UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Wavelet Transforms and PDE Techniques in Image Compression (Ph.D. Thesis)

Hao-Min Zhou

June 2000

CAM Report 00-21

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

UNIVERSITY OF CALIFORNIA
Los Angeles

Wavelet Transforms and PDE Techniques in Image Compression

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

by

Hao-Min Zhou

2000

To my wife and my parents and my brother and sister

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

I would like to express my sincere and deepest gratitude to Tony Chan, my advisor and mentor for his guidance, patience, and support. I am indebted to him for all his help in all aspects of my life during my graduate studies at UCLA. His energy, diligence and intelligence are inspiring. I would like to thank other members of my committee: Stanley Osher, Bjorn Engquist, and Abeer Alwan for their insights, support and comments in research. I would also like to thank Raymond H. Chan and Zhen-Huan Teng for introducing me to apply mathematics and their long time support and encouragement. I would especially like to further thank Stanley Osher for his encouragement, support in all aspects of my life.

I would like to thank other people for their support and discussions on various math topics. Thanks goes to Ming Gu, Bo Li, Danping Peng, Wing Lok Wan, Chiu-Kwong Wong, Jianhong Shen, Luminita Vese, Paul Burchard, Ron Fedkiw, Hong-Kai Zhao, Barry Merriman, Myungjoo Kang, Susan Chen, Richard Tsai, Frederic Gibou, and Duc Nguyen. Special thanks goes to Li-Tien Cheng for his help and discussion on various issues.

This work was partially supported by grants ONR-N00017-96-1-0277, ONR-N00014-97-1-0027, and NSF DMS-96-26755.

VITA

1969	Born, Shaanxi, P. R. China.
1991	B.S. (Mathematics), Peking University, Beijing China.
1994	M.S. (Computational Mathematics), Peking University, Beijing China.
1994–1996	Teaching Assistant, Department of Mathematics, The Chinese Univeristy of Hong Kong, Hong Kong.
1996	M.Phil. (Mathematics), The Chinese Univeristy of Hong Kong, Hong Kong.
1996–present	Teaching/Research Assistant, Mathematics Department, UCLA.
1997	M.S. (Mathematics), UCLA.

ABSTRACT OF THE DISSERTATION

Wavelet Transforms and PDE Techniques in Image Compression

by

Hao-Min Zhou

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2000

Professor Tony Chan, Chair

In this thesis, we propose using Partial Differential Equation (PDE) techniques in wavelet image processing to reduce edge artifacts generated by wavelet approximations. We have been exploring in two directions: the first is to apply the main idea of the well developed ENO schemes for numerical shock capturing to modify standard wavelet transforms so that a fewer number of large high frequency coefficients are generated. Therefore less distortion near edges is created when the coefficients are thresholded. Another direction is to use minimization techniques, in particular the minimization of total variation (TV), to select and modify the retained standard wavelet coefficients so that the reconstructed images have less oscillations near edges. Numerical experiments show that both approaches improve the reconstructed image quality in wavelet compression and in denoising.

CHAPTER 1

Introduction

Wavelet analysis is a well developed field in mathematics. It has many successful applications, especially in digital image processing. There are many ways to introduce wavelet transforms. One way is to view wavelets as basis functions of the L^2 space. This means that every function in L^2 space can be uniquely decomposed in terms of the wavelet basis. The coefficients of such a decomposition fall into two categories: the low frequency part which describes the local average properties of the function, and the high frequency part which represents the local differences. There are many papers and books on wavelets and their applications. For some mathematically oriented books, see [15], [18], [33] and [42].

It is well known that wavelet linear approximations (i.e. truncating the high frequency coefficients) can approximate smooth functions very efficiently: it can achieve arbitrary high accuracy through the selection of an appropriate wavelet basis, through concentrating the energy to low frequencies, and in addition, it has a multiresolution framework with associated fast transform algorithms. But standard wavelet linear approximation techniques cannot achieve similar results for general images which typically are piecewise continuous functions connected by large jumps (edges). Many problems arise near these jumps, caused primarily by the well-known edge artifacts (Gibbs' phenomenon [33]).

Several approaches have been proposed to overcome these problems. Within the wavelet pyramidal filtering framework, non-linear data-dependent approxima-

tions, which selectively retain certain high frequency coefficients, are often used, e.g. the commonly used hard and soft thresholding techniques of Donoho[19]. A more fundamental approach is to modify the wavelet transforms so as not to generate large wavelet coefficients near jumps. Claypoole, Davis, Sweldens and Baraniuk use an adaptive lifting scheme which lowers the order of approximation near jumps [16], thus reducing edge artifacts. All these approaches have their limitations, and some residual Gibbs' phenomenon still exists, especially in high ratio image compression and in denoising of highly noisy images. Another approach, due to Donoho, is to construct an orthonormal basis such as wedgelets [20], rigdelets [21] and curvelets [14] to represent the edges.

In this thesis, we propose applying some PDE techniques in wavelet image processing to overcome these problems. In fact, PDE techniques have been commonly used in image processing since the end of the 1980's, for example, Mumford-Shah's functional [35] in segmentation, Rudin-Osher-Fatemi's Total Variation in restoration [39], Alveraz-Morel's formalization in image analysis [2], Perona-Malik's diffusion model [38], Sapiro-Tannenbaum's affine scale space [40], Alveraz-Guichard-Lions-Morel's fundamental equations for image processing [1], Morel-Solimini's book on variational method for segmentation [34], and some more recent works such as Chan-Vese's active contour [9], Chambolle-Lions's total variation analysis [7], Bertalmio-Sapiro-Caselles-Ballester's image inpainting [3] and the papers collected in the IEEE special issue on PDE and geometry-driven diffusion image processing [25]. A crucial observation which makes these methods successful is viewing images as piecewise smooth functions connected by large jumps (edges) and realizing the similarity between images and piecewise smooth solutions of certain kinds of PDE's. Then well-developed PDE techniques can be used to handle the edges. Based on this observation, we apply several PDE techniques to wavelet image processing to reduce the edge oscillations. In particular, we have

been exploring in two directions: the first is to apply the main idea of Essentially Non-Oscillatory (ENO) schemes for numerical shock capturing to modify the standard wavelet transforms so that a fewer number of large high frequency coefficients are generated. Therefore less distortion is created near edges when the coefficients are thresholded. Another direction is to use minimization techniques to modify the retained standard wavelet coefficients so that the reconstructed images have fewer oscillations near the edges. We will briefly introduce these two approaches in the following paragraphs.

For the first direction, we have designed an adaptive ENO-wavelet transform for approximating piecewise continuous functions without oscillations near the jumps. Our approach is to apply the main idea from ENO schemes for numerical shock capturing to standard wavelet transforms. The crucial point is that the wavelet coefficients are computed without differencing function values across jumps. However, we accomplish this in a different way than in standard ENO-schemes. Whereas in standard ENO schemes, the stencils are adaptively chosen, in the ENO-wavelet transforms, we adaptively change the function and use the same uniform stencils. The ENO-wavelet transform retains the essential properties and advantages of standard wavelet transforms such as concentrating the energy to the low frequencies, obtaining an arbitrary high order of accuracy uniformly and having a multiresolution framework and fast algorithms, all without any edge artifacts. We have obtained a rigorous approximation error bound which shows that the error in the ENO-wavelet approximation depends only on the size of the derivative of the function away from the discontinuities. Numerical results have illustrated these error estimates. The extra cost (in floating point operations) over standard wavelet transforms is proportional to the number of jumps. Numerical examples show that the ENO-wavelet compressed images retain more detail and sharper edges than that of standard wavelet compressed images.

To reduce edge oscillations in wavelet based image compression, we have been working on another approach which applies variational image restoration techniques, particularly total-variation (TV), in wavelet compression and denoising to select and modify the retained wavelet coefficients in a such way that the reconstructed image has less edge artifacts.

In this direction, our first attempt is advocating the total variation denoising method, followed by the standard wavelet compression for high noise level images to obtain high ratio compression [12]. Numerical experiments demonstrate that TV-denoising can bring more wavelet coefficients closer to zero thereby making the compression more efficient while the salient features (edges) of the images can still be retained.

It has been shown through many simulations that the TV model can effectively suppress the noise while retaining the edges in images. This is because it allows the existence of discontinuities. In [6], Chambolle *et* al propose to minimize some Besov norms which are approximations to the TV norm for denoising and compression in wavelet spaces. In fact, edge oscillations caused by standard wavelet thresholding significantly increase the TV norm of the reconstructed images. All this motivates us to embed the TV model into the wavelet compression algorithms, i.e., minimize the TV norm to find the best m coefficients, where m is a given maximum allowed number of nonzero coefficients, to form the least edge artifact approximation. In general, minimizers of such variational problems can be found by solving their associated Euler-Lagrangian equations, which are PDE's. Many methods have been advocated to solve the PDE's, such as the time marching method [39], the fixed-point iterative scheme [45], the primal-dual method [8], and the level-set approach [37]. We also study some numerical issues related to this proposed TV norm model for wavelet thresholding. We will show

numerical experiments in the applications of compression and denoising.

