Copyright © 1999, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

IMAGE DENOISING AND INTERPOLATION BASED ON COMPRESSION AND EDGE MODELS

by

Sai-Hsueh Grace Chang

Memorandum No. UCB/ERL M99/57

15 November 1999

IMAGE DENOISING AND INTERPOLATION BASED ON COMPRESSION AND EDGE MODELS

by

Sai-Hsueh Grace Chang

Memorandum No. UCB/ERL M99/57

15 November 1999

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Image Denoising and Interpolation based on Compression and Edge Models

by

Sai-Hsueh Grace Chang

B.S. (Massachusetts Institute of Technology) 1993
M.S. (University of California, Berkeley) 1995

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Martin Vetterli, Co-Chair
Professor Bin Yu, Co-Chair
Professor Jitendra Malik

1998

The dissertation of Sai-Hsueh Grace Chang is approved:

Co-Chair Date

Co-Chair Date

Date

University of California at Berkeley

1998

Image Denoising and Interpolation based on Compression and Edge Models

Copyright 1998
by
Sai-Hsueh Grace Chang

Abstract

Image Denoising and Interpolation based on Compression and Edge Models

by

Sai-Hsueh Grace Chang

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California at Berkeley

Professor Martin Vetterli and Professor Bin Yu, Co-Chair

This thesis investigates some innovative approaches to inverse problems in image restoration and enhancement. The specific problems addressed include image denoising and image interpolation. Before developing an algorithm, the first step is to find an appropriate image model to use. To achieve this, we identify two successful domains of image processing, namely, *image compression* and *edge analysis*, from which ideas can be applied to image denoising and interpolation. The underlying framework for signal analysis and algorithm development is based on *wavelets*, which conveniently provides a multiresolution, localized space-frequency representation of the signal.

Wavelet thresholding is a simple and effective denoising method that has been studied extensively in recently years. Most of the significant insights have stemmed from the statistics community, and thus, not much have been researched on finding an appropriate model for images and the corresponding wavelet thresholding strategy. We use a Bayesian model for the distribution of the wavelet coefficients, namely, the Generalized Gaussian distribution which has been widely used for image compression. From this distribution, we propose a near-optimal threshold selection. This threshold value is used in the various denoising algorithms in this thesis that incorporate wavelet thresholding with several image models motivated by compression methods and edge analysis.

One of the first ideas we examine is using lossy compression for removing noise from corrupted images. Previously proposed approaches were either unclear about the choice of the coder or were less than a true lossy compression. We make a connection between lossy compression and wavelet thresholding, and develop a systematic lossy compression method

to achieve simultaneous compression and denoising.

Next, we develop a spatially adaptive algorithm for image denoising. Images typically consist of edges, textures and smooth regions. In the wavelet transform domain, the first two features are characterized by clusters of high energy transform coefficients and the latter by low energy coefficients. Because edges are among the most important features in an image, typical coders allocate the most resources for these high energy coefficients. Distortion due to edge blurring is very noticeable; distortion due to additive random noise, however, is not as discernible in the edge region. This edge preservation idea from coding can be applied to denoising, with the edge coefficients being only slightly modified to preserve the edge sharpness, and the flat region coefficients being significantly smoothed to guarantee the removal of most of the noise.

To conclude the denoising topic, we investigate the best strategy to combine multiple noisy copies of the same image. Typically, multiple sets of noisy observations of the same data are averaged to obtain the best estimate of the noiseless version. Since wavelet thresholding is effective for denoising one set of noisy observations, it is worthwhile to incorporate it with weighted averaging when multiple noisy copies are available. In particular, we investigate which sequential ordering of the averaging and wavelet thresholding operation would yield a final result with the lowest mean squared error. The result shows that, under the assumed Laplacian distribution for the coefficients (a special, simple case of the Generalized Gaussian), the ordering is dependent on the distribution parameter, the noise power, and the number of noisy copies.

Lastly, we develop an edge-preserving image interpolation algorithm. The available image is modeled as a low resolution image of some higher resolution image which we wish to estimate. The additional details needed to obtain the desired image is estimated by extrapolating edge characteristics from the low resolution image. The problem model and the edge analysis can be developed very naturally in the wavelet framework.

To my parents,

who have made great sacrifices to give Henry and me what we have today.

Contents

List of Figures

List of Tables

Acknowledgements

It has been a long journey and there have been so many people who were encouraging and helpful along the way. First and foremost, I would like to thank my two wonderful advisors, Martin Vetterli and Bin Yu. Besides providing invigorating technical discussions, they were often more friends than authoritative figures. It is this humanness that made the working rapport so enjoyable and endearing. Perhaps I have not always been easy to work with, and was at times wayward and spoiled, but I think we all got along pretty well. I also thank Martin for giving me the opportunity to live in Lausanne for half a year. This experience has changed my perspective in life perhaps more than any other thing I have learned in grad school. Bin has not only been a supportive mentor but a good friend as well. We have had many personal conversations which I did not imagine possible with an advisor, and she has given many invaluable advice on my personal life.

All my colleagues throughout grad school have been wonderful. Zoran Cvetković, my colleague-mentor during my early years at Berkeley, brought me up to speed with my first project, and was always very pleasant to be around. Masoud Khansari, the most devoted, selfless and supportive friend one could ask for, has listened to my cries of joy and pain through times of thick and thin. Masoud is also a walking encyclopedia from whom I can always gather lots of technical information, career advice, stock insights, and, more recently, HP gossips. I would also like to thank Cormac Herley of HP Labs and Balas Natarajan formerly of HP Labs for their initiatives in providing guidance in my research during the few months after I came back from Switzerland, and for opening up to me the tremendous resources of HP Labs. Lastly, the people of the Wavelet Group in Berkeley and LCAV in EPFL have made my working environment very enjoyable for the last five years.

I have made many special friends, in Berkeley and Lausanne, who have made grad school such a unique and exciting learning experience. You know who you are, so I will not list your names here because, well, it's just not my style. Being in Berkeley or Lausanne has been a transient period for many of us, and some of you have already left, and so have I. Fortunately, since most of us are engineers, it is likely that we either stay in or converge to the Silicon Valley, so there will be lots more chances to take trips or hit the bar scenes together.

Mom, Dad — I finished!

IMAGE DENOISING AND INTERPOLATION BASED ON COMPRESSION AND EDGE MODELS

by

Sai-Hsueh Grace Chang

Memorandum No. UCB/ERL M99/57

15 November 1999

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Chapter 1

Introduction

Image restoration and enhancement is a useful but often difficult area, due to the need to estimate and to reverse an unknown degradation process. An image is often corrupted during an intermediate process such as transmission or acquisition, and depending on the specific goals and applications, reversing this degradation may be only partially achievable. To ameliorate the degradation, it is necessary to first devise a problem model, including that of the degradation process and the image, and then to estimate the original image from this model. This model of the problem and the image is application dependent, and we investigate two applications in this thesis: image denoising and image interpolation.

The term "denoising" has been coined in recent years to refer to the classic problem of removing noise from a corrupted signal, and it has gained a surge of interest partially due to a simple yet effective technique called *wavelet thresholding* [22, 21, 23, 24, 16]. The Wiener filter is a traditional approach which results in the optimal *linear* least squares estimator of the original signal. The non-linear wavelet thresholding, when appropriate parameters are chosen, often yield images visually better than those from Wiener filtering. Many results on wavelet denoising stem from statistics and provide valuable theoretical insights into the performance of wavelet thresholding under different signal and noise models. In this work, we approach denoising from a more image processing point of view, and, in our algorithms, combine wisdoms from both the theoretical works and the image processing insights to be discussed shortly.

Image interpolation, often for purposes of magnification or zooming, is another classic problem investigated in this thesis. The most simple-minded algorithms are zero-order hold (or pixel replication) and linear interpolation, both known to produce blurry

and jagged images. Interpolation using higher order polynomials and splines often yield visually more pleasant images. However, they all assume some smoothness constraints on the underlying signal, which may not always be valid (such as in the case of interpolating a step edge). Our approach is to devise constraints adaptive to the local image characteristics (based on an appropriate image model) rather than to employ presumed smoothness constraints.

The initial step in addressing these restoration and enhancement problems is to devise an image model. Image modeling is a daunting task in itself, and there is really no consensus on a general model which can well describe an arbitrary image. Most of the time, the model used is application specific. For example, in analyzing and synthesizing texture images, a combination of deterministic periodic components and random field is frequently used [25, 26]. This approach, however, may not be suitable for an image which has no periodic components but with many edges. A lack of general image models has prompted us to ask what branches of image processing have been successful. *Image compression* and *edge analysis* are two such branches and they provide the basis for numerous motivations in this work.

A coder which compresses an image well must provide a good model of the image since it can represent the image concisely. Such a coder exploits the predictable structures in a typical image to reduce the redundancy in the coded bits. White noise, on the other hand, is not compressible because it does not have correlated structures. Thus, compression can provide a suitable model which distinguishes between a structured data (a typical image) and a sequence of random noise.

A framework that has enabled excellent performance for both compression and edge analysis is the *wavelet analysis*. Its ability to provide localized information in the space and frequency domain and a multiresolution structure has made it an attractive framework. A notable predecessor of wavelet-based coders is the pyramid scheme [6]. Later, image compression based on the pyramid scheme or wavelet analysis (collectively called *subband coding*) became popular as its superiority over DCT-based compression became clear (see [67] for a survey of subband coding). The breakthrough in wavelet-based image compression was the embedded zerotree wavelet (EZW) coder [55], which was based on the observation that insignificant transform coefficients tend to occur in a predictable tree structure. The SPIHT coder by [53] uses the same intuition. Some later wavelet-based algorithms (for example [37, 68]) used classification or on-the-fly prediction to adapt the coder to spatially changing

energies in the transform coefficients. Wavelets are also the backbone framework for several top contenders in the JPEG2000 image compression standard. Thus, the wavelet analysis, along with other ideas and motivations extracted from various compression methods, are deemed to provide a suitable framework for the image denoising problem.

Edges are among the most important features in an image, for they are the component most accountable for making an object recognizable. Edge analysis, thus, is an important part in many image processing applications. The edge detection methods of [7, 42] can also be formulated in the wavelet framework [40, 41], as the detection of significant local extrema or of zero-crossings in the wavelet transform. In [40, 41], keeping these significant local extrema was used for the purposes of compression or denoising. In the interpolation problem, edge analysis is important if one wishes to preserve the regularity of the edges and not create an overly smoothed image.

In this thesis, the goal is to investigate how ideas from image compression and edge analysis combined with the wavelet framework can give a new way of thinking about image restoration and enhancement. A large part is dedicated to developing different approaches of image denoising, within the wavelet thresholding paradigm. We commence with background materials in Chapter 2, including an introduction of the well-known wavelet transform, and the idea of wavelet thresholding along with a brief survey of that literature. Subsequently, in Chapter 3 we present our Bayesian approach to wavelet thresholding, tailored for denoising images. It has been widely accepted in the image coding community that the subband coefficients collectively form a histogram which is sharply peaked at zero and symmetric about zero. This distribution has generally been described by the Laplacian distribution for simplicity or the more encompassing Generalized Gaussian distribution. Using this distribution, we propose a threshold which is close to the optimal threshold that minimizes the expected squared error for the soft-thresholding estimator. Building on this result, in Chapter 4 we develop a lossy compression method which achieves simultaneous denoising and compression when both features are desired. The idea of using lossy compression as a means to denoise has been proposed in several work [54, 49, 12, 13, 35]. A typical image has a predictable structure that can be highly exploited by a coder, while a sequence of random noise does not and thus is not compressible. This disparity suggests that a compression method can distinguish the noise from the image. The compression algorithm for denoising developed in this thesis incorporates wavelet thresholding, coefficient quantization, and entropy coding, of which the decision on certain parameters are based on the *Minimum*

Description Length principle [50]. In Chapter 5, we investigate the spatial adaptivity of threshold selection, an area not explored in the literature. This is motivated by the intuition that the knowledge of the spatially changing characteristics of the image can yield a threshold selection adaptive on a pixel-by-pixel base, which in turn generates a significantly better denoised image than that due to a uniform threshold (measured both visually and in the mean squared error sense). The adaptivity of the threshold value is based on context modeling, a commonly used technique in compression methods for adapting the coder on-the-fly to local image characteristics. Lastly, in Chapter 6, we extend wavelet thresholding denoising to situations when multiple corrupted copies are observed. The most straightforward recovery method is to simply compute a weighted average of the noisy copies. We explore whether an additional thresholding step would improve the performance, and investigate the preferred ordering (averaging first or thresholding first) which yields a lower mean squared error.

To conclude this thesis, in Chapter 7 we present an edge-preserving interpolation algorithm, based on estimating higher resolution information from the available image. The idea is to observe that extrema points in the wavelet transform propagate across scales, and the higher resolution information can be obtained by extrapolating this trend into the finer scale. In Chapter 8, we summarize the findings in this thesis and propose related future directions.

Chapter 2

Introduction to Wavelet Transform and Wavelet-based Denoising

The basic analysis tool used in this thesis is the wavelet transform. Its usefulness and efficient implementation has made it ubiquitous in the signal processing community. It offers an alternative to Fourier analysis and provides information which have been shown to be suitable for applications such as image compression, denoising and edge characterization (see, for example, [55, 53, 37, 68, 22, 24, 39, 38] and other work referenced therein). While wavelets have deep roots in mathematics, in Section 2.1 we only briefly describe here their notations, functionalities, and implementations, and refer the readers to standard literatures for a detailed discussion [19, 20, 39, 58, 60].

A substantial part of this thesis is based on extensions of the *wavelet thresholding* technique for signal denoising. Thus, Section 2.2 presents a survey of the different thresholding methods proposed in the literature. We first describe the seminal work of Donoho and Johnstone [22] on wavelet thresholding, and its asymptotic near-optimality. A Bayesian approach allows one to incorporate some prior knowledge of both the signal and noise. Thus, this will be our preferred framework since for a large class of natural images, the wavelet transform coefficients are often well-described by the Generalized Gaussian distribution, a piece of information which aids us in making an appropriate threshold selection. Non-parametric methods such as cross-validation and its variant have been proposed as well. Of particular interest is the work in [54] which combines thresholding with a non-parametric model selection criterion based on the Minimum Description Length principle [50]. This

approach is related to our compression-based denoising work described in Chapter 4 and will be revisited there.

2.1 Multiresolution Analysis and the Wavelet Transform

To introduce the wavelet transform, it is perhaps easiest to make analogies with the Fourier transform. Just as Fourier analysis is an expansion of a signal into sinusoids of different frequencies, the wavelet transform decomposes a signal into a *wavelet* basis of different spatial and frequency support. For now we will focus on the *discrete-time wavelet series* (sometimes also called the *discrete-time wavelet transform* in the literature) rather than the continuous-time wavelet transform.

The discrete-time wavelet series expansion can be implemented by an *analysis* filter bank shown in Figure 2.1 (a), which is a cascade of a two-channel filtering. Each stage of the two-channel bank consists of a highpass filter $H_1(z)$ and a lowpass filter $H_0(z)$, and it is iterated on the lowpass channel. This tree structure, often called an *octave-band filter bank* because each successive highpass output contains an octave of the input bandwidth, achieves the tiling of the time-frequency plane given in Figure 2.2 (shown for a tree of depth four). Each coefficient in the expansion corresponds to the result of projecting onto a basis representing one of the tiles[1]. In the first stage of the decomposition, the basis functions span a short time period, but a large frequency range (from $\pi/2$ to π). In the second stage, the basis functions span a time period twice as long as that in the first stage, but half the size of the frequency range (from $\pi/4$ to $\pi/2$). This recursive iteration thus results in a logarithmic tiling of the time-frequency plane.

The *synthesis* part (or the inverse transform) is shown in Figure 2.1 (b), where the analysis filters, $H_0(z)$ and $H_1(z)$, and the synthesis filters, $\tilde{H}_0(z)$ and $\tilde{H}_1(z)$, together must satisfy the *perfect reconstruction* property:

$$\begin{aligned} \tilde{H}_0(z)H_0(z) + \tilde{H}_1(z)H_1(z) &= 2 \\ \tilde{H}_0(z)H_0(-z) + \tilde{H}_1(z)H_1(-z) &= 0 \ . \end{aligned} \tag{2.1}$$

[1]The division of the time-frequency plane is ideal as shown in Figure 2.2 for the sake of illustration. In reality, it is not possible to be both time-limited and band-limited.

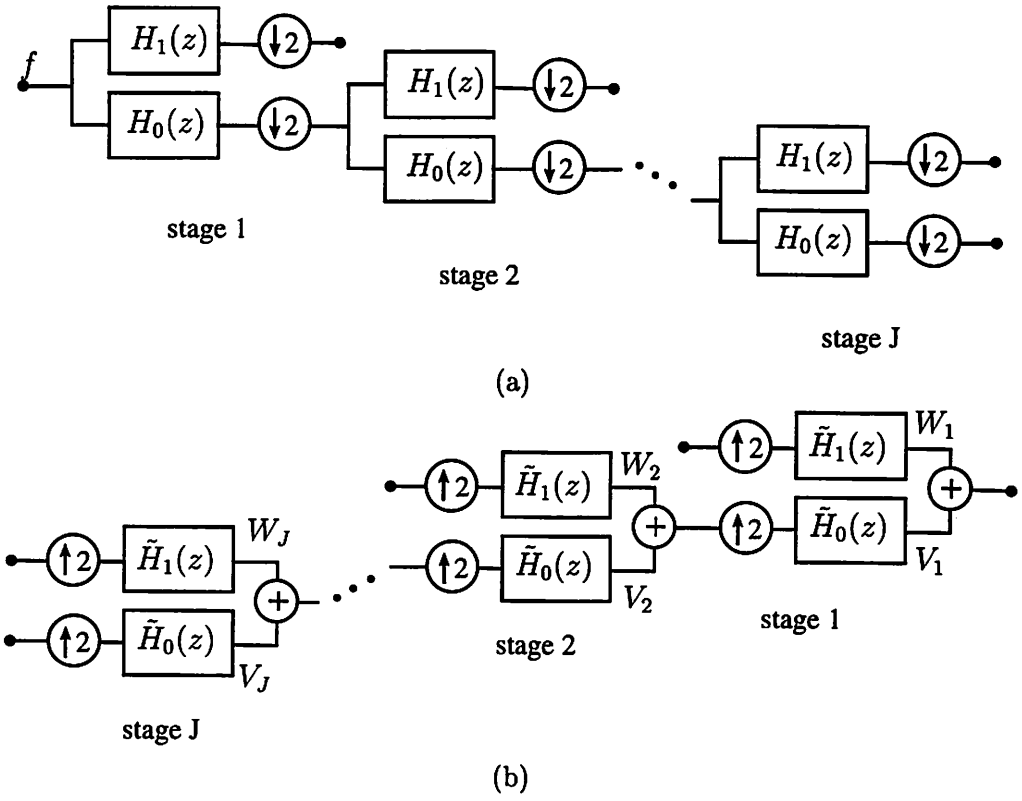


Figure 2.1: An octave-band filter bank of J stages, implementing the discrete-time wavelet series expansion. The decomposition spaces V_i and W_i are labelled. (a) Analysis stages (forward transform). (b) Synthesis stages (inverse transform).

2.1.1 Multiresolution Interpretation

The octave-band filter bank has a multiresolution interpretation that is often useful for signal analysis. At each stage, the two-channel filter bank splits the input into a lowpass component (or the coarser resolution part) and a highpass component (or the finer resolution part). This recursive application of the two-channel split on the lowpass part results in a hierarchical structure, called a *multiresolution decomposition.* The idea of viewing a signal at various resolutions has been explored for quite some time in the computer vision and the image processing community [66, 32]. Burt and Adelson [6] introduced the pyramid coding scheme which builds a signal from a low resolution version plus a sequence of finer and finer details. Daubechies [18] and Mallat [38] provided the first links between signal processing and the wavelet theory by recognizing that the pyramid scheme is closely related to wavelet theory and multiresolution analysis, and also that filter banks and subband coding can

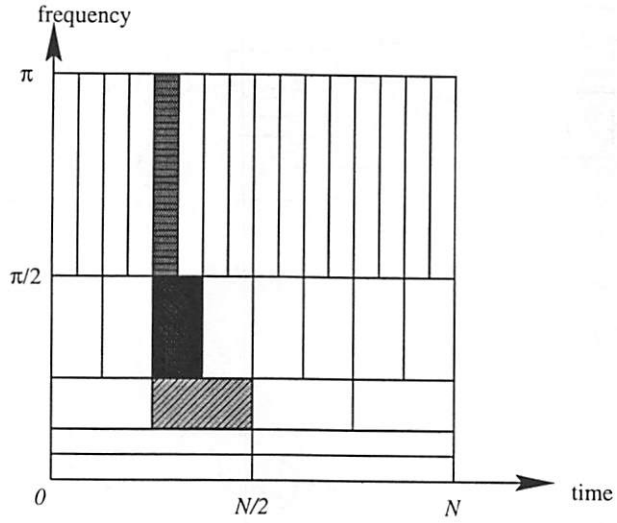


Figure 2.2: Tiling of the time-frequency plane achieved by the 1-D wavelet transform.

be used for efficient computation of wavelet decompositions. As aforementioned, here we concentrate on the discrete case, and will introduce the continuous decomposition when it is relevant to our image interpolation algorithm in Chapter 7.

The formalization of the multiresolution analysis is as follows. Let V_0 be the space of all square-summable sequences,

$$V_0 = \ell_2(\mathbb{Z}).$$

A multiresolution analysis consists of a sequence of *embedded* closed spaces

$$V_J \subset \cdots \subset V_2 \subset V_1 \subset V_0 \ .$$

The orthogonal complement of V_{j+1} in V_j is denoted by W_{j+1} and

$$V_j = V_{j+1} \oplus W_{j+1}$$

with $V_{j+1} \perp W_{j+1}$. Suppose there exists a sequence $g_0[n] \in V_0$ such that $\{g_0[n-2k]\}_{k\in\mathbb{Z}}$ is an orthogonal basis for V_1. Then it can be shown that for $g_1[n] = (-1)^n g_0[-n+1]$, $\{g_1[n-2k]\}_{k\in\mathbb{Z}}$ provides a basis for W_1. That is, $\{g_0[n-2k], g_1[n-2k]\}_{k\in\mathbb{Z}}$ is an orthonormal basis for V_0. This splitting of two orthogonal subspaces is iterated on V_j, and after J stages, V_0 can be written as

$$V_0 = W_1 \oplus W_2 \oplus \cdots \oplus W_J \oplus V_J.$$

The space V_j's are called the *approximation* spaces and W_j's the *detail* spaces. The index j is called the *scale*. At a large (or coarse) scale, one views the signal on a broad global level, and

at a small (or fine) scale, one looks at the signal on a local, detailed level. Another important notion is the *resolution* of a signal, which, for a finite-length signal, is the minimum number of a samples required to represent it [60]. This notion can be explained more clearly through examples of multirate systems. When a signal is filtered by a halfband lowpass filter, the scale remains unchanged, but the resolution is said to be halved, since there is a loss of information in general. When a signal is upsampled by two, followed by a halfband lowpass filter, the scale is halved (because the frequency has been effectively scaled by 1/2), and the resolution remains the same (because there is no gain or loss of information). Lastly, when a signal is filtered by a halfband lowpass filter followed by a downsampler of a factor of 2, the scale is doubled and the resolution is halved.

The multiresolution decomposition can be readily computed by the filter bank in Figure 2.1. Suppose the analysis filters are the time-reversed versions of $g_0[n]$ and $g_1[n]$, then the octave-band filter bank computes the inner product of the input with the basis functions of $W_1, W_2, \ldots, W_J$ and V_J. In the synthesis, we start from the component V_J at the coarsest scale, and sequentially add to it more and more details residing in the space $W_j, j = J, \ldots, 1$. From the previous discussion, it is clear that the output of the filter bank is the expansion onto an orthogonal basis. With this orthogonal basis, this expansion will be referred to as the orthogonal wavelet transform in this thesis.

2.1.2 Overcomplete Wavelet Expansion

At times, it may be desirable to have an *overcomplete* expansion, rather than a basis expansion, of the input signal. That is, the number of functions used in the expansion is more than needed for a basis, thus resulting in a redundant representation where the functions are linearly dependent. For compression purposes, such a redundancy may not be desirable since it increases the number of transform coefficients to code. In other applications, an overcomplete expansion may be more suitable than a basis expansion. For example, an overcomplete expansion places looser requirements on the filters, and may allow the design of filters with better frequency selections and/or symmetry properties. The wavelet expansion implemented by the *critically sampled* filter bank in Figure 2.1 is a time-varying system. With a non-subsampled filter bank, for example, this time-variance can be completely avoided.

The overcomplete expansion that will be relevant for this work is the expansion

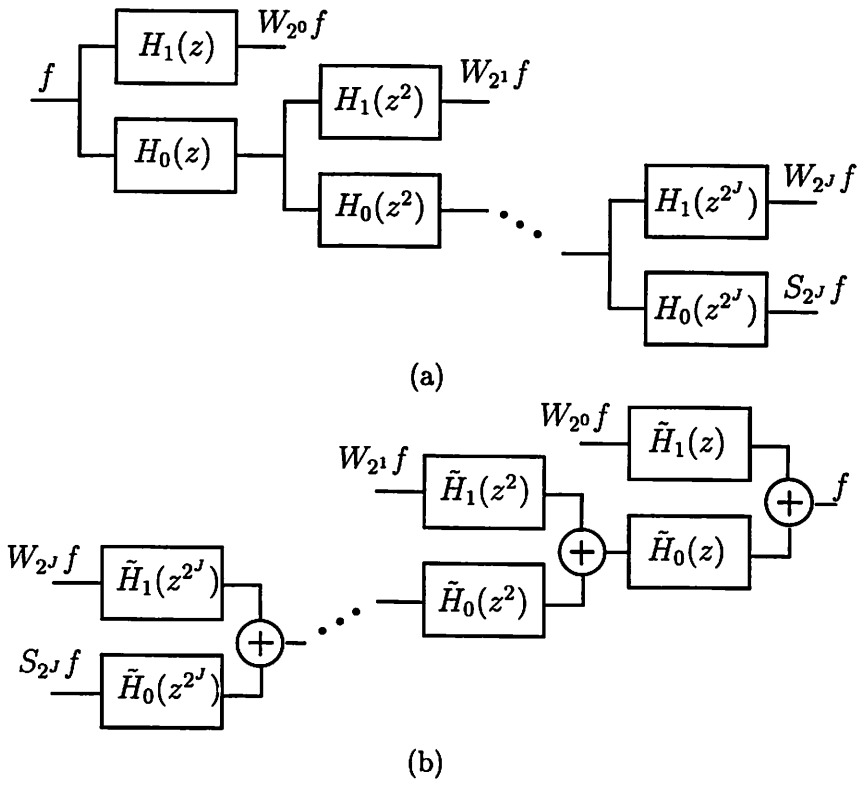


Figure 2.3: 1-D Non-subsampled filter bank. (a) Analysis. (b) Synthesis.

implemented by the *non-subsampled filter bank*, whose analysis and synthesis parts are shown in Figure 2.3. Schematically, the only difference between Figure 2.3 and 2.1 is the removal of the downsamplers in the analysis stage and the upsamplers in the synthesis stage. For the filters to be perfect reconstructing, they must satisfy

$$\tilde{H}_0(z)H_0(z) + \tilde{H}_1(z)H_1(z) = 1. \tag{2.2}$$

Note that (2.2) is a less stringent requirement than (2.1). Thus, the filters satisfying the perfect reconstruction properties of the critically sampled filter bank in (2.1) satisfy (2.2) as well (up to a scaling factor), but the converse is not true in general.

2.1.3 Two-Dimensional Wavelet Expansion

The most commonly used 2-D wavelet expansion is accomplished by separable 1-D filtering in both the horizontal and vertical directions. The separable 2-D filters are easier

to design and suffice for most applications and thus are used here. Interested readers in nonseparable filters are referred to [33, 60].

The separable 2-D octave-band wavelet transform is implemented by the filter bank in Figure 2.4 (a), where each stage is composed of a cascade of horizontal and vertical filtering. The frequency tiling of a one-stage decomposition is shown as well. Starting with $LL_0 = f$, the original image, the first stage decomposition generates 4 subbands, HH_1, HL_1, LH_1, and LL_1. The labelling, for example, HL_1, means the output from a highpass horizontal filtering and a lowpass vertical filtering, at stage 1. Subsequent stages are iterated on $LL_j, j = 1, 2, ..., J$. For the sake of visualization and storage, it is often convenient to arrange the subband coefficients in Figure 2.5. The synthesis filter bank is also an iterated filter bank, each stage composing of a cascade of vertical and horizontal filtering (in that order). Figure 2.4 (b) shows one stage of synthesis filter bank.

As with the 1-D case, in some applications it is desirable to have an overcomplete 2-D expansion. The non-subsampled 2-D filter bank is similar to Figure 2.4, but with the downsampler removed, and one stage of each of the analysis and synthesis filter banks are shown in Figure 2.6.

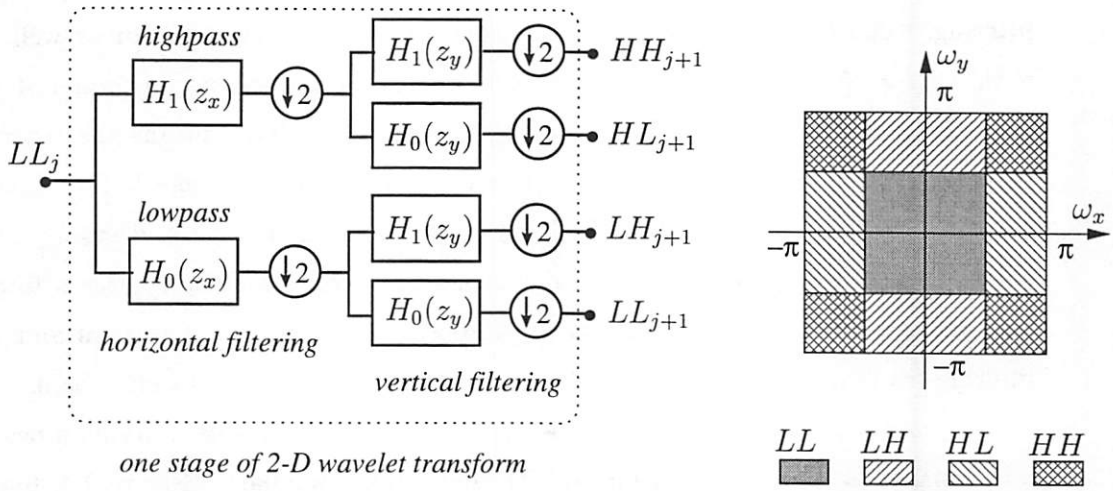
2.2 Wavelet Thresholding: Overview of Existing Work

In many engineering problems, for reasons such as finite precision or compression, it is necessary to consider coefficients below a certain threshold as negligible. This idea of thresholding coefficients is often more of an art than science. In recent years, this simple technique has been applied to removing noise from corrupted signals, or "denoising", and it it has been shown to have near-optimal theoretical properties. The theoretical formalization of the threshold denoising technique, particularly in the context of removing noise via thresholding wavelet coefficients, was pioneered by Donoho and Johnstone [22]. The simplicity and effectiveness of wavelet thresholding has spawned much interest in both theory and practice.

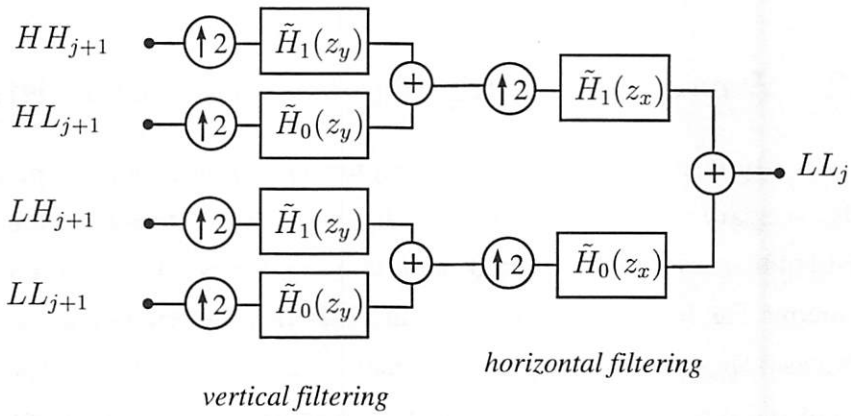
The problem at hand is that we observe a corrupted image

$$g_{ij} = f_{ij} + \varepsilon_{ij}, \quad i, j = 1, \ldots, N, \tag{2.3}$$

where $\{f_{ij}\}$ is the original image we wish to recover, $\{\varepsilon_{ij}\}$ are independent and identically distributed (*iid*) as normal $N(0, \sigma^2)$ and independent of $\{f_{ij}\}$, and N is an integral power



(a)



(b)

Figure 2.4: One stage of the 2-D separable filter bank shown in (a), with the partition of the frequency spectrum. The octave-band division of the frequency is achieved by iterating on the LL_j channel. (b) One stage of the synthesis filter bank.

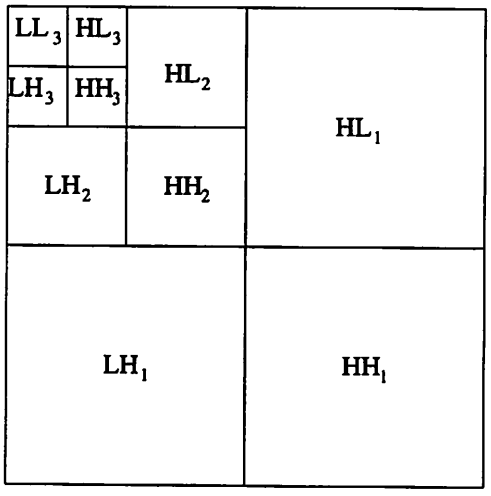


Figure 2.5: The subband coefficients (shown for $J = 3$) for the filter banks in Figure 2.4 are often arranged in this fashion for ease of visualization and storage.

of 2. To simplify notations, the image is assumed to be a square of size $N \times N$, though it is not a necessary requirement. The goal is to remove the noise and to obtain an estimate $\{\hat{f}_{ij}\}$ of $\{f_{ij}\}$, or to denoise $\{g_{ij}\}$.

Let $\boldsymbol{g} = \{g_{ij}\}_{i,j}$, $\boldsymbol{f} = \{f_{ij}\}_{i,j}$, $\boldsymbol{\varepsilon} = \{\varepsilon_{ij}\}_{i,j}$, that is, the boldfaced letters will denote the matrix representation of the signals under consideration. Let $\boldsymbol{Y} = \mathcal{W}\boldsymbol{g}$ denote the matrix of the wavelet coefficients of $\boldsymbol{g}$, where $\mathcal{W}$ is the two-dimensional orthogonal wavelet transform operator, and similarly $\boldsymbol{X} = \mathcal{W}\boldsymbol{f}$ and $\boldsymbol{V} = \mathcal{W}\boldsymbol{\varepsilon}$. Note that since the transform is orthogonal, $\{V_{ij}\}$ are also *iid* $N(0, \sigma^2)$.

Define the *soft-threshold* function to be

$$\eta_T(x) = \text{sgn}(x) \cdot \max(|x| - T, 0),$$

which takes the argument and shrinks it towards zero by the value T, called the *threshold.* A popular alternative is the *hard-threshold* function,

$$\psi_T(x) = x \cdot \mathbf{1}\{|x| > T\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. The hard-threshold function keeps the input if it is larger than the threshold T; otherwise, it is set to zero.

