

- [46] G. E. Legge and J. M. Foley, “Contrast masking in human vision,” *J. Optical Society of America*, vol. 7, pp. 1458–1471, 1980.
- [47] N. Moumkine, A. Tamtaoui, and A. A. Ouahman, “Integration of the contrast sensitivity function into wavelet codec,” in *Proc. of Eurasip ISCCSP*, 2006.
- [48] S. Winkler and P. Vandergheynst, “Computing isotropic local contrast from oriented pyramid decompositions,” in *Proc. of IEEE Intern. Conf. on Image Processing*, vol. 4, pp. 420–424, 1999.
- [49] S. Daly, W. Zeng, J. Li, and S. Lei, “Visual masking in wavelet compression for JPEG2000,” in *Proc. SPIE Image and Video Communications and Processing*, 2000.
- [50] H. A. Spang and P. M. Schultheiss, “Reduction of quantizing noise by use of feedback,” *IRE Trans. Communications Systems*, vol. 10, pp. 373–380, 1962.
- [51] J. Proakis and D. Manolakis, *Digital Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2006.
- [52] S. Tewksbury and R. Hallock, “Oversampled, linear predictive and noise-shaping coders of order $n > 1$,” *IEEE Trans. Circuits and System*, vol. 25, pp. 436–447, 1978.
- [53] H. Bolcskei and F. Hlawatsch, “Noise reduction in oversampled filter banks using predictive quantization,” *IEEE Trans. Information Theory*, vol. 47, pp. 155–172, 2001.
- [54] F. W. Wheeler, R. T. Hoctor, and E. B. Barrett, “Super-resolution image synthesis using projections onto convex sets in the frequency domain,” in *SPIE Symposium on Electronic Imaging, Conference on Computational Imaging*, pp. 479–490, 2005.
- [55] Y. Yang and N. P. Galatsanos, “Removal of compression artifacts using projections onto convex sets and line process modelling,” *IEEE Trans. Image Processing*, vol. 6, pp. 1345–1357, 1997.
- [56] D. Donoho, “De-noising by soft-thresholding,” *IEEE Trans. Information Theory*, vol. 41, pp. 613–627, 1995.
- [57] S. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Processing*, vol. 9, pp. 1532–1546, 2000.
- [58] Z. Wang and E. Simoncelli, “Translation insensitive image similarity in complex wavelet domain,” in *Proc. of IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, pp. 573–576, 2005.
- [59] R. M. Figueras and E. P. Simoncelli, “Statistically driven sparse image approximation,” in *Proc. IEEE Conf. on Image Processing*, pp. 461–464, 2007.
- [60] D. Percival and A. Walden, *Wavelet Method for Time Series Analysis*. Cambridge, UK: Cambridge University Press, 2000.
- [61] “The USC-SIPI image database,” Available <http://sipi.usc.edu/database/index.html>.

Appendix A

Additional examples by perceptual noise shaping



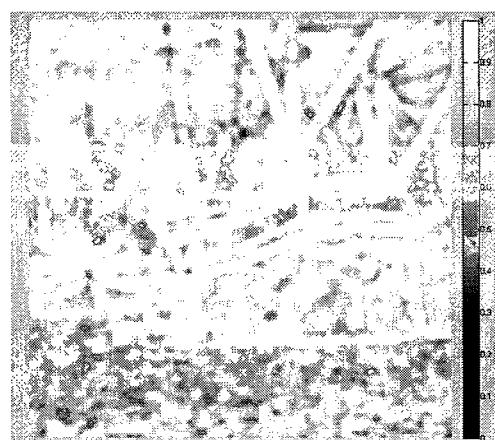
(a) Classical noise shaping (overall SSIM = 0.835)



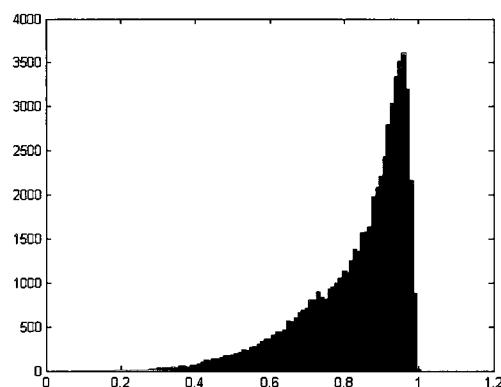
(b) Perceptual noise shaping (overall SSIM = 0.856)