The thesis is arranged as follows: In chapter 2, we study the first topic, which is applying the ENO idea so as not to generate large high frequency coefficients. In chapter 3, we discuss the TV regularized wavelet compression model for wavelet thresholding techniques.

CHAPTER 2

Adaptive ENO-wavelet Transforms and Its Applications

2.1 Introduction

In this chapter, we develop new wavelet algorithms to approximate piecewise continuous functions, for instance, piecewise smooth functions connected by large jumps. It is well known that wavelet linear approximation (i.e. truncating the high frequencies) can approximate smooth functions very efficiently: it can achieve arbitrary high accuracy by selecting appropriate wavelet basis, it can concentrate the large wavelet coefficients in the low frequencies, and it has a multiresolution framework and associated fast transform algorithms. All these properties have led to their wide applications, such as in partial differential equations (PDE's) and image processing.

Standard wavelet linear approximation techniques cannot achieve similar results for functions which are not smooth, for example piecewise continuous functions with large jumps in function value or in its derivatives. Several problems arise near jumps, primarily caused by the well-known Gibbs' phenomenon. The jumps generate large high frequency wavelet coefficients and thus linear approximation cannot get the same high accuracy near the points of discontinuity as in the smooth region. In fact, the jump points generate oscillations which cannot

be removed by mesh refinement.

To overcome these problems within the standard wavelet transform framework, non-linear data-dependent approximations, which selectively retain certain high frequency coefficients, are often used, e.g. hard and soft thresholding techniques, see [19], [24], [23], [33] and corresponding references listed there. The main idea of these thresholding approximations is to truncate the wavelet coefficients by their magnitudes, not frequencies. For instance, hard thresholding sets all wavelet coefficients whose magnitudes are less than a given tolerance to zero and retains the other coefficients unchanged. However, these techniques often require more complicated data structure to record the location of the retained wavelet coefficients and still cannot remove the effects of Gibbs' phenomenon completely.

A more fundamental approach is to modify the wavelet transform to not generate large wavelet coefficients near jumps. A few papers in the literature have discussed this approach. Claypoole, Davis, Sweldens and Baraniuk [16] proposed an adaptive lifting scheme which lowers the order of approximation near jumps, thus minimizing the Gibbs' effect. Of course, this scheme suffers from reduced approximation accuracy near jumps, and some residual Gibbs' phenomenon still exists. Another way due to Donono is to contruct orthonormal basis such as wedgelets [20] and rigdelets [13], [21] to represent the discontinuities.

In this chapter, we develop a new wavelet algorithm by borrowing the well developed Essentially Non-Oscillatory (ENO) technique for shock capturing in computational fluid dynamics (e.g. see [29] and [41]) to modify the standard wavelet transform near discontinuities in order to overcome the above mentioned difficulties. ENO schemes are systematic ways of adaptively defining piecewise polynomial approximations of the given functions according to their smoothness.

There are two crucial points in designing ENO schemes. The first is to use one-sided information near jumps, and never differencing across the discontinuities. The second is to adaptively form the divided difference table and select the smoothest *stencil* (the support of the basis) for every grid point. ENO schemes lead to uniform high accuracy approximations for each smooth piece of the function. We will only use the first point in our design of the ENO-wavelet transforms.

Combining the ENO idea with the multiresolution data representation is a natural way to avoid oscillations in the approximations. In fact, it has been explored by Harten in his general framework of multiresolution [26], [27] and [28], which is similar to the lifting scheme of Sweldens [44]. His approach is to directly blend the two ideas, and to fully implement the ENO schemes at every point. However, his method cannot be directly applied to the more interesting and generally used pyramidal filtering algorithms which the standard wavelet transforms are implemented in because in this context we have to work with a fixed size filter and the adaptive divided difference algorithm does not fit into this framework.

Our goal is to design a more direct functional replacement of the standard wavelet transforms such that there are no oscillations at the discontinuities in the approximations. We want to stick with the pyramidal filtering framework because they are easy to use, and have been successfully applied in many applications. Compared to Harten's multiresolution approach, which is more flexible and easier to adaptively implement the ENO idea, the standard wavelet transforms are more regular and rigid in algorithmic structure, therefore directly applying the ENO idea would leads to a more drastic perturbation of the underlying pyramidal filtering algorithms. This is the challenge we face.

The way we accomplish this is to not change the wavelet transforms or the

filter coefficients, which most data dependent multiresolution algorithms do, but instead locally change the function near the discontinuities in such a way that the standard filters are only applied to smooth data. By recording how the changes are make, the original discontinuous function can be exactly recovered by using the original inverse filters. Indeed, by applying the idea of using one-sided information near the discontinuities, we directly extend the functions from both sides of the discontinuities, thus we can apply the standard wavelet transforms on these extended values such that there are no large coefficients generated in the high frequencies and the low frequency approximations are essentially non-oscillatory, and therefore Gibbs' phenomenon can be completely avoided.

In addition, in this modified wavelet transform, the low frequency part preserves the piecewise smoothness of the original function. In particular, the jumps in the low frequency part is not spread widely as in the standard transform. Therefore, the same ENO idea can be recursively used for the coarser levels of the low pass coefficients. By doing so, the multiresolution framework can be kept too.

We show that the resulting wavelet transform retains all the desirable properties of the standard transform: it can have uniform arbitrarily high order of approximation (with a rigorous uniform order error bound), it concentrates the large wavelet coefficients to the low frequencies, it preserves the multiresolution framework and fast transform algorithms, and it is easy to implement. Furthermore, since we do not fully adopt the ENO schemes, in particular, we do not build the divided difference table and compare the smoothness of all possible stencils at every point, the extra cost (in floating point operations) required by the modified ENO-wavelet transforms is insignificant. In fact, it is of the order $O(dl)$ where d is the number of discontinuities and l the stencil length. Compared to the cost of

the standard wavelet transform, which is of the order $O(nl)$ where n is the size of the data, the ratio of the extra cost over that of the standard transform is of the order $O(\frac{d}{n})$ which is independent of l and negligible when n is large.

There are many possible applications of the ENO-wavelet transforms, for example, in image compression, image restoration, data representation and initial-boundary value problems for PDE's. We will show some examples of applications to image compression.

The arrangement of the chapter is as follows. In section 2.2 , we review the standard continuous and discrete wavelet transforms. In section 2.3, we give a general algorithm to implement the ENO-wavelet transform discretely. In section 2.4, we prove an error bound for the ENO-wavelet approximation which shows that the error in the ENO-wavelet approximation depends only on the size of the derivative of the function *a*way from the discontinuities. Finally, in section 2.5, we give some numerical examples to illustrate the main advantage of the ENO-wavelet transforms, including some examples in image compression.

2.2 Wavelet Transforms

Before we introduce the adaptive ENO-wavelet transforms, we briefly review the standard wavelet transforms, e.g. see [15], [18], [33] and [42]. We will go over both continuous and discrete wavelet transforms, because we will present our ENO-wavelet transforms in the discrete form and prove the approximation error bound by using the continuous form.

2.2.1 Continuous Wavelet Transforms

First, we review the standard continuous wavelet transforms. To simplify the notation, we assume zeros have been padded to the data at the boundaries.

The standard wavelet transforms are based on translation and dilation. Suppose $\phi(x)$ and $\psi(x)$ are the scaling function and the corresponding wavelet respectively with finite support $[0, l]$ where l is a positive integer. It's well known that $\phi(x)$ satisfies the basic dilation equation:

$$\phi(x) = \sqrt{2}\sum_{s=0}^{l} c_s \phi(2x - s); \tag{2.1}$$

and $\psi(x)$ satisfies the corresponding wavelet equation:

$$\psi(x) = \sqrt{2}\sum_{s=0}^{l} h_s \phi(2x - s); \tag{2.2}$$

where the c_s's and h_s's are constants called low pass and high pass filter coefficients respectively.

We assume that $\psi(x)$ has p vanishing moments:

$$\int \psi(x) x^j dx = 0, \qquad for \quad j = 0, 1, \cdots, p-1. \tag{2.3}$$

We will use the following standard notations:

$$\phi_{j,k}(x) = 2^{\frac{j}{2}} \phi(2^j x - k), \tag{2.4}$$

and

$$\psi_{j,k}(x) = 2^{\frac{j}{2}} \phi(2^j x - k). \tag{2.5}$$

Consider the subspace V_j of L^2 defined by:

$$V_j = Span\{\phi_{j,k}(x), k \in Z\},$$

and the subspace W_j of L^2 defined by:

$$W_j = Span\{\psi_{j,k}(x), k \in Z\}.$$

The subspaces V_j's, $-\infty < j < \infty$, form a multiresolution of L^2 with the subspace W_j being the difference between V_j and V_{j+1}. In fact, the L^2 space has an orthonormal decomposition as:

$$L^2 = V_J \oplus \sum_{j=J}^{\infty} W_j. \tag{2.6}$$

The projection of a L^2 function $f(x)$ onto the subspace V_j is defined by:

$$f_j(x) = \sum_k \alpha_{j,k}\phi_{j,k}(x), \tag{2.7}$$

where

$$\alpha_{j,k} = \int f(x)\phi_{j,k}(x)dx. \tag{2.8}$$

Similarly, we can project $f(x)$ onto W_j by:

$$w_j(x) = \sum_k \beta_{j,k}\psi_{j,k}(x), \tag{2.9}$$

where

$$\beta_{j,k} = \int f(x)\psi_{j,k}(x)dx. \tag{2.10}$$

Therefore, the function $f(x)$ can be decomposed by:

$$f(x) = f_j(x) + \sum_{i=j}^{\infty} w_i(x). \tag{2.11}$$

The projection $f_j(x)$ is called the linear approximation of the function $f(x)$ in the subspace V_j.