The wavelet thresholding procedure for denoising as proposed by Donoho and Johnstone consists of three stages:

1. Take the wavelet transform of the observation: $\boldsymbol{Y} = \mathcal{W}\boldsymbol{g}$.

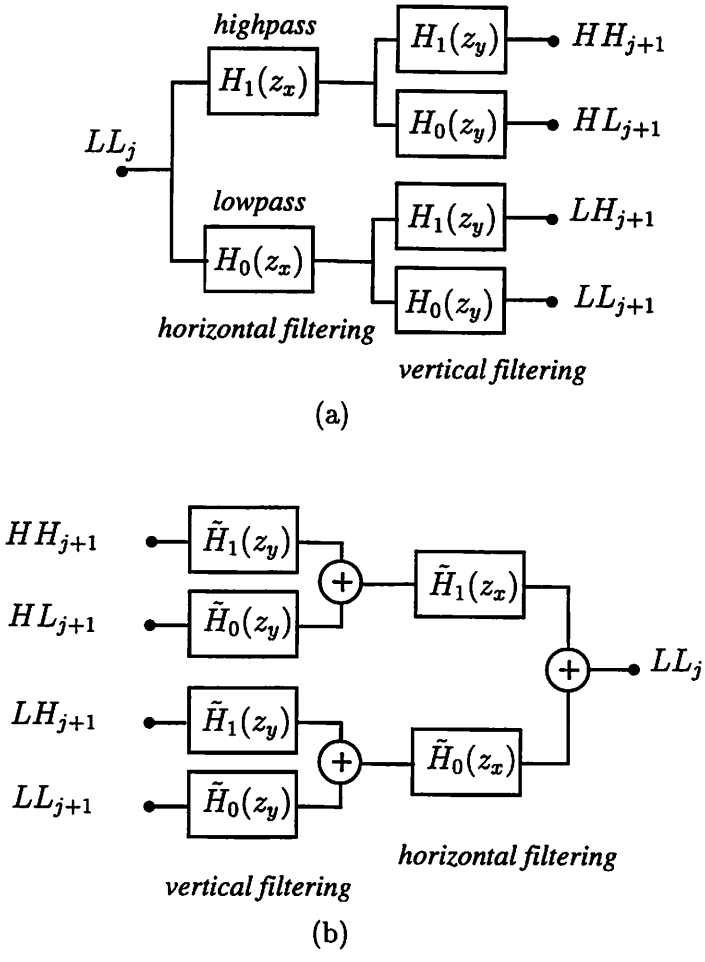


Figure 2.6: One stage of 2-D non-subsampled filter bank. (a) Analysis. (b) Synthesis.

2. Threshold the wavelet coefficients (except the lowest resolution subband LL_J) by either the hard- or soft-thresholding function for a chosen threshold T: $\hat{X}_{ij} = \eta_T(Y_{ij})$

3. The denoised estimate $\hat{f}$ is the inverse wavelet transform of the thresholded coefficients: $\hat{f} = \mathcal{W}^{-1}\hat{X}$.

Threshold denoising is especially effective for signals with sparse representations in the transform domain. Like the Fourier transform, the wavelet transform also has good energy compaction properties, so in general, large coefficients correspond to dominant signal features, while small coefficients correspond to fine details. When noise is added, the wavelet coefficients are perturbed. If the noise energy is low, then the perturbation is small, and only the very small coefficients should be killed. On the other hand, if the noise energy is high, only the very large coefficients should be kept so at least the dominant features are discernible in the recovered signal. The threshold thus acts as an oracle determining whether a coefficient should be kept or modified (because it has more signal contribution than noise) or be killed (because noise dominates).

The threshold choice is one of the most researched areas in the wavelet thresholding literature. Depending on the signal and noise models, there are various proposed methods for selecting a threshold. In the following, we provide a brief overview of several major methods found in the literature. The following notations are for 1-D signals to make the notations less cumbersome, but the idea can be extended in a straightforward manner to 2-D signals. That is, for the remainder of this chapter, the noisy observations of the 1-D signal is

$$g_i = f_i + \varepsilon_i \quad i = 1, \ldots, N,$$

where N is an integral power of 2. The noise samples $\{\varepsilon_i\}$ is *iid* $N(0, \sigma^2)$, independent of $\{f_i\}$, unless mentioned otherwise. The wavelet coefficients of $\{g_i\}$, $\{f_i\}$, and $\{\varepsilon_i\}$ will be denoted by $\{Y_i\}$, $\{X_i\}$, and $\{V_i\}$, respectively.

2.2.1 Deterministic Signal with Random Noise

Donoho and Johnstone [22] proposed an universal threshold $T_U = \sigma\sqrt{2\log N}$ for hard-thresholding when N samples have been corrupted by *iid* noise of $N(0, \sigma^2)$. This threshold is chosen because for a large N, the maximum amplitude of the noise coefficients, $\{V_i\}$, has a high probability of being smaller than, but close to, $\sigma\sqrt{2\log N}$. More precisely,

for $T_U = \sigma\sqrt{2\log N}$,

$$\lim_{N\to+\infty} \Pr\left(T_U - \frac{\sigma \log\log N}{\log N} \leq \max_{0\leq i<N} |V_i| \leq T_U\right) = 1.$$

Thus, thresholding with T_U has a high probability of removing the noisy coefficients in the asymptotic sense, and it is a conservative choice. To assess the performance of the estimator, $\hat{\boldsymbol{X}}^U$, based on hard-thresholding using the threshold T_U, let us first define the ideal diagonal projection estimator[2]. Consider among all diagonal projection estimators of the form $\hat{X}_i^{DP} = \gamma_i Y_i$ with $\gamma_i = 0$ or 1. The ideal estimator (minimizing $E\|\hat{\boldsymbol{X}}^{DP} - \boldsymbol{X}\|^2$) is obtained by setting $\gamma_i = 1\{|X_i| > \sigma\}$ and the associated expected squared error, or *risk*, is

$$E\|\hat{\boldsymbol{X}}^{DP} - \boldsymbol{X}\|^2 = \sum_{i=1}^{N} \min(X_i^2, \sigma^2).$$

Such an ideal estimator cannot be used since it requires the knowledge of the original signal $\boldsymbol{X}$, but it serves as an useful benchmark. The estimator $\hat{\boldsymbol{X}}^U$ can be shown to yield a risk which satisfies

$$E\|\hat{\boldsymbol{X}}^U - \boldsymbol{X}\|^2 \leq (2\log N + 1)(\sigma^2 + \sum_{i=1}^{N} \min(X_i^2, \sigma^2)).$$

That is, its risk comes to within a factor of $\log N$ of the risk due to the ideal diagonal projection estimator. Furthermore, for all estimators of the form $\tilde{X}_i = \theta(Y_i)$ for any function $\theta(\cdot)$,

$$\inf_{\tilde{\boldsymbol{X}}} \sup_{\boldsymbol{X}\in\mathbb{R}^N} \frac{1}{2\log N} \frac{E\|\tilde{\boldsymbol{X}} - \boldsymbol{X}\|^2}{\sigma^2 + \sum_i \min(X_i^2, \sigma^2)} \longrightarrow 1 \text{ as } N \to \infty.$$

This means that the best estimator $\tilde{\boldsymbol{X}}$ yields a maximum risk over all $\boldsymbol{X} \in \mathbb{R}^N$ that grows as $2\log N$ of the ideal risk. Thus, the hard-threshold estimate with T_U is asymptotically optimal in the *minimax* sense (minimizing the maximum error). A similar result can also be obtained for using soft-thresholding with T_U. In [21], $\hat{\boldsymbol{X}}^U$ was also shown to be near-minimax for various smoothness classes (such as Besov, Holder, Sobolev and Triebel classes).

The aforementioned method yields a single threshold T for all coefficients, regardless of the scale and spatial location. Thresholds which are dependent on the scale of the wavelet transform were also addressed by [24] and [30], with the latter considering correlated noise. Another notable threshold, the SURE threshold [23], is derived by minimizing

[2]Since the transform is orthogonal, the performance of the estimator can be discussed in either the signal domain or the transform domain.

Stein's unbiased risk estimator [57]. The hybrid threshold refers to switching between the SURE and the universal threshold, depending on the energy level of the coefficients. The method *SureShrink* refers to using the hybrid threshold in a scale dependent manner [23], and it has been found to perform better than using the universal threshold, while retaining asymptotically optimal properties.

2.2.2 Probabilistic Bayesian Modeling of the Signal

A large class of signals and natural images has been observed to have decaying spectra. This means that most of the signal energy is concentrated in the low frequency portion, or, visually, the slowly-varying smooth part of the signal. The high frequency energy corresponds to additional details manifested in sharp transitions such as edges or busy textures. In the wavelet domain, where the detail coefficients capture the local-varying nature of the signal, this translates to many small value coefficients and a relatively few large coefficients, resulting in a distribution with a peak at zero and often symmetric about zero. It has been widely accepted in the image subband coding community that the coefficients in each detail subband collectively form a histogram well described by a Generalized Gaussian distribution (GGD) (see, for example, [64, 39, 56, 37, 68]). This distribution has a density function

$$GG_{\alpha,\beta}(x) = C(\alpha,\beta)\, e^{-(\alpha|x|)^{\beta}}, \quad -\infty < x < \infty, \tag{2.4}$$

where $C(\alpha,\beta) = \frac{\alpha\beta}{2\Gamma(\frac{1}{\beta})}$ and $\Gamma(t) = \int_0^{\infty} e^{-u}u^{t-1}du$ is the gamma function. In applications of compression, the estimates of the parameters α and β are used to adapt the coder and the quantizer. For more tractable analysis, this assumption is often simplified to the Laplacian distribution, which is $GG_{\alpha,1}(x)$ (for example, see [59, 68]).

For denoising applications, Simoncelli and Adelson [56] used GGD to model the distribution of the wavelet coefficients of the original image, and to find the Bayesian estimate of the image. In the statistics community, there have also been many works on using a Bayesian model on the wavelet coefficients, with distributions mimicking the property of being symmetric about zero and having a sharp peak at zero. Vidakovic [61] used the Laplacian distribution to determine the shrinkage factor, which is a number multiplied to each wavelet coefficient and whose magnitude depends on the magnitude of the considered coefficient. Clyde *et al* [15], Chipman *et al* [14] and Abramovich *et al* [1] used a scaled mixture of normal priors to find scale-dependent shrinkage factors for Bayesian estimates.

Ruggeri and Vidakovic [52] examined the hard-thresholding rule for various combinations of different distributions for the signal and noise (such as a Laplacian-distributed signal with Gaussian noise) and found the corresponding thresholds based on minimizing the expected squared error.

2.2.3 Nonparametric Methods

Cross-validation Cross-validation is a classical statistics method used in various statistical settings to automatically choose the parameters of the problem at hand. The general paradigm is to minimize the prediction error generated by comparing a prediction, based on a subset of the data, to the remainder of the data. Some general references on cross-validation can be found in [5, 62]. In the setting of the threshold selection, it allows the threshold to be selected using the data only, without the knowledge of the noise energy, σ^2. In [48], Nason used cross-validation to find the threshold for the soft-thresholding rule in the following manner.

From the noisy observations, $\{g_1, g_2, \ldots, g_N\}$, a subsequence is formed from the even-indexed samples:

$$g_i^{\text{EVEN}} = g_{2i}, \quad i = 1, 2, \ldots, N/2 \; .$$

Let $\{\hat{f}_{T,i}^{\text{EVEN}}, i = 1, 2, \ldots, N/2\}$ be the wavelet threshold estimates of the even samples of $\{f_i\}$ (using a particular threshold T) based on $\{g_i^{\text{EVEN}}\}$. An interpolated version of the odd-indexed samples is computed as $\{\bar{g}_i^{\text{ODD}}\}$:

$$\bar{g}_i^{\text{ODD}} = \begin{cases} \frac{1}{2}(g_{2i-1} + g_{2i+1}) \; , & i = 1, 2, \ldots, N/2 - 1 \\ \frac{1}{2}(g_1 + g_{N-1}) \; , & i = N/2 \end{cases}$$

A similar computation is performed to yield $\{\hat{f}_{T,i}^{\text{ODD}}\}$, the threshold estimate based on the odd-indexed subsequence $\{g_i^{\text{ODD}}\}$, and $\{\bar{g}_i^{\text{EVEN}}\}$, the interpolated version of the even-indexed samples. A cross-validatory estimate of the mean squared error is

$$\hat{M}(T) = \sum_{i=1}^{N/2} \left[(\hat{f}_{T,i}^{\text{EVEN}} - \bar{g}_i^{\text{ODD}})^2 + (\hat{f}_{T,i}^{\text{ODD}} - \bar{g}_i^{\text{EVEN}})^2 \right] \; .$$

In this way, the threshold estimate based on the even-index of the data ($\{\hat{f}_{T,i}^{\text{EVEN}}\}$) is compared to the interpolated estimate of the odd-index of the data ($\{\hat{g}_i^{\text{ODD}}\}$), and vice versa.

Let T^* be the argument which minimizes $\hat{M}(T)$. Notice that $\hat{M}(T)$ relies on the estimates $\{\hat{f}_{T,i}^{\text{EVEN}}\}$ and $\{\hat{f}_{T,i}^{\text{ODD}}\}$, each of which is based on $N/2$ data points rather than N

points. To correct for this sample size[3], a heuristic adjustment is made. Specifically, the threshold T^* is multiplied by

$$C_N = \left(1 - \frac{\log 2}{\log N}\right)^{-1/2}$$

to yield the final cross-validatory threshold of Nason, where C_N is the constant satisfying $T_U(N) = C_N \cdot T_U(N/2)$, and $T_U(N) = \sigma\sqrt{2\log N}$ is the universal threshold for N data points. This cross-validatory threshold has been reported to be close to the optimal one (in the sense of minimizing the mean squared error) but tend to overfit the noisy data [61]. It also does not perform well in heavy-tailed noise distribution [48].

Weyrich and Warhola [65] proposed a generalized cross-validation criterion which finds a threshold T minimizing the expression

$$GCV(T) = \frac{\frac{1}{N}\|\boldsymbol{Y} - \eta_T(\boldsymbol{Y})\|^2}{(\frac{N_0}{N})^2},$$

where N_0 is the number of coefficients that have been set to zero by the thresholding procedure. Jansen *et al* [29] showed that this threshold choice is asymptotically optimal in the mean squared sense. That is, the minimizer of $GCV(T)$ also minimizes the mean squared error for a large N. Other variants of cross-validation and generalized cross-validation can be found in [63, 28].

Combination with MDL Saito [54] approached the threshold selection problem by posing it as a model selection problem. The two most important issues encountered when modeling a set of data are the choice of the model family and the order selection of the model. One solution is to use the MDL principle [50] to make this decision. In [54], a large library of orthogonal bases is available for the wavelet decomposition and the signal is denoised by wavelet thresholding. The question becomes which basis to choose and how many coefficients to be thresholded to zero. A criterion based on the MDL principle is used to make these choices. Saito's approach is related to our algorithm in Chapter 4, and will be discussed in more details therein. Several subsequent works also addressed the threshold selection problem from the MDL standpoint [34, 2, 45, 46].

[3]This correction is done to conform to the universal threshold of Donoho and Johnstone which is dependent on N and is asymptotically $\sigma\sqrt{2\log N}$.

Chapter 3

Bayesian Threshold Selection for Image Denoising

As surveyed in Section 2.2, there are many works especially in the statistics literature addressing different signal and noise models and the corresponding threshold selection or shrinkage factor. Most of these works experimented on one-dimensional signals, and thus the signal models may not be appropriate for images. Some Bayesian-based works played with different combinations of probability distributions, more for the sake of trying different models rather than examining real signals. Thus, due to a lack of models tailored for images, we proceed to find a threshold more suitable for our framework of image denoising. Our approach to finding the threshold is Bayesian, where *a priori* each detail subband of the signal is modeled with the Generalized Gaussian distribution (GGD) with fixed unknown parameters, also used widely in the image processing literature [64, 39, 56, 37, 68]. Within each subband, the goal is to find the threshold which minimizes the mean squared error among soft-threshold estimators. We propose an adaptive estimation of the threshold which is nearly optimal and is easy to compute. This threshold adapts based on the GGD parameter estimation for each subband, thus resulting in a different threshold for each subband. It will also be shown that with the chosen prior, the optimal soft-threshold estimator yields a lower mean squared error than the optimal hard-threshold estimator, and hence we use soft-thresholding in the image denoising algorithms in this thesis. The Bayesian framework and threshold selection described in this chapter will be the basis for the various denoising algorithms developed in Chapters 4, 5, and 6.

3.1 Signal Modeling and Threshold Selection

Recall that the noisy observation in (2.3) is

$$g_{ij} = f_{ij} + \varepsilon_{ij}, \quad i,j = 1,\ldots,N,$$

where $\{\varepsilon_{ij}\}$ are *iid* noise distributed as $N(0,\sigma^2)$, independent of the original signal $\{f_{ij}\}$. This is the setting which will be used throughout this thesis. The goal is to obtain an estimate $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$ which minimizes the mean squared error (MSE),

$$\frac{1}{N^2}\sum_{i,j}(\hat{f}_{ij} - f_{ij})^2.$$

Since the wavelet transform we choose is orthogonal, minimizing the MSE in the space domain is equivalent to minimizing the MSE in the transform domain. Thus, in the subsequent text we will work mostly in the wavelet domain. The wavelet coefficients for $\boldsymbol{g}$, $\boldsymbol{f}$, and $\boldsymbol{\varepsilon}$ are $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{V}$, respectively.

Firstly, we choose the soft-threshold estimate over the hard-threshold estimate. In practice, because the hard-thresholding rule tends to yield "blips" (or spikes) in the recovered image especially when the noise energy is significant, soft-thresholding is preferred here since it yields visually more pleasant images even if it tends to smooth out the image slightly more. These blips are typically in the forms of ringing around the edges or shot-noise like appearances in the smooth regions. These artifacts are more apparent under a hard-thresholding operation because it is a discontinuous function, whereas the soft-thresholding function is continuous. Furthermore, for the Bayesian prior assumed in this work, the optimal soft-thresholding estimator yields a smaller MSE than the optimal hard-thresholding estimator, as will be shown later. While the mean squared error is not necessarily a good measure for discriminating image qualities, it is nevertheless the most widely used standard in the literature, and thus will be employed here as well.

With the estimates restricted to the class of soft-threshold estimates, $\hat{X}_{ij} = \eta_T(Y_{ij})$, the next step is to find the appropriate threshold T. To do this, we use a Bayesian setting, where, for a given subband, each coefficient X_{ij} is viewed as a random variable having the Generalized Gaussian distribution with unknown parameters (see Section 2.2.2 for a qualitative justification for the choice of GGD). For completeness, this probability density function is repeated here:

$$\mathrm{GG}_{\alpha,\beta}(x) = C(\alpha,\beta)\, e^{-(\alpha|x|)^\beta}, \quad -\infty < x < \infty, \tag{3.1}$$

where $C(\alpha,\beta) = \frac{\alpha\beta}{2\Gamma(\frac{1}{\beta})}$ and $\Gamma(t) = \int_0^\infty e^{-u}u^{t-1}du$ is the gamma function. The parameter β controls the shape of the density function, and the parameter α controls the spread. With a probability distribution, the MSE can be approximated by the *expected* squared error,

$$\frac{1}{\tilde{N}^2}\sum_{i,j}(\hat{X}_{ij} - X_{ij})^2 \approx E(\hat{X} - X)^2 \ ,$$

where $\tilde{N}^2$ is the number of terms in the particular subband under consideration. The expectation $E(\cdot)$ is taken with respect to $X \sim GG_{\alpha,\beta}(x)$ and $V \sim N(0,\sigma^2)$, and the estimator is $\hat{X} = \eta_T(Y)$, with $Y = X + V$. The threshold selection then corresponds to finding the value which minimizes the expected squared error. Note that for each detail subband, the GGD has a different set of parameters α and β, thus this procedure results in a subband-adaptive threshold.

Consider now only coefficients from one particular detail subband. Let the parameters α and β be known for now. The distortion criterion to be minimized is the expected squared error, or *risk*, rewritten as

$$R(T) = E_X E_{Y|X}(\hat{X} - X)^2 \ ,$$

where $Y|X \sim N(x,\sigma^2)$. The optimal threshold T^* is the argument which minimizes $R(T)$. To our knowledge, there is no closed form solution for T which minimizes $R(T)$ for this chosen prior. Thus, we resort to numerical calculations to find the optimal answer.

Before examining the general case, it is insightful to consider two special cases of the GGD: the Gaussian and the Laplacian distributions. The Laplacian case is particularly interesting, because it is frequently used as the simplified distribution for wavelet coefficients to make analysis more tractable.

Case 1: (Gaussian) For $\beta = 2$ and α parameterized as $\alpha = 1/(\sqrt{2}\sigma_x)$ (where σ_x is the standard deviation), we have the Gaussian distribution, $X \sim N(0,\sigma_x^2)$. It is straightforward to verify that

$$\begin{aligned} E_X E_{Y|X}(\hat{X} - X)^2 &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\eta_T(y) - x)^2 p(y|x)\,p(x)dy\,dx \\ &= \sigma^2 w(\frac{\sigma_x^2}{\sigma^2}, \frac{T}{\sigma}) \end{aligned} \tag{3.2}$$

where

$$w(\sigma_x^2, T) = \sigma_x^2 + 2(T^2 + 1 - \sigma_x^2)\,\bar{\Phi}\left(\frac{T}{\sqrt{1+\sigma_x^2}}\right) - 2T(1+\sigma_x^2)\phi(T, 1+\sigma_x^2),$$

with $\phi(x,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{x^2}{2\sigma^2})$ and $\bar{\Phi}(x) = \int_x^\infty \phi(t,1)\,dt$.

A good approximation of the optimal threshold T^* is found to be

$$\tilde{T} = \frac{\sigma^2}{\sigma_x}. \tag{3.3}$$

Figure 3.1 (a) compares T^* and $\tilde{T}$, parameterized by σ_x on the horizontal axis, and $\sigma = 1$. Their expected risks are shown in Figure 3.1 (b), where the maximum deviation from the optimal risk is less than 1% when using the threshold $\tilde{T}$. For a further comparison, the risk for hard-thresholding is also calculated. After some algebra, it can be shown that the risk for hard-thresholding is

$$r_h(T) = \sigma^2 + (\sigma^2 - \sigma_x^2)(2T\ \phi(T,\sigma_x^2+\sigma^2) + 2\bar{\Phi}(\frac{T}{\sqrt{\sigma_x^2+\sigma^2}}) - 1). \tag{3.4}$$

By setting to zero the derivative of (3.4) with respect to T, the optimal threshold is found to be

$$T_h^* = \begin{cases} 0 & \text{if } \sigma_x > \sigma \\ \infty & \text{if } \sigma_x < \sigma \\ \text{anything} & \text{if } \sigma_x = \sigma \end{cases},$$

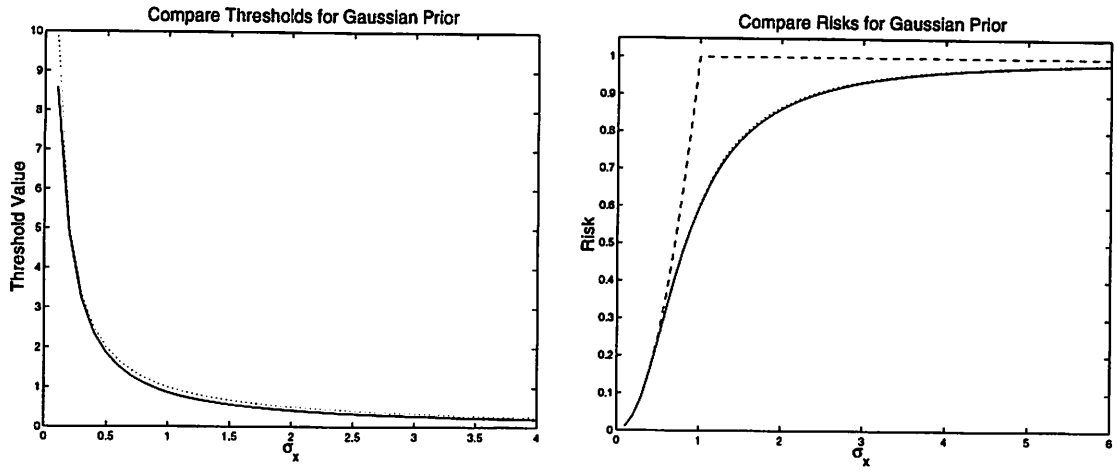
with the associated risk

$$R_h(T_h^*) = \begin{cases} \sigma^2 & \text{if } \sigma_x > \sigma \\ \sigma_x^2 & \text{if } \sigma_x \le \sigma \end{cases}.$$

Figure 3.1(b) shows that both the optimal and near-optimal soft-threshold estimators, $\eta_{T^*}(\cdot)$ and $\eta_{\tilde{T}}(\cdot)$, achieve lower risks than the optimal hard-threshold estimator.

The threshold $\tilde{T} = \sigma^2/\sigma_x$ is not only nearly optimal but also has an intuitive appeal. For such a choice, the normalized threshold $\tilde{T}/\sigma$ is inversely proportional to σ_x, the standard deviation of X, and proportional to σ, the noise standard deviation. When σ/σ_x is small relative to 1, the signal is much stronger than the noise, thus $\tilde{T}/\sigma$ is chosen to be small in order to preserve most of the signal and remove some of the noise; vice versa, when σ/σ_x is much larger than 1, the noise dominates and the normalized threshold is chosen to be large to remove the noise which has overwhelmed the signal. Thus, this threshold choice adapts to both the signal and noise characteristics reflected in the parameters σ and σ_x.

Case 2: (Laplacian) For $\beta = 1$ and $C(\alpha,\beta) = \alpha/2$, we have the Laplacian distribution $\mathrm{LAP}(x) = \frac{\alpha}{2}e^{-\alpha|x|}$. Note that the variance of X is $2/\alpha^2$.



(a) (b)

Figure 3.1: Thresholding for the Gaussian prior, with $\sigma = 1$. (a) Compare the optimal threshold $T^*(\sigma_x)$ (solid —) and the threshold $\tilde{T}(\sigma_x)$ (dotted $\cdots$) as a function of the standard deviation σ_x on the horizontal axis. (b) Compare the risks of optimal soft-thresholding (—), soft-thresholding with $\tilde{T}$ ($\cdots$), and optimal hard-thresholding ($- - -$).

Without loss of generality, let $\sigma = 1$. The optimal threshold T^* found by minimizing the risk[1] is plotted against the standard deviation $\sigma_x = \sqrt{2}/\alpha$ on the horizontal axis in Figure 3.2 (a). The curve corresponding to T^* (in solid line —) is compared with the approximate threshold $\tilde{T} = 1/\sigma_x = \alpha/\sqrt{2}$ (in dotted line $\cdots$) in Figure 3.2 (a). Their corresponding expected risks are shown in Figure 3.2 (b), and the deviation is less than 0.8%. This suggests that the risk at the minimum is not too sensitive to the threshold value.

For a general value of σ, the parameters T and α are replaced by T/σ and $\sigma\alpha$, respectively, and the proposed threshold is

$$\tilde{T} = \frac{\sigma^2}{\sigma_x} = \frac{\sigma^2\alpha}{\sqrt{2}}, \tag{3.5}$$

which has the same form as the Gaussian case in Equation (3.3), but with different parameters.

The threshold choice

$$\tilde{T}_h = \frac{2\sqrt{2}\sigma^2}{\sigma_x} = \frac{2\sigma^2}{\alpha}$$

[1]Note that for numerical calculation, it is more robust to obtain the value of T^* from locating the zero-crossing of the derivative, $R'(T)$, than from directly minimizing $R(T)$.

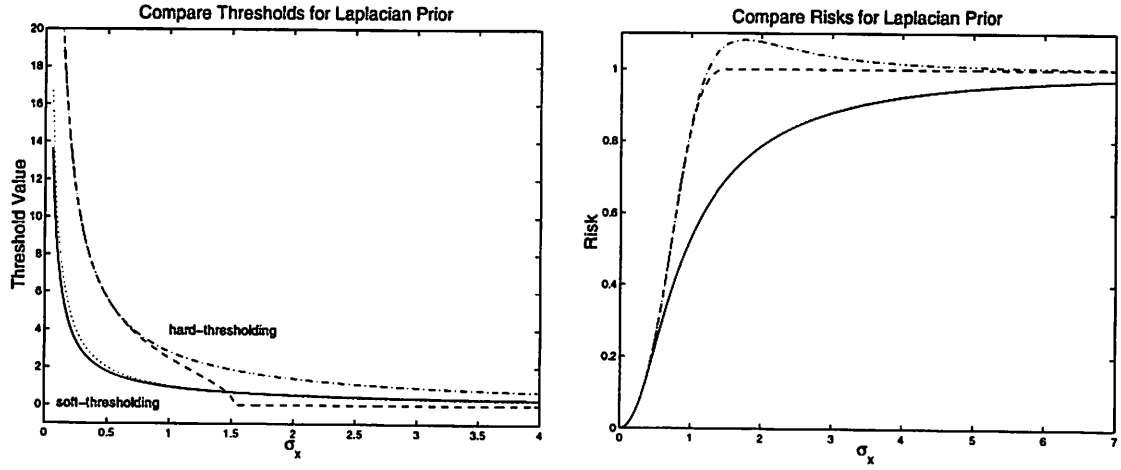


Figure 3.2: Thresholding for the Laplacian prior, with $\sigma = 1$. (a) Compare the optimal soft-threshold T^* (—), the approximation $\tilde{T}$ $(\cdots)$, the optimal hard-threshold T_h^* $(- - -)$, and its approximation $\tilde{T}_h$ $(- \cdot -)$ as a function of the standard deviation, σ_x, on the horizontal axis. (b) Their corresponding risks.

was found independently in [52] for approximating the optimal hard-threshold T_h^* using the same prior. Figure 3.2 compares the optimal soft- and hard-thresholds and their approximations, and it shows the soft-thresholding rule to yield a lower risk for this chosen prior. In fact, for σ_x larger than approximately 1.3σ, the risk of hard-thresholding with the approximate threshold, $\tilde{T}_h$, is worse than if no thresholding were performed (which has a risk of σ^2).

Case 3: (Generalized Gaussian) Similarly, our proposed near optimal threshold is

$$\tilde{T}(\alpha, \beta) = \frac{\sigma^2}{\sigma_x} = \sigma^2 \sqrt{\frac{\alpha^2 \Gamma(\frac{1}{\beta})}{\Gamma(\frac{3}{\beta})}}$$

for the GGD case. Let $\sigma = 1$. In Figure 3.3 (a), each dotted line $(\cdots)$ is the optimal threshold $T^*(\alpha, \beta)$ for a given fixed β, plotted against σ_x on the horizontal axis, with α varying. The proposed threshold $\tilde{T} = 1/\sigma_x$ is plotted with the solid line (—). The plot of the optimal threshold that lies closest to $\tilde{T}$ is the curve for $T^*(\alpha, \beta = 1)$, the Laplacian case, while other curves deviate from $\tilde{T}$ as β moves away from 1. Figure 3.3 (b) shows the corresponding risks. The deviation between the optimal risk $R(T^*)$ and $R(\tilde{T})$ grows as β moves away from 1, but the error is still within 5% for the curves shown in Figure

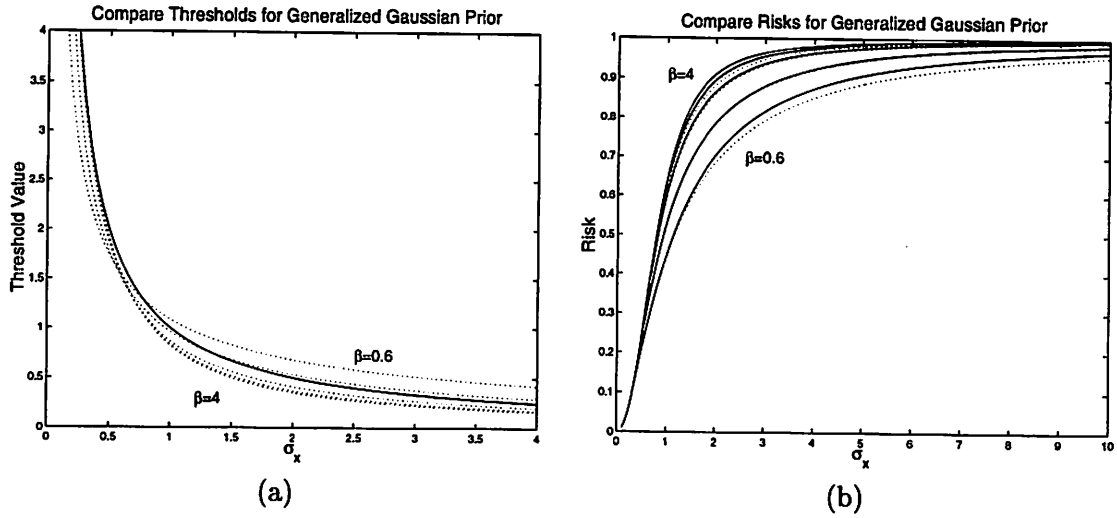


Figure 3.3: Thresholding for the Generalized Gaussian prior, with $\sigma = 1$. (a) Compare the approximation $\tilde{T} = \sigma^2/\sigma_x$ (—) with the optimal threshold for $\beta = 0.6, 1, 2, 3, 4$ $(\cdots)$. The horizontal axis is the standard deviation, σ_x. (b) The optimal risks for each β are plotted in $(\cdots)$, and the approximation in (—).

3.3 (b). Because the threshold $\tilde{T}$ depends only on the standard deviation and not on the shape parameter β, it may not yield a good approximation for other values of β than the range tested here, and the threshold may need to be modified to incorporate β. However, since in practice the values of $\beta = 1, 2$ are typically used in modeling wavelet coefficients of real images, well within the range of β tested here, the simple form of the threshold $\tilde{T}$ is appropriate for our purpose. The curve of expected squared error is very flat near the optimal threshold T^*, implying that the error is not very sensitive to a slight perturbation near T^*.

3.1.1 Parameter Estimation for Threshold

In the discussion thus far, we have assumed the parameters of the distribution to be known. We now discuss the estimation of these parameters, which in turn yield thresholds $\tilde{T}$ adaptive to different subband characteristics.

The first step is to estimate the noise variance, σ^2. In some practical cases, it is possible to measure σ^2 based on information other than the corrupted observation. If this is not the case, we estimate it by using the robust median estimator in the highest subband

of the wavelet transform,

$$\hat{\sigma} = \frac{\text{Median}(|Y_{ij}|)}{0.6745}, \quad Y_{ij} \in \text{ subband } HH_1, \tag{3.6}$$

also used in [22, 23].

Next, to obtain an estimate of σ_x, recall that our model is $Y = X + V$, with X and V being zero-mean and independent of each other, therefore,

$$\begin{aligned} \text{Variance}(Y) &= \text{Variance}(X) + \text{Variance}(V) \\ &= \sigma_x^2 + \sigma^2 \ . \end{aligned}$$

Thus, $\hat{\sigma}_x$ can be obtained by

$$\hat{\sigma}_x^2 = \max(\hat{m}_2 - \hat{\sigma}^2, 0) \ ,$$

where $\hat{m}_2$ is the estimate of the second moment of Y,

$$\hat{m}_2 = \frac{1}{\tilde{N}^2} \sum_{i,j} Y_{ij}^2,$$

and $\tilde{N}^2$ is the number of coefficients in this subband. In the rare case that $\hat{\sigma}^2 \geq \hat{m}_2$, the threshold is effectively set to ∞; that is, all coefficients are set to 0.