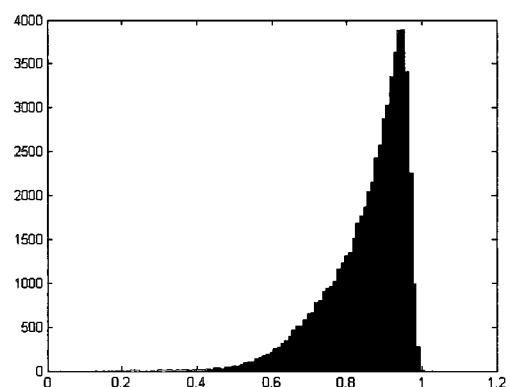
(c) SSIM map of (a) (SSIM range from 0.193 to 0.997)



(d) SSIM map of (b) (SSIM range from 0.228 to 0.996)

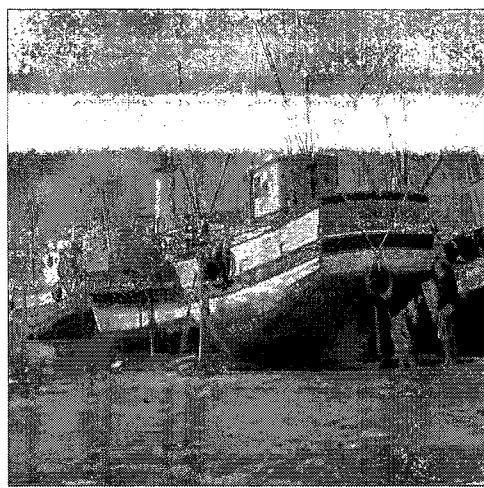


(e) Histogram of (a)

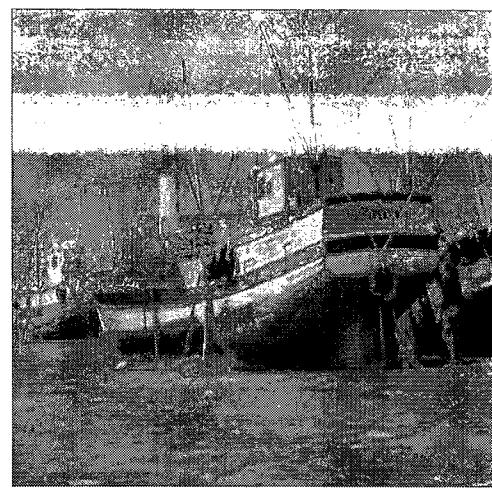


(f) Histogram of (b)

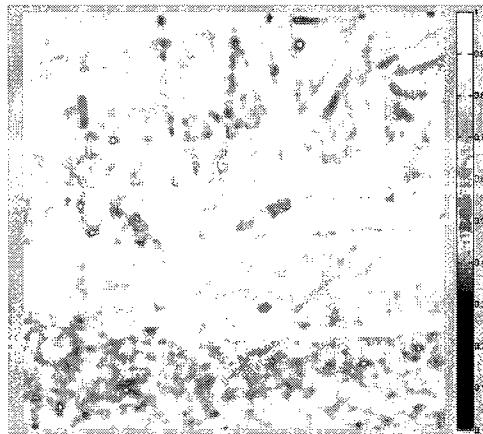
Figure A.1: “Boat” image with 3000 non-zero coefficients



(a) Classical noise shaping (overall SSIM = 0.892)



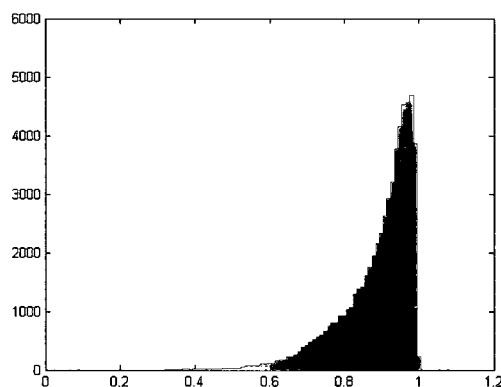
(b) Perceptual noise shaping (overall SSIM = 0.909)