From (2.4) and (2.5), the projection coefficients $\alpha_{j,k}$ and $\beta_{j,k}$ of $f(x)$ in the subspaces V_j and W_j can be easily computed by the so called fast wavelet transform:

$$\alpha_{j,i} = \sum_{s=0}^{l} c_s \alpha_{j+1,2i+s}; \tag{2.12}$$

and

$$\beta_{j,i} = \sum_{s=0}^{l} h_s \alpha_{j+1,2i+s}. \tag{2.13}$$

2.2.2 Discrete Wavelet Transforms

In practice, discrete wavelet transforms are often used. Unlike in the continuous case where the wavelet transform is applied to the L^2 function $f(x)$, in the discrete case, we start by considering a set of discrete numbers which are the low frequency coefficients of the L^2 function $f(x)$ at a fine level subspace V_{j+1}. In many applications, this set of numbers are sample values of the function $f(x)$ on a fine grid (although in [42], this is called a "wavelet crime").

Let us define the following matrices:

$$L = \begin{pmatrix} c_0 & c_1 & \cdots & c_{l-1} & & & & \\ & & c_0 & c_1 & \cdots & c_{l-1} & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & c_0 & c_1 & \cdots & c_{l-1} \end{pmatrix},$$

and

$$H = \begin{pmatrix} h_0 & h_1 & \cdots & h_{l-1} & & & & \\ & & h_0 & h_1 & \cdots & h_{l-1} & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & h_0 & h_1 & \cdots & h_{l-1} \end{pmatrix}.$$

We also denote $\vec{\alpha}_j = (\cdots, \alpha_{j,k}, \alpha_{j,k+2}, \cdots)^T$ and $\vec{\beta}_j = (\cdots, \beta_{j,k}, \beta_{j,k+2}, \cdots)^T$.

The fast wavelet transform equations (2.12) and (2.13) in discrete form can be written as:

$$\vec{\alpha}_j = L\vec{\alpha}_{j+1}; \tag{2.14}$$

and

$$\vec{\beta}_j = H\vec{\alpha}_{j+1}. \tag{2.15}$$

It is well known that the wavelet transform matrices L and H are orthogonal:

$$L^*L + H^*H = I. \tag{2.16}$$

It follows that the inverse wavelet transform is simply:

$$\vec{\alpha}_{j+1} = L^*\vec{\alpha}_j + H^*\vec{\beta}_j. \tag{2.17}$$

In this discrete case, the linear approximation refers to reconstructing $\vec{\alpha}_{j+1}$ by discarding the last term in (2.17).

The standard linear wavelet approximation achieves arbitrary high accuracy away from discontinuities, but it oscillates near the jumps. The reason for the oscillations is that some stencils cross jumps and cause the corresponding high frequency coefficients to becoming large and therefore, more information is lost when the high frequency coefficients are discarded.

In Fig 2.1, we display a piecewise continuous function (left) and its DB-4 wavelet coefficients (right). From the right picture, we see that most of the high frequency coefficients are zero, except for a few large coefficients which these coefficients are computed near jumps. Fig 2.2 displays the linear approximation (solid line) compared to the initial function (dotted line). The right picture is the zoom-in to show the approximation behavior near a jump. In this figure, we clearly see oscillations (people call them Gibbs' phenomenon) near discontinuities.

Since the oscillations are generated by discarding large high frequency coefficients which are computed on the stencils crossing discontinuities, to get rid of the oscillations, we want to avoid stencils crossing discontinuities. This motivates us to apply the ENO idea to avoid stencils crossing jumps.

Before we introduce the ENO-wavelet transforms, we give the following definition which we will use in the later sections. Given a function $f(x)$ which has

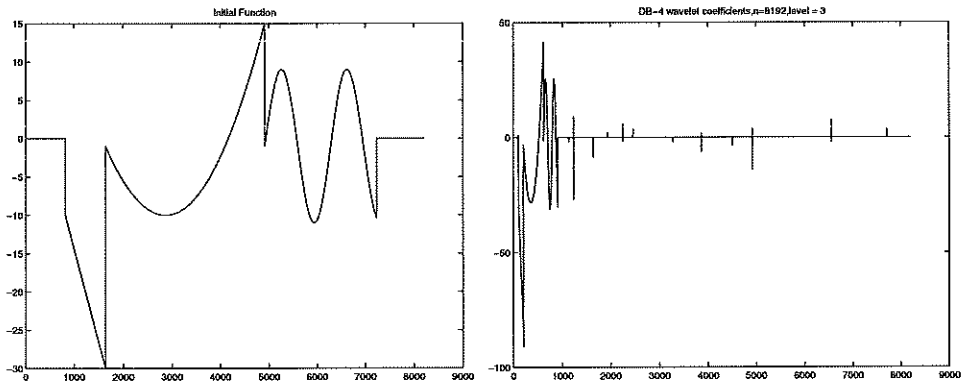

Figure 2.1: The initial function (left) and its DB4 coefficients (right). Most of the high frequency coefficients (right part) are zero except for a few large coefficients computed near the jumps.

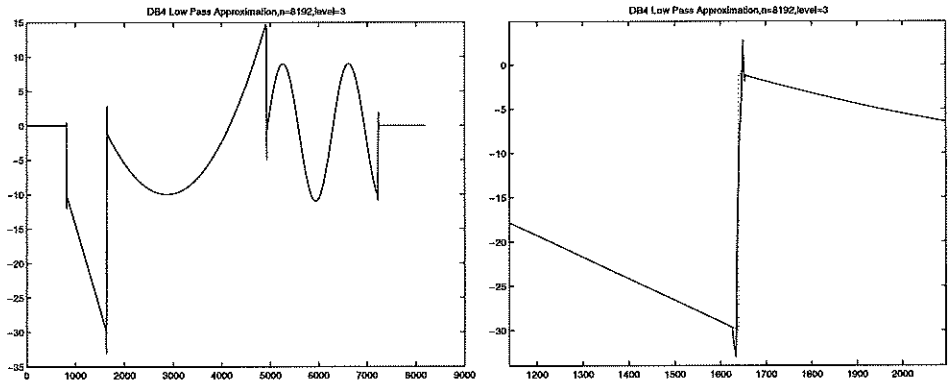

Figure 2.2: The approximation function (left) and its zoom in (right), Oscillations are generated near the discontinuities in the linear approximation.

discontinuous set D, i.e.

$$D = \{x_i : f(x) \quad is \quad discontinuous \quad at \quad x_i\}.$$

Denote t as the closest distance between any two discontinuous points, i.e.

$$t = \inf\{|x_i - x_j| : x_i, x_j \in D\}.$$

Definition 1 *For a given wavelet filter with stencil length l, we say a projection of $f(x)$ in space V_j with spatial step $\Delta x = 2^{-j}$ satisfies the* ***D****iscontinuity Separation Property (DSP) if $(l+2)\Delta x < t$.*

A projection satisfying the DSP implies that any one discontinuity is located at least one stencil and two data points away from other discontinuities. In other words, there are no two consecutive stencils containing two discontinuities. We assume that all projections we consider in this paper satisfy the DSP. Since our ENO-wavelet transform is essentially using ENO techniques to modify the standard wavelet transform near discontinuities, this property will avoid the modifications near one discontinuity interacting with the modifications near other discontinuities.

Remark: For any piecewise discontinuous function, a projection will satisfy this DSP if j is sufficiently large, i.e. if the discretization is fine enough. On the other hand, at the place where the DSP is invalid, the approximations produced by the ENO-wavelet transforms are comparable to that by the standard wavelet transforms. We will show numerical examples in section 2.5 illustrating this point.

2.3 ENO-wavelet Transforms

In this section, we design the ENO-wavelet transforms. In addition to the standard wavelet transforms, our ENO-wavelet transforms are composed of two

phases: locating the jumps and forming the approximations at the discontinuities. Firstly, assuming that the location of the jumps are known, we give the ENO-wavelet approximations at the discontinuities by using one-sided information to avoid oscillations. Then, we give the methods to detect the exact location of the discontinuities.

2.3.1 ENO-wavelet Approximation at Discontinuities

In this subsection, we assume that the exact location of the discontinuities are known. We want to modify the standard wavelet transforms near the jumps such that oscillations can be avoided in the approximation. From ENO schemes, we borrow the idea of using one-sided information to form the approximation and avoid applying the wavelet filters crossing the discontinuities. Since we assume the DSP is satisfied by the given projection of the function $f(x)$, we can just consider the local modification near one jump. The main tool which we use to modify the standard wavelet transforms at the discontinuities is function extrapolation in the function spaces or in the wavelet spaces.