For the proposed threshold $\tilde{T} = \sigma^2/\sigma_x$, it suffices to have the estimates $\hat{\sigma}_x$ and $\hat{\sigma}^2$. However, to be complete, we describe the method to obtain estimates of α and β as well. These parameters can be found from the second and the fourth moments of the distribution [56]:

$$m_2 = \int_{-\infty}^{\infty} y^2 p(y) dy \quad \text{and} \quad m_4 = \int_{-\infty}^{\infty} y^4 p(y) dy.$$

Since $Y = X + V$, it can be derived that

$$m_2 = \sigma^2 + \frac{\Gamma(\frac{3}{\beta})}{\alpha^2 \Gamma(\frac{1}{\beta})} \quad \text{and} \quad m_4 = 3\sigma^4 + \frac{6\sigma^2 \Gamma(\frac{3}{\beta})}{\alpha^2 \Gamma(\frac{1}{\beta})} + \frac{\Gamma(\frac{5}{\beta})}{\alpha^4 \Gamma(\frac{1}{\beta})}. \tag{3.7}$$

The moments are found empirically by

$$\hat{m}_2 = \frac{1}{\tilde{N}^2} \sum_{i,j} Y_{ij}^2 \quad \text{and} \quad \hat{m}_4 = \frac{1}{\tilde{N}^2} \sum_{i,j} Y_{ij}^4 \ .$$

The parameters α and β can be found by solving (3.7) with $\hat{m}_2$ and $\hat{m}_4$ in place of m_2 and m_4.

The Generalized Gaussian distribution offers more flexibility in the description of the subband coefficients. In practice, the Laplacian prior performs well, and it also leads to simple closed-form equations, thus it is sometimes preferred in coding applications. For example, Birney and Fischer [4] showed that in image coding, the quantizer based on the Generalized Gaussian distribution gives marginal improvements over that based on the Laplacian so they recommend the Laplacian distribution to be used for its simplicity and analytical tractability. For the Laplacian case, $\beta = 1$, and $m_2 = \sigma^2 + \frac{2}{\alpha^2}$, the parameter α can be estimated as

$$\hat{\alpha} = \sqrt{\frac{2}{\hat{m}_2 - \hat{\sigma}^2}}.$$

3.2 Summary

In this chapter, we addressed the threshold selection in a Bayesian approach. In each subband, the wavelet coefficients of the signal is modeled by the Generalized Gaussian distribution with unknown parameters. We found that the simple threshold

$$\tilde{T} = \frac{\sigma^2}{\sigma_x}$$

is nearly optimal and is simple to compute. This simple and effective threshold will be used in the denoising algorithms to be discussed in subsequent chapters, where we will also show its performance on real images. For now, the threshold is developed with the assumption that the wavelet transform coefficients in each subband collectively form a histogram distributed as the GGD. In Chapter 5, the spatially adaptive algorithm will discuss how to model the coefficients as random fields (with changing parameter), and thus yield a threshold selection that is not only subband-adaptive, but also pixel-wise adaptive as well.

Chapter 4

Lossy Compression and Wavelet Thresholding for Denoising

An obvious question which arises during denoising is how one distinguishes between signal and noise. If appropriate models exist for both the signal and noise, then this can be done effectively (for example, if the power spectrum of the signal and noise are known, then Wiener filtering can be used.) However, it is not straightforward to devise a general model for images, since they are rather complicated objects. Stochastic models are oftened used to represent an image as samples of a random field. Such models are often used for image restoration and data compression (see [27] for a survey of stochastic models and their applications). While these random field representations can be applied to image restoration and compression, by themselves they do not amount to a general image model. For example, a random field representation can describe self-similar or texture-like images [25, 26], but it may not predict a sharp transition (such as an edge) because it is an "unexpected" event. Thus, without additional modeling, a stochastic representation can only model a rather restrictive class of images.

For a more general model, we look into an area of image processing which has been rather successful, namely, image compression. Notably, subband coding such as EZW [55] and its variants have achieved high compression rate with good visual qualities. The fact that these compression methods are able to capture important image features with a concise representation implies that they achieve an efficient modeling of the image, where efficiency is quantified in terms of the description complexity. On the other hand, an image

of uncorrelated white noise is hard to compress for any coder, because there is no structural correlation or redundancy to exploit. Hence, a good compression method can provide a suitable model for distinguishing between signal and noise.

The idea of using a lossy compression algorithm for denoising has been proposed in several works [54, 49, 12, 13, 35]. Saito [54] viewed wavelet (hard-) thresholding as a means to achieve "simultaneous noise suppression and signal compression." The noise suppression nature of wavelet thresholding is already clear from the discussion in Section 2.2 and Chapter 3. It also achieves compression because after thresholding, there are less non-zero coefficients left to be coded. Natarajan's Occam filter [49] accomplishes denoising by coding the signal at a distortion equal to the noise strength, σ^2. The coder is chosen arbitrarily, as long as it is a "reasonable" one. Liu and Moulin [35] proposed a "complexity-regularized" denoising method, which codes the signal at a particular slope on the rate-distortion curve. The value of the slope is derived from an MDL-like criterion, while the coder is chosen also arbitrarily. To avoid disrupting the flow of this introduction, we save the details of these algorithms until Section 4.1 where previous works in the literature are surveyed, and also Section 4.2 where they are relevant to our work.

One main purpose of the work in this chapter is to explain and to further substantiate the theory that lossy compression can be appropriate for denoising. Most coders operate in an orthogonal transform domain such as wavelet or DCT, and this is also what we assume. Specifically, by posing quantization as an approximation to wavelet thresholding, we show that quantization (a common step in compression) of wavelet transform coefficients achieves denoising. We do not claim that lossy compression is the best way to denoise an image, but rather we want to show how to achieve both compression and denoising when both features are desired.

To make analogies between wavelet thresholding and lossy compression, we reiterate here the essence behind the idea of threshold denoising. The thresholding method compares the transform coefficients to a given threshold and set it to zero if its magnitude is less than the threshold; otherwise, it is kept or modified (depending on the thresholding rule). The idea is that coefficients insignificant relative to the threshold are likely due to noise, whereas significant coefficients are important signal structures. Thresholding essentially creates a region around zero where the coefficients are considered negligible. Outside of this region, the thresholded coefficients are kept to full precision.

Analogously, in a typical transform domain lossy compression method, negligible

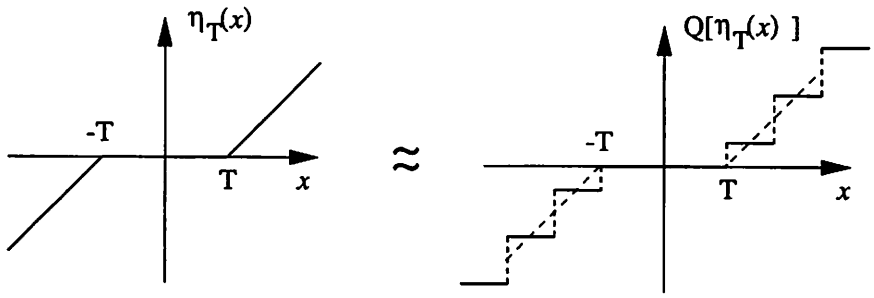


Figure 4.1: The thresholding function can be approximated by quantization with a zero-zone.

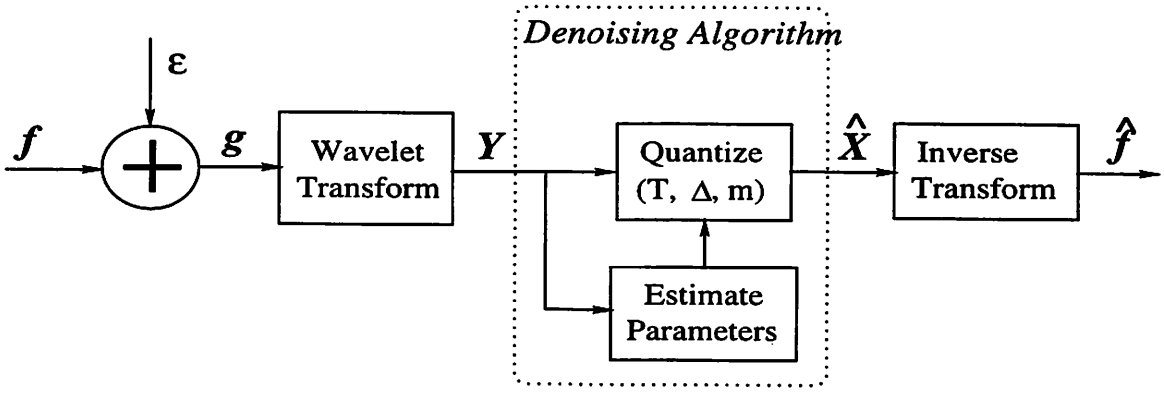


Figure 4.2: Problem formulation and proposed method for denoising. The noisy observation is the signal with additive noise. Denoising is achieved in the wavelet transform domain by a combination of soft-thresholding and quantizing the wavelet coefficients, with the specifications based on estimated model parameters.

coefficients are set to zero, creating what is called a "zero-zone" or "dead-zone", and coefficients outside of this zone are quantized. Our thesis is that *an appropriate quantization scheme (and hence compression) achieves denoising because it is an approximation to the thresholding operation* (see Figure 4.1). Furthermore, the effectiveness of denoising is mainly due to the zero-zone, and the full precision of the thresholded coefficients is of secondary importance. Thus, a comparable level of denoising performance can be achieved by quantizing the coefficients with a zero-zone and a few number of quantization levels outside of the zero-zone.

The problem formulation and our proposed denoising method are shown in Figure 4.2. The noise samples $\{\varepsilon_{ij}\}$ are *iid* normal $N(0, \sigma^2)$ and independent of the signal $\{f_{ij}\}$. Our denoising operation is done in the wavelet transform domain of the observed corrupted

signal. In essence, it is a two-stage quantization involving the zero-zone and the region outside of the zero-zone. Furthermore, the quantization procedure is adaptive to each subband, where we first estimate parameters to characterize the subband, and then use this information to determine the quantization specifications. For each subband, the size of the zero-zone is set by the chosen threshold value T, and then the region outside is quantized with $2m$ symmetric bins of width Δ. The quantized coefficients are then transformed back to yield the estimate. Thus, the two main issues in the quantization stage are "How to choose the threshold (and hence the zero-zone) ?" and "How to quantize outside of the zero-zone?"

We answer the first question with the Bayesian threshold we developed in Chapter 3. That is, the transform coefficients from each subband are modeled as random variables with Laplacian distribution (which is more tractable than the GGD). Based on this characterization, a simple threshold is used in the first stage soft-thresholding.

After being thresholded, the non-zero coefficients are good estimates of the original signal and thus close to being Laplacian-distributed. They are then quantized with uniform bin sizes and centroid reconstruction, and the number of bins is determined by a criterion derived from Rissanen's MDL principle. This criterion achieves a compromise in the trade-off between the compression rate (from coding the bins) and the distortion, and has a nice interpretation of operating at a fixed slope on the rate-distortion curve of the coder.

This chapter is organized as follows. In Section 4.1, the other previous works on lossy compression-based denoising will be discussed in more detail. Next, in Section 4.2 we develop our lossy compression method for denoising, which incorporates wavelet thresholding, coefficient quantization, and entropy coding. Experimental results on several test images will be shown in Section 4.3. In Section 4.4, we make some concluding remarks about our findings and possible future directions.

4.1 Related Previous Work

In Saito's work, the idea is to hard-threshold the wavelet coefficients of the noisy observation to achieve both denoising and compression. The decision on the number of coefficients to keep is made by deriving a criterion based Rissanen's MDL principle [50] (see (4.3)) and evaluating this criterion. The formulation is related to ours and thus will be expounded in Section 4.2. While this work proposes an interesting idea, we feel that it does

not achieve true compression, since the coefficients are not quantized (which is necessary in any practical coder).

In the denoising method of Liu and Moulin [35], the MDL criterion is viewed as a trade-off between rate and distortion during coding, operated at a particular slope on the rate-distortion curve. Their contribution is finding this slope, but no guidelines is provided for choosing the coder. Rather, they merely presented a comparison of several popular coders such as JPEG and SPIHT operated at this slope, against the Occam filter and Donoho's hard-thresholding with the universal threshold T_U. Their work is also related to ours and it will be revisited in Section 4.2. In our algorithm, however, we will present a systematic approach to choosing the coder.

Natarajan's Occam filter [49] removes noise by coding the noisy observation (with an arbitrary coder) at a distortion equal to σ^2. This particular choice of distortion is based on the following intuition. When the distortion is small, the coder tracks the small details in the signal (and thus the noise); when the distortion is large, it tracks the more global structure of the signal. At the distortion point σ^2, there is a "knee" on the rate-distortion curve, representing the change in the tracking behavior of the signal. This knee refers to a rapid change in the slope of the rate-distortion curve (manifested as the maximum of its second derivative). Thus, by examining the rate-distortion characteristics of the observation, one first estimates σ^2 by locating the knee, then compresses it at this distortion. One can also interpret this method as finding an estimate of the signal on the hyper-sphere of radius σ centered at the noisy observation $\boldsymbol{g}$ [31]. The original signal must reside on this hyper-sphere because it was corrupted by noise of energy σ^2. However, since the dimensionality is so large (about a quarter of a million for a typical 512×512 image), it may be very difficult to find an estimate in the vicinity of the original signal. Our experience with the Occam filter using standard coders such as JPEG and SPIHT [53] has not been satisfactory, especially with large values of σ^2 (a range of σ values between 10 and 20 were tested for *iid* Gaussian noise of $N(0, \sigma^2)$, with greyscale test images). The resulting images are extremely distorted. Furthermore, we have found the knee to exist in the rate-distortion curve of a noiseless image as well, making the knee argument dubious. Nevertheless, the intuition posed by this work gives an invaluable insight for motivating compression-based denoising.

4.2 Quantization and Compression using the MDL Principle

If there is no compression required, then the adaptive thresholding rule discussed in Section 3.1 suffices to effectively denoise the image. To achieve the dual purpose of denoising and compression, however, there is an additional step of quantization. Recall that compression achieves denoising because the zero-zone in the quantization step (typical in compression methods) effectively removes the noise. Hence, after the zero-zone has been fixed, there is an additional step of quantizing the thresholded coefficients.

Consider again only one detail subband of the wavelet transform, with the coefficients modeled as Laplacian distributed. Suppose that the parameters σ, α and the threshold T^* have been estimated (see Section 3.1 for the parameter estimation and the threshold choice $\tilde{T} = \sigma^2/\sigma_x$). There remains the questions of how to quantize the coefficients outside of the zero-zone and how to compress them.

When compressing a signal, two important objectives are to be kept in mind. On the one hand, the distortion between the compressed signal and the original should be kept low; on the other hand, the description of the compressed signal should use as few resources as possible (e.g. use the least number of bits to code). Typically, these two criteria are conflicting requirements. In order to reach a compromise, there needs to be a criterion for selecting the most suitable outcome. Rissanen's *Minimum Description Length* principle serves exactly this purpose [50].

4.2.1 The MDL Principle

Let $\mathcal{M}$ be a library or class of models from which we choose the "best" one to represent the data. According to the MDL principle, given a sequence of observations, the best model is one which yields the shortest description length for describing the data using the model, where the description length is the number of bits needed for encoding. This description can be accomplished by a two-part code: one part to describe the model and the other the description of the data using the model. To develop some intuition about this principle, let us consider several scenarios. In order to minimize the distortion between the signal and the model, we can choose the signal itself as the model, in which case the signal is represented exactly, but at the cost of using many bits to encode the signal. In our case where the input of the coder is the corrupted observation, this choice implies no denoising, thus it is useless. At the other extreme, we can use the zero function, which

needs essentially zero bits to encode, at the expense of high distortion (unless the signal is also identically zero). In the middle ground, a parametric model (such as polynomial fitting) may be chosen, yielding a total description length to be the number of bits needed to code the data given the model (e.g. the residual), plus the bits needed to specify the number of parameters and the parameter values. The idea is that the chosen model should establish a compromise between fitting the data well and having low complexity, that is, having a simple representation or a reasonable number of parameters.

Example 1 *For a sequence of discrete random variables, $u_1, u_2, \ldots, u_N$, with distribution $p(u), u \in \mathcal{U}$, where $\mathcal{U}$ is a finite or a countable set, the shortest code-length on average is the well-known Shannon code, $L(u) = -\log p(u)$, with base 2 in the logarithm (for the rest of the chapter, the* log *function is of base 2). For the entire sequence, the shortest code-length is*

$$L(\boldsymbol{u}) = \sum_{i=1}^{N} L(u_i) = -\sum_{i=1}^{N} \log p(u_i).$$

Example 2 *Now let us consider when $\{u_i\}$ are continuous variables, with density $p(u)$, $u \in \mathcal{U}$, and $\mathcal{U}$ is a subset of $\mathbb{R}$. The set $\mathcal{U}$ can be discretized into equal intervals of size or precision δ. Let u_i^δ be the discretized random variable, then the Shannon code for the vector $\boldsymbol{u}^\delta$ is $-\log(p(\boldsymbol{u}^\delta) \cdot \delta^N) = -\log p(\boldsymbol{u}^\delta) - N \log \delta$. This term can be viewed as the ideal code-length for coding $\boldsymbol{u}$ at precision δ. With the value δ fixed, $-\log p(\boldsymbol{u})$ is called the* idealized code-length. *Rissanen [50] showed that the optimal δ is on the order of $1/\sqrt{N}$ for parametric models.*

Let us now state the MDL principle. Given the set of observations $\boldsymbol{Y}$, we wish to find a model $\hat{\boldsymbol{X}}$ to describe it. The MDL principle chooses $\hat{\boldsymbol{X}}$ which minimizes the two-part code-length

$$L(\boldsymbol{Y}, \hat{\boldsymbol{X}}) = L(\boldsymbol{Y}|\hat{\boldsymbol{X}}) + L(\hat{\boldsymbol{X}}), \tag{4.1}$$

where $L(\boldsymbol{Y}|\hat{\boldsymbol{X}})$ is the code-length for $\boldsymbol{Y}$ based on $\hat{\boldsymbol{X}}$, and $L(\hat{\boldsymbol{X}})$ is the code-length for $\hat{\boldsymbol{X}}$.

The first term on the right-hand side of (4.1) is the idealized code-length,

$$L(\boldsymbol{Y}|\hat{\boldsymbol{X}}) = -\log p(\boldsymbol{Y}|\hat{\boldsymbol{X}}). \tag{4.2}$$

In the original MDL, further truncation of the model parameters is considered. Suppose that there are K parameters and each parameter X_i is truncated up to precision δ, yielding

X_i^δ, then Rissanen showed that the optimal precision δ^* is $1/\sqrt{N}$ and

$$\min_\delta L(\boldsymbol{Y}, \hat{\boldsymbol{X}}, \delta) = L(\boldsymbol{Y}|\hat{\boldsymbol{X}}) + L([\hat{\boldsymbol{X}}]) + \frac{K}{2}\log N + O(K),$$

where $L([\hat{\boldsymbol{X}}])$ denotes the code-length of the integer part of $\hat{\boldsymbol{X}}$, and $(1/2)\log N$ bits are used to represent the decimal part of each of the K parameter values. When K is much less than N, the last term $O(K)$ is negligible, and the MDL criterion is

$$MDL(\boldsymbol{Y}, \hat{\boldsymbol{X}}) = L(\boldsymbol{Y}|\hat{\boldsymbol{X}}) + L([\hat{\boldsymbol{X}}]) + \frac{K}{2}\log N. \tag{4.3}$$

In practice, the parameter values in MDL are rarely optimally truncated but kept to full machine precision.

In Saito's simultaneous compression and denoising method [54], the hard-threshold function was used to generate the models $\hat{X} = \psi_T(Y)$, where the number of K non-zero coefficients to retain is determined by minimizing the MDL criterion. The first term $L(\boldsymbol{Y}|\hat{\boldsymbol{X}})$ is the idealized code-length with the normal distribution (see (4.4)), and the second term $L([\hat{\boldsymbol{X}}])$ is taken to be $K \log N$, which are the bits needed to indicate the location of each non-zero coefficient (assuming a uniform indexing). Although compression has been achieved in the sense that a fewer number of nonzero coefficients are kept, it still does not address the issue that in a practical compression setting, the coefficients usually need to be quantized more coarsely. Thus, our criterion will be developed from a coding point of view, and the minimization of $L(\boldsymbol{Y}, \hat{\boldsymbol{X}})$ is restricted to $\hat{\boldsymbol{X}}$ belonging to the set of quantized signals, whose construction will become clear in the following.

4.2.2 The MDL Principle for Compression-based Denoising: The MDLQ Criterion

Consider only one particular subband, which is of size $\tilde{N} \times \tilde{N}$. Since the noisy wavelet transform coefficients are $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{V}$, where V_{ij} are *iid* $N(0, \sigma^2)$, then $Y_{ij}|X_{ij} \sim N(x_{ij}, \sigma^2)$. Thus,

$$\begin{aligned} L(\boldsymbol{Y}|\boldsymbol{X}) &= -\sum_{i,j=1}^{\tilde{N}} \log p(Y_{ij}|X_{ij}) \\ &= \frac{1}{2\sigma^2 \ln 2}\sum_{i,j=1}^{\tilde{N}} (Y_{ij} - X_{ij})^2 + \frac{1}{2}\log(2\pi\sigma^{2\tilde{N}^2}). \end{aligned} \tag{4.4}$$

The second term in (4.4) is a constant, and thus can be ignored in the minimization. The expression in (4.4) was also derived in [54, 35], though in [54] the estimation of σ^2 is integrated into the criterion as well. The main deviation between their works and ours is the different ways of estimating $\boldsymbol{X}$.

Let $\mathcal{M}$ be the set of quantized coefficients, $\boldsymbol{X}^Q$, and $\boldsymbol{X}$ be constrained in $\mathcal{M}$, then (4.4) (with constant terms removed) becomes

$$\begin{aligned} L(\boldsymbol{Y}|\hat{\boldsymbol{X}}^Q) &= -\sum_{i,j=1}^{\tilde{N}} \log p(Y_{ij}|\hat{X}_{ij}^Q) \\ &= \frac{1}{2\sigma^2 \ln 2} \sum_{i,j=1}^{\tilde{N}} (Y_{ij} - \hat{X}_{ij}^Q)^2. \end{aligned} \tag{4.5}$$

There are many possibilities for the second term $L(\hat{\boldsymbol{X}}^Q)$ in (4.1), since there are many ways for coding quantized coefficients. Here we propose a simple method suitable for subband coding of images.

Since the observation Y_{ij} has Gaussian noise embedded, it is not strictly Laplacian distributed. However, after thresholding as discussed in Section 3.1, the (non-zero) thresholded coefficients $\hat{\boldsymbol{X}} = \eta_T(\boldsymbol{Y})$ can be seen to be close to Laplacian distributed (see the text in Section 4.3 and the referenced plot in Figure 4.4), and thenceforth are quantized with this distribution. The problem of quantizing Laplacian random variables has been well-studied and the design of the entropy-constrained scalar quantizer (ECSQ) for a Laplacian distribution is discussed by Sullivan [59]. Furthermore, it was shown that the uniform threshold quantizer (UTQ) achieves nearly the performance of the ECSQ, and has the additional benefit of being simple to design.

Hence the UTQ is used here on the non-zero thresholded coefficients, with m levels of equal intervals of Δ on each side, and with the centroids being the reconstruction values (see Figure 4.3). The quantized coefficients are denoted by $\hat{X}_{ij}^Q$, and there are a total of $2m+1$ quantization levels (one zero-zone plus m symmetric levels on each positive and negative side), which are indexed as $\ell = -m, -m+1, \ldots, -1, 0, 1, \ldots, m-1, m$. Consider the positive side and let $b_0, b_1, \ldots, b_m$ denote the boundaries of the quantization bins, with reconstruction values $\gamma_1, \gamma_2, \ldots, \gamma_m$. Note that $b_0 = 0$ and $b_m = \infty$. The value of γ_ℓ with

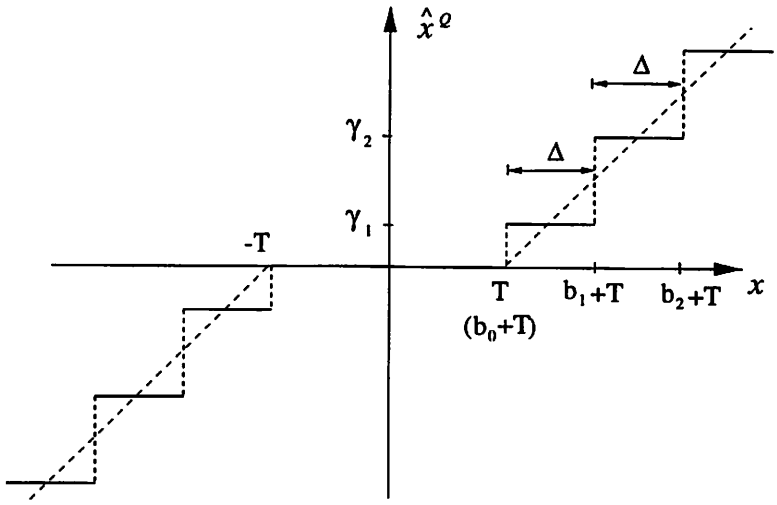


Figure 4.3: Illustrating the quantizer.

boundaries $b_{\ell-1}$ and b_ℓ is

$$\gamma_\ell = \frac{\int_{b_{\ell-1}}^{b_\ell} xp(x)dx}{\int_{b_{\ell-1}}^{b_\ell} p(x)dx} = \frac{1}{\alpha} + \frac{b_{\ell-1}e^{-\alpha b_{\ell-1}} - b_\ell e^{-\alpha b_\ell}}{e^{-\alpha b_{\ell-1}} - e^{-\alpha b_\ell}}.$$

The negative side is quantized in a symmetric way. Note that the zero coefficients resulting from thresholding are kept as zeros, and that the quantization of the non-zero coefficients does not set any additional coefficients to zero. On average, the smallest number of bits needed to code $\hat{\boldsymbol{X}}^Q$ is the Shannon code, and the code-length for coding the bin indices is

$$L(\hat{\boldsymbol{X}}^Q|m,\Delta) = -\sum_{\ell=-m}^{m} K_\ell \log \frac{K_\ell}{\tilde{N}^2},$$

where K_ℓ is the number of coefficients in bin ℓ. The additional parameters m and Δ need to be coded also, but we suppose that any positive values are equally likely, thus a fixed number of bits are allocated for $L(m,\Delta)$.

Now we state our model selection criterion:

$$MDLQ(\boldsymbol{X}^Q,m,\Delta) = \frac{1}{2\sigma^2 \ln 2}\sum_{i,j=1}^{\tilde{N}}(Y_{ij} - \hat{X}_{ij}^Q)^2 + L(\hat{\boldsymbol{X}}^Q|m,\Delta). \tag{4.6}$$

To find the best model, we minimize (4.6) over values of m and Δ to find the corresponding set of quantized coefficients, $\boldsymbol{X}^Q$.

This thresholding-quantization scheme is applied to each subband independently. First the noise variance $\hat{\sigma}^2$ is estimated. Then the parameter $\hat{\alpha}$ and the threshold $\tilde{T}(\hat{\alpha})$ are

calculated, and (4.6) is minimized over m and Δ to find the desired quantized coefficients $\boldsymbol{X}^Q$. The coarsest subband LL_J is quantized differently in that it is not thresholded, and the quantization with (4.6) uses the uniform distribution. The LL_J coefficients are essentially local averages of the image, and are not characterized by distributions with a peak at zero. Thus the uniform distribution is used for generality. The mean is subtracted from the LL_J coefficients to make the distribution centered at zero. The zero-zone is also of width Δ, with reconstruction value 0, and the reconstruction values in other zones are the midpoints of the intervals.

The MDLQ criterion in (4.6) has the additional interpretation of operating at a specified point on the rate-distortion curve, as also pointed out by Liu and Moulin [35]. For a given coder, one can obtain a set of operational rate-distortion points (R, D). When there is a rate or a distortion constraint, the constraint problem can be formulated into a minimization problem with a Lagrange multiplier, $\lambda D + R$. In this case, (4.6) can be interpreted as operating at

$$\lambda = \frac{1}{2\sigma^2 \ln 2} \, .$$

In the related works, Natarajan's coder operates at a constrained distortion, $D \leq \sigma^2$ [49], while Liu and Moulin's coder operates at the slope $\lambda = \frac{1}{2\sigma^2 \ln 2}$ on the R-D curve [35]. Both works merely recommend the use of any "reasonable" coder. In contrast, our work pinpoints the effectiveness of using compression for denoising to come from the zero-zone in the compression schemes.

4.3 Experimental Results

The 512×512 images "goldhill" and "lena", with various levels of noise $\sigma = 5, 10, 15, 20$, are used as test data. Daubechies' least asymmetric compactly-supported wavelet with 8 vanishing moments [19] is used in the wavelet transform, and four levels of decomposition are computed. The coefficient values of this *Symmlet8* wavelet are listed in Appendix A.

Firstly, to show that the thresholded coefficients are nearly Laplacian, the Q-Q plot of the HL_1 subband of goldhill is shown in Figure 4.4. The Q-Q plot is an effective statistical tool to verify that a data set is close to the assumed distribution. Let $x_1, x_2, \ldots, x_N$ be the sample values of $x \sim \text{LAP}(\alpha)$, and we order them by increasing magnitude, denoted by $y_1, y_2, \ldots, y_N$, called the order statistics of the absolute values. When $\alpha = 1$, $y = |x| \sim$

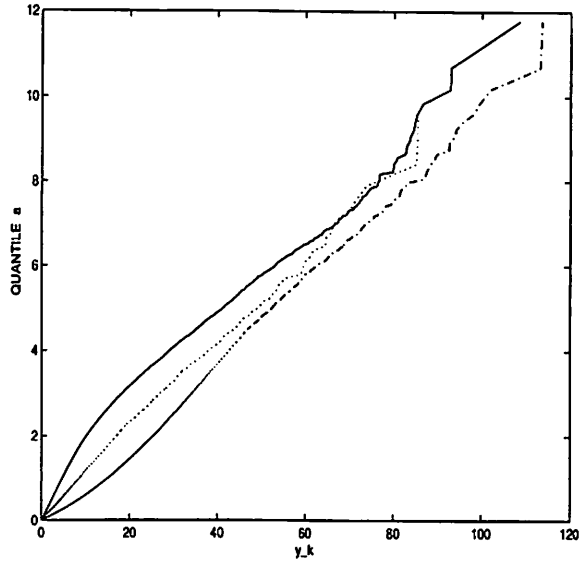


Figure 4.4: Q-Q plot of subband HL_1 of goldhill. Compares the original uncorrupted coefficients (—), the noisy coefficients (– · –·), and the non-zero thresholded coefficients (···).

EXP(1) $= e^{-y}$. The cumulative distribution function (cdf) of EXP(1) is $F(a) = \int_0^a e^{-y}dy = q$, and $a = F^{-1}(q)$ is called the q-th *quantile*. Then $a = -\ln(1-q)$. The Q-Q plot graphs the pairs $(y_k, -\ln(1 - \frac{k-0.5}{N})), k = 1, 2, \ldots, N$, where the number 0.5 is inserted to keep the log function well defined at the boundaries. The straighter the line is, the closer the samples are to the assumed distribution. For a general α, the line is of slope $1/\alpha$. Figure 4.4 compares the plot of the original uncorrupted coefficients, the corrupted coefficients, and the non-zero thresholded coefficients. The thresholded coefficients follows a straighter line than the noisy observation, suggesting a closer match to the Laplacian distribution. For this subband, the threshold $\tilde{T}$ is large, thus the lines deviate substantially from each other. For other subbands where the thresholds are small (also implying that the noise power is small relative to the signal), the three lines are close to each other, and show a good match to the Laplacian distribution.

To assess the performance of soft-thresholding using the adaptive threshold $\tilde{T}$, we compare it with soft-thresholding using the *oracle threshold* defined as

$$T_{\text{orc}} = \arg\min_T \sum_{i,j} (\eta_T(Y_{ij}) - X_{ij})^2 \tag{4.7}$$

where X_{ij} are assumed to be known, and a different T_{orc} is found for each detail subband.

In Table 4.1, the first column is the MSEs of the noisy observations, and the next two columns compare the MSEs of soft-thresholding with T_{orc} and $\tilde{T}$, respectively, averaged over 20 runs. The MSEs resulting from $\tilde{T}$ are very close to those from T_{orc}, indicating that the Laplacian pdf is a good model and that the threshold selection is appropriate. Visually, the two sets of images are also very similar, as shown in Figure 4.5 (b) and (c) for goldhill and $\sigma = 15$. These images are also available on the Internet at
`http://www-wavelet.eecs.berkeley.edu/~grchang/compressDenoise/`.

The fourth column in Table 4.1 shows the MSEs of the quantized signal using $\tilde{T}$ as the zero-zone threshold. The quantized goldhill image with $\sigma = 15$ is shown in Figure 4.5 (d), where the quantization noise is quite visible. The last column of Table 4.1 shows that, as expected, the quantized signal uses much less bits than the 8 bits per pixel (bpp) of the original greyscale image, but at the expense of some degradation. On average, the quantized signal loses about 1-1.5 dB in SNR over the unquantized thresholded signal, although it still has a much lower MSE than the noisy image. This suggests that mainly the zero-zone is responsible for filtering the noise. Note that the first-order entropy coding, $L(\hat{\boldsymbol{X}}^Q|m,\Delta)$, for the bitrate of the quantized coefficients is a rather loose estimate. With more sophisticated coding methods (e.g. predictive coding, pixel classification), the same bitrate could yield a higher number of quantization level m, thus resulting in a lower MSE.

Table 4.2 gives the values of m chosen by $MDLQ$ for each subband of the goldhill image, $\sigma = 15$, averaged over 20 runs. Recall that each subband has $2m + 1$ quantization levels. The $MDLQ$ criterion allocates more levels in the coarser, more important levels, as would be the case in a practical subband coding situation.

4.4 Summary

In this chapter, we demonstrated the connection between lossy compression and wavelet thresholding to explain why compression is suitable for denoising. Specifically, it is the zero-zone in coefficient quantization that is the main agent in removing the noise. Although the setting in this chapter was the wavelet domain, the idea can be extended to other transform domains such as DCT, which also relies on energy compaction and sparse representation properties to achieve good compression. Thus, lossy compression has the dual purpose of both removing the noise and compressing the input into fewer bits. Furthermore, the denoising experiments using our proposed thresholds for wavelet thresholding images,

Table 4.1: MSE of (1) the noisy observed image, (2) oracle soft-thresholding, (3) soft-thresholding with thresholds $\tilde{T}$, and (4) quantized signal with zero-zone thresholds $\tilde{T}$. The last column shows the entropy bitrate (bits per pixel) of the quantized image. Averaged over 20 runs.

MSE	observ.	T_{orc}	$\tilde{T}$	$\tilde{T}$, Quant.	bitrate (bpp)
goldhill σ=5	25	16.33	17.72	29.57	1.458
σ=10	100	41.15	41.82	58.64	1.058
σ=15	225	64.85	66.46	87.04	.679
σ=20	400	86.51	88.98	112.22	.445
lena σ=5	25	12.39	13.47	20.66	1.186
σ=10	100	28.15	29.58	42.49	.725
σ=15	225	43.80	45.75	63.73	.490
σ=20	400	59.13	61.44	83.79	.358

Table 4.2: The value of m (averaged over 20 runs) for the different subbands of Goldhill, with noise strength $\sigma = 15$.

Orientation	Scale			
	1 (fine)	2	3	4 (coarse)
HH	0	2.10	3.65	6.10
HL	2.75	3.95	5.95	20.05
LH	2.70	3.50	6.35	12.05
LL	34.65			

Figure 4.5: Comparing the performance of the various methods. Clockwise from top left: (a) Original. (b) Noisy image, $\sigma = 15$. (c) Oracle soft-thresholding. (d) Thresholding with $\tilde{T}(\hat{\alpha})$. (e) Our method of thresholding followed by quantization.

based on Laplacian distribution modeling of the subband coefficients, showed that these thresholds perform very close to the oracle results.