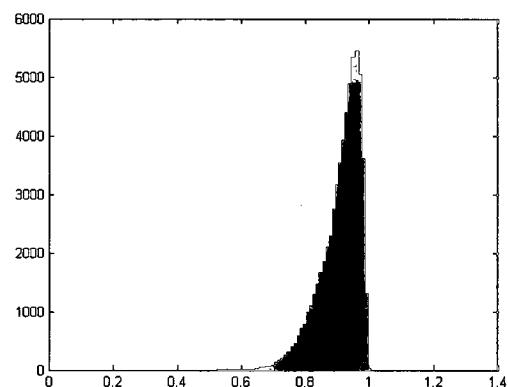
(c) SSIM map of (a) (SSIM range from 0.365 to 0.998)



(d) SSIM map of (b) (SSIM range from 0.526 to 0.996)



(e) Histogram of (a)



(f) Histogram of (b)

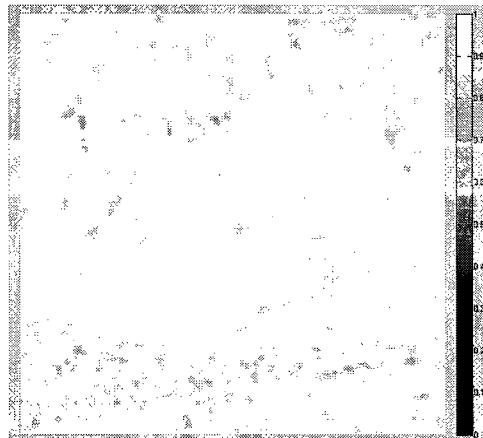
Figure A.2: “Boat” image with 5000 non-zero coefficients



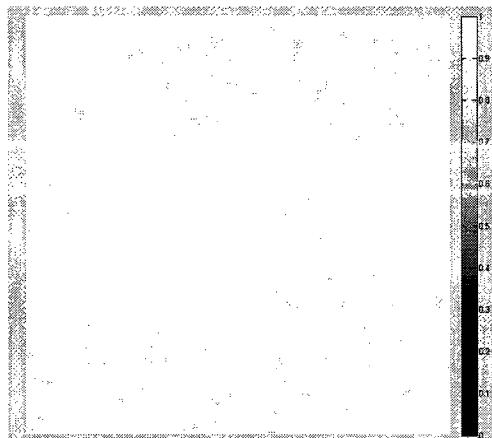
(a) Classical noise shaping (overall SSIM = 0.948)



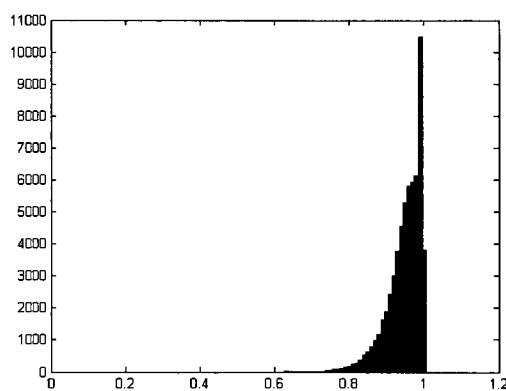
(b) Perceptual noise shaping (overall SSIM = 0.956)



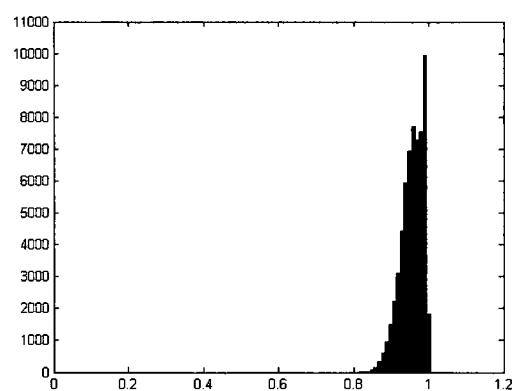
(c) SSIM map of (a) (SSIM range from 0.631 to 0.999)



(d) SSIM map of (b) (SSIM range from 0.818 to 0.999)



(e) Histogram of (a)



(f) Histogram of (b)

Figure A.3: “Boat” image with 10000 non-zero coefficients