Direct Function Extrapolation

The first way is to extend the function directly at the discontinuity by extrapolation from both sides. Then we can apply the standard wavelet transforms on the extended functions and avoid computing wavelet coefficients using information from both sides.

To maintain the same approximation accuracy near the discontinuity as that for away from the discontinuity, the extrapolation has to be $(p-1)$-th order accurate if the wavelet functions have p vanishing moments. For instance, we

use constant extrapolation for Haar wavelet, $(p-1)$-th order extrapolation for Daubechies-$2p$ orthogonal wavelets which have p vanishing moments.

We use the diagram in Fig 2.3 to show how to extend the function and compute the ENO-wavelet coefficients.

As shown in Fig 2.3, the discontinuity is located between $\{x(2i+l-2), x(2i+l-1)\}$. We extend the function from both sides of the discontinuity using $(p-1)$-th order extrapolation, i.e. we use the information from the left side of the jump to extrapolate the function over $\hat{x}(2i+l-1), \cdots, \hat{x}(2i+2l-2)$; use the information from the right side to extrapolate the function over $\bar{x}(2i), \cdots, \bar{x}(2i+l-2)$. And then for $i \leq m \leq i+k-2$, we can compute the wavelet coefficients $\hat{\alpha}_{j,m}$ and $\hat{\beta}_{j,m}$ from the left side, and compute $\bar{\alpha}_{j,m}$ and $\bar{\beta}_{j,m}$ from the right side by using the standard wavelet transforms respectively.

In general, we have the low frequency wavelet coefficients on the finer levels instead of knowing the function values themselves near the discontinuities. We extrapolate these finer level coefficients from both sides of the discontinuities to obtain the values of $\hat{\alpha}_{j+1,m}$ and $\bar{\alpha}_{j+1,m}$, and use the fast wavelet transforms (2.12) and (2.13) to compute the coarser level coefficients. For instance, we can compute $\hat{\alpha}_{j,i}$ and $\hat{\beta}_{j,i}$ by:

$$\begin{aligned}
\begin{pmatrix} \hat{\alpha}_{j,i} \\ \hat{\beta}_{j,i} \end{pmatrix} &= \begin{pmatrix} \sum_{s=0}^{l-2} c_s \alpha_{j+1,2i+s} + c_{l-1}\hat{\alpha}_{j+1,2i+l-1} + c_l \hat{\alpha}_{j+1,2i+l} \\ \sum_{s=0}^{l-1} h_s \alpha_{j+1,2i+s} + h_{l-1}\hat{\alpha}_{j+1,2i+l-1} + h_l \hat{\alpha}_{j+1,2i+l} \end{pmatrix} \\
&\equiv \begin{pmatrix} \delta_{j,i} \\ \gamma_{j,i} \end{pmatrix} + A \begin{pmatrix} \hat{\alpha}_{j+1,2i+l-1} \\ \hat{\alpha}_{j+1,2i+l} \end{pmatrix}, \qquad (2.18)
\end{aligned}$$

where $\delta_{j,i}$ and $\gamma_{j,i}$ are $\sum_{s=0}^{l-2} c_s \alpha_{j+1,2i+s}$ and $\sum_{s=0}^{l-1} h_s \alpha_{j+1,2i+s}$ respectively and depend only on the unextrapolated values of $\alpha_{j+1,m}$, and A a 2×2 matrix defined

by the filter coefficients as:

$$A = \begin{pmatrix} c_{l-1} & c_l \\ h_{l-1} & h_l \end{pmatrix}.$$

By symmetry, we can compute $\bar{\alpha}_{j,m}$'s and $\bar{\beta}_{j,m}$'s from the right side in a similar way.

There are many methods to extrapolate the extended values. For example, a straightforward way is to use p-point polynomial extrapolation. Least square extrapolation can be used too [46].

There is a storage problem for this direct function extrapolation. Indeed, it doubles the number of the wavelet coefficients near every discontinuity. To retain the perfect invertible property, we need to store the ENO-wavelet coefficients $\hat{\alpha}_{j,m}$ and $\hat{\beta}_{j,m}$ from the left side, also $\bar{\alpha}_{j,m}$ and $\bar{\beta}_{j,m}$ from the right side. Thus, the output sequences are no longer the same size as the input sequences. In many applications, such as image compression, this extra storage requirement definitely needs to be avoided.

Remark: In the least square extrapolation case, it is possible to reduce the demands of the extra storage because not all the wavelet coefficients $\hat{\alpha}_{j,m}$, $\hat{\beta}_{j,m}$, $\bar{\alpha}_{j,m}$ and $\bar{\beta}_{j,m}$ are linearly independent [46]. However, it requires complicated extra computation.

Our Approach: Coarse Level Extrapolation

To avoid computing the wavelet coefficients using the information from both sides of the discontinuities, to maintain the same high order accuracy near the discontinuities as away from the discontinuities, and also to keep the size of the output sequences the same as that of the input sequences without significant extra

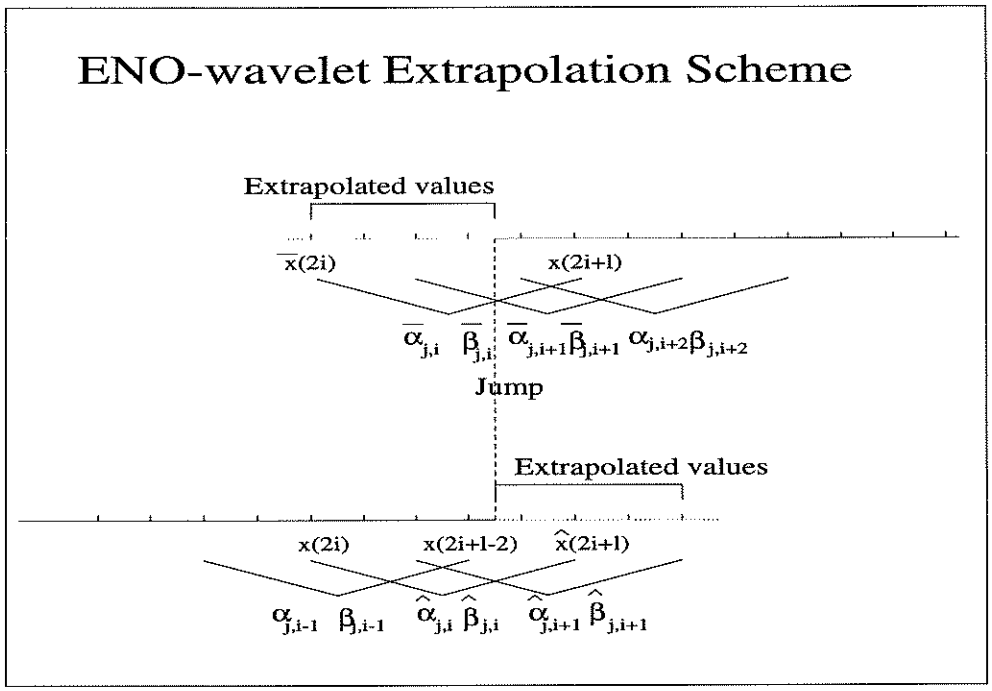

Figure 2.3: Coarse Level Extrapolation Illustration. From the left side of the discontinuity, we extrapolate the low frequency coefficients $\hat{\alpha}_{j,m}$ to determine corresponding high frequency coefficients $\hat{\beta}_{j,m}$ and store them. From the right side of the discontinuity, we extend the high frequency coefficients $\bar{\beta}_{j,m}$ to determine and store the low frequency coefficients $\bar{\alpha}_{j,m}$.

computation, we introduce the coarse level extrapolation schemes. The idea is to extrapolate the coarser level wavelet coefficients near the discontinuities instead of the function values or the finer level wavelet coefficients.

We still use Fig 2.3 to illustrate these schemes. We consider the left side of the jump first.