There are several interesting directions worth pursuing. The current scheme selects the threshold (i.e. zero-zone size) $\tilde{T}$ and the quantization bin size Δ in a two-stage process. In typical image coders, however, the zero-zone is chosen to be the same size or twice the size as other bins. Thus it would be interesting to jointly select these two values and analyze their dependencies on each other. Furthermore, a more sophisticated coder is likely to produce better compressed images than the current scheme, which uses the first order entropy to code the bin indices. With an improved coder, an increase in the number of quantization bins would not increase the bitrate penalty by much, and thus the coefficients would be quantized at a finer resolution than the current method. The model family $\mathcal{M}$ could also be expanded. For example, one could use a collection of wavelet bases for the wavelet decomposition, rather than using just one chosen wavelet, to allow possibly better representations of the signals. Lastly, the combination of the spatially adaptive thresholding method developed in Chapter 5 with coefficient quantization could yield a more sophisticated compression-based denoising algorithm. This is likely to both improve the denoising performance and reduce the bitrate.

Chapter 5

Spatially and Scale-Adaptive Image Denoising

Most of the wavelet thresholding literature thus far has concentrated on developing threshold selection methods, with the threshold being uniform or at best using a different threshold for each subband. Very little has been done on developing thresholds that are adaptive to different spatial characteristics. Other works investigate the choice of wavelet basis or expansion for the thresholding framework. One particularly interesting result is that (uniform) thresholding in a shift-invariant expansion (dubbed *translation-invariant (TI) denoising* by Coifman and Donoho [16]) eliminates some of the unpleasant artifacts introduced by the modification of the orthogonal wavelet expansion coefficients. In this chapter, we use the wisdom that thresholding in a shift-invariant, overcomplete representation outperforms thresholding in an orthogonal basis, and investigate an issue that has not been explored, namely, the spatial adaptivity of the threshold value.

To motivate spatially-adaptive thresholding, consider the example in Figure 5.1, where a square function has been corrupted by additive noise, and the goal is to recover the original function. The wavelet coefficients of the original and the noisy function are displayed in Figure 5.1(a). The noisy coefficients are soft-thresholded by a single threshold in Figure 5.1(b), and one can see that, especially in the finest scale, there are some coefficients corresponding to noise which have not been set to zero, and that some of these noisy coefficients are larger in magnitude than those coefficients corresponding to the signal. Thus, with a uniform threshold, it may not be feasible to have both the benefits of keeping

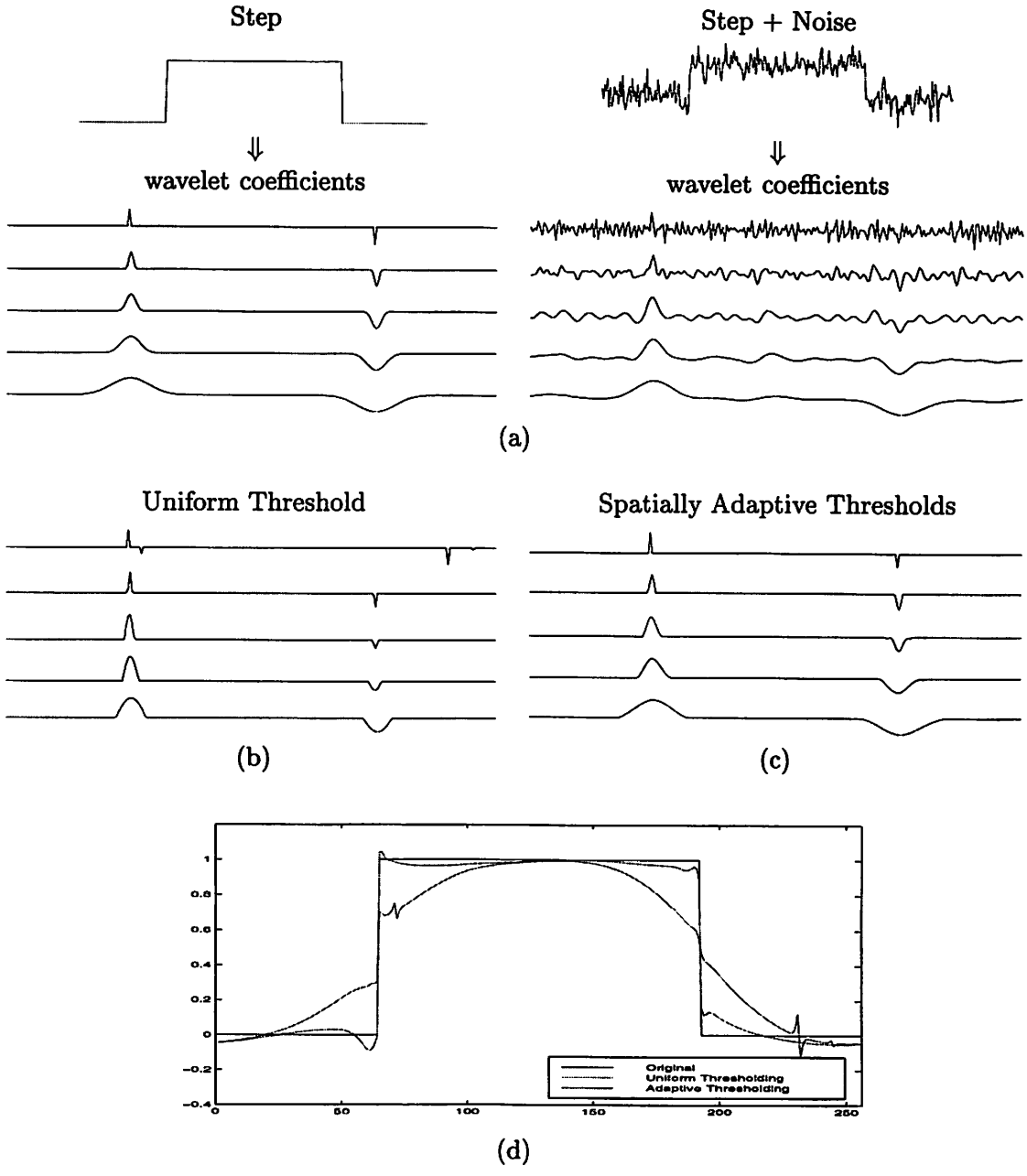


Figure 5.1: Motivation for adaptive thresholds. (a) shows a step function and its noisy version, along with their wavelet decomposition of 4 scales. The wavelet coefficients are thresholded by a uniform threshold in (b) and spatially adaptive thresholds in (c). The original and the reconstructions from (b) and (c) are shown in (d).

the important signal features and killing the noisy coefficients. On the other hand, one can reap both benefits with adaptive thresholds by choosing the threshold value to be very small in the regions of the peaks due to the step function, and large otherwise (see Figure 5.1(c))[1]. The reconstructed signals are shown in Figure 5.1(d), where it is clear that the adaptively thresholded reconstruction is much better than that due to uniform thresholding, especially in the area of the sharp transitions. The question, then, is how can one distinguish between the coefficients that are mainly due to the signal and those mainly due to the noise? Also, how should the thresholds be adjusted pixel by pixel? These are the questions that we will answer in this chapter with our proposed algorithm.

Most natural images have very different local properties, since they typically consist of regions of smoothness and sharp transitions. These regions of varying characteristics can be well differentiated in the wavelet domain, as can be seen in the wavelet decomposition of the *lena* image in Figure 5.2. One observes areas of high and low energy (or large and small coefficient magnitude), represented by white and black pixels, respectively. Areas of high energy correspond to signal features of sharp variation such as edges and textures; areas of low energy correspond to smooth regions. When noise is added, it tends to increase the magnitude of the wavelet coefficient on average. Specifically, in smooth regions, one expects the coefficients to be dominated by noise, thus most of these coefficients should be removed, especially since noise is highly visible there. In regions with sharp transitions, the signal has the main contribution to the high energy coefficients, while noise has less. These coefficients should be kept, or modified only a little, to ensure that most of the signal details are retained, and also because noise is not so visible here. Thus, the idea is to distinguish between the low and high energy regions, and modify the coefficients using a *spatially adaptive* thresholding strategy.

To accomplish spatially adaptive thresholding, we model each wavelet coefficient as GGD random variable whose parameter is to be estimated. This parameter in turn is used to find the appropriate threshold. Instead of using one parameter for each subband level, several wavelet-based image coders have achieved better performances by modeling the wavelet coefficients as a mixture of GGD random variables with unknown slowly spatially varying parameters [37, 68]. The estimation of the parameter for a given coefficient is conditioned on a function of its neighboring coefficients, a method called *context-modeling*

[1]For the sake of illustrating the effectiveness of varying thresholds, the regions of the true peaks are assumed to be known in this example.

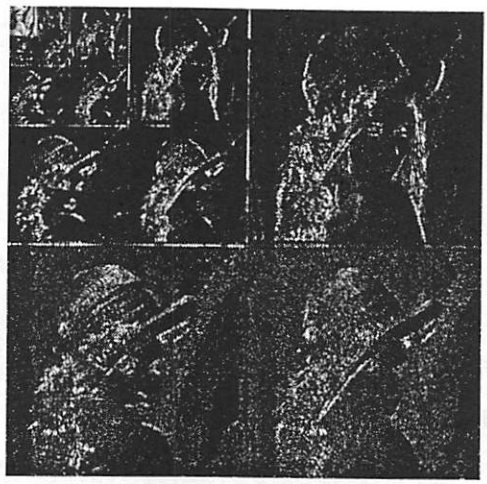


Figure 5.2: Four level wavelet decomposition of *lena*. White pixels indicate large magnitude coefficients, and black signifies small magnitude.

frequently used in compression for differentiating pixels of varied characteristics and adapting the coder. Context modeling also allows one to group pixels of similar nature but not necessarily spatially adjacent, and to gather statistical information from these pixels. Now, given that one can estimate the parameter for each coefficient, the next step is to use them to calculate the threshold. In Chapter 3, we found that when the signal coefficients are modeled as Generalized Gaussian random variables and the noise as Gaussian, the threshold $\tilde{T} = \sigma^2/\sigma_x$ is a good approximation to the optimal threshold which minimizes the mean squared error of the soft-thresholding estimator, where σ^2 is the noise power, and σ_x is the standard deviation of the signal. The simplicity of this threshold makes it easy to achieve spatial adaptivity: one uses context modeling to quantify the local characteristic in σ_x, which in turn yields a threshold $\tilde{T}$ adaptive on a pixel-by-pixel manner.

Our proposed adaptive algorithm is based on using adaptive thresholding in the overcomplete wavelet expansion. It outperforms both using only adaptive thresholding in the orthogonal expansion or using only uniform thresholding in the overcomplete expansion like the TI denoising. That is, *by combining spatially adaptive thresholding and overcomplete expansion, we achieve results which are significantly superior than either method alone.* Firstly, the adaptive threshold selection does a good job at removing noise in smooth regions, while not disturbing too much the edge and texture regions. Secondly, thresholding in the overcomplete expansion acts as an additional averaging which further attenuates the remaining noise.

The organization of this chapter is as follows. In Section 5.1.1, we introduce modeling each subband coefficient as a Generalized Gaussian distributed random variable with a different parameter. Because this threshold selection is based on the *iid* noise assumption, the discussion will first be set in the orthogonal wavelet transform. Then context modeling is introduced in Section 5.1.2 to allow the parameters to be estimated on a pixel level, which in turn yields a spatially adaptive threshold. The final adaptive algorithm will be complete when we discuss how to extend the adaptive thresholding in the orthogonal expansion to the overcomplete expansion in Section 5.1.3. There are several alternative approaches and related work which we have explored and the findings are discussed in Section 5.1.4. In Section 5.2, we will compare the spatially adaptive results with those from the best uniform thresholding strategy (in the mean squared error sense, and based on knowing the original image), in both the orthogonal and an overcomplete expansion. Results will show that the combination of using spatially adaptive thresholding and overcomplete expansion yields significantly better results in both visual quality and mean squared error.

5.1 Spatially Adaptive Algorithm

The adaptive algorithm will be developed in the following manner. To make this thresholding approach spatially adaptive, *each coefficient* (rather than each subband) is modeled as a GGD random variable with a different unknown parameter which is estimated via context modeling. This spatial mixture of distributions allows the image characteristics to be quantified locally in the distribution parameters, which are then used to adjust the threshold for each coefficient. Lastly, since the aforementioned algorithm is developed in the orthogonal expansion where the coefficients are uncorrelated, the algorithm will need to be modified to extend to the overcomplete expansion where coefficients are correlated. Several remarks will also be made about related alternative approaches.

5.1.1 Coefficient Modeling and Threshold Selection

The corrupted image modeled in (2.3) will be re-iterated here for completeness. The observed degraded image is

$$g_{ij} = f_{ij} + \varepsilon_{ij}, i, j = 1, \ldots, N,$$

where $\{f_{ij}\}$ is the original image, and $\{\epsilon_{ij}\}$ are *iid* $N(0,\sigma^2)$ and independent of $\{f_{ij}\}$. The observations $\{g_{ij}\}$ are transformed to the wavelet domain for threshold denoising. Let $\{Y_{ij}^{(s,o)}, i,j = 1,\ldots,N/2^s\}$, denote the wavelet coefficients of $\{y_{ij}\}$ at a particular scale s and orientation o, where $s = 1,2,\ldots,J$ and $o \in \{HL, LH, HH, LL\}$ (see Figure 2.5). Also let $\{X_{ij}^{(s,o)}\}$ and $\{V_{ij}^{(s,o)}\}$ denote the wavelet coefficients of the original signal $\{f_{ij}\}$ and the noise $\{\epsilon_{ij}\}$, respectively. Notice that here we have introduced the extra notations to denote the scale and orientation of the wavelet decomposition, since they will be needed in the following. The estimate of each coefficient $X_{ij}^{(s,o)}$ is the soft-threshold estimator, $\hat{X}_{ij}^{(s,o)} = \eta_{T_{ij}}(Y_{ij})$. The threshold T_{ij} has been written explicitly as a function of the indices i and j to denote a different threshold for each location.

Let us rewrite the zero-mean Generalized Gaussian distribution (2.4) in a more convenient form,

$$GG_{\sigma_x,\beta}(x) = C(\sigma_x,\beta)\, e^{-(\alpha(\sigma_x,\beta)|x|)^\beta} \tag{5.1}$$

where

$$\alpha(\sigma_x,\beta) = \sigma_x^{-1}\left[\frac{\Gamma(\frac{3}{\beta})}{\Gamma(\frac{1}{\beta})}\right]^{1/2}, \quad C(\sigma_x,\beta) = \frac{\beta\,\alpha(\beta,\sigma_x)}{2\Gamma(\frac{1}{\beta})}.$$

The parameter σ_x is the standard deviation and the parameter β is the shape parameter. As demonstrated in Section 3.1, the optimal threshold T^* defined as

$$T^* = \arg\min_T E_{Y|X,X}(\eta_T(Y) - X)^2 \tag{5.2}$$

where $Y|X \sim \phi(y-x,\sigma^2)$ and $X \sim GG_{\sigma_x,\beta}(x)$, can be well approximated by

$$\tilde{T} = \frac{\sigma^2}{\sigma_x}.$$

This threshold, $\tilde{T}$, can easily be adjusted to the signal and noise energy as reflected in σ_x and σ.

To achieve a spatially adaptive thresholding strategy, the wavelet coefficients are modeled as components in a discrete random field, with a collection of independent zero-mean GGD random variables whose parameters β and σ_x are spatially varying. As discussed previously, mainly the parameter σ_x is of interest since $\tilde{T}$ depends on it, and β is assumed to be in the range for which this threshold is appropriate. In the next section, the technique of context modeling is used to estimate σ_x at every pixel, thus yielding adaptive thresholds.

5.1.2 Context Modeling for Spatial Adaptivity

The parameter σ_x needs to be estimated for each coefficient to make the threshold $\tilde{T}$ spatially adaptive. This can be accomplished by *context modeling*, an idea used frequently in image compression for adapting the coder to changing image characteristics. That is, the statistical model for a given coefficient is conditioned on a function of its neighbors. Several model-based coders have utilized information from causal quantized neighbors to determine the context model for each coefficient [37, 68]. The coder in [37] estimates the shape and standard variation parameters by the maximum likelihood (ML) estimator from the quantized coefficients within a causal neighborhood, allowing essentially an infinite mixture of distributions[2]. In the wavelet-based compression scheme in [68], context modeling was used to classify coefficients into several classes of Laplacian distributions with different values of σ_x. The conditioning was based on the weighted average of the coefficient magnitude in a causal neighborhood, and each class was formed by clustering coefficients whose associated weighted averages fall within a specified range. The distribution parameter is estimated from the coefficients for each class, which is then used to adapt the coder. Since the parameter and the description of each class need to be sent as overhead, only four classes were used in [68]. For the denoising problem, there is no need to conserve bits, thus it is not necessary to explicitly classify the pixels, and parameters can be estimated for each coefficient via a moving window, resulting in virtually an infinite mixture of distributions.

Consider one particular subband with $\tilde{N}^2$ coefficients, $\{Y_{ij}^{(s,o)}\}$. To simplify notation, we drop the superscript (s,o) now, and resume its usage when necessary for clarity. Each coefficient Y_{ij} is a random variable whose variance can be estimated as follows. Consider a neighborhood of size p around Y_{ij}, and place the *absolute value* of these p elements in a $p \times 1$ vector $\boldsymbol{u}_{ij}$. One possible choice is the eight nearest neighbors of Y_{ij} in the same subband, plus its parent coefficient $Y_{\lceil i/2 \rceil, \lceil j/2 \rceil}^{(s+1,o)}$ (see Figure 5.3 for the definition of parent-child relationship). To characterize the activity level of the current pixel, we calculate a weighted average of the absolute value of the neighbors as

$$Z_{ij} = \boldsymbol{w}^t \, \boldsymbol{u}_{ij}.$$

[2]To be exact, the shape parameter are restricted to be a member of a discrete set. The quantizer is pre-designed for a fixed discrete set of shape parameter β and slope λ/σ_x^2 where λ is determined by the target rate, so that the coding can be done quickly with the aid of a lookup table.

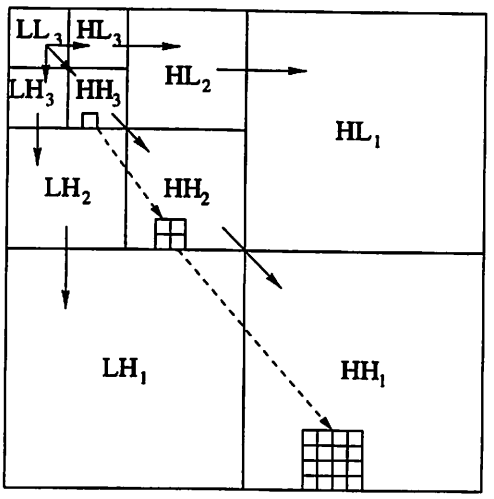


Figure 5.3: The parent-child relationship in the orthogonal wavelet transform. Each arrow points from the parent to its children, which are in the same orientation, but in the adjacent finer scale (except the children of coefficients in LL_3). For example, a coefficient in HH_3 is the parent of the four coefficients in HH_2 corresponding to the same spatial location, each of which is the parent of four coefficients in HH_1.

The weight $\boldsymbol{w}$ is found by using the least squares estimate, that is,

$$\begin{aligned} \boldsymbol{w}_{LS} &= \arg\min_{\boldsymbol{w}} \sum_{i,j} (|Y_{ij}| - \boldsymbol{w}^t \, \boldsymbol{u}_{ij})^2 \\ &= (\boldsymbol{U}^t \boldsymbol{U})^{-1} \boldsymbol{U}^t \, |\boldsymbol{Y}| \end{aligned} \tag{5.3}$$

where $\boldsymbol{U}$ is a $\tilde{N}^2 \times p$ matrix with each row being $\boldsymbol{u}_{ij}^t$, for all i, j, and $\boldsymbol{Y}$ is the $\tilde{N}^2 \times 1$ vector containing all coefficients Y_{ij}. Notice that the absolute values of the neighbors, rather than their original values, are used in the averaging. This is because orthogonal wavelet coefficients are uncorrelated, and thus an average of the neighbors does not yield much information about the coefficient of interest. However, the absolute value or the squared values of neighboring coefficients are correlated [55], and therefore their averages are useful in collecting information about other coefficients in the vicinity.

The variance of the random variable Y_{ij} is estimated from other coefficients whose contexts lie in an interval around Z_{ij}. To develop an intuition for this, it is helpful to examine Figure 5.4, which plots the pairs $\{Z_{ij}, Y_{ij}\}, i, j = 1, \ldots, \tilde{N}$. The points are clustered within a cone shape whose peak is at the origin. Taking an interval of small valued Z_{ij}, the associated coefficients $\{Y_{ij}\}$ have a small spread; on the other hand, an interval of large valued Z_{ij} has corresponding $\{Y_{ij}\}$ with a larger spread (the intervals are of different widths to capture the

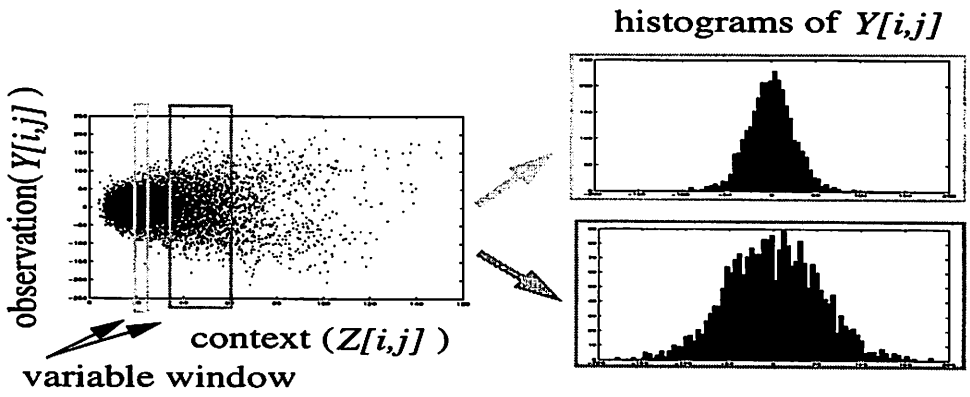


Figure 5.4: A sample plot of $\{Z_{ij}, Y_{ij}\}$, where Y_{ij} is the noisy wavelet coefficient, and Z_{ij} is its context. A collection of Y_{ij} with small values of Z_{ij} have a smaller spread than those with large values of Z_{ij}, suggesting that context modeling provides a good variability estimate of Y_{ij}.

same number of points). This suggests that the context provides a good indication of local variability. Thus, for a given Y_{i_0,j_0} at location $[i_0, j_0]$, we place an interval around Z_{i_0,j_0}, and the variance of Y_{i_0,j_0} is estimated from the points Y_{ij} whose context falls within this window. In particular, we take L closest points above Z_{i_0,j_0} and L closest points below, resulting in a total of $2L+1$ points, where we choose $L = \max(50, M^2/10)$ to ensure that enough points are used to estimate the variance. Note that this is a moving window rather than the fixed classes in [68], and thus allows a continuous range of estimate values. Let $\mathcal{B}_{i_0 j_0}$ denote the set of points $\{Y_{ij}\}$ whose corresponding $\{Z_{ij}\}$ fall in the moving window. The estimate of the variance $\sigma_x^2[i_0, j_0]$ is then

$$\hat{\sigma}_x^2[i_0, j_0] = \max\left(\left(\frac{1}{2L+1}\sum_{[k,\ell]\in\mathcal{B}_{i_0 j_0}} Y_{k\ell}^2\right) - \sigma^2, 0\right). \tag{5.4}$$

The term σ^2 needs to be subtracted because $\{Y_{ij}\}$ are the noisy observations, and the noise is independent of the signal, with variance σ^2. The threshold at location $[i_0, j_0]$ is then

$$\tilde{T}_{i_0,j_0} = \frac{\sigma^2}{\hat{\sigma}_x[i_0, j_0]}.$$

Calculating the threshold $\tilde{T}_{ij}$ for every location $[i, j]$ yields a spatially adaptive threshold. In the implementation, $\{Z_{ij}\}$ are first sorted, and a moving window is placed over them, so the set $\mathcal{B}_{ij}$ and the variance estimate $\hat{\sigma}_x^2[i, j]$ can be updated efficiently. The noise variance can be estimated by the median estimator in (3.6).

5.1.3 Thresholding in Overcomplete Expansion

Thresholding in the orthogonal wavelet domain has been observed to produce significantly noticeable artifacts such as Gibbs-like ringing and blips. To ameliorate this unpleasant phenomenon, Coifman and Donoho [16] proposed the *translation-invariant (TI) denoising*. Let $Shift_{k,\ell}[g]$ denote the operation of circularly shifting the input image g by k indices in the vertical direction and ℓ indices in the horizontal, and let $Unshift_{k,\ell}[g]$ be a similar operation but in the opposite direction. Also, let $Denoise[g,T]$ denote the operation of taking the DWT of the input image g, threshold it with a chosen uniform threshold T, then transform it back to the space domain. Then TI denoising yields an output which is the average of the thresholded copies over all possible shifts:

$$\hat{f} = \frac{1}{N^2} \sum_{k,\ell=0}^{N-1} Unshift_{k,\ell}[Denoise[Shift_{k,\ell}[g], T]] \ .$$

The rationale is that since the orthogonal wavelet transform is a time-varying transform and thresholding the coefficients produces ringing-like phenomena, thresholding a shifted input would produce ringing at different locations, and averaging over all different shifts would yield an output with more attenuated artifacts than a single copy alone. TI denoising can be shown to be equivalent to thresholding in the overcomplete representation implemented by the non-subsampled filter bank as discussed in Section 2.1.3, shown in Figure 2.6, up to a scaling in the thresholds. It has been shown empirically to remove some of the ringing artifacts, because denoising in the redundant expansion can be interpreted as an additional averaging. Thus we proceed to extend our spatially adaptive algorithm to this redundant expansion.

The adaptive algorithm in the orthogonal basis described above can easily be extended to the overcomplete expansion. Now consider the same orthogonal filters but used in a filter bank without downsampler. The filters are renormalized by $1/\sqrt{2}$ so that coefficient energy stays the same. This decomposition is a redundant representation, and there are correlations between the decomposition coefficients. For example, at the first level of decomposition, the odd and even coefficients (in each direction) are correlated. Thus, we can separate the coefficients into four sets of uncorrelated coefficients, namely, $\{Y_{2i,2j}\}, \{Y_{2i,2j+1}\}, \{Y_{2i+1,2j}\}$ and $\{Y_{2i+1,2j+1}\}$. For the s-th level decomposition, the coefficients can be separated into 2^{2s} sets, each containing uncorrelated coefficients, and they are $\{Y_{2^s i+k_1, 2^s j+k_2}\}_{i,j}, k_1, k_2 = 0, 1, \ldots, 2^s - 1$. Since each set contains uncorrelated coefficients,

the noise are also *iid* within each set as well, and thus the adaptive algorithm can be used for each set of coefficients. This approach lets us still use the independent noise assumption and circumvent the issue of denoising correlated signal coefficients with correlated noise, which is not an easy task. That is, if the coefficients are correlated, then one can conceivably do better than thresholding each coefficient independently; one could look at the numerous correlated coefficients and do a *joint thresholding*. This is still an open problem and is worth investigating. For correlated noise, near-optimal minimax properties were derived in [30] for a modified universal threshold, $\sigma^{(s)}\sqrt{2\log N}$, where $\sigma^{(s)}$ is the standard deviation of the noise at the decomposition level s. The framework is for a deterministic signal, however, and not the Bayesian framework used here. Thus, for simplicity, we separate the coefficients into groups of uncorrelated coefficients before using the thresholding algorithm.

There are two other minor details in this implementation. Firstly, one needs to alter the noise power σ^2 at each decomposition scale to $\sigma^2/4^s$ due to the renormalization of the filters. Secondly, the definition of the parent coefficient used in the neighborhood of the context is slightly changed: the parent of a coefficient at $[i,j]$ in scale s is simply the coefficient at the same spatial location $[i,j]$ in scale $s+1$.

5.1.4 Alternative Methods

There are several other possible alternative approaches which will be discussed below.

1. One may ask why the local variance is not estimated from, say, a local window around Y_{ij} (as in [37]), but rather from an indirect way of grouping the coefficients first via its context. Estimating from a local neighborhood is simple, and, as demonstrated by the good performance of the image coder in [37], it yields an estimate good enough for adapting the coder. However, our experience with noisy images shows that such an estimate yields considerably more unreliable variance estimates, $\sigma_x^2[i,j]$, and also a blotchily denoised image. This is because the estimate is highly sensitive to the window size we choose: a small window contains few points and thus yields unreliable estimates; a large window adapts slowly to changing characteristics. The context-based grouping allows one to congregate those coefficients with similar context though not necessarily spatially adjacent. It also allows a large number of coefficients to be used in the variance estimation, thus yielding a more reliable estimate. Simulations

show that the performance is not sensitive to the neighborhood choice $\mathcal{B}_{ij}$ and the weight $\boldsymbol{w}$ used in the context calculation, as a simple equally weighted average of the eight nearest neighbors in the same subband yield approximately the same result.

2. The method we have proposed is a *two-pass* process: the first pass calculates the weighted average $\{Z_{ij}\}$ of the absolute values of the neighboring *noisy* coefficients, and then $\{Z_{ij}\}$ are sorted; the second pass collects the noisy coefficients with similar values of Z_{ij}, estimates the signal variance, $\sigma_x^2[i,j]$, from the noisy coefficients and then the thresholds for the thresholding function. It is worthwhile to investigate the algorithm performance when the context modeling and the parameter estimation are performed on the *denoised* coefficients instead, since, intuitively, if the coefficients are really denoised, they should yield more reliable information. This simple intuition is, however, not as straightforward to implement as it seems. To do this in a two-pass algorithm is difficult, since Z_{ij} is a weighted average of neighboring denoised coefficients, but the threshold used to denoise these coefficients are estimated from other denoised coefficients with similar context. A simple-minded alternative solution is to use a *one-pass* modification of our algorithm, where the conditioning and estimation are based on the *causal, denoised* coefficients, much along the same philosophy as one-pass compression methods conditioning on causal quantized data [37, 68]. Assume a scanning order of row by row, and initialize the first coefficient as already denoised, that is, $\hat{X}_{1,1} = Y_{1,1}$. For every new coefficient at location $[i,j]$, the context is conditioned on $Z_{ij} = \boldsymbol{w}^t \boldsymbol{u}_{ij}$ where $\boldsymbol{u}_{ij}$ is now the vector containing the absolute value of denoised coefficients $\hat{X}_{ij}$ in a causal neighborhood, and the elements of $\boldsymbol{w}$ are simply the equal weights. These choices are made for simplicity since the denoising performance is not too sensitive to the neighborhood selection and weight vector $\boldsymbol{w}$. The GGD parameter $\sigma_x[i,j]$ is estimated from past denoised coefficients whose contexts are similar, and $2L+1$ coefficients are used (or all of the available coefficients so far if less than $2L+1$ coefficients have been denoised.) Since the coefficients are already denoised, the estimation of $\sigma_x[i,j]$ is

$$\hat{\sigma}_x^2[i,j] = \frac{1}{2L+1} \sum_{[k,\ell] \in \mathcal{B}_{ij}} \hat{X}_{k\ell}^2$$

instead of (5.4). Simulations show this approach to run into problems especially when the noise power σ^2 is large, causing many coefficients to be denoised to zero. Having

too many consecutive zero coefficients is likely to cause $\hat{\sigma}_x[i,j]$ to be zero, which then translates to an infinite threshold (i.e. Y_{ij} is thresholded to zero). This in turn may cause all the subsequent coefficients to be thresholded to zero. This phenomenon is frequently encountered in backward adaptive compression methods which adapts based on causal quantized coefficients: a run of zero coefficients may cause all subsequent coefficients to be quantized to zero as well. Some work ameliorate this problem by looking ahead to identify *unpredictable sets*, coefficients whose neighbors are zero, but who should not be quantized to zero [37, 68]. This logic can be applied to the denoising framework as well. When the algorithm computes Z_{ij}, it identifies the locations $[k,\ell]$ in the causal neighborhood $\mathcal{B}_{ij}$ for which $\hat{X}_{k\ell} = 0$ but $|Y_{k\ell}| \geq \sigma$, and these $Y_{k\ell}$ are substituted for the zero $\hat{X}_{k\ell}$ to be used in the computation of Z_{ij} and $\hat{\sigma}_x[i,j]$. Simulations show the resulting images to yield slightly worse MSEs than the previously proposed method, and they are visually considerably more noisy.

Another variation is to use the denoised coefficient for context modeling, but the observed noisy coefficients for estimating $\sigma_x^2[i,j]$ as in (5.4). Again, without taking some caution about the runs of zero coefficients, the variance estimate may be inadequate for several rows (recall the scanning is row by row) before having enough non-zero causal neighbors for collecting valid information. The denoised images are also similar to the ones described above, having slightly worse MSEs than the proposed two-pass algorithm, and are visually more noisy.

3. A central part of our spatially adaptive algorithm is based on modeling the variance $\sigma_x^2[i,j]$ to be non-constant and varying throughout the image. This is reminiscent of the *heteroscedasticity*, or non-constant variance, problem in statistics. Let $\{Y_{ij}\}$ be the observed noisy wavelet coefficients, and each Y_{ij} a random variable whose variance γ_{ij}^2 is non-constant. A common approach to the heteroscedasticity problem is to model γ_{ij}^2 as a function of some design vector, $\boldsymbol{u}_{ij}$. Traditionally there are two approaches in estimating this function: parametric and non-parametric. Since we have an assumed distribution on the wavelet coefficients (i.e. GGD), the parametric approach will be used here. The readers are referred to [8, 47] and related literatures for more details on heteroscedasticity models. Using a parametric function to describe the variance γ_{ij}^2 has the advantage that it allows a compact representation of the non-constant variance, useful for image analysis and understanding. In contrast, although the non-

parametric approach described in Section 5.1.2 works well, it does not lend itself to any tractable analysis.

In the previous section, we have described the noisy coefficient Y_{ij} as a sum of two random variables, $X_{ij} \sim$ GGD and $V_{ij} \sim$ Gaussian. Unless X_{ij} is a Gaussian distributed random variable, there is no closed form expression for the distribution of Y_{ij}. However, often one observes the wavelet coefficients for images to be sharply peaked at zero, better described by the Laplacian density function. Furthermore, the noisy coefficients also form a histogram which is sharply peaked at zero. Thus, for simplicity and for the sake of tractable analysis, we assume the noisy coefficient Y_{ij} to be Laplacian distributed, or, alternatively, $|Y_{ij}|$ be exponentially distributed. Similar to the context modeling framework in Section 5.1.2, let the design vector $\boldsymbol{u}_{ij}$ at location $[i,j]$ be the vector containing the absolute value of the eight closest neighboring (noisy) coefficients, $\boldsymbol{w}$ be the unknown regression parameter (i.e. the weights for the weighted average of the neighboring coefficients contained in $\boldsymbol{u}_{ij}$), and the variance for Y_{ij} be a function of $\boldsymbol{w}^t\boldsymbol{u}_{ij}$. Formally, our heteroscedasticity model is

$$|Y_{ij}| \sim \frac{1}{\gamma_{ij}} e^{(-y/\gamma_{ij})}, y \geq 0,$$

where the standard deviation is

$$\gamma_{ij} = K_{\boldsymbol{\theta}}(\boldsymbol{w}^t\boldsymbol{u}_{ij}),$$

$K_{\boldsymbol{\theta}}(\cdot)$ is a smooth function such as a polynomial of order r, with unknown parameter $(r+1) \times 1$ vector $\boldsymbol{\theta}$. Modeling γ_{ij} as a function of $\boldsymbol{w}^t\boldsymbol{u}_{ij}$ can again be justified by observing that the plot of $\{(\boldsymbol{w}^t\boldsymbol{u}_{ij}, Y_{ij})\}_{i,j}$ often resides within a cone shape (see Figure 5.4), implying that the variability of Y_{ij} depends highly on $\boldsymbol{w}^t\boldsymbol{u}_{ij}$.