In the direct function extrapolation case, the computation process is to directly extrapolate the finer level wavelet coefficients $\hat{\alpha}_{j+1,m}$, $(2i+l-1) \leq m \leq (2i+2l-2)$, then compute the extended coarser level wavelet coefficients $\hat{\alpha}_{j,m}$

and $\hat{\beta}_{j,m}$, $i \leq m \leq (i+k-2)$ using the standard filters. We reverse the order of this process in our coarse level extrapolation. More precisely, we extrapolate the coarser level low frequency coefficients $\hat{\alpha}_{j,m}$ using the known low frequency coefficients from the left, and extend the coarser level high frequency coefficients $\hat{\beta}_{j,m}$ to zero, then determine the extended finer level wavelet coefficients. For example, in the direct function extrapolation, we extrapolate finer level values $\hat{\alpha}_{j+1,m}$ and then compute the coarser level coefficients $\hat{\alpha}_{j,i}$ and $\hat{\beta}_{j,i}$ by (2.18). On the contrary, we can first extend the coarser level coefficients $\hat{\alpha}_{j,i}$ and $\hat{\beta}_{j,i}$ and then determine the finer level values. Indeed, if the matrix A is nonsingular, we can uniquely determine the finer level values by solving (2.18). In this case, we can prescribe both the coarser level coefficients simultaneously. However, in Daubechies' orthogonal wavelet transforms, the matrix A is singular, because:

$$\frac{h_{l-1}}{c_{l-1}} = \frac{h_l}{c_l}. \tag{2.19}$$

Thus, in order to have a solution of (2.18), we must extend the coarser level coefficients $\hat{\alpha}_{j,i}$ and $\hat{\beta}_{j,i}$ in a way such that they satisfy:

$$\begin{pmatrix} \hat{\alpha}_{j,i} \\ \hat{\beta}_{j,i} \end{pmatrix} - \begin{pmatrix} \delta_{j,i} \\ \gamma_{j,i} \end{pmatrix} \in R(A),$$

where $R(A)$ is the range space of A. This requirement implies that:

$$\begin{pmatrix} -1 & \frac{c_l}{h_l} \end{pmatrix} \left[\begin{pmatrix} \hat{\alpha}_{j,i} \\ \hat{\beta}_{j,i} \end{pmatrix} - \begin{pmatrix} \delta_{j,i} \\ \gamma_{j,i} \end{pmatrix} \right] = 0,$$

which we can also rewrite as:

$$\hat{\beta}_{j,i} = \gamma_{j,i} + \frac{h_l}{c_l}(\hat{\alpha}_{j,i} - \delta_{j,i}), \tag{2.20}$$

or

$$\hat{\alpha}_{j,i} = \delta_{j,i} + \frac{c_l}{h_l}(\hat{\beta}_{j,i} - \gamma_{j,i}). \tag{2.21}$$

Therefore, we cannot prescribe both $\hat{\alpha}_{j,m}$ and $\hat{\beta}_{j,m}$ simultaneously. Thus we have two choices:

(1) We can extrapolate the low frequency coefficients $\hat{\alpha}_{j,m}$ first, then determine the corresponding high frequency coefficients $\hat{\beta}_{j,m}$ by (2.20);

(2) Or we can extend $\hat{\beta}_{j,m}$ to zero first, then determine the corresponding $\hat{\alpha}_{j,m}$ by (2.21).

Remark: We notice that in both cases (2.20) and (2.21), the coefficients are computed by applying the standard filters to the extended data which is smooth. This implies that there are no large coefficients generated by them, which also means that the extension is stable, although we do not explicitly use the extended finer level values in the computation formulas.

By symmetry, we have two analogous choices for the right side of the jump.

Using this coarse level extrapolation technique, we can easily solve the storage problem which we have in the direct function extrapolation. In fact, we just need to store the high frequency coefficients $\hat{\beta}_{j,m}$ for choice (1) and the low frequency coefficients $\hat{\alpha}_{j,m}$ for choice (2). In our implementation, we use choice (1) for the left side of the jumps and choice (2) for the right side of the jumps, therefore we store $\hat{\beta}_{j,m}$ and $\bar{\alpha}_{j,m}$ for every m. This satisfies the standard wavelet storage scheme, i.e. storing one α and one β for every stencil.

Remark: We select choice (1) from the left side of the jumps and choice (2) from the right side because we want to keep half of the output sequence to be α's and half β's. It is possible to select choice (1) or choice (2) for both sides of the jumps, but that will not give equal number of α's and β's in the output sequence.

Since we know the way we extend the data at the discontinuities, we can easily extrapolate the low frequency coefficients $\hat{\alpha}_{j,m}$ from the left sides of the discontinuities. Using them together with the stored high frequency coefficients $\hat{\beta}_{j,m}$, we can exactly recover data at the left sides by applying the standard inverse filters. Similarly, the data at right sides of the discontinuities can also be exactly restored. Moreover, since all the data are smoothly extended, the reconstruction process is also stable.

Remark: In the ENO-wavelet transforms, to retain the perfect invertibility property, we have to store all adaptive information, i.e. the locations of the discontinuities. In our implementation, we just use one extra bit for each stencil near the discontinuities to indicate it contains a discontinuity.

For each stencil crossing a jump, an extra cost (in floating point operation) is required in the extrapolation low frequency coefficients, which is of the order $O(1)$ per stencil, and in the computation of the corresponding high and low frequency coefficients by (2.20) and (2.21), which is of the order $O(l)$ per stencil. Overall, the extra cost over the standard wavelet transform is of the order $O(dl)$ where d is the number of discontinuities. Compared to the cost of the standard wavelet transform, which is of the order $O(nl)$ where n is the size of data, the ratio of the extra cost over that of the standard transform is $O(\frac{d}{n})$, which is independent of l and negligible when n is big.

2.3.2 Locating the Discontinuities

In the previous subsection, we showed how to modify the standard wavelet transforms at the discontinuities to avoid oscillations if we know the exact location of the jumps. In this subsection, we introduce the methods to detect the exact location of the discontinuities for piecewise smooth functions with and without

noise. First we give a method for smooth data.

Piecewise Smooth Functions

Our purpose is to avoid wavelet stencils crossing discontinuities. Theoretically, a discontinuity can be characterized by comparing the left and right limit of $f^{(i)}(x)$ at the given point x, i.e. we call a point x a discontinuity if for some i, we have:

$$f^{(i)}(x-) \neq f^{(i)}(x+).$$

It is well known that the high pass filters in wavelet transforms measure the smoothness of functions: they produce smaller values at smoother regions, and larger values at rougher regions. In fact, at smooth regions, the magnitudes of high frequency coefficients $|\beta_{j,i}|$ have the order of $|f^{(p)}(x)|O(\Delta x^p)$. On the other hand, if a stencil contains a discontinuity, no matter a discontinuity in function value or in its derivatives, the magnitude of the corresponding high frequency coefficient $|\beta_{j,i}|$ is at most of the order of $|f^{(p-1)}(x)|O(\Delta x^{(p-1)})$. Therefore, instead of fully adopting the ENO comparison idea which compares the magnitudes of divided differences on all possible stencils, we can use the magnitudes of the high frequency coefficients as our criterion to identify the discontinuities.

The obvious way, also the cheapest way, to identify the discontinuities is to compare the magnitudes of the high frequency coefficients on the current standard stencils $|\beta_{j,i}|$ with that on the previous standard stencils $|\beta_{j,i-1}|$. As we already know that for the smooth functions, we have $|\beta_{j,i}| = |f^{(p)}(x)|O(\Delta x^p)$, this implies that at smooth regions, we have:

$$|\beta_{j,i}| = (1 + O(\Delta x))|\beta_{j,i-1}|. \tag{2.22}$$

In contrast, the magnitudes of high frequency coefficients $|\beta_{j,i}|$ based on the stencils containing the discontinuities are at least one order lower than that at

the smooth regions. Thus, we can design a method to detect the discontinuities as follows: For each standard stencil, suppose we know that the previous standard stencil does not contain any discontinuities, if we have $|\beta_{j,i}| \leq a|\beta_{j,i-1}|$, where $a > 1$ is a given constant, then we treat the current stencil as a smooth stencil. Otherwise, we conclude that there are discontinuities contained in it.

To completely avoid oscillations, we also need to know the exact locations of the discontinuities so that we can avoid computing the wavelet coefficients crossing them. In fact, the above comparison method based on the magnitudes of high frequency coefficients can also help us to locate the exact positions of the discontinuities. We will use the diagram in Fig 2.4 to explain how to find the exact jump positions.

Assume we consider the wavelet filters with length $(l+1)$ where $l = 2k-1$. We compare the magnitude of the high frequency coefficient $|\beta_{j,i}|$ on the current stencil, which starts at $x(2i)$ with $|\beta_{j,i-1}|$ on the previous stencil. If we have $|\beta_{j,i}| > a|\beta_{j,i-1}|$, we identify the discontinuity lying in the current stencil. Since there are no discontinuities in the previous stencils, we know that this discontinuity must be located between $\{x(2i+l-2), x(2i+l)\}$, where it only has two possible positions: between $\{x(2i+l-2), x(2i+l-1)\}$ or between $\{x(2i+l-1), x(2i+l)\}$. In fact, we can determine the exact position of the jump by continue comparing the subsequent values of $\beta_{j,m}$. As shown in Fig 2.4, we must have at least $(k-1)$ consecutive "large" $\beta_{j,m}$, $i \leq m \leq (i+k-2)$, because the subsequent $(k-1)$ stencils also include the discontinuity. We compute $\beta_{j,i+k-1}$ and $\beta_{j,i+k}$ on the corresponding standard stencils, if we have $|\beta_{j,i+k-1}| > a|\beta_{j,i+k}|$, then we have k consecutive stencils containing the discontinuity, which implies that the discontinuity is located between $\{x(2i+l-1), x(2i+2l-1)\}$ (see Fig 2.4 (a)). If we have exactly $(k-1)$ consecutive standard stencils containing the discontinuity,

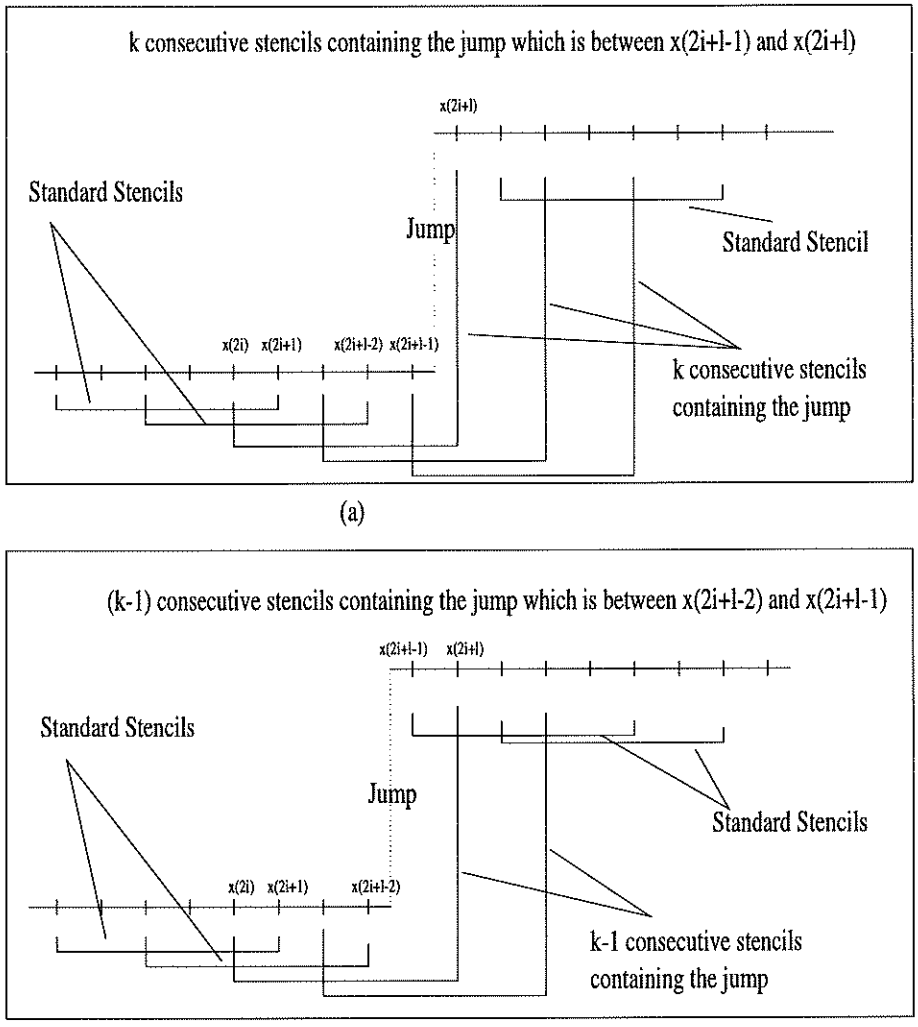

Figure 2.4: Locating the exact position of the jump by counting the number of consecutive stencils containing the jump. (a) If k stencils contain the jump, then the jump position is between $x(2i+l-1)$ and $x(2i+l)$. (b) If $(k-1)$ consecutive stencils contain the jump, the jump is located between $x(2i+l-2)$ and $x(2i+l-1)$.

which implies that the jump must be located between $\{x(2i+l-2), x(2i+l-1)\}$ (see Fig 2.4(b)). We summarize the above arguments in the following lemma.

Lemma 1 *Consider the wavelet filters with length $l = 2k-1$. For a given index i, assume we have $|\beta_{j,i-1}| \leq a|\beta_{j,i-2}|$ but $|\beta_{j,i}| > a|\beta_{j,i-1}|$. Then*

(1) if $|\beta_{j,i+k-1}| > a|\beta_{j,i+k}|$, which means there are k consecutive standard stencils containing the jump, then the discontinuity is located between $\{x(2i+l-1),\ x(2i+l)\}$.

(2) else we have $|\beta_{j,i+k-1}| \leq a|\beta_{j,i+k}|$, which implies that there are $(k-1)$ consecutive standard stencils containing the jump, then the discontinuity is located between $\{x(2i+l-2), x(2i+l-1)\}$.

Remark: The constant a in the comparison scheme is proportional to $\frac{1}{\Delta x}$. It also depends on the norms of the function and of its derivatives at smooth regions $\|f^{(i)}(x)\|_{(a,b)\setminus D}$ where $0 \leq i \leq p$ and D is the set of the discontinuities.

The extra cost introduced by this comparison jump identification method over the standard wavelet transforms is just the comparison $|\beta_{j,i}| > a|\beta_{j,i-1}|$ for each stencil. In section 2.5, we use this detection method for all noise free numerical examples.

Noisy Data

The above described detection method may not be reliable if the function is polluted by noise, especially when the noise is "large". This is because the high frequency coefficients β's may not be able to measure the correct order of smoothness of the functions. Indeed, the high frequency coefficients have the

order $\|f^{(p)}(x) + \sigma n^{(p)}(x)\|O(\Delta x^p)$, where $n(x)$ is the random noise and σ a positive number indicating the noise level. In general, the derivatives of the noise $n^{(p)}(x)$ have large values. The noise term $\sigma n^{(p)}(x)$ can dominate the function term $f^{(p)}(x)$ if the noise level σ is large and thus, the high frequency coefficients β's may not be able to detect certain discontinuities, e.g. if the jump is small or the discontinuity is in the higher derivatives. In this situation, we need to use heuristics to locate the exact position of the essential discontinuities. Here, we give a simple method to detect the significant large jumps in function values in noisy data.

In many applications such as in image processing, large discontinuities in function value are the most significant features. Using the standard wavelet transforms, these large discontinuities will generate high frequency coefficients which can be much larger than those generated by the noise. (this is also the fundamental principle in the design of wavelet thresholding). A simple way to detect this kind of discontinuities is to look for these large magnitude high frequency coefficients and then compare the data values in the corresponding stencils to locate the exact jump positions. For example, we can look for the places which have the largest difference between two adjacent data values within the stencils. In section 2.5, we will show an example using this method.

Remark: Other jump detection methods can be used for noisy data. As long as the exact location of the discontinuities are correctly determined, the coarse level ENO-wavelet approximations can be formed at the discontinuities, and our experience shows that it is not sensitive to the presence of noise.

2.3.3 Forward and Inverse Transform Algorithms

In this subsection, we explicitly present the complete forward and inverse ENO-wavelet transform algorithms for the noise free piecewise smooth data.

We consider the forward transform algorithms first. We denote by $\{c_0, \cdots, c_l\}$ and $\{h_0, \cdots, h_l\}$ the standard wavelet filter coefficients, and $\{r_0, \cdots, r_l\}$ and $\{d_0, \cdots, d_l\}$ the corresponding inverse filter coefficients. We use a one-bit variable s_i to indicate whether a stencil contains a jump in our algorithms.

Forward Transform Algorithm

For each i,

(i) Compute $\beta_{j,i}$ by (2.13).

(ii) If $|\beta_{j,i}| \geq a|\beta_{j,i-1}|$ and $|\beta_{j,i}| \geq \epsilon$, then,

- Compute $\beta_{j,i+k}$ and $\beta_{j,i+k+1}$ by (2.13).
- Find the exact location of the jump by Lemma 1. For $i \leq m \leq i+k$ or $i \leq m \leq i+k+1$,
  - For the left side of the jump, compute $\hat{\alpha}_{j,m}$ by extrapolation, compute $\hat{\beta}_{j,m}$ by (2.20). Then set:

$$\beta_{j,m} = \hat{\beta}_{j,m}, s_i = 1;$$

  - For the right side of the jump, set $\bar{\beta}_{j,m} = 0$ and compute $\bar{\alpha}_{j,m}$ by (2.21). Set:

$$\alpha_{j,m} = \bar{\alpha}_{j,m}.$$

(iii) Otherwise, compute $\alpha_{j,i}$ by (2.12). Set $s_i = 0$.

In the algorithm, ϵ is a predefined small positive number which is used to prevent the numerical instability caused by small $\beta_{j,i}$. More precisely, if both $\beta_{j,i}$ and $\beta_{j,i-1}$ are less that the given tolerance ϵ, we treat the current standard stencil as a smooth stencil.

In step (ii), it is possible to use any extrapolation techniques to handle the discontinuities.

Inverse Transform Algorithm

For each i,

(i) If $s_i = 0$ and $s_j = 0, j = i - l, \cdots, i - 1$, then the standard inverse wavelet transforms are applied,

$$\alpha_{j+1,2i} = \sum_{s=0}^{l} (r_{2s+1}\alpha_{j,i-s} + d_{2s+1}\beta_{j,i-s}), \tag{2.23}$$

and

$$\alpha_{j+1,2i+1} = \sum_{s=0}^{l} (r_{2s}\alpha_{j,i-s} + d_{2s}\beta_{j,i-s}). \tag{2.24}$$

(ii) If $s_j = 1$, $i - l \leq j \leq i$,

- Use Lemma 1 to locate the position of the jump by counting the number of consecutive $s_i = 1$.
- Extrapolate $\hat{\alpha}_{j,i}$ from the left side of the jump.
- Set $\bar{\beta}_{j,i}$ as zero for the right side of the jump.
- Using $\hat{\alpha}_{j,k}$ and $\beta_{j,k}$ to restore the left side by (2.23).
- Using $\alpha_{j,k}$ and $\bar{\beta}_{j,k}$ to restore the right side of the jump by (2.24).