To estimate the parameters $\boldsymbol{\theta}$ and $\boldsymbol{w}$, we use the likelihood approach. The negative log-likelihood of $|Y_{ij}|$ is

$$\log K_{\boldsymbol{\theta}}(\boldsymbol{w}^t\boldsymbol{u}_{ij}) + \frac{|Y_{ij}|}{K_{\boldsymbol{\theta}}(\boldsymbol{w}^t\boldsymbol{u}_{ij})}.$$

For $\{|Y_{ij}|\}_{i,j=1,\ldots,N}$, the negative log-likelihood, or the *likelihood function*, is

$$L(\boldsymbol{\theta}, \boldsymbol{w}) = \sum_{i,j=1}^{N} \left(\log K_{\boldsymbol{\theta}}(\boldsymbol{w}^t\boldsymbol{u}_{ij}) + \frac{|Y_{ij}|}{K_{\boldsymbol{\theta}}(\boldsymbol{w}^t\boldsymbol{u}_{ij})} \right).$$

The likelihood function is minimized over both parameters $\boldsymbol{\theta}$ and $\boldsymbol{w}$ to find their optimal values. One way to do this is to start with an initial $\boldsymbol{w}$ being the linear least squares estimate, $\boldsymbol{w}_{LS}$, in (5.3). Then $\boldsymbol{\theta}$ is estimated as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \hat{\boldsymbol{w}}_{LS}).$$

The regression parameter $\boldsymbol{w}$ is refined one step further as

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} L(\hat{\boldsymbol{\theta}}, \boldsymbol{w}).$$

After obtaining $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{\theta}}$, the standard deviation of Y_{ij} is estimated by $\hat{\gamma}_{ij} = K_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{w}}^t \boldsymbol{u}_{ij})$, and the variance estimate of the clean coefficient X_{ij} is $\hat{\sigma}_x^2[i,j] = \max(0, \hat{\gamma}_{ij}^2 - \hat{\sigma}^2)$. The threshold is then calculated as before to be $\tilde{T}_{ij} = \hat{\sigma}^2 / \hat{\sigma}_x[i,j]$.

Polynomials of order $r = 1, 2$ for $\boldsymbol{\theta}$ are experimented with, and a different set of polynomial parameters is found for each subband. Simulations show this parametric estimation of γ_{ij}^2 to differentiate well between regions of high energy (e.g. edges and textures) and smooth areas. That is, the variance estimate is larger in the edge and texture region, and smaller in the smooth regions. However, these values are not appropriate since the subsequently calculated variance estimate of X_{ij}, $\hat{\sigma}_x^2[i,j]$, results in zero in many subbands, which in turn translates to killing all the coefficients in the thresholding. This phenomenon may be due to the disparity between this parametric modeling of the non-constant variance and the noisy observation modeling: in the parametric approach, the observed noisy coefficients are modeled as Laplacian distributed, whereas in the original framework, the observations are *sums* of a Laplacian and Gaussian random variable. Nevertheless, the likelihood approach to the heteroscedasticity problem may be valuable to other applications.

5.2 Experimental Results

We use the images *barbara* and *lena* as test images. *iid* Gaussian noise at different levels of σ^2 are generated using *randn* in MATLAB. For the orthogonal wavelet transform, four levels of decomposition are used, and the wavelet employed is Daubechies' symmlet with 8 vanishing moments [19]. There are four methods that we compare, and the MSE results are shown in Table 5.1. The *AdaptDWT* method refers to the proposed adaptive thresholding

using the orthogonal transform DWT, and *AdaptNS* refers to adaptive thresholding using the non-subsampled wavelet transform. These two are compared against the best uniform thresholding techniques (in the MSE sense) when the original uncorrupted image is assumed to be known. For thresholding with DWT, in each subband, we find the oracle threshold T_{orc} as in (4.7). This method is labeled *OrcUnifDWT* in Table 5.1. Similarly, this is extended to the non-subsampled wavelet transform, where a different threshold is found for each set of uncorrelated coefficients within each subband (thus 2^{2s} thresholds for a subband at scale s). This method is labeled *OrcUnifNS*. Figure 5.5 shows a magnified region in the *barbara* image for $\sigma = 25$ and the *lena* image for $\sigma = 22.5$. The *AdaptNS* method outperforms all the other methods in both visual quality and MSE performance. It yields significantly less ringing artifacts and blotchiness than the methods using DWT. The *OrcUnifNS* method using uniform thresholds in the non-subsampled framework still shows significant noise in the smooth background. Thus, it is both the spatially adaptive thresholds and the overcomplete representation that contribute to the superior quality of the *AdaptNS* method. The adaptive methods denoise better especially in the flat regions, where the uniform methods yields images with much noise and "blips". Note that although the MSEs for the *lena* image is similar between the adaptive and uniform oracle methods, the visual quality in the adaptive method is far superior as it produces a denoised image that is smooth in the flat regions and has less artifacts around the edges as well. Interested readers can find Figure 5.5 available on the website `http://www-wavelet.eecs.berkeley.edu/~grchang/SpatialDenoise.html`.

5.3 Summary

We have proposed a simple and effective spatially and scale-wise adaptive method for denoising via wavelet thresholding in an overcomplete expansion. The adaptivity is based on context-modeling which enables a pixel-wise estimation of the signal variance and thus of the best threshold. The issue of spatially adapting the threshold values has not been addressed much in the literature. As we have shown in this chapter, adapting the threshold values to local signal energy allows us to keep much of the edge and texture details, while eliminating most of the noise in smooth regions, something that may be hard to achieve with a uniform threshold. The results showed substantial improvement over the oracle uniform thresholding assuming the original image known, both in visual quality and mean squared error.

Table 5.1: Comparing the MSE of the spatially adaptive algorithm with optimal subband uniform threshold in the DWT and the overcomplete expansion for various test images and σ.

σ	12.5	15	17.5	20	22.5	25
	barbara					
AdaptDWT	61.4	78.3	94.0	111.6	127.5	144.8
OrcUnifDWT	62.2	80.7	99.2	117.3	136.8	155.0
AdaptNS	**43.5**	**56.0**	**68.7**	**83.1**	**97.5**	**112.2**
OrcUnifNS	51.2	66.3	81.0	96.7	112.0	128.2
	lena					
AdaptDWT	36.1	42.7	50.2	58.1	66.5	72.9
OrcUnifDWT	36.1	43.7	51.3	58.8	67.4	73.7
AdaptNS	**27.5**	**32.7**	**38.4**	**44.1**	**51.1**	**56.5**
OrcUnifNS	29.8	35.9	42.3	48.7	55.7	61.2

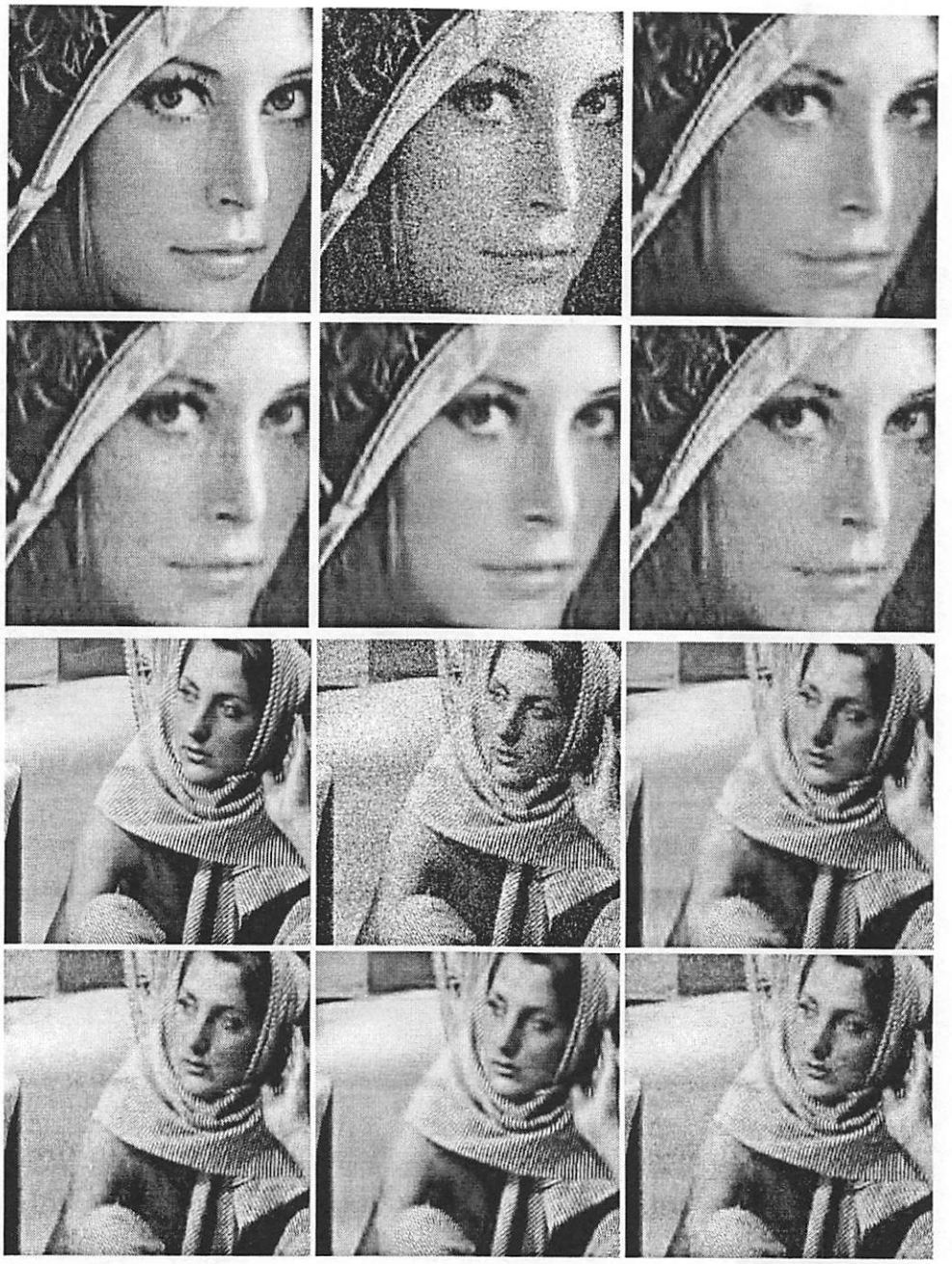


Figure 5.5: Comparing results of various denoising methods, for *lena* corrupted by noise $\sigma = 22.5$ and *barbara* by noise $\sigma = 25$. Clockwise from top left: original, noisy observation, adaptive thresholding in DWT basis ($AdaptDWT$), uniform thresholding in DWT basis ($OrcUnifDWT$), spatially adaptive thresholding in overcomplete expansion ($AdaptNS$), and uniform thresholding in overcomplete expansion ($OrcUnifNS$).

Chapter 6

Multiple Copy Image Denoising via Wavelet Thresholding

Most of the threshold denoising literatures are for applications in which there is only one set of observations (i.e. one sequence of time series or one still image). However, in numerous applications there are multiple copies of the same or similar images, thus it is necessary to investigate denoising techniques for removing noise from multiple corrupted copies of the same signal. For a corrupted video sequence, suppose that there are several consecutive frames in which the motion is not significant and that the registration problem has been corrected, one can view the frames as multiple noisy copies of the same image. Another example is when one scans a picture, but with unsatisfactory result, thus one does multiple scans, and then combines these copies to obtain the most noise-free copy possible. Since wavelet thresholding has worked well for one copy, it is natural to consider its extension to multiple copies.

The standard method for combining the multiple copies is to simply compute their weighted sum. One can only do better by incorporating a thresholding step. The question is, which ordering is better, thresholding first or averaging first, and what is the threshold value for each method? These are the issues to be addressed in this chapter. With the coefficients of each subband modeled as samples of a Laplacian random variable and the noise as samples of a Gaussian variable, we will show that the optimal ordering (in the mean squared error sense) depends on the number of available copies and the ratio between the noise power and the signal power. Moreover, we propose near-optimal subband adaptive

thresholds for both orderings. Results show that with the optimal or the proposed near-optimal thresholds, the two methods yield very similar performance, and both outperforms weighted averaging substantially.

6.1 Denoising Algorithm

The noisy observation model is the same as (2.3), but we add an additional index to denote the various noisy copies. Let $\boldsymbol{f}$ denote the $N \times N$ matrix of the original image to be recovered. The signal $\boldsymbol{f}$ has been transmitted over a Gaussian additive noise channel M times, and at the receiver we have M copies of noisy observations,

$$\boldsymbol{g}^{(m)} = \boldsymbol{f} + \boldsymbol{\varepsilon}^{(m)}, \quad m = 1, \ldots, M \ .$$

For the m-th copy, the $\{\varepsilon_{ij}^{(m)}\}_{ij}$ pixels are *iid* Gaussian $N(0, \sigma_m^2)$, where σ_m^2 is the variance of the m-th copy of noise. The noise samples between different copies are also assumed independent. The goal is to find an estimator $\hat{\boldsymbol{f}}$ which minimizes the mean squared error (MSE), $\frac{1}{N^2}\sum_{i,j=1}^{N}(\hat{f}_{ij} - f_{ij})^2$.

As in previous chapters, the image recovery is done in the orthogonal wavelet transform domain (at least the thresholding part of the algorithm). Let the wavelet transform of the noisy observation $\boldsymbol{g}^{(m)} = \boldsymbol{f} + \boldsymbol{\varepsilon}^{(m)}$ be denoted by $\boldsymbol{Y}^{(m)} = \boldsymbol{X} + \boldsymbol{V}^{(m)}$. Coefficients from each detail subband of $\boldsymbol{X}$ are modeled as samples of a centered Laplacian random variable with an unknown parameter. In this chapter, the subband coefficients are modeled using the Laplacian distribution, rather than GGD, for tractability. That is, the coefficients X_{ij} and $Y_{ij}^{(m)}$ at location index $[i, j]$ are modeled as random variables $X \sim p(x) = \mathrm{LAP}(\alpha) \triangleq \frac{\alpha}{2}e^{-\alpha|x|}$ and $Y^{(m)}|X \sim N(x, \sigma_m^2)$, respectively. In the following, the most straightforward method of weighted sum will be discussed first. Subsequently, we investigate the estimators which minimizes the expected square error for the two thresholding strategies, and compare their performance.

6.1.1 Recovery by Weighted Averaging

When there are multiple copies available, the standard method is to use the (pixel-wise) weighted average as the estimate. To simplify the notation, the subscript ij denoting the pixel indices will be dropped, since it should be clear that the denoising algorithm combines pixels of the same indices from the multiple copies.

Let $V^{(m)} \sim N(0, \sigma_m^2), m = 1, \ldots, M$, be the random variables representing the m-th copy noise, define Z to be the weighted sum of the M random variables $Y^{(m)} = X + V^{(m)}$,

$$Z = \sum_{m=1}^{M} \beta_m Y^{(m)} = X + \sum_{m=1}^{M} \beta_m V^{(m)},$$

where $\sum_m \beta_m = 1$. The optimal values of β_m are found by minimizing

$$E_{V^{(1)},\ldots,V^{(M)}|X}(Z - X)^2$$

subject to $\sum_m \beta_m = 1$ (by using a Lagrange multiplier and setting to zero the derivatives with respect to β_m, $m = 1, \ldots, M$), and they are

$$\beta_m^* = \frac{\frac{1}{\sigma_m^2}}{\sum_{i=1}^{M} \frac{1}{\sigma_i^2}} \quad , \quad m = 1, \ldots, M \ , \tag{6.1}$$

with the resulting MSE

$$\sigma_{\text{total}}^2 = \text{Var}(Z - X) = \text{Var}\left(\sum_{m=1}^{M} \beta_m^* V^{(m)}\right) = \frac{1}{\sum_{m=1}^{M} \frac{1}{\sigma_m^2}}. \tag{6.2}$$

Now let us incorporate thresholding into averaging. The weighted sum Z is essentially a new random variable and $Z|X \sim N(x, \sigma_{\text{total}}^2)$. Since this is exactly the setting for one copy thresholding, the next straightforward step is to simply find the best threshold and apply it on Z. However, can we do better than that? More specifically, since we have two operations here — averaging and thresholding — it is natural to ask which ordering is best in the mean squared sense.

6.1.2 Thresholding and Averaging

Consider the special case when $\sigma_1 = \sigma_2 = \cdots = \sigma_M \triangleq \sigma$. Thus, $\beta_1 = \cdots = \beta_M = \frac{1}{M}$. To make references more convenient, let $\mathcal{A}(\cdot)$ denote the weighted average operation and $\mathcal{T}(\cdot)$ the threshold operation, and we give the following notation to the two possible orderings of these operations:

$$\begin{aligned} \mathcal{A}(\mathcal{T}(Y^{(1)}, \ldots, Y^{(M)})): &\quad \hat{X}_{\mathcal{AT}}(T) = \tfrac{1}{M} \textstyle\sum_{m=1}^{M} \eta_T(Y^{(m)}) \\ \mathcal{T}(\mathcal{A}(Y^{(1)}, \ldots, Y^{(M)})): &\quad \hat{X}_{\mathcal{TA}}(T) = \eta_T\left(\tfrac{1}{M} \textstyle\sum_{m=1}^{M} Y^{(m)}\right). \end{aligned}$$

The MSE or *risk* of the $\mathcal{A}(\mathcal{T}(\cdot))$ method is

$$\begin{aligned} R_{\mathcal{A}\mathcal{T}}(T) &= E_X E_{Y^{(1)},\ldots,Y^{(M)}|X}(\hat{X}_{\mathcal{A}\mathcal{T}}(T) - X)^2 \\ &= \frac{1}{M} E_X E_{Y|X}(\eta_T(Y) - X)^2 + \frac{M-1}{M} E_X \left[E_{Y|X}(\eta_T(Y) - X)\right]^2, \end{aligned} \quad (6.3)$$

where $Y|X \sim N(x, \sigma^2)$ and (6.3) follows from the fact that $\{Y^{(1)}, \ldots, Y^{(M)}\}$ conditioned on X are independent. The risk of $\mathcal{T}(\mathcal{A}(\cdot))$ is

$$R_{\mathcal{T}\mathcal{A}}(T) = E_X E_{Y^{(1)},\ldots,Y^{(M)}|X}(\hat{X}_{\mathcal{T}\mathcal{A}}(T) - X)^2 = E_X E_{Z|X}(\eta_T(Z) - X)^2,$$

where $Z|X \sim N(x, \frac{\sigma^2}{M})$. The optimal threshold is the argument which minimizes the risk, that is,

$$T^*_{\mathcal{A}\mathcal{T}} = \arg\min_T R_{\mathcal{A}\mathcal{T}}(T) \quad \text{and} \quad T^*_{\mathcal{T}\mathcal{A}} = \arg\min_T R_{\mathcal{T}\mathcal{A}}(T) \ .$$

To compare the risks of these two methods, we look at the scaled MSE difference

$$\frac{R_{\mathcal{A}\mathcal{T}}(T^*_{\mathcal{A}\mathcal{T}}) - R_{\mathcal{T}\mathcal{A}}(T^*_{\mathcal{T}\mathcal{A}})}{\sigma^2}$$

as a function of M (the number of copies available) and of the ratio σ_x/σ, illustrated in Figure 6.1. For each $M \leq 5$, there is a cutoff point, C^*_M, below which $R_{\mathcal{A}\mathcal{T}}(T^*_{\mathcal{A}\mathcal{T}}) > R_{\mathcal{T}\mathcal{A}}(T^*_{\mathcal{T}\mathcal{A}})$, and above which $R_{\mathcal{A}\mathcal{T}}(T^*_{\mathcal{A}\mathcal{T}}) < R_{\mathcal{T}\mathcal{A}}(T^*_{\mathcal{T}\mathcal{A}})$. For $M > 5$, however, the $\mathcal{T}(\mathcal{A}(\cdot))$ method is better for any value of σ_x/σ. The values of C^*_M is tabulated in Table 6.1. This finding indicates that *the best method depends on the relative power between the noise and signal, and also on the value of* M. With the optimal thresholds, the improvement of one method over the other is small, on the order of $10^{-3}\sigma^2$. The $\mathcal{T}(\mathcal{A}(\cdot))$ method requires much less computation than the $\mathcal{A}(\mathcal{T}(\cdot))$ method, since the former can be implemented by computing the wavelet transform once, whereas the latter computes it M times. Thus if computation is an issue, the $\mathcal{T}(\mathcal{A}(\cdot))$ method is preferred.

We do not have closed form solutions for $T^*_{\mathcal{T}\mathcal{A}}$ and $T^*_{\mathcal{A}\mathcal{T}}$, thus their values would need to be numerically computed each time or be tabulated. However, we have found that they can be well approximated by simple closed form expressions. For the $\mathcal{T}(\mathcal{A}(\cdot))$ estimator, the threshold is simply a modification of $\tilde{T}$ for one copy denoising, but with a change in the noise variance,

$$\tilde{T}_{\mathcal{T}\mathcal{A}} = \frac{\sigma^2/M}{\sigma_x}. \quad (6.4)$$

For the $\mathcal{A}(\mathcal{T}(\cdot))$ method, we use the approximation

$$\tilde{T}_{\mathcal{A}\mathcal{T}} = \frac{\sigma^2/M^{(3/4)}}{\sigma_x}. \quad (6.5)$$

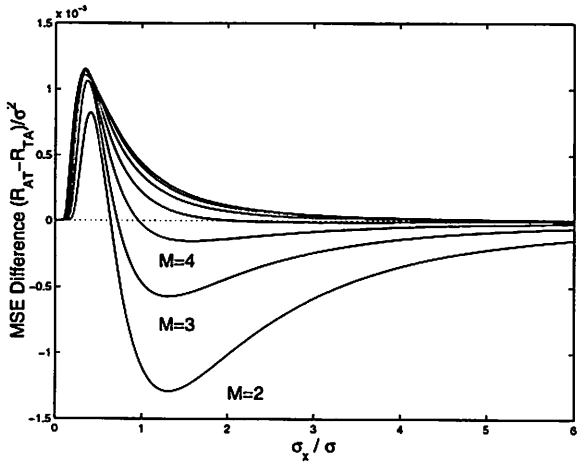


Figure 6.1: Scaled MSE difference $(R_{\mathcal{A}T}(T^*_{\mathcal{A}T}) - R_{T\mathcal{A}}(T^*_{T\mathcal{A}}))/\sigma^2$ as a function of M and σ_x/σ.

The exponent 3/4 in (6.5) yields a good fit to $T^*_{\mathcal{A}T}$, though it is by no means an optimal result. Notice that the threshold for $\mathcal{A}(\mathcal{T}(\cdot))$ decreases as M increases, even though at the thresholding stage, each copy is thresholded independently of the other copies. To explain this, the inner expectation of $R_{\mathcal{A}T}(T)$ is written as

$$E_{Y^{(1)},\ldots,Y^{(M)}|X}(\hat{X}-X)^2 = \frac{1}{M}E_{Y|X}\left[\eta_T(Y) - E_{Y|X}\eta_T(Y)\right]^2 + \left[E_{Y|X}\eta_T(Y) - X\right]^2 \quad (6.6)$$

The first term in (6.6) is the variance due to thresholding, while the second term is the square of the bias. The optimal threshold is obtained from the tradeoff between the variance term (which decreases with increasing T) and the bias term (which increases with increasing T). As M becomes larger, the variance term decreases due to the $1/M$ factor while the bias term stays the same. Thus, T needs to be decreased as well to obtain the minimum total.

Figure 6.2 compares the optimal and approximate thresholds for both methods as a function of M, for $\sigma = 1$ and $\sigma_x = 1$. Using the approximate thresholds $\tilde{T}_{T\mathcal{A}}$ and $\tilde{T}_{\mathcal{A}T}$ results in less than .2% loss of MSE optimality for any M. Figure 6.3 compares the optimal threshold $T^*_{\mathcal{A}T}$ and the approximation $\tilde{T}_{\mathcal{A}T}$ as a function of σ_x/σ for $M = 2,\ldots,6$. It shows that the approximation is good for large σ_x/σ but not as well for very small σ_x/σ, especially for large M. The loss of MSE optimality is less than 3.5% for $\sigma_x/\sigma < 1$ and less than 0.1% for $\sigma_x/\sigma > 1$. However, since typically the signal power is much larger than the noise power, inaccurate approximations for small σ_x/σ are acceptable. The use of the

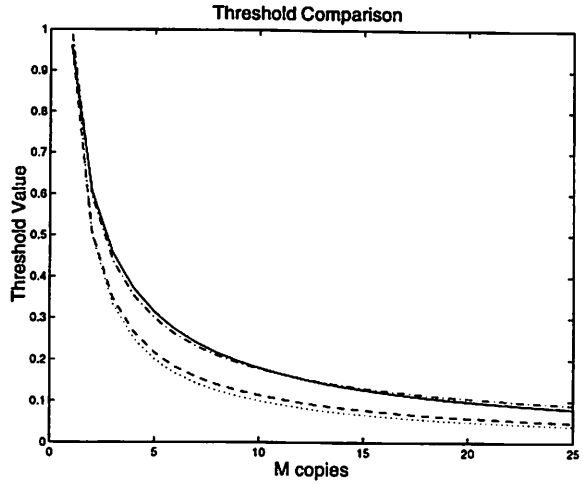


Figure 6.2: Comparing $T^*_{\mathcal{TA}}$ $(- - -)$ versus $\tilde{T}_{\mathcal{TA}}$ $(\cdots)$, and $T^*_{\mathcal{AT}}$ $(—)$ versus $\tilde{T}_{\mathcal{AT}}$ $(- \cdot -\cdot)$, when $\sigma = 1$ and $\sigma_x = 1$.

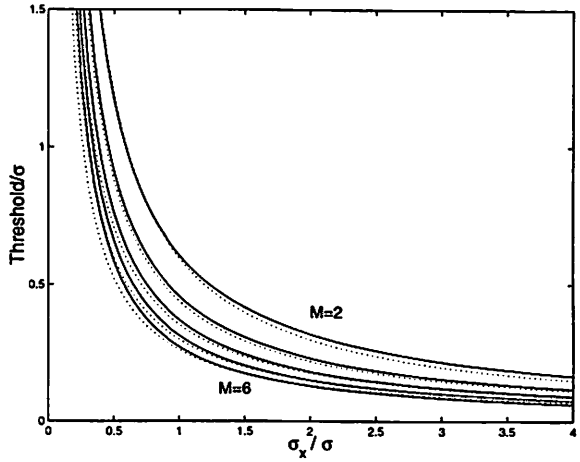


Figure 6.3: Comparing $T^*_{\mathcal{AT}}$ $(—)$ and $\tilde{T}_{\mathcal{AT}}$ $(\cdots)$ for $\sigma_1 = \cdots, \sigma_M \stackrel{\Delta}{=} \sigma$ as a function of σ_x/σ and $M = 2, \ldots, 6$.

Table 6.1: Cutoff values (in unit σ_x/σ) for each N, where C_N^* is the cutoff value for when using the optimal thresholds, and $\tilde{C}_N$ (listed only for $N \leq 5$) is the cutoff value when using the proposed thresholds, $\tilde{T}_{\mathcal{AT}}$ and $\tilde{T}_{\mathcal{TA}}$.

	C_N^*	$\tilde{C}_N$
$N = 2$	.6367	.1379
$N = 3$	.7154	.7654
$N = 4$	.9601	.9466
$N = 5$	1.9768	1.0884
$N > 5$	∞	> 1.23

thresholds $\tilde{T}_{\mathcal{TA}}$ and $\tilde{T}_{\mathcal{AT}}$ yield a different set of cutoff values $\tilde{C}_M$ (tabulated in Table 6.1), but the scaled MSE difference $(R_{\mathcal{AT}}(\tilde{T}_{\mathcal{AT}}) - R_{\mathcal{TA}}(\tilde{T}_{\mathcal{TA}}))/\sigma^2$ is similar to the curves shown in Figure 6.1 for optimal thresholds and is of the same order of magnitude. Thus, the use of the approximations $\tilde{T}_{\mathcal{TA}}$ and $\tilde{T}_{\mathcal{AT}}$ does not perturb the previous results substantially.

Parameter Estimation We now discuss how to estimate the noise variance, σ^2, and the standard deviation, σ_x, of the signal from the noisy observations. For both methods, these two parameters are estimated the same way for a fair comparison. First the noise variance, σ_m^2, of the m-th copy is estimated by the robust median estimator in (3.6), then $\hat{\sigma}^2$ is taken to be the average of these M estimates. Since the noise is independent from the signal,

$$\mathrm{Var}(Z) = \mathrm{Var}(X) + \sigma^2/M = \sigma_x^2 + \sigma^2/M \ .$$

Thus, for each subband of $Z = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{Y}^{(n)}$, the sample variance estimate of $\mathrm{Var}(Z)$ is calculated, and the estimate of the standard deviation of the signal is

$$\hat{\sigma}_x = \sqrt{(\mathrm{Var}(Z) - \hat{\sigma}^2/M)} \ .$$

Note that as in previous chapters, the estimate of σ_x and the threshold selection is done for each subband to yield subband-adaptive thresholds.

Heterogeneous Noise Variances Now consider the case when the noise variances σ_n^2 are different. This extension is straightforward in the $\mathcal{T}(\mathcal{A}(\cdot))$ case. The multiple copies are averaged with the coefficients β_m^* in (6.1), and the threshold is $\tilde{T}_{\mathcal{TA}}$ in (6.4) but with σ^2/M replaced by σ^2_{total} in (6.2).

For the $\mathcal{A}(\mathcal{T}(\cdot))$ method, one needs to find the optimal threshold for each copy

and the optimal weights in the summation. By minimizing the risk

$$E_X E_{Y^{(1)},\ldots,Y^{(M)}|X}\left(\sum_{m=1}^{M}\beta_m\eta_{T_m}(Y^{(m)})-X\right)^2$$

with respect to $\beta_1,\ldots,\beta_M$ subject to $\sum_m \beta_m = 1$, and also with respect to $T_1,\ldots,T_M$, one can find their optimal values. The optimal values of β_m are found to be very close to those in (6.1), and the optimal thresholds can be approximated by

$$\tilde{T}_{\mathcal{AT}}^{(m)} = \frac{\sigma_m^{1/2}}{\sigma_x}\left(\frac{1}{\sum_{i=1}^{M}\frac{1}{\sigma_i^2}}\right)^{3/4}, \quad m=1,\ldots,M,$$

which yields $\tilde{T}_{\mathcal{AT}}$ in (6.5) when $\sigma_1 = \sigma_2 = \cdots = \sigma_M$. For a given set of σ_m's, this approximation is good for the threshold corresponding to the smallest σ_m, and it worsens for thresholds corresponding to larger σ_m. This inaccuracy is mitigated by the fact that the weights β_m^*'s for copies with large σ_m's are small, thus the overall MSE is still close to the optimal MSE.

6.2 Experimental Results

To validate the theory, we take as the test image a 256×256 block from the image *barbara*, with $\sigma_1 = \ldots = \sigma_M = \sigma = 30$, using Daubechies' least unsymmetric wavelet with 8 vanishing moments (tabulated in Appendix A) and 4 scales of wavelet transform. The parameters σ_x and σ are estimated as in the previous discussion. We compare the MSEs of four methods for a range of M: averaging, $\mathcal{A}(\mathcal{T}(\cdot))$, $\mathcal{T}(\mathcal{A}(\cdot))$, and switching between the two thresholding methods (only for $M \leq 5$) with cutoff values $\tilde{C}_M$ (thus the switching method becomes $\mathcal{A}(\mathcal{T}(\cdot))$ for $M > 5$). The resulting MSEs are shown in Figure 6.5. The three thresholding methods show significant improvement over merely averaging, ranging from 70% to 30% reduction in MSE for M varying from 2 to 30. The removal of noise due to thresholding is also visually significant, especially for small M (see Figure 6.4, also available at `http://www-wavelet.eecs.berkeley.edu/~grchang/multThreshImages.pgm`). Among the thresholding methods, the $\mathcal{T}(\mathcal{A}(\cdot))$ method is the best in terms of MSE, even better than switching, suggesting that perhaps the $\mathcal{A}(\mathcal{T}(\cdot))$ method is more sensitive to model errors and threshold estimation errors. For $1 < M \leq 5$, the switching method yields MSEs that are between those of $\mathcal{A}(\mathcal{T}(\cdot))$ and $\mathcal{T}(\mathcal{A}(\cdot))$. Visually, one does not discern any

difference between the results from these three thresholding methods. The $\mathcal{T}(\mathcal{A}(\cdot))$ method also requires the least amount of computation since it can be implemented with only one wavelet transform. Thus, in practice, this method suffices to combine multiple noisy copies.

It is interesting to investigate if an additional stage of thresholding can have a significant improvement. It cannot do worse, since we can always choose the second stage threshold to be zero. To test this idea, we take the output of $\mathcal{A}(\mathcal{T}(\cdot))$ and optimally threshold it assuming that we have the original. The resulting MSE is only slightly better than the $\mathcal{T}(\mathcal{A}(\cdot))$, suggesting that thresholding of the weighted sum yields a sufficiently denoised image already. Furthermore, finding the optimal thresholds of a two-stage thresholding operation is difficult.

6.3 Summary

In this chapter we addressed the issue of image recovery from multiple copies of noisy images, and explored the idea of combining the wavelet thresholding technique with the more traditional averaging operation. The investigation showed that the optimal ordering of these two operations is not so straightforward and is in fact a function of the number of available copies and of the relative energy between noise and signal. We also proposed near-optimal thresholds for each ordering. With these thresholds, the performances were similar, and for computational reasons, averaging followed by thresholding is recommended. Furthermore, all of these thresholding methods showed substantial improvement over mere averaging, both visually and in the MSE sense.

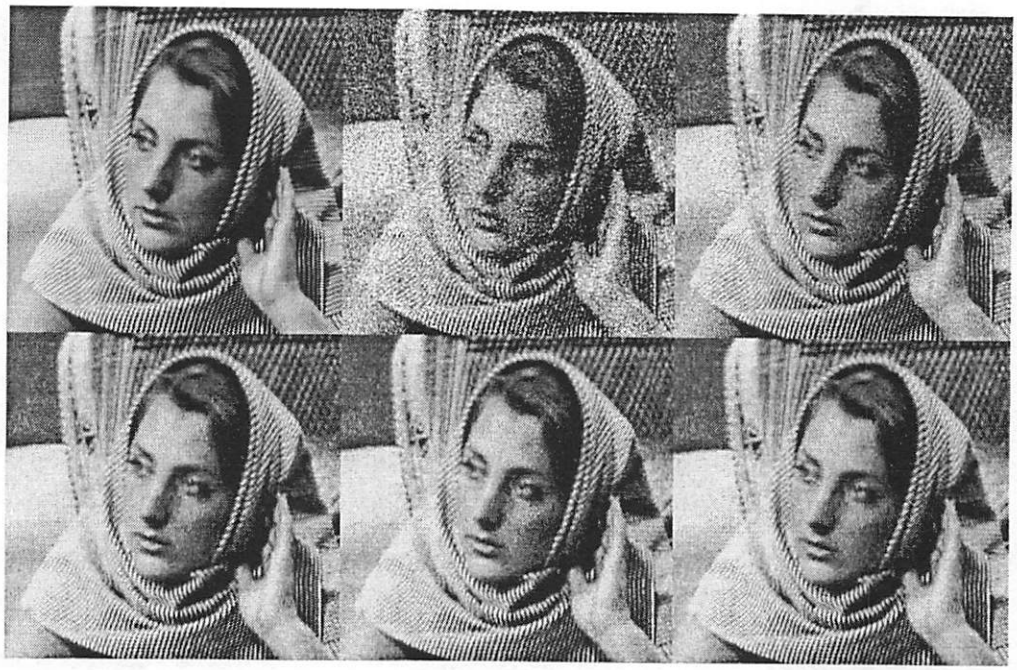


Figure 6.4: Denoised images, for $M = 5$. From top left, clockwise: original, noisy image with $\sigma = 30$, averaging, switching, $\mathcal{A}(\mathcal{T}(\cdot))$, and $\mathcal{T}(\mathcal{A}(\cdot))$.