Two Simple Examples

We give two simple examples in the ENO-Haar case to illustrate the algorithms. First, we consider computing the transform coefficients of the following initial data:

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 \end{pmatrix}.$$

The standard Haar produces the low and high frequency coefficients:

$$\alpha = \begin{pmatrix} \frac{2}{\sqrt{2}} & \frac{3}{\sqrt{2}} & \frac{4}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} 0 & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}.$$

The corresponding linear approximation is:

$$\begin{pmatrix} 1 & 1 & \frac{3}{2} & \frac{3}{2} & 2 & 2 \end{pmatrix},$$

which cannot recover the discontinuity correctly.

Using the ENO-Haar wavelet, we break the initial data sequence into two smooth pieces as shown in the following two rows:

$$\begin{pmatrix} & & y & 2 & 2 & 2 \\ 1 & 1 & 1 & x & & \end{pmatrix},$$

where x and y are some smooth extensions of the corresponding pieces. In fact, we extend x in a way such that the low frequency coefficient $\hat{\alpha}_2$ (boxed in (2.25)) based on the stencil $(1, x)$ is the same as the previous α_1, which is based on the stencil $(1, 1)$ giving $x = 1$. Similarly, we extend y in a way such that the high frequency coefficient $\bar{\beta}_2$ (boxed in (2.25)) is zero giving $y = 2$. Therefore we compute the high frequency coefficients $\hat{\beta}_2$ based on stencil $(1, x)$ and the low frequency coefficients $\bar{\alpha}_2$ based on stencil $(y, 2)$ by using the corresponding standard filters giving $\hat{\beta}_2 = 0$ and $\bar{\alpha}_2 = \frac{4}{\sqrt{2}}$. Thus we have the coefficients:

$$\alpha = \begin{pmatrix} & \frac{4}{\sqrt{2}} & \frac{4}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} & \boxed{\frac{2}{\sqrt{2}}} & \end{pmatrix}, \beta = \begin{pmatrix} & \boxed{0} & 0 \\ 0 & 0 & \end{pmatrix}. \tag{2.25}$$

Since we know how we extended $\hat{\alpha}_2$ and $\bar{\beta}_2$, we do not need to store them. In fact, we just need to store the low and high frequency coefficients as:

$$\alpha = \begin{pmatrix} \frac{2}{\sqrt{2}} & \frac{4}{\sqrt{2}} & \frac{4}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix},$$

which have the same storage schemes as the standard Haar wavelet transform.

When we reconstruct the linear approximation, we can first recover $\hat{\alpha}_2$ and $\bar{\beta}_2$ by the same way as in the forward transform, and then apply the standard inverse filters to the smooth data to build the approximation. In fact, in this case the linear approximation is exactly the initial data.

In the next example, we show a similar example in which the ENO-Haar linear approximation is not exactly the same as the initial data but it still preserves the jump well. The initial data is given as:

$$\begin{pmatrix} 0 & 1 & 2 & 10 & 11 & 12 \end{pmatrix}.$$

The standard Haar low and high frequency coefficients are:

$$\alpha = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{12}{\sqrt{2}} & \frac{23}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{8}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Notice that in this case, we have a large high frequency coefficient (middle) which corresponds to the discontinuity. If we discard the high frequency part, the corresponding linear approximation is

$$\begin{pmatrix} 0.5 & 0.5 & 6 & 6 & 11.5 & 11.5 \end{pmatrix},$$

and the discontinuity cannot be preserved.

If we use the ENO-Haar wavelet transform, we have the coefficients:

$$\alpha = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{20}{\sqrt{2}} & \frac{23}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

The recovered linear approximation is

$$\begin{pmatrix} 0.5 & 0.5 & 0.5 & 10 & 11.5 & 11.5 \end{pmatrix}.$$

In this case, although the linear approximation is not the same as the initial data, it forms a much more accurate approximation than that of the standard Haar transform. More importantly, this approximation preserves the discontinuity sharply in contrast to the standard Haar wavelet which takes the average at the discontinuity.

Remarks

(i) The ENO-wavelet transforms are just simple modifications of the standard wavelet transforms near discontinuities. The computational complexity of the algorithms remains $O(n)$ and they are relatively easy to implement.

(ii) In the transform algorithms and the corresponding inverse algorithms, the extended low frequency coefficients $\hat{\alpha}_{j,m}$ and the high frequency coefficients $\bar{\beta}_{j,m}$ can be computed by other extrapolation schemes such as least square extrapolation. This may be more robust, especially for noisy data.

(iii) The adaptive ENO wavelets idea can also be used for other kind of wavelets. They do not necessarily have to be orthogonal wavelets.

(iv) Like other wavelet transforms, 2-dimensional or even higher dimensional transforms can be formed by tensor products. In the numerical example section, we will give a 2-dimensional example.

(v) The adaptive ENO-wavelet idea can be recursively used even if the projections do not satisfy the DSP. In such case, of course we will not get the nice error bound (see section 2.4), but the approximation error are comparable

to that of the standard wavelet transforms. Also, it is easy to modify the algorithms such that the standard wavelet transforms are applied at the place where the DSP is invalid.

2.4 Approximation Error

In this section, we consider the ENO-wavelets approximation error for piecewise continuous functions.

Given a function $f(x)$ in L^2, in standard wavelet theory [33] [18] [42], it can be linearly approximated by its projection $f_j(x)$ in V_j as in (2.7) and (2.8). This linear approximation has a standard error estimate which we state in the following theorem, see also [42].

Theorem 1 *Suppose the wavelet $\psi(x)$ generated by scaling function $\phi(x)$ has p vanishing moments, $f_j(x)$ is the approximation of $f(x)$ in V_j with basis $\phi_{j,k}(x)$, then,*

$$\|f(x) - f_j(x)\| \leq C(\Delta x)^p \|f^{(p)}(x)\|, \tag{2.26}$$

where $\Delta x = 2^{-j}$ and C is a constant which is independent of j.

This theorem holds for the L^2 norm in general. Moreover, if the scaling function and the wavelets have finite support, then it also holds for the L^∞ norm.

In this theorem, we can see that the approximation error is controlled by two factors. One is the p-th power of the spatial step Δx; the other is the norm of the p-th derivative of the function. This implies that the approximation could be poor for irregular functions even if the spatial step Δx is small. For piecewise continuous functions, especially functions with large jumps, this error

estimate does not give a real upper bound because $\|f^{(p)}(x)\|$ is infinite. In fact, in the standard approximation function $f_j(x)$, oscillations are generated near the discontinuous points and they will not disappear even if the spatial step size is reduced (Gibbs' phenomenon).

In contrast, in our ENO-wavelet transforms, since no approximation coefficients are computed using information from both sides of the discontinuities, we can obtain a similar error estimate without taking derivatives across the jumps. In the next theorem, we state the estimation and prove it in the rest of this section.

Theorem 2 *Suppose the scaling function $\phi(x)$ and its $\psi(x)$ have finite support in $[0, l]$, $\psi(x)$ has p vanishing moments, $f(x)$ is a piecewise continuous function in $[a, b]$, and $f_j(x)$ is its j-th level ENO-wavelet projection obtained by using any one of the three extrapolation methods given in section 2.4. If the projection $f_{j+1}(x)$ satisfies the DSP, then*

$$\|f(x) - f_j(x)\| \leq C(\Delta x)^p \|f^{(p)}(x)\|_{(a,b)\setminus D}, \tag{2.27}$$

where $\Delta x = 2^{-j}$ and D is the set where $f(x)$ has jumps. The norm $\|\cdot\|$ can be either the L^2 or the L^∞ norm.

Proof: We prove the inequality (2.27) under the L^∞ norm, which implies the L^2 result.

The DSP allows us to separate the discontinuities and individually consider a small neighborhood around each jump. Therefore, without loss of generality, we consider a piecewise function $f(x)$ with one jump at the origin. i.e.

$$f(x) = \begin{cases} f_1(x) & a \leq x < 0 \\ f_2(x) & 0 \leq x \leq b, \end{cases}$$

where $f_1(x) \in C^p[a,0]$ and $f_2(x) \in C^p[0,b]$. The small neighborhood affected by the ENO decision is $[-l\Delta x, l\Delta x]$. In fact, the ENO wavelet coefficients only depend on one sided information and therefore, by symmetry, we just need to prove (2.27) in $[-l\Delta x, 0]$.

Before we prove that (2.27) holds for the three types of extrapolation methods, namely direct function extrapolation and the two choices of coarse level extension ((1) and (2) in section 2.3.1), we give some notations which we will frequently use in the proof.