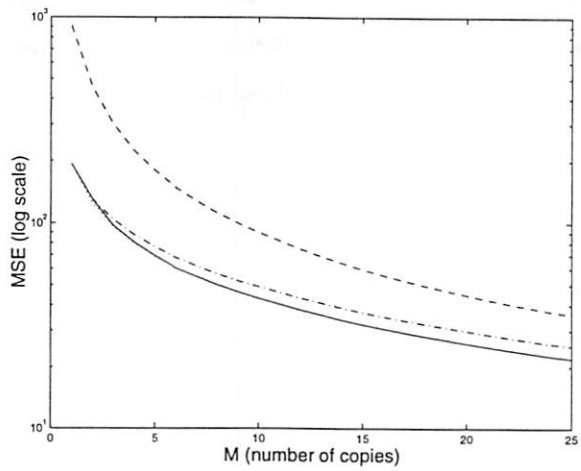


Figure 6.5: Comparing for each M the MSE (on a log 10 scale) of averaging $(- - -)$, $\mathcal{A}(\mathcal{T}(\cdot))$ $(- \cdot -\cdot)$, $\mathcal{T}(\mathcal{A}(\cdot))$ $(\cdots)$, and switching (—), for $\sigma = 30$. Note that the latter two curves are overlapped.

Chapter 7

Wavelet-based Image Interpolation

The classic problem of image interpolation refers to extracting information from the given image to fill in the extra pixels whose values we wish to know. It is useful for magnification and zooming purposes, which are the applications we have in mind here. The challenge is to process the image in such a way as to keep the magnified image looking sharp. Traditional methods such as bilinear and spline interpolations inherently assume smoothness constraints on the signal. As a result, they typically generate blurred images since they do not try to preserve some important image features. For example, a sharp edge in the smaller image would become a gradual ramp in the interpolated image when using these methods without taking precautions to keep it a sharp edge. To deblur these images, one could use the standard approach of unsharp masking [27] or other filtering techniques to boost the high frequencies needed to make the image look sharper. These post-processing approaches are somewhat *ad hoc*, however. In this chapter, we propose a wavelet-based method which extracts information of edges or points of sharp variations and preserves this information in the magnification process.

Points of sharp variations, or singularities, are among the most meaningful features of a signal. For images, these points typically correspond to edges, or boundaries between regions, and for many image enhancement applications, it is important to detect these points. Information about these points can be obtained by multi-scale edge detection methods developed in the computer vision community [51, 42, 66, 7]. The multiscale edge detection can be formulated in the wavelet framework, as the Canny edge detector [7] is equivalent to finding the local maxima in the wavelet transform. This multiscale edge characterization framework will be used here, as it allows both a convenient analysis of edges

and a model for the interpolation problem which will be introduced shortly.

Information about sharp variation points can be obtained from examining the evolution of the wavelet transform across scales. For a family of wavelets, the wavelet transform modulus maxima capture the sharp variation points of a signal, and their evolution across scales can be characterized by the local *Lipschitz regularity* of the signal [43, 40, 41]. For example, Figure 7.1 shows a 1-D signal and its wavelet transform for several scales. This signal includes singularities such as a step and an impulse, and other sharply varying regions. Each of these sharp variations induces peaks in the wavelet transform across scales, and the values of these peaks can be characterized by a mathematical equation with unknown parameters.

The interpolation problem can be viewed as estimating some "higher resolution" information. That is, the given image resides in the approximation space V_0, and the "desired" image is an element of the higher resolution space V_{-1}. Thus the essence of the problem is to estimate the detail signal in W_0 (recall from Section 2.1 that $V_{-1} = V_0 \oplus W_0$). In the context of the previous discussion on propagating peaks in the wavelet transform, the estimation of the detail signal entails the extrapolation of this propagation to the finer resolution. With this in mind, we now describe a regularity-preserving interpolation algorithm.

The proposed interpolation algorithm will first capture and characterize sharp variation points based on the multiscale wavelet analysis. This characterization is then used to estimate the higher resolution information necessary to preserve the regularity of the edge points. From the model of the problem, one can identify constraints on the estimate and thus refine the estimate iteratively.

The chapter is organized as follows. Section 7.1 will introduce the wavelet transform framework, and relate the multiscale edge detection to the wavelet analysis. The discussion will start in continuous time, followed by issues due to discretization. In Section 7.2.1, details of the interpolation problem model and algorithm will be discussed in the one-dimensional case for clarity. This algorithm is extended to reconstructing 2-D images in Section 7.2.3. Results and comparisons with traditional interpolation methods will be shown in Section 7.3.

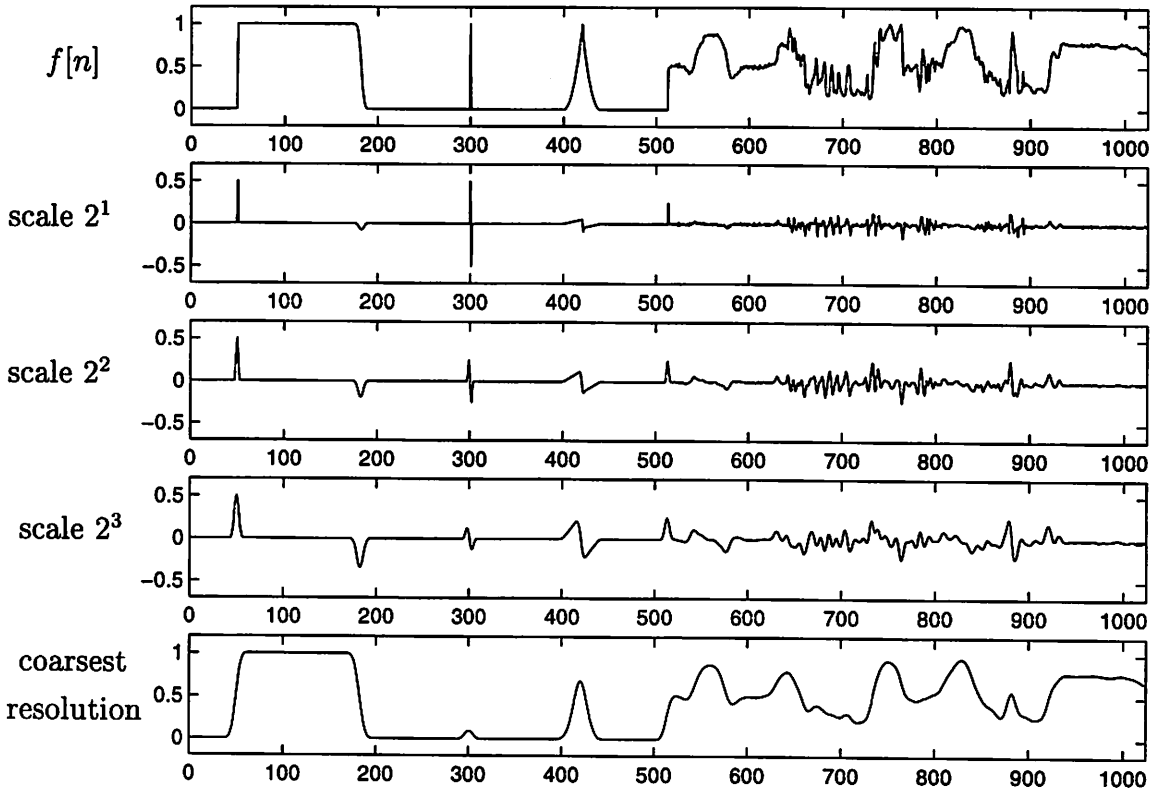


Figure 7.1: A 1-D waveform and its wavelet transform for three scales, showing the propagation of extrema points across the scales.

7.1 Multiscale Edges

In this section, we introduce the relationship between edge detection and the wavelet transform, and the characterization of multiscale edges. The readers are referred to [42, 51, 66, 7] for more details on edge detection, and [43, 40, 41] for multiscale edges in wavelet analysis. This section contains the background review and notations of multiscale edge and wavelet transform as presented in [41], and readers familiar with this material can peruse it and skip forward to Section 7.2.

7.1.1 Edge Detector and its Relation to the Wavelet Transform

Most traditional edge detectors extract sharp variation points by examining the first or second derivatives of the signal or its smoothed version. This is because an inflection point indicates a neighborhood of signal variation, and an inflection point in the signal

domain correspond to the local extremum of its first derivative and to the zero-crossing of its second derivative. Furthermore, local extrema points (of the first derivative) with large magnitude correspond to regions of sharp variation in the signal domain, while those with small magnitude correspond to regions of slow transition. This edge detection strategy can be formulated in the wavelet framework as follows.

Define a *smoothing function* $\theta(x)$ which satisfies

$$\lim_{x \to \pm\infty} \theta(x) = 0 \ ,$$

and

$$\int_{-\infty}^{\infty} \theta(x)dx = 1 \ .$$

Assume that $\theta(x)$ is differentiable and define a function $\psi(x)$ as the first derivative of $\theta(x)$:

$$\psi(x) = \frac{d\theta(x)}{dx} \ .$$

A *wavelet* is defined to be any function which integrates to 0. Hence, $\psi(x)$ can be considered as a wavelet. Now let $\psi_s(x)$ denote the dilated version of the wavelet function

$$\psi_s(x) = \frac{1}{s}\psi(\frac{x}{s}) \ ,$$

where s is the scale. The wavelet transform of $f(x)$ at scale s and position x is denoted by $W_s f(x)$, where

$$W_s f(x) = f(x) * \psi_s(x) \ ,$$

and $*$ is the convolution operator[1]. From the linearity of convolution and differentiation, it is easy to verify that

$$W_s f(x) = f(x) * (s\frac{d\theta_s(x)}{dx}) = s\frac{d}{dx}(f * \theta_s)(x) \ , \tag{7.1}$$

where $\theta_s(x)$, the dilation of $\theta(x)$, is defined similarly as $\psi_s(x)$. In words, equation (7.1) says that taking the wavelet transform of the signal at scale s is equivalent (up to a constant) to taking the first derivative of $f * \theta_s$, the signal smoothed at scale s.

As elucidated in [32], the notion of viewing an image at different scales is very natural for its understanding and analysis. The role of the scale s determines how global or local the signal features are that we want to capture. When s is small, the smoothing

[1]This is the continuous wavelet transform, with continuous scale and space parameter. It is different from the discrete-time wavelet series introduced in Section 2.1

function $\theta_s(\cdot)$ is localized in space and thus provides little smoothing, and $W_s f(x)$ yields information about local fluctuations in $f(x)$. On the other hand, when s is large, $\theta_s(\cdot)$ has a large spatial support and removes small local fluctuations, thus $W_s f(x)$ conveys information of signal variation on a more global scale. At a given scale, an extremum point in $W_s f(x)$ of large magnitude has the physical meaning of locating a sharp transition region in $f * \theta_s$, while an extremum of small magnitude indicates a region of relatively slow variation. One can also define a wavelet which is the second derivative of $\theta(x)$, and use the zero-crossing to detect edges. However, using the local extrema has added advantages since the magnitude of the extrema points conveys how sharply the signal is changing, whereas the zero-crossings do not. In the case that $\theta(x)$ is Gaussian, the detection of zero-crossings correspond to the Marr-Hildreth edge detector [42], and extrema points correspond to the Canny edge detector [7]. Furthermore, a Gaussian $\theta(\cdot)$ is the unique function with the property of not creating additional spurious extrema points at larger scales [32]. Therefore, for edge characterization, it is important to choose a filter which is Gaussian or approximately Gaussian.

The extension of the multiscale edge detection to two dimensions is straightforward. Let $\theta(x,y)$ be a smoothing function which integrates to 1 and converges to 0 at infinity, and let $\theta_s(x,y)$ denote the dilation of $\theta(x,y)$,

$$\theta_s(x,y) = \frac{1}{s^2}\theta(\frac{x}{s}, \frac{y}{s}).$$

The image $f(x,y)$ is smoothed by $\theta_s(x,y)$, and its gradient $\vec{\nabla}(f*\theta_s)(x,y)$ is computed. The direction of the gradient vector at (x,y) is the direction at which $f(x,y)$ has the sharpest variation. An edge point is defined to be a point (x_0,y_0) at which $|\vec{\nabla}(f*\theta_s)(x,y)|$ is the maximum along the direction of the gradient vector, and it is an inflection point of $f*\theta_s$.

To relate multiscale edges to the 2-D wavelet transform, first define

$$\psi^1(x,y) = \frac{\partial\theta(x,y)}{\partial x} \text{ and } \psi^2(x,y) = \frac{\partial\theta(x,y)}{\partial y}.$$

The wavelet transform of $f(x,y)$ consists of two components,

$$W_s^1 f(x,y) = f * \psi_s^1(x,y) \text{ and } W_s^2 f(x,y) = f * \psi_s^2(x,y),$$

and it is related to the gradient vector by

$$\begin{bmatrix} W_s^1 f(x,y) \\ W_s^2 f(x,y) \end{bmatrix} = s \begin{bmatrix} \frac{\partial}{\partial x}(f*\theta_s)(x,y) \\ \frac{\partial}{\partial y}(f*\theta_s)(x,y) \end{bmatrix} = s\vec{\nabla}(f*\theta_s)(x,y).$$

The edge points are where the *modulus*,

$$M_s f(x,y) = \sqrt{|W_s^1 f(x,y)|^2 + |W_s^2 f(x,y)|^2},$$

is maximum in the direction of the gradient vector. In the rest of the paper, these points will be referred to as the *modulus maxima*. The time-domain regions represented by these modulus maxima will be called loosely as edge points, singularities, or sharply-varying points, interchangeably.

7.1.2 Characterizing Multiscale Edges

From the previous discussion, it is clear that the value of the wavelet transform at scale s measures the smoothness of the signal smoothed at scale s. Furthermore, a sharp variation induces a local maximum (in the absolute value of the wavelet transform) which propagates across scales. To illustrate, we return to Figure 7.1 which shows a waveform and its wavelet transform at the dyadic scales $s = 2^j$, for $j = 1, 2, 3$. This waveform consists of a step edge, an impulse, their smoothed versions, and one row taken from the image *Lena*. Each isolated singularity produces extrema points which propagate across scales, and this evolution can be characterized in the wavelet transform by the local Lipschitz regularity, which measures the smoothness and differentiability of a continuous function.

Definition 1 *Let* $0 \leq \alpha \leq 1$. *A function* $f(x)$ *is uniformly Lipschitz* α *over an interval* (a,b) *if and only if there exists a constant* K *such that for any* $x_0, x_1 \in (a,b)$

$$|f(x_0) - f(x_1)| \leq K|x_0 - x_1|^\alpha \quad .$$

The uniform Lipschitz regularity *of* $f(x)$ *is the supremum* α_0 *over all* α *for which* $f(x)$ *is uniform Lipschitz* α.

The value of the uniform Lipschitz regularity measures the differentiability and smoothness of the function in a local neighborhood. For example, if $f(x)$ is differentiable at x_0, then it is Lipschitz regularity 1. The larger the α, the more *regular* or smooth the function is. If $f(x)$ is discontinuous but bounded in the neighborhood of x_0, then $\alpha_0 = 0$. A step function is Lipschitz 0 at the discontinuity. The following result states that the Lipschitz exponent can be measured from the evolution of the absolute values of the wavelet transform across scales [43].

Theorem 1 *A function $f(x)$ is uniformly Lipschitz α over an interval (a,b) if and only if there exists a constant $K > 0$ such that for all $x \in (a,b)$, the wavelet transform satisfies*

$$|W_s f(x)| \leq K s^{\alpha} . \tag{7.2}$$

Note that the values K and α depend on the particular singularity at x.

The above result for functions with Lipschitz regularity $\alpha \in [0,1]$ can be extended to tempered distributions such as a Dirac function, which has a negative Lipschitz exponent, $\alpha = -1$. That is, a distribution $f(x)$ is said to have a uniform Lipschitz regularity equal to α on (a,b) if and only if its primitive is $\alpha + 1$ on (a,b). The primitive of a Dirac function at x_0 is a step function at x_0, which has $\alpha = 0$, and thus a Dirac has $\alpha = -1$. The results in Theorem 1 can be proven for negative Lipschitz regularity as well.

Often signals have points of sharp variations rather than discontinuities. An example is the smoothed edge in Figure 7.1. The previous discussion can be extended to smoothed singularities as well. Suppose a local smooth sharp variation at x_0 is modeled as the result from convolving a singularity at x_0 with a Gaussian function with variance σ^2. That is, a signal $f(x)$ with a sharp variation at x_0 is modeled as $f(x) = h * g_\sigma(x)$, where $h(x)$ has a local singularity at x_0 whose uniform Lipschitz regularity is α_0, and $g_\sigma(x)$ is a zero-mean Gaussian function with variance σ^2. Further suppose that the smoothing function $\theta(x)$ is close to Gaussian in the sense that $\theta_s * g_\sigma(x) \approx \theta_{s_0}(x)$ where $s_0 = \sqrt{s^2 + \sigma^2}$, then the wavelet transform of $f(x)$ and $h(x)$ can be related by

$$W_s f(x) = s\frac{d}{dx}(\phi * \theta_{s_0})(x) = \frac{s}{s_0} W_{s_0} h(x). \tag{7.3}$$

Thus, by combining (7.3) and (7.2), the results in Theorem 1 can be extended for the smoothed sharp variation in $f(x)$ for any scale $s > 0$:

$$|W_s f(x)| \leq K s \cdot s_0^{\alpha - 1}, \text{ where } s_0 = \sqrt{s^2 + \sigma^2}.$$

7.1.3 Discretization Issues

In practice, any implementation must be discrete, and thus the previous discussion in the continuous space and scale domain needs some discretization considerations. These issues include the discretization of a continuous-time signal and its wavelet transform, and the discrete implementation.

For discrete processing, any continuous-time signal must also be sampled before being processed. Thus, a signal is measured at a finite resolution. Its wavelet transform can only be computed over a countable and finite range of scales. In many applications, it suffices to compute the wavelet transform at the *dyadic* scale, $s = 2^j$, with $j = 1, 2, \ldots$, which also allows a fast discrete computation. The fast computation algorithm, the design of the discrete filters and their relations with the continuous filters $\theta(x)$ and $\phi(x)$ are well explained in [41], to which the readers are referred for more details. Here only the necessary results and notations will be introduced.

Thus, let the finest scale be $s = 1$, and the coarsest scale computed be $s = 2^J$. Define a smoothing operator at scale $s = 2^j$ to be

$$S_{2^j} f(x) = f * \phi_{2^j}(x), \quad j = 0, 1, \ldots, J \quad .$$

The function $\phi(x)$ satisfies certain properties such that the difference, or *details*, between $S_{2^j} f$ and $S_{2^{j+1}} f$ is $W_{2^j} f$ defined in (7.1). Now let $D = \{d_n\}_{n \in \mathbf{Z}}$ be a discrete sequence such that there exists a (non-unique) continuous function $f(x) \in L_2(\mathbb{R})$ satisfying

$$S_1 f(n) = d_n, \ \forall n \in \mathbb{Z}.$$

Hence, we assume that the underlying signal is the continuous function $f(x)$, but only the discretized version, $S_1 f(n)$, is available for processing. For a particular class of wavelets, one can compute from the discrete sequence $D = \{S_1 f(n)\}_{n \in \mathbf{Z}}$ the uniform sampling (in x) of the wavelet transform of $f(x)$ at dyadic scales $s \geq 1$. Let the following notations denote these discrete samples,

$$W^d_{2^j} f = \{W_{2^j} f(n + \epsilon)\}_{n \in \mathbf{Z}} \quad \text{and} \quad S^d_{2^j} f = \{S_{2^j} f(n + \epsilon)\}_{n \in \mathbf{Z}}$$

where ϵ is the shift due to convolution with ϕ_{2^j} and ψ_{2^j}. The set of signals

$$\left\{ (W^d_{2^j} f)_{1 \leq j \leq J}, S^d_{2^J} f \right\}$$

is the *discrete dyadic wavelet transform* of $D = \{S_1 f(n)\}_{n \in \mathbf{Z}}$. Henceforth the discussion will concern discrete sequences, thus to simplify notations, the discrete sequence $f[n]$ will denote the samples $S_1 f[n]$, and $W_{2^j} f[n]$ will denote the discrete dyadic transform of $f[n]$ (note the omittance of the superscript "d").

The discrete dyadic wavelet transform allows a fast implementation to be described below and whose 1-D filter bank interpretation is shown in Figure 2.3. We describe the

algorithm here, not to be repetitive, but because there is the issue of the *multiplicative constants* which was not a concern before, but is important for estimating the Lipschitz regularity in discrete-time.

The forward transform is characterized by two filters: a lowpass filter $h_0[n]$ and a highpass filter $h_1[n]$. Let $h_0^{(j)}[n]$ and $h_1^{(j)}[n]$ be the filters obtained by upsampling $h_0[n]$ and $h_1[n]$, respectively, by a factor of 2^j (i.e. inserting $2^j - 1$ zeros between the coefficients). The wavelet transform of a signal $f \in l_2(\mathbb{Z})$ can be computed through the convolution with $h_0^{(j)}[n]$ and $h_1^{(j)}[n]$ in a recursive manner:

$$\begin{aligned} W_{2^j} f &= \tfrac{1}{\lambda_{j-1}} S_{2^{j-1}} f * h_1^{(j-1)} \\ S_{2^j} f &= S_{2^{j-1}} f * h_0^{(j-1)} \end{aligned} \qquad j = 1, 2, \ldots, J, \tag{7.4}$$

where $S_1 f = f$, $h_0^{(0)} = h_0$, and $h_1^{(0)} = h_1$. Let the wavelet transform operator $\mathcal{W}$ denote the linear operator mapping f to $\{S_{2^J} f, W_{2^j} f, j = 1, \ldots, J\}$. The operator $\mathcal{W}$ can be implemented by the octave-band non-subsampled filter bank shown in Figure 2.3(a), provided the multiplication with λ_j are incorporated appropriately. The multiplicative constant λ_j appears here because discretization introduces deviation in the estimation of the Lipschitz regularity, and scaling factors are needed to make the correction. More specifically, the constants λ_j are multiplied to the detail levels of the wavelet transform, $W_{2^j} f$, and these constants are found empirically so as to make the discrete time step function have Lipschitz regularity $\alpha = 0$. Obviously, the values of λ_j are dependent on the chosen wavelet. The quadratic spline filters (see [41] for the derivation) are used for our work because they approximate coarsely the Gaussian function and its first derivative and they also can be used in the fast implementation of the discrete dyadic wavelet transform. These filters are shown in Figure 7.2. Their coefficients and the associated constants λ_j are tabulated in Appendix A.

For perfect reconstruction to be possible, it is necessary and sufficient that there exists a synthesis pair $\tilde{h}_0[n]$ and $\tilde{h}_1[n]$ which satisfy the perfect reconstruction condition

$$H_0(z)\tilde{H}_0(z) + H_1(z)\tilde{H}_1(z) = 1 \,, \tag{7.5}$$

where $H_0(z), H_1(z), \tilde{H}_0(z)$, and $\tilde{H}_1(z)$ are the z-transform of the filters $h_0[n], h_1[n], \tilde{h}_0[n]$, and $\tilde{h}_1[n]$, respectively. The inverse wavelet transform reconstructs the original signal by progressively adding finer and finer details onto the coarse residual signal $S_{2^J} f$. It can be

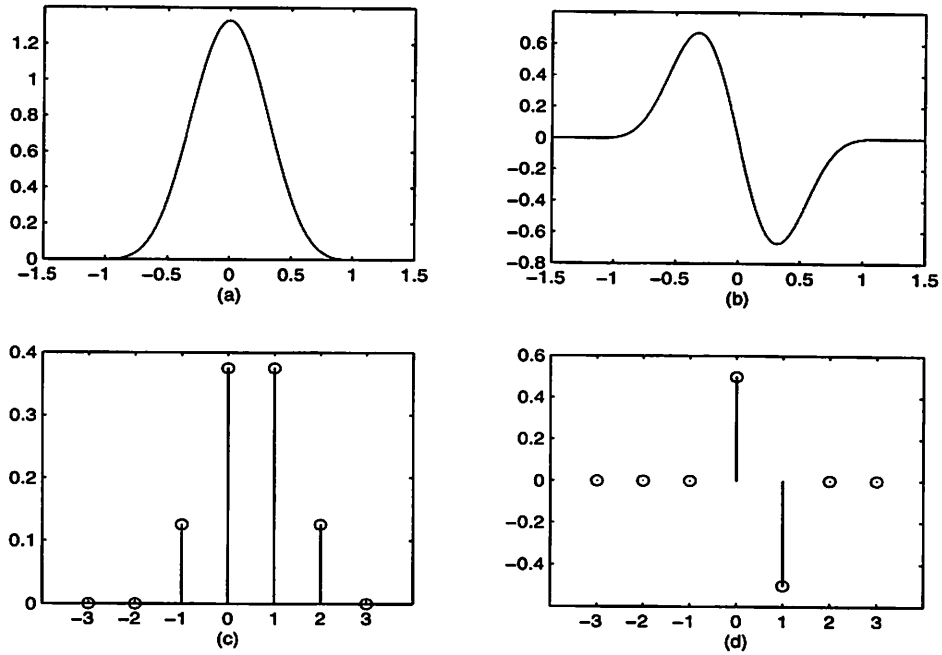


Figure 7.2: The quadratic spline wavelet and smoothing function used in this work. The continuous-time smoothing function $\phi(x)$ in (a), and wavelet $\psi(x)$ in (b). The corresponding FIR coefficients of the smoothing function (lowpass filter $h_0[n]$) in (c), and of the wavelet (highpass filter $h_1[n]$) in (d).

calculated recursively as

$$S_{2^{j-1}} f = \lambda_j W_{2^j} f * \tilde{h}_1^{(j-1)} + S_{2^j} f * \tilde{h}_0^{(j-1)} \quad , \quad j = J, J-1, \ldots, 1 \; , \tag{7.6}$$

where $\tilde{h}_0^{(0)} = \tilde{h}_0$ and $\tilde{h}_1^{(0)} = \tilde{h}_1$. The inverse wavelet transform operator $\mathcal{W}^{-1}$ can be implemented as a non-subsampled synthesis octave band filter bank in Figure 2.3 (b). Note again that the λ_j constants are needed to offset the scaling in the wavelet transform equation (7.4).

The discrete dyadic wavelet transform is an overcomplete, or redundant, representation of a function. An arbitrary set of sequences $\{g_j\}_{j=1,\ldots,J+1}$ is not necessarily the wavelet transform of some function f in $l_2(\mathbb{Z})$. It is the wavelet transform of some function $f \in l_2(\mathbb{Z})$ if and only if

$$\mathcal{W}(\mathcal{W}^{-1}(\{g_j\}_{j=1,\ldots,J+1})) = \{g_j\}_{j=1,\ldots,J+1} \; . \tag{7.7}$$

If the set of sequences $\{g_j\}_{j=1,\ldots,J+1}$ satisfies (7.7), then we say that it belongs to the range

of the wavelet transform operator $\mathcal{W}$. The operator $\mathcal{W}\mathcal{W}^{-1}$ is thus the projection operator onto the range of the wavelet transform.

In practice, there are only a finite number N of available samples $f[n]$, which creates a problem at the boundary in the computation of the wavelet transform. To mitigate this problem, the signal is extended with mirror symmetry. This periodization avoids creating a spurious discontinuity at the boundaries.

For the 2-D wavelet transform, a particular class of 2-D wavelets is used here. Specifically, we choose separable filters for the 2-D wavelets, where the 1-D filters $H_0, H_1, \tilde{H}_0$, and $\tilde{H}_1$ are the same as in the 1-D wavelet transform. An additional filter L is needed, whose Fourier transform satisfies

$$L(\omega) = \frac{1 + H_0(\omega)\tilde{H}_0(\omega)}{2} .$$

The 2-D forward and inverse wavelet transform can be computed in a recursive manner similar to the 1-D case, implemented with the non-subsampled filter banks shown in Figure 7.3. Filtering with $H_1(z_x)$, for example, means convolving with $h_1[n]$ in the horizontal direction. Similarly, filtering with $H_1(z_y)$, for example, denotes convolving with $h_1[n]$ in the vertical direction. Note that this filter bank is different from the 2-D non-subsampled filter bank discussed in Section 2.1.3. In Section 2.1.3, each stage has 4 channels of output. Here each stage of the filter bank has 3 channels, and it emulates the horizontal and vertical derivatives and the lowpass versions of the image at various scales.

7.1.4 Edge Points as Signal Representation

Several works have proposed to reconstruct a signal based on only the information about its edge points, characterized as modulus maxima or zero-crossing representations in the wavelet domain[2] [3, 41, 17]. The zero-crossing representation includes the location of the zero-crossings, and the integral values between each pair of zero-crossings. Marr and Mallat conjectured that such the local extrema or zero-crossings representation defines uniquely a signal, a belief which was later disproved by a counter-example from Meyer [44] and Berman [3] (the latter in discrete analysis). The completeness of this representation depends on the chosen wavelet, and is unstable at high frequencies.

With the quadratic spline wavelet used in this work, the wavelet transform modulus maxima representation does not provide a complete representation. Nevertheless, recon-

[2]Of course, different families of wavelets need to be used for these two representations, namely, functions which are the first- and second-derivatives, respectively, of a smoothing function.

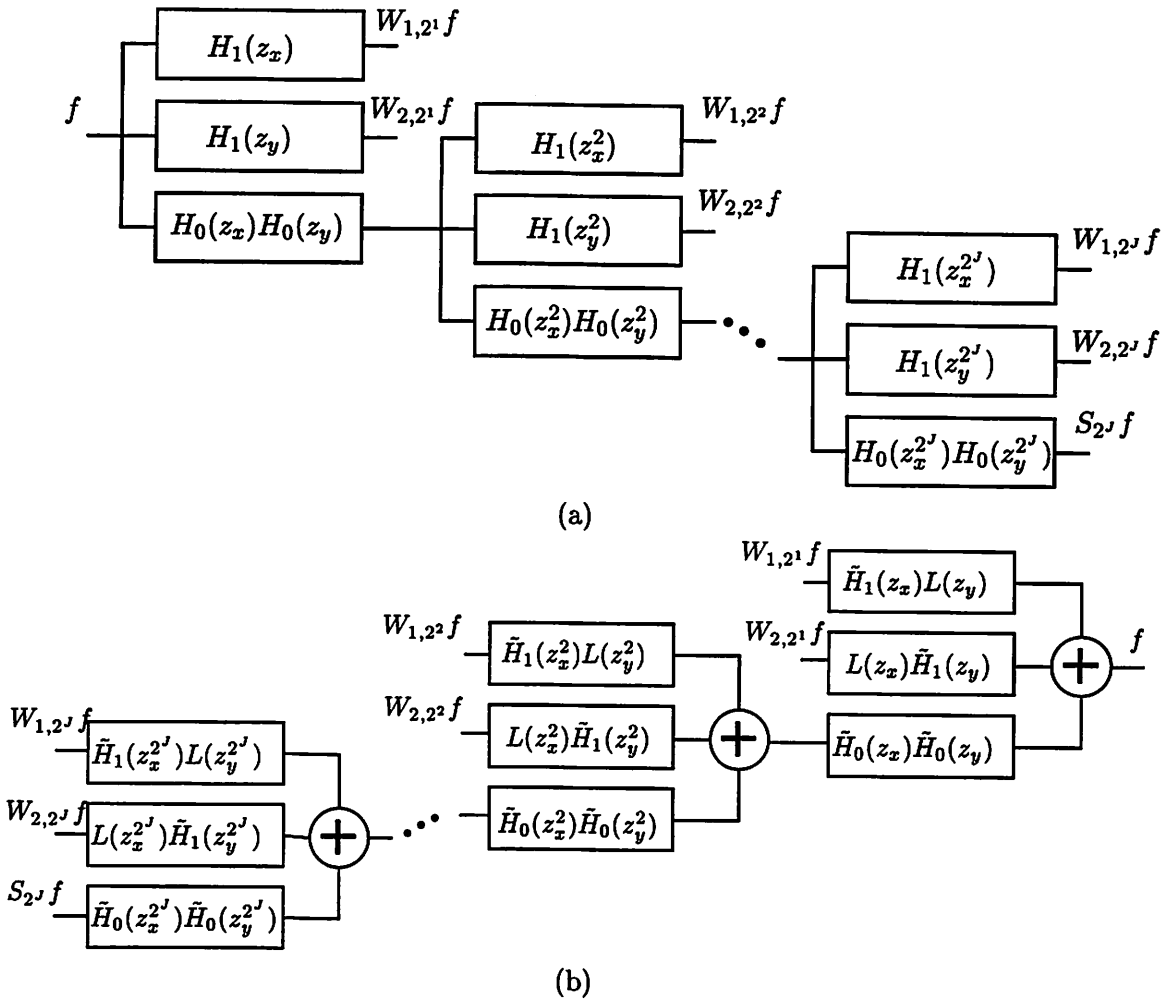


Figure 7.3: The 2-D discrete dyadic wavelet transform. (a) The forward transform. (b) The inverse transform.

struction from this representation has been shown to be satisfactory and has been applied to image coding [41, 17]. In [41], the local maxima of the absolute value of the wavelet transform (or local modulus maxima in 2-D) are kept in the representation, because local minima correspond to slowly varying regions. This creates a difficulty in the reconstruction since the representation is not a convex set. In [17], both the local maxima and minima of the wavelet transform are kept to allow a convex representation, which then allows a simpler reconstruction algorithm. Also, the 2-D case is treated as separable 1-D problems. In this work we adopt the latter representation, so that a simple reconstruction algorithm could be used.

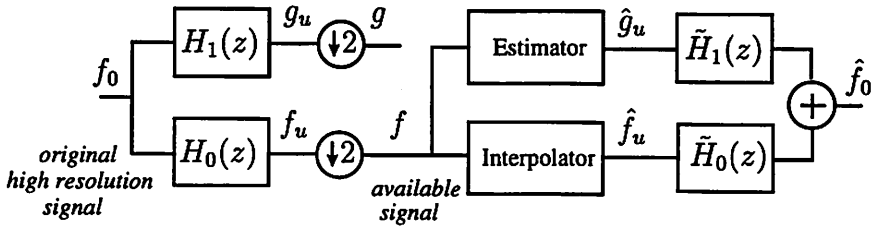


Figure 7.4: Interpolation problem model for 1-D. The available signal f is modeled as the subsampled lowpass component of a higher resolution signal f_0, which is the desired signal.

7.2 Enhancement Algorithm

The enhancement algorithm is first explained in one-dimension for clarity before extending it to the two-dimensional case. The discussion concentrates on magnification by a factor of 2, although larger magnifications (for factor which are powers of 2) can be achieved through iteratively performing this algorithm. First the main concepts will be introduced in Section 7.2.1, and the details will be given in Section 7.2.2. The 2-D algorithm will be developed in Section 7.2.3.

7.2.1 Algorithm Overview

The model of the interpolation problem is shown in Figure 7.4. The available signal $\{f[n], n = 0, \ldots, N-1\}$ is modeled to be obtained from the high resolution signal $\{f_0[n], n = 0, \ldots, 2N-1\}$ which we wish to recover, by lowpass filtering followed by downsampling by a factor of 2. This is a reasonable model since a higher resolution signal is often lowpassed before sampling to avoid aliasing. Naturally, one does not assume the exact knowledge of the lowpass filter used in the sampling process. We conjecture that as long as it is reasonable (i.e. a good lowpass/highpass pair of filters), the result of our algorithm will not depend strongly on the choice of filters. Furthermore, we have at our disposal a pair of a lowpass filter $H_0(z)$ and a highpass filter $H_1(z)$ such that the two filters, together with a synthesis pair $\tilde{H}_0(z)$ and $\tilde{H}_1(z)$, constitute a perfect reconstruction non-subsampled filter bank (i.e. they satisfy the perfect reconstruction condition (7.5)). With this model, the goal of the interpolation algorithm is to estimate the signals f_u and g_u at the output of $H_0(z)$ and $H_1(z)$, and then reconstruct an estimate of f_0 via the synthesis filters. The algorithm consists of two stages: initial estimation and refinement.

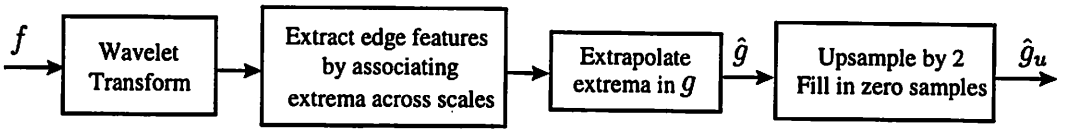


Figure 7.5: Estimation of g_u based on f.

Initial Estimation

An initial estimate $\hat{f}_u$ of the low frequency component f_u can be obtained by simply interpolating f using, for instance, linear or spline interpolation. To find an initial estimate of the high frequency component g_u, first notice that it contains information that would add "sharpness" to f. That is, if there were a sharp edge in the length $2N$ signal f_0, then the length N component f would contain a smoothed edge in this region. The reconstruction based solely on f_u would not be as sharp as the original edge in f_0. The information about the additional sharpness resides in g_u, whose essence is well captured by local extrema points (supposing that the filters used are appropriate for multiscale edge characterization). Thus the central part of the initial estimation is to find the values and positions of the local extrema in g_u. The detailed procedures are illustrated in Figure 7.5.

The first step in estimating g_u is to identify the edge regions via analysis of the available signal f. This identification is based on extracting local extrema of the wavelet transform of f which propagate across scales, and estimating the parameters in Equation (7.2) which characterize this propagation. The knowledge of an edge location in f conveys knowledge about the edge location in g_u as well (up to a possible ambiguity of ± 1 in location), since the wavelet transform of f is the decimated version (by a factor of 2) of the wavelet transform of f_0 starting from the scale $s = 2^2$ (see Figure 7.6):

$$W_{2^j} f_0[2n] = W_{2^{j-1}} f[n], \quad j = 2, 3, \ldots . \tag{7.8}$$

An edge information at $f[x_0]$ extracted from the analysis of $\{W_{2^j} f\}_{j=1,2,\ldots,J}$ and characterized in the parameters K and α of (7.2) translates to an edge at $f_0[2x_0]$. That is, an extremum in $W_{2^1} f_0$ can be estimated to be

$$W_{2^1} f_0[2x_0] = W_{2^0} f[x_0] = K \ .$$

Naturally, the downsampling operation in (7.8) introduces some ambiguity which needs to be addressed in the estimation process. More specifically, the true extrema points of

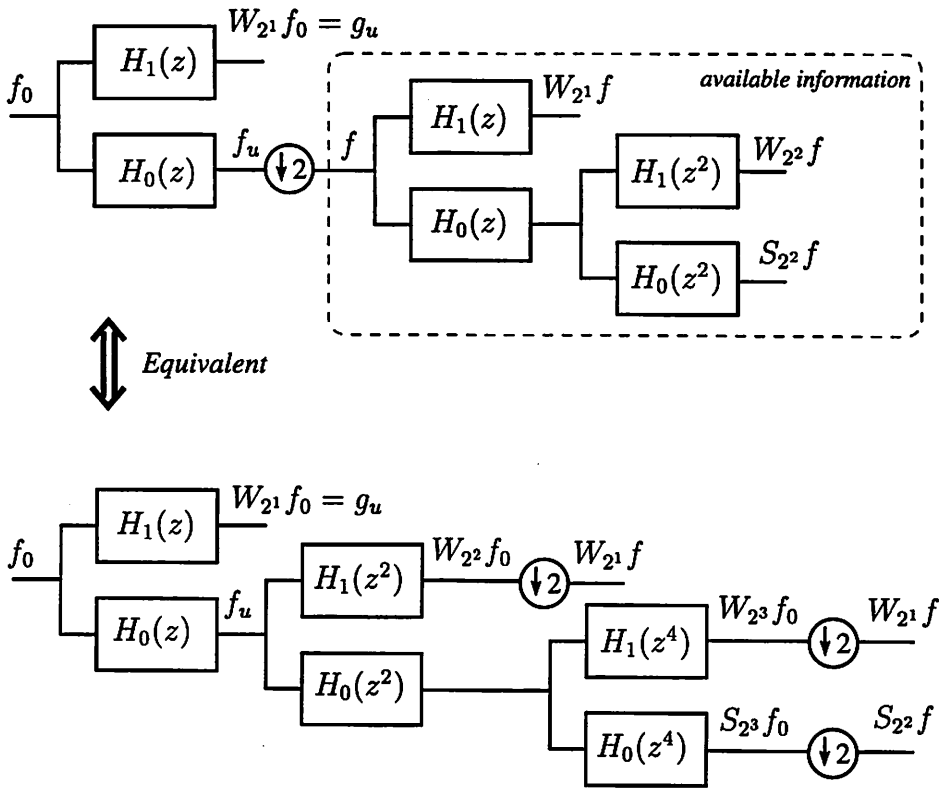


Figure 7.6: Illustrating the equivalence between the wavelet transform of f and the decimated version of the wavelet transform of f_0 starting from scale $s = 2^2$.

$\{W_{2^j} f_0\}_{j=1,\ldots,J+1}$ may not have been sampled in the downsampling process. Thus the edge identified at $f[x_0]$ may actually be at one of $\{f_0[2x_0 - 1], f_0[2x_0], f_0[2x_0 + 1]\}$. In Section 7.2.2, we will discuss constraints which allow possible corrections to this ambiguity.

The edge characterization allows the estimation of significant extrema points of g_u. To obtain an initial estimate of g_u that may be closer to the real g_u, the points in between are then filled in by linearly interpolating between the extrema points.

Refinement by Alternating Projection

The initial estimates of f_u and g_u can be further refined by identifying constraints which they should obey. These constraints define convex sets and one can utilize the POCS (projection onto convex sets) method to find a solution existing in the intersection of these sets, called the *reconstruction set*. The POCS method alternatingly projects the signal onto the various convex sets until it converges to a solution in the reconstruction set

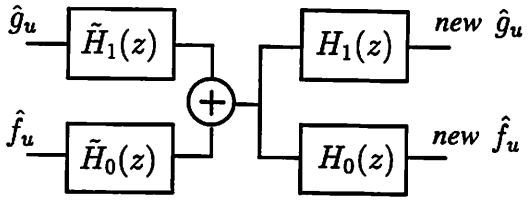


Figure 7.7: The projection operator, $P_{\mathcal{V}}$, onto the subspace $\mathcal{V}$, the range of the wavelet transform.

(provided that it is nonempty). Any solution in the reconstruction set is called a *consistent reconstruction* and it satisfies all the imposed constraints. There are three convex sets identified, labelled by $\mathcal{V}, \mathcal{S}$ and $\mathcal{E}$, respectively:

1. $\mathcal{V}$: The waveforms $\{\hat{f}_u, \hat{g}_u\}$ must belong to the subspace $\mathcal{V}$ of $l_2(\mathbb{Z})$, where $\mathcal{V}$ denotes the range of the wavelet transform.

2. $\mathcal{S}$: $\hat{f}_u$ must belong to a set $\mathcal{S}$, which comprises of length $2N$ signals whose downsampled version is consistent with f, the available signal.

3. $\mathcal{E}$: The edge points of f_0 (estimated from the analysis of f) should be reflected in local extrema of $\hat{g}_u$. $\mathcal{E}$ comprises of signals whose structure is consistent with the edge information, and $\hat{g}_u$ should reside in $\mathcal{E}$.

The first two items are hard constraints in that they follow from the consistency of the problem model in Figure 7.4. The third constraint is based on the estimation of how the signal should be at finer scales, and its purpose is to enhance the resolution of the reconstructed signal beyond that achieved by the first two constraints.

To speak of projections, it is more convenient to define the projection operator onto each convex set. The projection operator $P_{\mathcal{V}}$ of the subspace $\mathcal{V}$ is the operator in (7.7) and is pictorially illustrated in Figure 7.7: it puts the pair $(\hat{f}_u, \hat{g}_u)$ through the synthesis filter bank, followed by the analysis filter bank, where the filters obey the perfect reconstruction property in (7.5).

The projection operator $P_{\mathcal{S}}$ for the convex set $\mathcal{S}$ needs to ensure that $\hat{f}_u$ is consistent with the available signal f. At the very least, $\hat{f}_u[2n] = f[n]$ must hold. In practice, better performance could be achieved by placing restrictions such as smoothness constraints on the odd samples $\hat{f}_u[2n+1]$ as well, especially in regions of sharp variation. The details of this operator will be discussed in the implementation section.

The high frequency component $\hat{g}_u$ must reside in the set $\mathcal{E}$, which consists of signals that are consistent with the estimated edge information. However, only the *estimated* edge information is available and thus one must allow some error tolerances. In Section 7.2.2, we discuss the structure of the set $\mathcal{E}$ which allows varying degrees of leniency on the values and locations of the wavelet transform extrema, and finding a corresponding projection operator $P_{\mathcal{E}}$ which projects $\hat{g}_u$ onto $\mathcal{E}$.

The enhancement algorithm iteratively improves the estimates with the three projection operators, $P_{\mathcal{V}}, P_{\mathcal{S}}$, and $P_{\mathcal{E}}$. Let $\{\hat{f}_u^{(0)}, \hat{g}_u^{(0)}\}$ denote the initial estimates of $\hat{f}_u$ and $\hat{g}_u$. At the end of the k-th iteration, the estimates of $\hat{f}_u$ and $\hat{g}_u$ are

$$\{\hat{f}_u^{(k)}, \hat{g}_u^{(k)}\} = P_{\mathcal{E}}(P_{\mathcal{S}}(P_{\mathcal{V}}(\{\hat{f}_u^{(k-1)}, \hat{g}_u^{(k-1)}\}))) \ .$$

7.2.2 Implementation Details

This section addresses the implementation details of the algorithm. The association of extrema points across scales and the characterization of Lipschitz regularities are not so simple and straightforward when we deal with real data. Wavelet transform extrema points due to closely-spaced sharp variations may interfere with each other and make association difficult. This interference also complicates the estimation of the parameters in (7.2), and these complications will be discussed. The estimation of g_u will be elaborated, as well as the the exact structure of the set $\mathcal{S}$ and $\mathcal{E}$ and their respective projection operators.

Associating Extrema Across Scales

To extrapolate the extrema points, we need to first select important singularities and associate the corresponding extrema points across scales. Since $W_{2^1} f$ contains an abundance of extrema which are not necessarily due to global structures, extrema selection is, instead, done at a coarser scale, $s = 2^2$. For each extremum at scale $s = 2^2$, the algorithm searches in the other scales for extrema points to associate to those in scale $s = 2^2$.

Due to various reasons, only some extrema are observed to propagate from scale 2^j to 2^{j+1}. Extrema points at fine scales induced by closely spaced singularities may merge into one extremum point at coarse scales. Also, because the wavelet transform is discretized in both scale and space, one may not always observe the extrema points evolve across scales. For these reasons, it is sometimes difficult to associate the extrema points and thus some ad hoc rules are used. Suppose we are analyzing the m-th singularity which induces extrema

points at location $x_m^{(j)}$ in scale $s = 2^j$. The values of $x_m^{(j)}$ are unknown except for $x_m^{(2)}$, since the association starts from $x_m^{(2)}$ in scale $s = 2^2$. We search in other scales in a small neighborhood around $x_m^{(2)}$ to find extrema points which obey several rules. These extrema must be of the same sign and must all be maxima (or minima). Furthermore, it is reasonable to assume that the extrema values should not differ by too much from scale to scale (i.e. they are approximately of the same order of magnitude), thus we restrict the ratio between two extrema points of consecutive scales to be within a range: $1/2^{1.5} \leq |W_{2^j} f[x_m^{(j)}]|/|W_{2^{j+1}} f[x_m^{(j+1)}]| \leq 2^{1.5}$. If the chain of extrema cannot be associated for at least the first two scales, then the search is aborted.

Estimating high frequency component g_u

Let us first rewrite (7.2) in discrete-time and explicitly show the dependence of the local Lipschitz parameters on the different singularities. This results in

$$W_{2^j} f[x_m^{(j)}] = K_m (2^j)^{\alpha_m} , \quad j = 1, \ldots, J , \tag{7.9}$$

where $x_m^{(j)}$ is the location of the local extremum at scale 2^j corresponding to the m-th singularity, α_m is the Lipschitz regularity of f at the singular point, and K_m is a nonzero constant. The objective is to estimate K_m and α_m, and then extrapolate to an extremum point at scale $s = 2^0$ through estimating its location $x_m^{(0)}$ and value $W_{2^0} f[x_m^{(0)}]$. Recall that the relation between g_u, $W_{2^j} f_0$ and $W_{2^j} f$ is

$$W_{2^{j+1}} f_0[2n] = W_{2^j} f[n] \qquad \text{and} \qquad g_u[2n] \triangleq W_{2^1} f_0[2n] = W_{2^0} f[n].$$

Thus this extrapolation provides the first step in obtaining an estimate of the high frequency component $W_{2^1} f_0$ (or g_u) by first estimating $W_{2^0} f$.

For those singularities whose sequence of extrema, $W_{2^j} f[x_m^{(j)}], j = 1, \ldots, J$, is available, the parameters α_m and K_m in (7.9) can be estimated via linear regression on

$$\log_2(W_{2^j} f[x_m^{(j)}]) = \log_2 K_m + j\alpha_m , \quad j = 1, \ldots, J .$$

An initial estimate of the extremum point of the wavelet transform of f at scale 2^0 is then given by

$$W_{2^0} f[x_m^{(0)}] = \hat{K}_m = g_u[2x_m^{(0)}] .$$

The extrema location in scales $s = 2^0$ and $s = 2^1$ are assumed to be the same, that is, we let $x_m^{(0)} = x_m^{(1)}$.

The extrema extrapolation yields an estimate of the extrema positions and values in $g_u[2x_m^{(0)}]$. An initial estimate of the remaining points are obtained by linearly interpolating between consecutive extrema points.

Projection operator $P_{\mathcal{S}}$ for $\mathcal{S}$

From the problem model in Figure 7.4, it follows that $P_{\mathcal{S}}$ must, at the very least, assign $\hat{f}_u[2n] = f[n]$. In practice, this constraint alone does not prevent the spurious oscillations which often occur in sharp variation regions. To ameliorate this artifact, each odd sample $\hat{f}_u[2n+1]$ is bounded within an interval determined by the smoothness of $\hat{f}_u[2n]$ in that vicinity.

Let $\tilde{f}_u[n]$ be a length $2N$ cubic spline interpolated version of $f[n]$. Also let the discrete Laplacian gradient of $f[n]$ be defined as $\vec{\nabla} f[n] = f[n] - \frac{1}{2}(f[n-1] + f[n+1])$. The upper bound on the odd samples of $\hat{f}_u$ is made to be

$$\mathrm{HI}_{f_u}[2n+1] = \tilde{f}_u[2n+1] + \varepsilon * (|\vec{\nabla} f[n]| + |\vec{\nabla} f[n+1]|).$$

The value of $\varepsilon = .5$ was used. Similarly, the lower bound $\mathrm{LO}_{f_u}[2n+1]$ is calculated as

$$\mathrm{LO}_{f_u}[2n+1] = \tilde{f}_u[2n+1] - \varepsilon * (|\vec{\nabla} f[n]| + |\vec{\nabla} f[n+1]|).$$

To summarize, the operator $P_{\mathcal{S}}$ modifies $\hat{f}_u$ by assigning the even samples to $f[n]$ and bound the odd samples to within the interval $[\mathrm{LO}_{f_u}[2n+1], \mathrm{HI}_{f_u}[2n+1]]$.

Projection operator $P_{\mathcal{E}}$ for $\mathcal{E}$

Being the highpass component, the waveform $\hat{g}_u$ should reflect sharp variations in f_0. From the analysis of the wavelet transform of f, we have some knowledge of the extrema values and positions in $\hat{g}_u$. Hence, the set $\mathcal{E}$ can be thought of as the set of waveforms minimizing a specified cost function which penalizes when the extrema values do not conform to this knowledge. The operator $P_{\mathcal{E}}$ modifies $\hat{g}_u$ in a way such that the result has a lower cost.

This edge information, however, is estimated, and thus prone to inaccuracy especially when using data containing more than just isolated singularities. The downsampling process introduces errors as well. Knowing that a certain set of points are edge points imply that the other points are not. Thus, one needs to be careful to prevent additional

spurious edges from being created during the reconstruction. With this in mind, there are various degrees of leniency that can be employed when constructing the cost function. We can either (a) constrain $\hat{g}_u$ to retain the initial estimates throughout the reconstruction, (b) allow the values to be within an allowable range, or (c) have no constraints at all on the values. Approaches (a) and (c) are extreme cases, assigning either infinite cost for wrong values or no cost at all. The allowed interval of approach (b) serves as a moderation, and yields better results. In the following, we will not construct explicitly an analytical cost function, but rather describe how $P_{\mathcal{E}}$ modifies the input to conform to the edge information.

Extrema Location Because the initial estimate of $\hat{g}_u$ is obtained by interpolating from the estimate of the subsampled waveform g, the sampling may be such that we miss the true extrema and obtain instead the adjacent points. Thus for each extremum of $\hat{g}_u$, the points immediately next to it are also allowed to be extrema points to account for this ambiguity. More specifically, if we initially determine $x_m^{(0)}$ to be an extremum point in the length-N signal $\hat{g}$ (which translates to location $2x_m^{(0)}$ in $\hat{g}_u$), then after the projection $P_{\mathcal{S}} \circ P_{\mathcal{V}}$, $2x_m^{(0)}$ may not be an extremum point of $\hat{g}_u$ any longer. If the point of interest is a maximum (minimum) point, then the abscissa corresponding to the greatest (smallest) of $\{\hat{g}_u[2x_m^{(0)} - 1], \hat{g}_u[2x_m^{(0)}], \hat{g}_u[2x_m^{(0)} + 1]\}$ is assigned as the new local maximum (minimum).

Between Extrema Points The points between adjacent extrema points need also to be constrained to prevent spurious oscillations during the reconstruction. For example, by definition, the points between a pair of adjacent maximum and minimum points should have values bounded by these extrema values, and, furthermore, the slopes of these in-between points should be monotonic so that there is no other extrema among them. Such a consistent reconstruction can be achieved by a simple algorithm proposed in [17] which reconstructs a signal from only its wavelet extrema points. For the interpolation problem, it has been found experimentally that these constraints are too restrictive for reconstructing g_u, since the extrema information is estimated and more leniency should be allowed. Therefore, "softer" constraints will be described.

In predicting the extrema points of g_u, only a subset of them could be extrapolated from the coarser scales, due to the fact that coarser scales typically have less extrema than finer scales. Thus, for each extremum predicted in $\hat{g}_u$, we only assume that it is valid *locally*. For each maximum (minimum) examined, the points in small neighborhood around it (a centered window of 7 is used here) are clipped to be less (greater) than or equal to this maximum (minimum) point. Since we are working with greyscale images, another

by the matrix

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

and the upsampled version of $|\vec{\nabla} f[n_1, n_2]|$ (upsampled by a factor of 2 in each direction). The lower bound $\mathrm{LO}_{f_u}[n_1, n_2]$ is defined similarly, but with a subtraction substituting the addition in (7.11). The operator P_S then bounds $\hat{f}_u[n_1, n_2]$ to be within $[\mathrm{LO}_{f_u}[n_1, n_2], \mathrm{HI}_{f_u}[n_1, n_2]]$.

Each of the $2N$ available rows of the row component $\hat{g}_{1,u}$ and the $2N$ available columns of the column component $\hat{g}_{2,u}$ is treated as a separate 1-D problem, and is project onto $\mathcal{E}$ using the 1-D operator $P_{\mathcal{E}}$ described in Section 7.2.1.

7.3 Experimental Results

The performance of the algorithm will be compared against several standard methods such as bilinear interpolation, bicubic spline interpolation, and bicubic spline followed by *unsharp masking* [27]. Unsharp masking is a commonly used method for boosting the high frequency portion of a signal. The general operation is to take the input $f[n_1, n_2]$ and yield

$$v[n_1, n_2] = f[n_1, n_2] + \lambda u[n_1, n_2]$$

where $\lambda > 0$ and $u[n_1, n_2]$ is a defined gradient at location $[n_1, n_2]$. A commonly used gradient is the discrete Laplacian defined in (7.10), and a commonly used value for λ is 1. The filters used in the wavelet decomposition are tabulated in Appendix A, and three levels of decomposition are computed.

In order to obtain MSE or PSNR measurements in addition to the visual judgement, we take a $2N \times 2N$ image, $f_0[n_1, n_2]$, filter it with some lowpass filter, $\varphi[n_1, n_2]$, and downsample it to obtain the available $N \times N$ image, $f[n_1, n_2]$. The choice of the filter $\varphi[n_1, n_2]$ is a parameter which we wish to test to see how sensitive the algorithm is to this choice.

The reconstructed image generally does not change much after 7-8 iterations, both in visual quality and in PSNR measurements. But to see its best performance, the measurements listed and images displayed are after 15 iterations.

Portions from the images *Barbara*, *Lena*, and *Baboon* are extracted as the original high resolution image f_0, shown in Figure 7.9. Each 2-D lowpass filter $\varphi[n_1, n_2]$ is a separable filter, $\varphi[n_1, n_2] = \varphi[n_1]\varphi[n_2]$. The three choices of $\varphi[n]$ are

$\varphi_1[n]$: 12-tap symmetric lowpass filter generated by MATLAB `fir1(11, 0.5)`

$\varphi_2[n]$: 11-tap symmetric lowpass filter generated by MATLAB `fir1(10, 0.5)`

$\varphi_3[n]$: the same filter $h_0[n]$ used in the wavelet analysis.

The even-length filter has a delay of 1/2, while the odd-length filter has delay 0. The reason for choosing $\varphi_3[n]$ is to obtain a benchmark, to see how well the algorithm can perform when we "cheat" by pretending to know the nature of degradation from f_0 to f.

Each set of experiments consist of taking one of the four test image and one of the three $\varphi_i[n]$ lowpass filters, and interpolate the images using four different interpolation methods. Here only one set of experiment for each test image will be shown and interested readers can view the rest at the website `http://www-wavelet.eecs.berkeley.edu/~grchang/Interpolation.html` . The *Barbara* experiment with filter $\varphi_1[n]$ is shown in Figure 7.10, *Lena* with $\varphi_2[n]$ in Figure 7.11, *Baboon-A* with $\varphi_2[n]$ in Figure 7.12, and *Baboon-B* with $\varphi_3[n]$ in Figure 7.13. In all the experiments, the wavelet interpolation approach yields images considerably sharper than those from linear and cubic spline interpolation. The unsharp masking method comes very close to producing images almost as sharp as those from the wavelet method, though in high frequency images such as *Baboon-A*, one can see that the unsharp masking method is slightly more blurry than the wavelet method. Visually, experiments from the three different filters $\phi_i[n]$ yield very similar results and conclusions, though the PSNR tells quite a different story. Though the PSNR is not a good indication of image quality, it is nevertheless frequently used, and the results are tabulated in Tables 7.1, 7.2, and 7.3. The best numbers are highlighted in bold. Note that in the *Barbara* experiments, the interpolated images show aliasing on the scarfs. This is through no fault of the interpolation algorithms, but rather that the downsampling operation used to obtain the test image f already introduced aliasing.

The PSNR results are very sensitive to the choice of lowpass filter $\varphi_i[n]$. For the even-length filter, $\varphi_1[n]$, the methods with the highest PSNR are either the wavelet or the linear method. When the odd-length filter, $\varphi_2[n]$, is used, unsharp masking yields the highest PSNR. With $\varphi_3[n] = h_0[n]$, not surprisingly, the wavelet approach yields the highest PSNR.

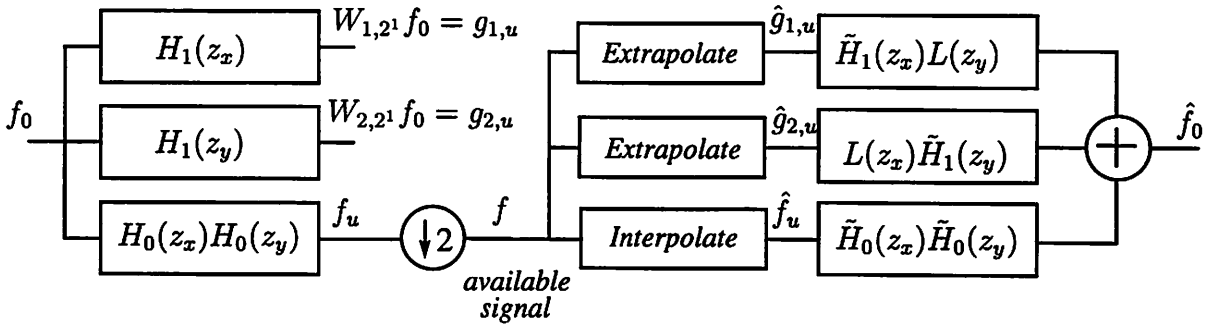


Figure 7.8: Interpolation problem model for 2-D.

optimization is to clip all the pixel values to be within [0,255]. These constraints are very lenient, and we prefer them over the more restrictive ones when analyzing real data, where it is difficult to ensure the robustness of capturing all the extrema points. In our previous work in [10], we used strict constraints such as bounding extrema values to be within an estimated range, and enforcing monotonicity between consecutive extrema points. This sometimes resulted in images with some unpleasant artifacts such as overly pronounced edges or small streaks. Here we find that the softer constraints yield much more pleasant looking results.

7.2.3 Enhancement Algorithm for 2-D Images

In general, analyzing a 2-D problem by treating the two coordinates independently is not an optimal approach. However, for computational reasons, we propose here to treat the two coordinates separately. The problem model for the 2-D case is analogous to the 1-D case, and is illustrated in Figure 7.8 for clarity. To iterate, the goal is to extrapolate from f information about f_u, $g_{1,u} \triangleq W_{1,2^1} f_0$ and $g_{2,u} \triangleq W_{2,2^1} f_0$, which are the necessary components of f_0.

Initial estimates

In the wavelet transform, the data is filtered by the separable 2-D filter bank as discussed earlier. The wavelet transform generates the row components $\{W_{1,2^j} f\}_{j=1,\ldots,J}$, the column components $\{W_{2,2^j} f\}_{j=1,\ldots,J}$, and the low resolution component $S_J f$, all of which are $N \times N$. Bicubic spline interpolation is used to obtain the initial estimate of size $2N \times 2N$ signal f_u. The i-th rows of $\{W_{1,2^j} f\}_{j=1,\ldots,J}$ are used to estimate the i-th row of

the scale $s = 2^0$ row component as in the 1-D case. The columns are processed likewise. After interpolating this row to length $2N$, we have an initial estimate of the $2i$-th row of $W_{1,2^1} f_0$.

Having only extrema constraints on the even lines may result in jagged edges during the reconstruction process. To ameliorate this artifact, we estimate the extrema of an odd row based on its two neighboring even rows. Typical images have smooth contours which traverse through numerous rows or columns. Thus, for a given extremum on the $2i$-th row, if there is an extremum on the $(2i + 2)$-th row which is of the same type (i.e. both maxima or both minima) and same sign, and is in a close proximity (within ± 4 pixels), then we assume there is an extremum of the same type and sign on the $(2i + 1)$-th row. The location and value are taken to be the average of the corresponding extrema on the neighboring rows. For simplicity, averaging is used rather than fitting a smoothed curve across these lines, since the considered neighborhood is small, and the difference in location is not significant.

A similar analysis is also done on the columns of $\{W_{2,2^j} f\}_{j=1,\ldots,J}$ to obtain an estimate of $W_{2,2^1} f_0$.

Alternating projections

The estimates $\hat{f}_u$, $\hat{g}_{1,u}$ and $\hat{g}_{2,u}$ are iteratively refined using constraints analogous to those proposed in the 1-D case. The 2-D version of $P_{\mathcal{V}}$, $P_{\mathcal{S}}$ and $P_{\mathcal{E}}$ will be described.

The projection operator $P_{\mathcal{V}}$ is simply a one-level 2-D inverse wavelet transform followed by a one-level 2-D forward wavelet transform. The operator $P_{\mathcal{S}}$ first makes the assignment $\hat{f}_u[2n_1, 2n_2] = f[n_1, n_2]$ for the even samples. To constrain the odd samples, we define $\tilde{f}_u[n_1, n_2]$ to be a $2N \times 2N$ bicubic spline interpolated version of $f[n_1, n_2]$. Also let the discrete Laplacian gradient of $f[n_1, n_2]$ be

$$\vec{\nabla} f[n_1, n_2] = f[n_1, n_2] - \frac{1}{4}(f[n_1 - 1, n_2] + f[n_1 + 1, n_2] + f[n_1, n_2 - 1] + f[n_1, n_2 + 1]). \quad (7.10)$$

The upper bound on the samples of $f_u[n_1, n_2]$ is taken to be

$$\mathrm{HI}_{f_u}[n_1, n_2] = \tilde{f}_u[n_1, n_2] + w[n_1, n_2] * \mathrm{Upsample}(|\vec{\nabla} f[n_1, n_2]|), \quad (7.11)$$

where the second term is the convolution between a weighting function $w[n_1, n_2]$ depicted

Figure 7.9: Four test images for the interpolation algorithm. Clockwise from top left: *Barbara*, *Lena*, *Baboon-A*, and *Baboon-B*.

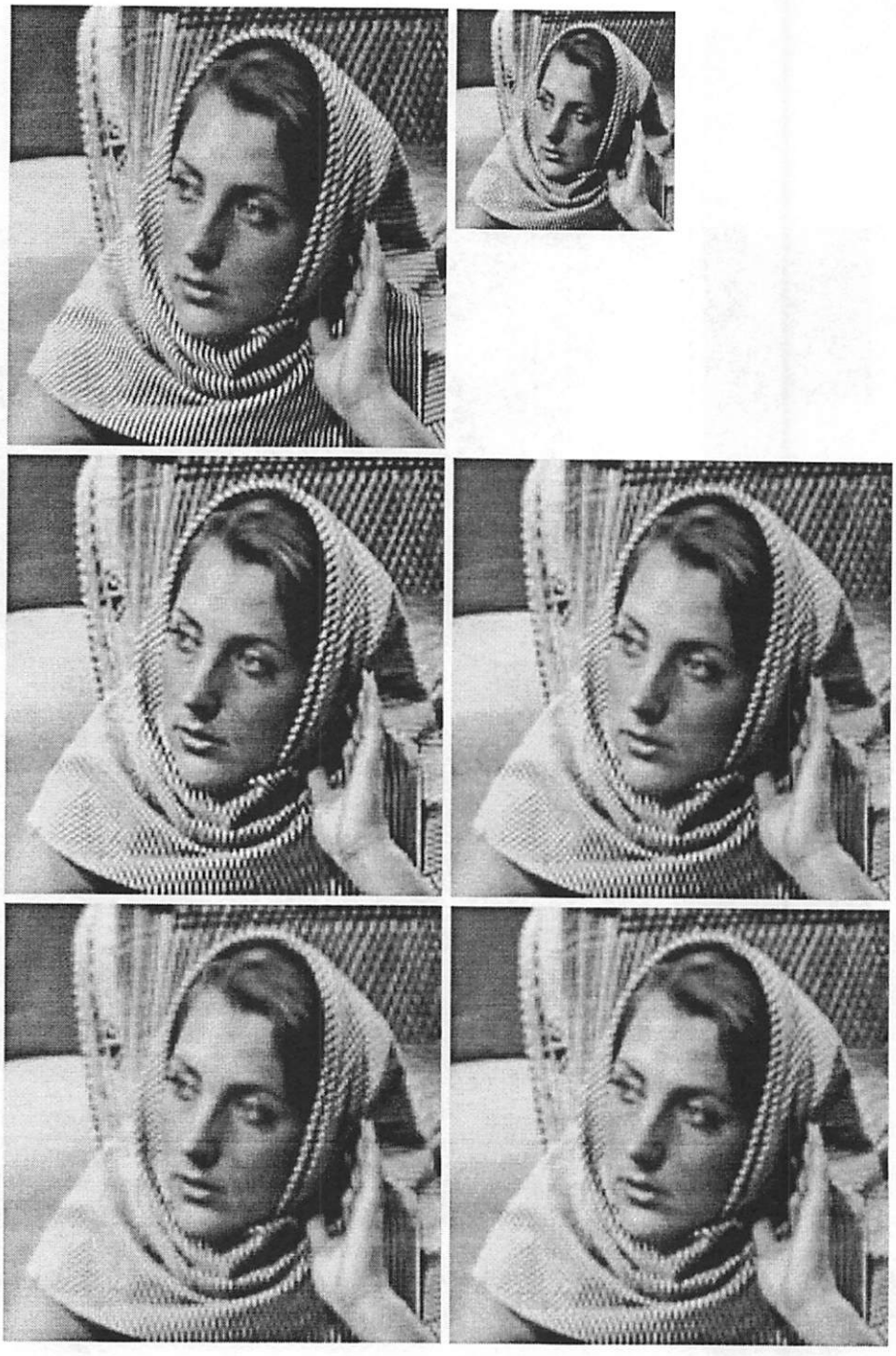


Figure 7.10: Interpolation of the *Barbara* image, with the even-length lowpass filter $\varphi_1[n]$. From left to right, top to bottom: (a) Original 256×256 image. (b) Lowpass, available image, 128×128. (c) Wavelet-based interpolation. (d) Cubic spline interpolation with unsharp masking. (e) Linear interpolation. (f) Cubic spline interpolation.

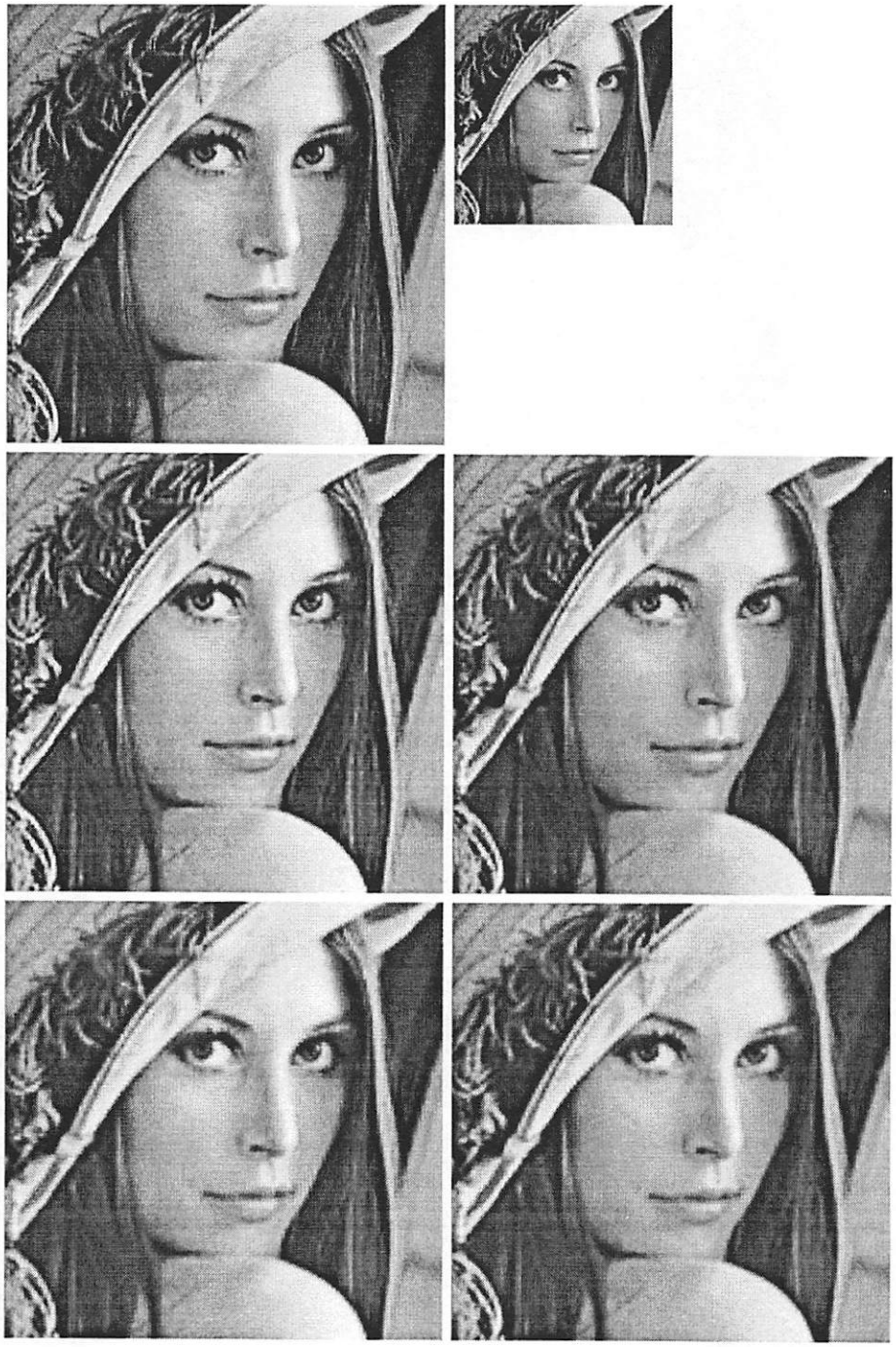


Figure 7.11: Interpolation of the *Lena* image, with the odd-length lowpass filter $\varphi_2[n]$. From left to right, top to bottom: (a) Original 256×256 image. (b) Lowpass, available image, 128×128. (c) Wavelet-based interpolation. (d) Cubic spline interpolation with unsharp masking. (e) Linear interpolation. (f) Cubic spline interpolation.

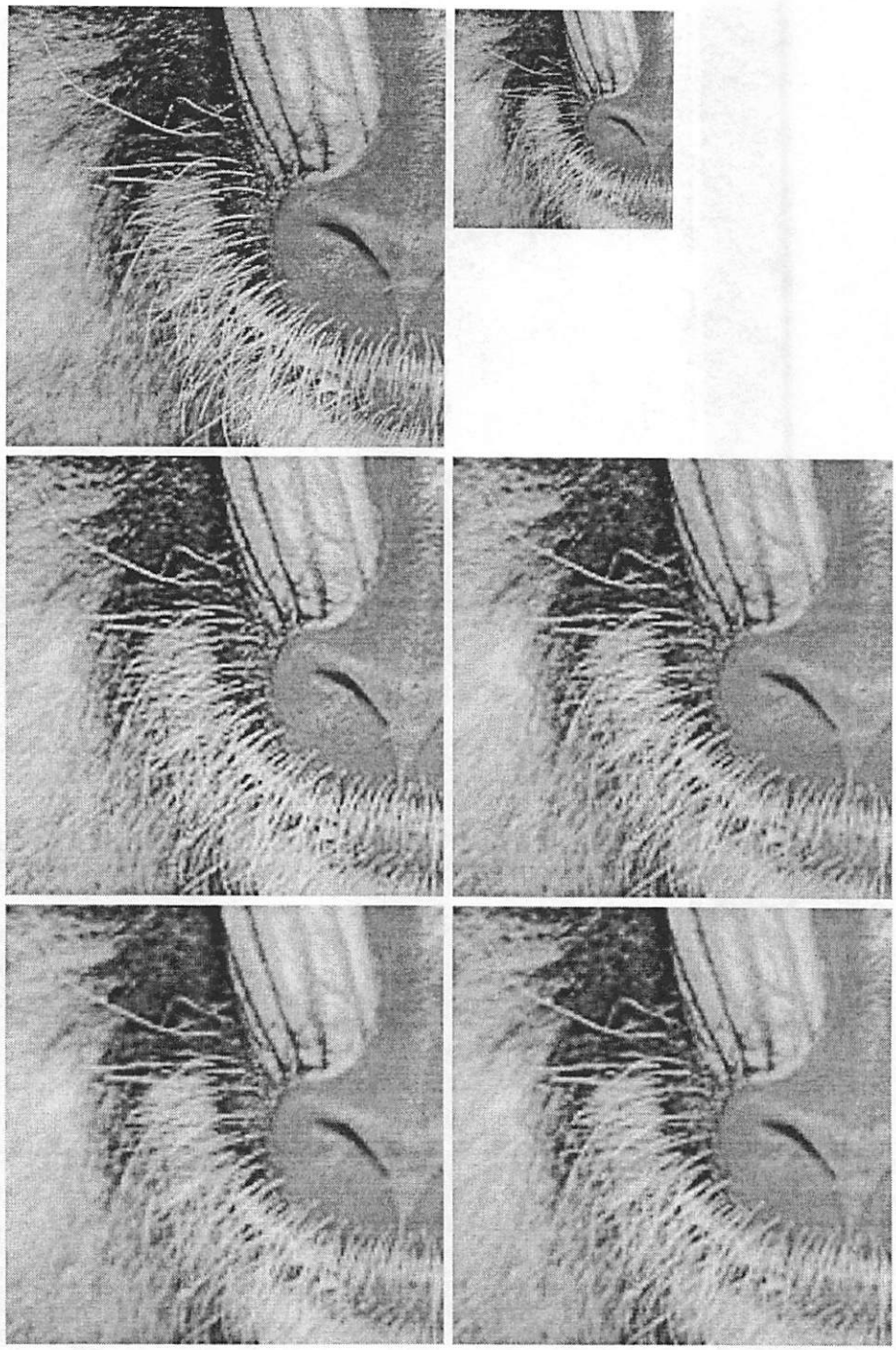


Figure 7.12: Interpolation of the *Baboon-A* image, with the odd-length lowpass filter $\varphi_2[n]$. From left to right, top to bottom: (a) Original 256×256 image. (b) Lowpass, available image, 128×128. (c) Wavelet-based interpolation. (d) Cubic spline interpolation with unsharp masking. (e) Linear interpolation. (f) Cubic spline interpolation.

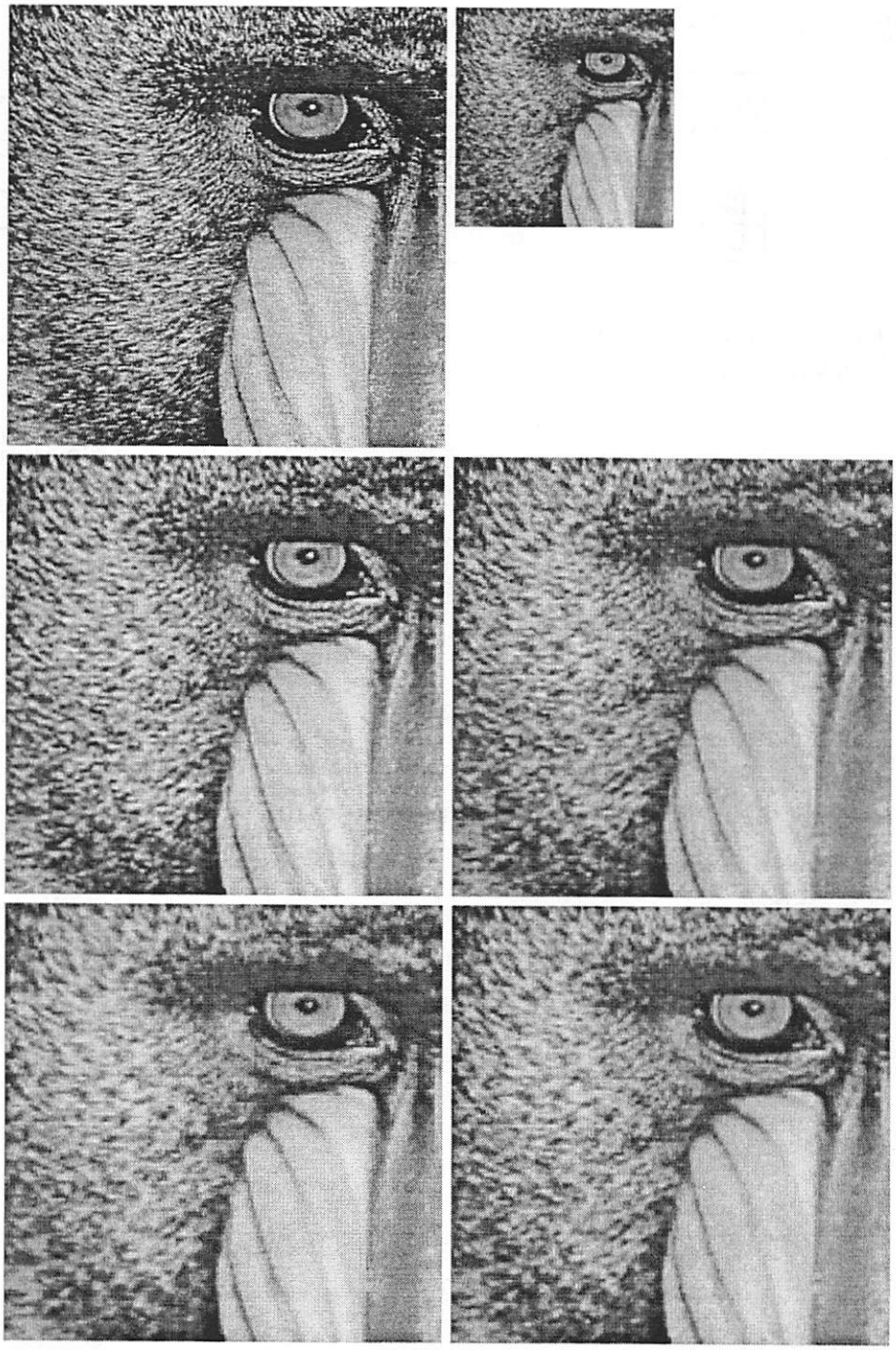


Figure 7.13: Interpolation of the *Baboon-B* image, with the lowpass filter $\varphi_3[n] = h_0[n]$. From left to right, top to bottom: (a) Original 256×256 image. (b) Lowpass, available image, 128×128. (c) Wavelet-based interpolation. (d) Cubic spline interpolation with unsharp masking. (e) Linear interpolation. (f) Cubic spline interpolation.

Table 7.1: Comparing PSNR of different methods when the given image is downsampled after lowpass filtering by the even-length filter $\varphi_1[n]$.

Image	Wavelet	Cubic	Linear	Cubic + UnsharpMask
Barbara	21.36	21.20	**21.36**	20.50
Lena	**22.71**	21.44	21.69	20.86
Baboon-A	21.86	21.66	**21.90**	20.91
Baboon-B	**19.80**	19.56	19.71	18.96

Table 7.2: Comparing PSNR of different methods when the given image is downsampled after lowpass filtering by the odd-length filter $\varphi_2[n]$.

Image	Wavelet	Cubic	Linear	Cubic + UnsharpMask
Barbara	24.64	26.79	26.55	**27.34**
Lena	27.48	32.16	31.07	**32.69**
Baboon-A	25.32	27.37	26.98	**27.79**
Baboon-B	22.41	23.59	23.31	**23.92**

Figures 7.14 shows the PSNR as a function of the iteration number for the images *Barbara*, *Lena*, *Baboon-A*, and *Baboon-B*. Each plot shows three curves, for the three choices of lowpass filter $\varphi_i[n]$. As mentioned previously, the reconstructed image remains visually indistinguishable after 7-8 iterations. The PSNR also shows quick convergence, though it is not always monotonically increasing. For filters $\varphi_1[n]$ and $\varphi_2[n]$, the PSNR actually decreases after the 3rd or 4th iteration, but for $\varphi_3[n]$, it is monotonically increasing. Again, we want to stress that these PSNR numbers are not necessarily a good measure, and the visual quality of the wavelet approach is the best in all cases.

7.4 Summary

We have proposed a wavelet based method for image interpolation which preserves the regularity of edge points. By characterizing edge points via the wavelet transform, we extrapolate the extrema needed at a finer scale for reconstruction of a higher resolution image. The result shows that the enhanced image is significantly sharper than simple schemes such as linear and cubic spline interpolation, and still noticeably sharper than unsharp masking.

The better performance comes at an expense of higher complexity and more com-

Table 7.3: Comparing PSNR of different methods when the given image is downsampled after lowpass filtering by the filter $\varphi_3[n] = h_0[n]$.

Image	Wavelet	Cubic	Linear	Cubic + UnsharpMask
Barbara	**26.94**	24.99	24.86	25.14
Lena	**32.46**	27.86	27.49	28.13
Baboon-A	**27.57**	25.53	25.35	25.72
Baboon-B	**26.06**	24.62	24.45	24.76

putation than the linear methods, and the nonlinearity of our method makes it difficult to characterize the behavior of the algorithm analytically. Because the theoretical framework is geared towards isolated singularities, this method is not necessarily appropriate for, say, texture images.

For future research, we could explore the potential of processing the image with 2-D neighborhoods instead of a separable 1-D approach. Since the method proposed here is for isolated singularities, a more comprehensive interpolation algorithm would be to segment the images into regions of isolated singularities and textures and process them differently.

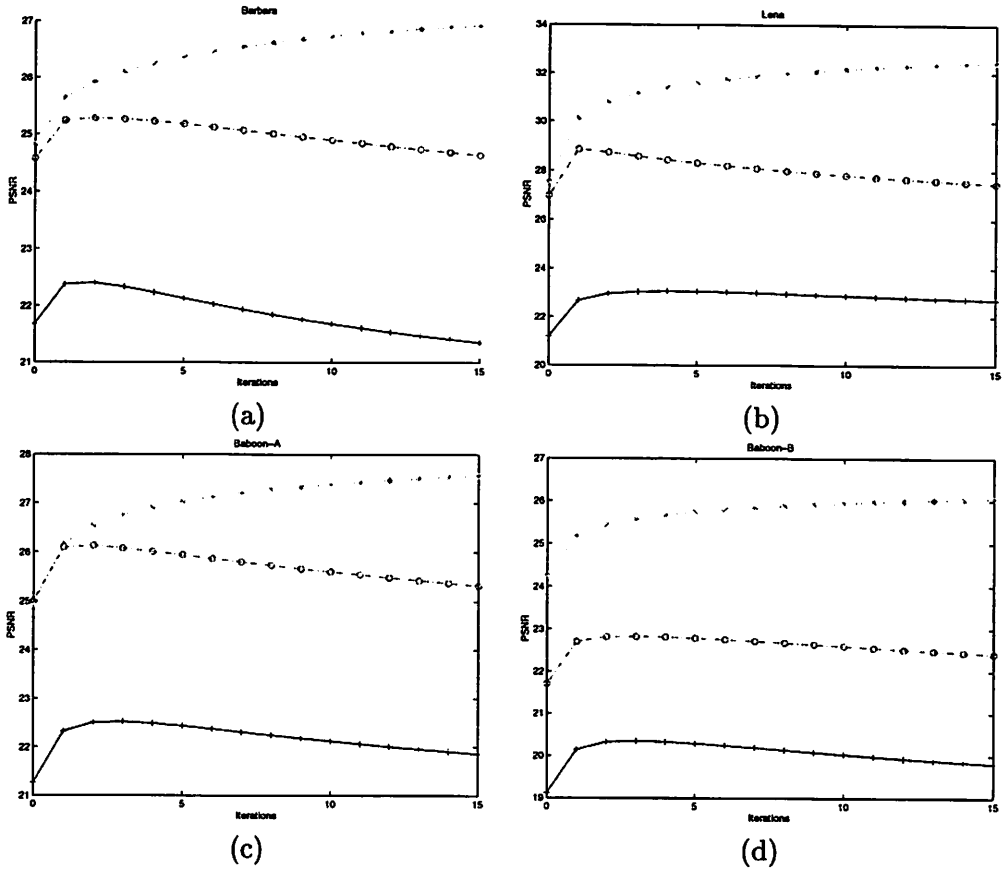


Figure 7.14: PSNR as a function of iterations. The curves are for images with φ_1 (+—+), φ_2 (o — · — o), and φ_3 (∗ · · · ∗). (a) *Barbara.* (b) *Lena.* (c) *Baboon-A.* (d) *Baboon-B.*

Chapter 8

Conclusion

This thesis explored various aspects of image denoising and interpolation. We summarize here the findings and possible future research directions.

8.1 Models for Image Restoration

Many ideas and techniques which have worked well in image compression and edge analysis were adapted to image restoration problems in this thesis. Results showed considerable success over conventional methods and some recent works in literature. These successes led to the conclusion that good image coders and edge analysis do provide good image models applicable to other areas of image processing. Our algorithms present a break-away from the more traditional filtering or stochastic modeling approaches in image restoration. While filtering and stochastic modeling are still important components of our algorithms, the combination with adaptive and non-linear techniques in image modeling and analysis make these algorithms outperform conventional methods. The basic analysis framework used the wavelet decomposition, which allowed viewing images in a natural multiresolution fashion and provided a convenient basis for the problem models. In the following text, the details of the various results are described.

8.1.1 Bayesian Approach to Threshold Selection

There have been many works in the literature addressing the issue of threshold selection for wavelet threshold denoising. We felt that none of them are ideal for real image denoising, thus we proposed our own approach to this problem. Our framework was to model

the signal coefficients as random variables with Generalized Gaussian distribution and to find the threshold which minimizes the mean squared error using the soft-threshold estimate. We proposed a near-optimal threshold which is effective, intuitive, and easy to compute. This threshold selection provided the basis for the subsequent denoising algorithms.

8.1.2 Lossy Compression for Denoising

Based on the intuition that typical images have predictable structures more easily compressible than random noise, several works have proposed using lossy compression to distinguish between signal and noise for noise removal. Such an approach also achieves the "kill two birds with one stone" benefit of simultaneous compression and denoising. Prior works were fuzzy on choosing a coder or did not achieve compression in a practical sense. In our work, we made the connection between compression and the wavelet thresholding denoising operation, and developed a systematic approach to achieve both denoising and compression. Results showed that, though some quantization noise was introduced, lossy compression did remove a considerable amount of the observation noise, especially when the noise was significant.

8.1.3 Spatially Adaptive Denoising Algorithm

Most successful image processing applications employ spatially adaptive algorithms, since images typically have changing spatial characteristics. Thus, we investigated a spatially adaptive version of the wavelet thresholding technique. The manner of adaptation is based on context modeling, a frequently used method for adapting coders to changing signal characteristics. The other part of the algorithm is based on smoothing the corrupted signal in its overcomplete expansion, which essentially provides an additional averaging for reducing the noise. The spatial adaptivity and overcomplete expansion together yielded results significantly better than either one alone, both in visual quality and mean square error measurements.

8.1.4 Multiple Noisy Copies Denoising

For applications where a receiver has available multiple noisy copies of the same image, we investigated the optimal ordering of averaging and wavelet thresholding for combining them into one denoised image. The finding showed that the ordering is less than

obvious, and it depends on the noise power, the signal model, and the number of copies available.

8.1.5 Edge-Preserving Interpolation

Conventional interpolation or magnification algorithms assume some smoothness constraints on the underlying image. This assumption may yield overly blurred images. We proposed a regularity-preserving interpolation algorithm which adapts to the local regularity. The available image was modeled as a low resolution image, from which we wished to obtain a higher resolution image. Edge analysis and extrapolation was performed on the available image to estimate the needed details. The experiments produced images much sharper than those from the conventional spline and linear interpolation. Furthermore, the problem model provided a justified framework for estimating the high frequency component, rather than an *ad hoc* post-process sharpening such as unsharp masking.

8.2 Research Directions

During the course of the thesis, there emerged many relevant issues which would be natural extensions of our work thus far. Below are some of these issues.

8.2.1 Other Noise Models

In order to obtain some theoretical results, we have assumed the *iid* Gaussian noise model. While many practical problems are modeled this way, there are other practial problems with very different noise behaviors. Examples include shot noise and "snow" noise (as on the television). Sometimes the noise samples may be correlated among each other, and may also be correlated with the image. Such is the case for a lossy-compressed image, where the error residuals shows a strong correlation with the image along the edge regions. Another example is the block-based compression (such as JPEG), which, at low bitrates, yields considerable artifacts along the boundaries of the blocks. It would be interesting to see if our methods are suitable or can be adapted to these different noise characteristics.

8.2.2 Texture Modeling

Many of the intuitions used in our algorithms (especially the edge-preserving interpolation) is based on isolated edge analysis. For textures, another paradigm of modeling would be required. Typically, a piece of texture is decomposed into a deterministic component and a random field component, which is a useful representation for characterizing and synthesizing. It merits an investigation to extend this representation to, say, the interpolation and denoising framework. For the interpolation problem, since the high resolution image and the available image is related by a lowpass filter followed by a downsampler, knowledge of the deterministic and stochastic behavior of the available image can be extended to the high resolution image as well. We did some preliminary studies towards this direction in [9], but did not probe it deep enough. For the denoising problem, one can extract the texture regions and denoise it with a texture model. We used a primitive image segmentation method in [11] to separate the image into different regions and denoise them differently. It showed promising results, and this idea can potentially be greatly improved with a more sophisticated image segmentation method and texture model.

8.2.3 Restoration from a Blurred and Noisy Image

Another domain of image restoration deals with recovering an image which has been degraded by both a blurring function and additive noise. Practical applications include removing the blur due to camera out-of-focus, motion blur, scatter blur (from X-ray, for example), to name a few. A simple-minded inversion of the blurring operator, even when it is known, is a bad idea since the inversion of a lowpass filter (which is essentially what a blur is) amplifies the high frequency noise. Thus, restoration typically comes in the form of *regularization*, where the blur and the noise are decreased little by little in a regularized fashion. We have done some preliminary experiments on using compression methods as regularization and the results were promising. That is, from the compressibility of the degraded image, we estimate how compressible the original image is. During the recovery process, this estimated information is kept consistent with the estimated image. This framework presents a very interesting approach to the image restoration and warrants further investigations. A similar idea has also been proposed independently by Liu and Moulin [36].

Appendix A

Wavelet Filter Coefficients

Daubechies' symmlet with 8 vanishing moments [19] has 16 coefficients as displayed below:

$$\begin{aligned}\text{Symmlet } 8 = \{ \quad & 0.002672793393, \quad -0.000428394300, \quad -0.021145686528, \\ & 0.005386388754, \quad 0.069490465911, \quad -0.038493521263, \\ & -0.073462508761, \quad 0.515398670374, \quad 1.099106630537, \\ & 0.680745347190, \quad -0.086653615406, \quad -0.202648655286, \\ & 0.010758611751, \quad 0.044823623042, \quad -0.000766690896, \\ & -0.004783458512\}\end{aligned}$$

The filter coefficients of H_0, H_1, $\tilde{H}_0$, $\tilde{H}_1$ and L corresponding to Mallat's quadratic spline wavelets [41], used in the interpolation algorithm in Chapter 7, are tabulated in Table A.1.

Table A.1: Filter coefficients of the quadratic spline wavelets.

n	H_0	H_1	$\tilde{H}_0$	$\tilde{H}_1$	L
-3				-0.001953125	0.0078125
-2			0.125	-0.01367125	0.046875
-1	0.125		0.375	-0.04296875	0.1171875
0	0.375	0.5	0.375	0.04296875	0.65625
1	0.375	-0.5	0.125	0.01367125	0.1171875
2	0.125			0.001953125	0.046875
3					0.0078125

Note that the filters H_1 and $\tilde{H}_1$ are different from those listed in [41] because we have normalized them such that $H_1(z = -1) = 1$.

The multiplicative constants, λ_j's, used in the non-subsampled filter bank in Chapter 7 are listed in Table A.

Table A.2: Multiplicative constants used in the non-subsampled filter bank using the quadratic spline wavelet.

j	λ_j
0	1.0
1	0.75
2	0.6875
3	0.6719
4	0.6680
5	0.6670
6	0.6668

Bibliography

[1] F. Abramovich, T. Sapatinas, and B. Silverman. Wavelet thresholding via a Bayesian approach. preprint, 1996.

[2] A. Antoniadis, I. Gijbels, and G. Grégoire. Model selection using wavelet decomposition and applications. *Biometrika*, 84(4):751–763, 1997.

[3] Z. Berman and J.S. Baras. Properties of the multiscale maxima and zero-crossings representations. *IEEE Trans. on Signal Processing, Special Issue on Wavelets and Signal Processing*, 41(12):3216–3231, December 1993.

[4] K.A. Birney and T.R. Fischer. On the modeling of DCT and subband image data for compression. *IEEE Trans. Image Processing*, 4(2):186–193, 1995.

[5] P. Burman. A comparative study of ordinary cross-validation, ν-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76:503–514, 1989.

[6] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, April 1983.

[7] J. Canny. A computational approach to edge detection. *IEEE Trans. Patt. Anal. Machine Intell.*, PAMI-8:679–698, 1986.

[8] R.J. Carroll. Adapting for heteroscedasticity in linear models. *Annals of Statistics*, 10(4):1224–1233, 1982.

[9] S.G. Chang. Image interpolation using wavelet-based edge enhancement and texture analysis. Master's thesis, U.C. Berkeley, May 1995. Also ERL Technical Memo M95/100.

[10] S.G. Chang, Z. Cvetković, and M. Vetterli. Resolution enhancement of images using wavelet transform extrema extrapolation. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 4, pages 2379–2382, Detroit, MI, May 1995.

[11] S.G. Chang and M. Vetterli. Spatial adaptive wavelet thresholding for image denoising. In *Proc. IEEE Int. Conf. Image Processing*, volume 2, pages 374–377, Santa Barbara, CA, November 1997.

[12] S.G. Chang, B. Yu, and M. Vetterli. Bridging compression to wavelet thresholding as a denoising method. In *Proc. Conf. Information Sciences and Systems*, pages 568–573, Baltimore, MD, March 1997.

[13] S.G. Chang, B. Yu, and M. Vetterli. Image denoising via lossy compression and wavelet thresholding. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 604–607, Santa Barbara, CA, November 1997.

[14] H. Chipman, E. Kolaczyk, and R. McCulloch. Adaptive Bayesian wavelet shrinkage. *Journal of the American Statistical Assoc.*, 92(440):1413–1421, 1997.

[15] M. Clyde, G. Parmigiani, and B. Vidakovic. Multiple shrinkage and subset selection in wavelets. Discussion paper 95-37, ISDS, Duke University, Durham, NC, 1995.

[16] R.R. Coifman and D.L. Donoho. Translation-invariant de-noising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, Lecture Notes in Statistics. Springer-Verlag, 1995.

[17] Z. Cvetković and M. Vetterli. Discrete-time wavelet extrema representation: Design and consistent reconstruction. *IEEE Trans. on Signal Processing*, 43(3):681–693, March 1995.

[18] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 41:909–996, November 1988.

[19] I. Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, Philadelphia, 1992.

[20] I. Daubechies. Orthonormal bases of compactly supported wavelets ii: Variations on a theme. *SIAM J. Math. Anal.*, 24:499–519, 1993.

[21] D.L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Information Theory*, 41(3):613–627, May 1995.

[22] D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[23] D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Assoc.*, 90(432):1200–1224, December 1995.

[24] D.L. Donoho, I.M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: Asymptopia? *Journal of the Royal Statistical Society, Ser. B*, 57(2):301–369, 1995.

[25] D.S. Ebert. *Texturing and Modeling: A procedural approach.* AP Professional, Cambridge, MA, 1994.

[26] J.M. Francos, A.Z. Meiri, and B. Porat. A unified texture model based on a 2-D Wold-like decomposition. *IEEE Trans. on Signal Processing*, 41(8):2665–2678, August 1993.

[27] A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[28] M. Jansen and A. Bultheel. Multiple wavelet threshold estimation by generalized cross validation for data with correlated noise. Technical report TW250, 1996.

[29] M. Jansen, M. Malfait, and A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56(1), 1996.

[30] I.M. Johnstone and B.W. Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the Royal Statistical Society, Ser. B*, 59, 1997.

[31] Masoud Khansari, 1996. Personal communication.

[32] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[33] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelets. *IEEE Trans. Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):533–555, March 1992.

[34] H. Krim and J.C. Pesquet. Multiresolution analysis of a class of nonstationary processes. *IEEE Transactions on Information Theory*, 41(4):1010–1020, 1995.

[35] J. Liu and P. Moulin. Complexity-regularized image denoising. In *Proc. IEEE Int. Conf. Image Processing*, volume 2, pages 370–373, Santa Barbara, CA, October 1997.

[36] J. Liu and P. Moulin. Complexity-regularized image restoration. In *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, October 1998.

[37] S. LoPresto, K. Ramchandran, and M. Orchard. Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework. In *Proc. Data Compression Conference*, Snowbird, Utah, March 1997.

[38] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbb{R})$. *Trans. Amer. Math. Soc*, 315:69–87, September 1989.

[39] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Pat. Anal. Mach. Intell.*, 11(7):674–693, July 1989.

[40] S. Mallat and W.L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory*, 38(2):617–643, March 1992.

[41] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. on PAMI*, 14(7):2207–2232, July 1992.

[42] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Royal Soc. London*, 207:187–217, 1980.

[43] Y. Meyer. *Ondelettes et Operateurs.* Hermann, New York, 1990.

[44] Y. Meyer. Un contre-exemple à la conjecture de Marr et à celle de S. Mallat. preprint, 1991.

[45] P. Moulin. Model selection criteria and the orthogonal series method for function estimation. In *Proc. IEEE Symp. on Info Theory*, page 252, Whistler, B.C., September 1995.

[46] P. Moulin. Signal estimation using adapted tree-structured bases and the MDL principle. In *Proc. IEEE Time-Frequency and Time-Scale Analysis*, pages 141–143, June 1996.

[47] H.-G. Müller and U. Stadtmüller. Estimation of heteroscedasticity in regression analysis. *Annals of Statistics*, 15(2):610–625, 1987.

[48] G. Nason. Choice of the threshold parameter in wavelet function estimation. In A. Antoniades and G. Oppenheim, editors, *Wavelets in Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 261–280. Springer-Verlag, 1995.

[49] B.K. Natarajan. Filtering random noise from deterministic signals via data compression. *IEEE Trans. on Signal Processing*, 43(11):2595–2605, November 1995.

[50] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific, 1989.

[51] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Trans. Comput.*, C-20:562–569, 1971.

[52] F. Ruggeri and B. Vidakovic. A Bayesian decision theoretic approach to wavelet thresholding. preprint, ftp://ftp.isds.duke.edu/pub/Users/brani/papers/Decision.ps, Duke University, Durham, NC, 1997.

[53] A. Said and W.A. Pearlman. A new fast and efficient image coder based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, pages 243–250, June 1996.

[54] N. Saito. Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion. In E. Foufoula-Georgiou and P. Kumar, editors, *Wavelets in Geophysics*, pages 299–324. Academic Press, 1994.

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

[56] E. Simoncelli and E. Adelson. Noise removal via Bayesian wavelet coring. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 379–382, Lausanne, Switzerland, September 1996.

[57] C.M. Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9(6):1135–1151, 1981.

[58] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley Cambridge Press, 1996.

[59] G. J. Sullivan. Efficient scalar quantization of exponential and Laplacian random variables. *IEEE Trans. on Information Theory*, 42(5):1365–1374, September 1996.

[60] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[61] B. Vidakovic. Nonlinear wavelet shrinkage with Bayes rules and Bayes factors. *Journal of the American Statistical Assoc.*, 93(441):173–179, 1998.

[62] G. Wahba. *Spline models for observational data.* SIAM, Philadelphia, 1990.

[63] Y. Wang. Function estimation via wavelets for data with long-range dependence. Technical report, Dept. of Statistics, University of Missouri-Columbia, 1994.

[64] P.H. Westerink, J. Biemond, and D.E. Boekee. An optimal bit allocation algorithm for sub-band coding. In *Proc. Int. Conf. on Acous., Speech and Signal Process.*, pages 1378–1381, Dallas, TX, April 1987.

[65] N. Weyrich and G.T. Warhola. De-noising using wavelets and cross-validation. In S.P. Singh, editor, *Approximation Theory, Wavelets and Applications*, volume 454 of *NATO ASI Series C*, pages 523–532. 1995.

[66] A. Witkin. Scale space filtering. In *Proc. Int. Joint Conf. Artificial Intell.*, 1983.

[67] J.W. Woods, editor. *Subband Image Coding.* Kluwer Academic Publishers, Boston, MA, 1991.

[68] Y. Yoo, A. Ortega, and B. Yu. Image subband coding using progressive classification and adaptive quantization. preprint, 1997.

Publications

[1] S.G. Chang, Z. Cvetković, and M. Vetterli, "Resolution enhancement of images using wavelet transform extrema extrapolation," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol.4, pp.2379-2382, Detroit, MI, May 1995.

[2] S.G. Chang, M.M. Goodwin, V.K. Goyal, and T. Kalker, *Solution Manual for "Wavelets and Subband Coding"* by Martin Vetterli and Jelena Kovačević. Prentice Hall, Englewood Cliffs, NJ, 1995, ISBN 0-13-461625-1.

[3] S.G. Chang, "Image interpolation using wavelet-based edge enhancement and texture analysis," M.S. Thesis, ERL Technical Memo M95/100, U.C. Berkeley, May 1995.

[4] S.G. Chang, M. Vetterli, and Z. Cvetković, "Image enhancement using wavelet modeling," in *Proc. IEEE Image and Multidimensional Signal Processing Workshop*, pp. 50-51, Belize City, Belize, March 1996.

[5] S.G. Chang and G. Yovanof, "A simple block-based lossless image compression scheme," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, November 1996.

[6] S.G. Chang, B. Yu, and M. Vetterli, "Bridging compression to wavelet thresholding as a denoising method," in *Proc. Conf. Information Sciences and Systems*, pp. 568-573, Baltimore, MD, March 1997.

[7] S.G. Chang and M. Vetterli, "Spatial adaptive wavelet thresholding for image denoising," in *Proc. IEEE Int. Conf. Image Processing*, Vol.2, pp. 374-377, Santa Barbara, CA, October 1997.

[8] S.G. Chang, B. Yu, and M. Vetterli, "Image denoising via lossy compression and wavelet thresholding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 604-607, Santa Barbara, CA, October 1997.

[9] S.G. Chang, B. Yu, and M. Vetterli, "Multiple copy image denoising via wavelet thresholding," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, October 1998.

[10] S.G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, October 1998.

[11] S.G. Chang, B. Yu, and M. Vetterli, "Lossy compressy and wavelet thresholding for image denoising," submitted to *IEEE Transactions on Image Processing*, 1998.

[12] S.G. Chang, B. Yu, and M. Vetterli, "Wavelet thresholding for multiple noisy image copies," submitted to *IEEE Transactions on Image Processing*, 1998.

[13] S.G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," submitted to *IEEE Transactions on Image Processing*, 1998.