Denote by $g_1(x)$ the $(p-1)$-th order polynomial which is the first p terms of the Taylor expansion of $f_1(x)$ at the origin, i.e.

$$f_1(x) = g_1(x) + \frac{f_1^{(p)}(\xi)}{p!}x^p, \tag{2.28}$$

where ξ is in interval $[-l\Delta x, 0]$. Also denote by $\alpha_{j,m}$ and $\beta_{j,m}$ the ENO-wavelet's low and high frequency coefficients respectively and $\bar{\alpha}_{j,m}$ the low frequency coefficients of the polynomial $g_1(x)$, i.e.,

$$\bar{\alpha}_{j,m} = \int g_1(x)\phi_{j,m}(x)dx$$

and

$$g_{1,j}(x) = \sum_m \bar{\alpha}_{j,m}\phi_{j,m}(x).$$

Now, we are ready to prove that (2.27) holds for the three types of extrapolation methods. We first prove (2.27) for direct function extrapolation.

Direct Function Extrapolation

The direct function extrapolation extends $f_1(x)$ to interval $[0, l\Delta x]$ by defining

$$f_d(x) = \begin{cases} f_1(x) & -l\Delta x \le x < 0 \\ g_1(x) & 0 \le x \le l\Delta x. \end{cases}$$

The corresponding ENO-wavelet low frequency coefficients $\alpha_{j,m}$ are computed by

$$\alpha_{j,m} = \int f_d(x)\phi_{j,m}(x)dx, \tag{2.29}$$

and the approximation function is defined as

$$f_{d,j}(x) = \sum_m \alpha_{j,m}\phi_{j,m}(x), \qquad x \in [-l\Delta x, 0]. \tag{2.30}$$

For any point $x_0 \in [-l\Delta x, 0]$, by using (2.28) and the fact that since $g_1(x)$ is a $(p-1)$-th order polynomial, $g_{1,j}(x) = g_1(x)$, we have:

$$\begin{aligned} |f_1(x_0) - f_{d,j}(x_0)| &\leq |f_1(x_0) - g_1(x_0)| + |g_{1,j}(x_0) - f_{d,j}(x_0)| \\ &\leq C(\Delta x)^p \|f_1^{(p)}\| + |g_{1,j}(x_0) - f_{d,j}(x_0)|. \end{aligned} \tag{2.31}$$

Let q be an integer in $[-l, 0]$ such that $x_0 \in [q\Delta x, (q+1)\Delta x)$, then the last term of (2.31) can be bounded by:

$$\begin{aligned} |g_{1,j}(x_0) - f_{d,j}(x_0)| &= |\sum_m (\bar{\alpha}_{j,m} - \alpha_{j,m})\phi_{j,m}(x_0)| \\ &\leq \sum_{q-l\leq m\leq q} |\bar{\alpha}_{j,m} - \alpha_{j,m}||\phi_{j,m}(x_0)| \\ &= \sum_{q-l\leq m\leq q} |\bar{\alpha}_{j,m} - \alpha_{j,m}||(\Delta x)^{-\frac{1}{2}}\phi(2^j x_0 - m)|. \end{aligned} \tag{2.32}$$

To prove (2.27), we now need to estimate $|\bar{\alpha}_{j,m} - \alpha_{j,m}|$. In fact, for every m, $-l \leq m \leq 0$, we have:

$$\begin{aligned} |\bar{\alpha}_{j,m} - \alpha_{j,m}| &= |\int (f_d(x) - g_1(x))\phi_{j,m}(x)dx| \\ &\leq |\int_{m\Delta x}^{0} (f_d(x) - g_1(x))\phi_{j,m}(x)dx| \\ &\quad + |\int_0^{(m+l)\Delta x} (f_d(x) - g_1(x))\phi_{j,m}(x)dx|. \end{aligned}$$

Because $f_d(x)$ is the same as $g_1(x)$ in $[0, (m+l)\Delta x]$, using (2.28), we have:

$$|\bar{\alpha}_{j,m} - \alpha_{j,m}| = |\int_{m\Delta x}^{0} (f_1(x) - g_1(x))\phi_{j,m}(x)dx|$$

$$
\begin{aligned}
&\leq (\int_{m\Delta x}^{0} |f_1(x) - g_1(x)|^2 dx)^{\frac{1}{2}} (\int_{m\Delta x}^{0} |\phi_{j,m}(x)|^2 dx)^{\frac{1}{2}} \\
&\leq C(\Delta x)^p \|f^{(p)}\| (\Delta x)^{\frac{1}{2}} \\
&\leq C(\Delta x)^{p+\frac{1}{2}} \|f^{(p)}\|.
\end{aligned}
$$

Therefore, combining this with (2.32), we have:

$$|g_{1,j}(x_0) - f_{d,j}(x_0)| \leq C(\Delta x)^p \|f^{(p)}\|.$$

This and (2.31) complete the proof of (2.27) for the case of direct function extrapolation.

Coarse Level Extrapolation

As described in section 2.4, there are two ways of extrapolating coefficients on the coarse level. One way is to set the extended high frequencies to zero. The other way is to extrapolate the low frequency coefficients by a $(p-1)$-th order polynomial in wavelet space. In the following part of the proof, we consider them separately.

We consider the high frequency zero extension first.

Similar to the direct function extrapolation, we also extend $f_1(x)$ to the interval $[0, l\Delta x]$ and denote it by

$$f_h(x) = \begin{cases} f_1(x) & x \in [-l\Delta x, 0] \\ g_h(x) & x \in (0, l\Delta x], \end{cases}$$

where $g_h(x)$ is implicitly defined such that it makes $f_h(x)$ satisfy

$$\int f_h(x)\psi_{j,m}(x)dx = 0, \qquad -l+1 \leq m \leq 0, \tag{2.33}$$

and

$$\int f_h(x)\phi_{j,m}(x)dx = \alpha_{j,m}, \qquad -l+1 \leq m \leq 0. \tag{2.34}$$

The difference between $f_d(x)$ and $f_h(x)$ is that in the direct function extrapolation $f_d(x)$, we know that $g_1(x)$ is the $(p-1)$-th order polynomial, but in this case, $g_h(x)$ is unknown.

Formally following the proof of (2.27) for the direct function extrapolation, (2.31) and (2.32) also hold for this case. Therefore, we only need to estimate $|\bar{\alpha}_{j,m} - \alpha_{j,m}|, -l < m \leq 0$. We consider $m = -l+1$ first. Unlike in the direct function extrapolation where $|\bar{\alpha}_{j,-l+1} - \alpha_{j,-l+1}|$ can be computed directly by the Taylor expansion, here we cannot bound $|\bar{\alpha}_{j,-l+1} - \alpha_{j,-l+1}|$ in the same way. Instead, we use the following trick to obtain the estimate we need.

From the dilation equation (2.1) and the wavelet equation (2.2), we have the following relationships:

$$\phi_{j,m}(x) = \sum_{s=0}^{l-1} c_s \phi_{j+1,s+2m}(x), \tag{2.35}$$

and

$$\psi_{j,m}(x) = \sum_{s=0}^{l-1} h_s \phi_{j+1,s+2m}(x). \tag{2.36}$$

Using (2.35), $\bar{\alpha}_{j,-l+1}$ and $\alpha_{j,-l+1}$ can be computed by

$$\alpha_{j,-l+1} = \int f_h(x)\phi_{j,-l+1} = \sum_{s=0}^{l-1} c_s \int_{\frac{\Delta x}{2}(s-2l+2)}^{\frac{\Delta x}{2}(s-l+2)} f_h(x)\phi_{j+1,s-2l+2}(x)dx$$

and

$$\bar{\alpha}_{j,-l+1} = \int g_1(x) phi_{j,-l+1} = \sum_{s=0}^{l-1} c_s \int_{\frac{\Delta x}{2}(s-2l+2)}^{\frac{\Delta x}{2}(s-l+2)} g_1(x)\phi_{j+1,s-2l+2}(x)dx.$$

Therefore, we have

$$\begin{aligned} |\bar{\alpha}_{j,-l+1} - \alpha_{j,-l+1}| \;\leq\; & |\sum_{s=0}^{l-2} c_s) \int_{\frac{\Delta x}{2}(s-2l+2)}^{\frac{\Delta x}{2}(s-l+2)} (g_1(x) - f_h(x))\phi_{j+1,s-2l+2}(x)dx| \\ & + |c_{l-1} \int_{\frac{\Delta x}{2}(-l+1)}^{\frac{\Delta x}{2}} (g_1(x) - f_h(x))\phi_{j+1,1-l}(x)dx|. \end{aligned} \tag{2.37}$$

We know that only the last term involves the value of $f_h(x)$ in $[0, \frac{\Delta x}{2}]$. The other terms use $f_h(x)$ in $[-l\Delta x, 0]$, which is $f_1(x)$. Then, by Taylor expansion and Schwartz inequality,

$$|\sum_{s=0}^{l-2} c_s \int_{\frac{\Delta x}{2}(s-2l+2)}^{\frac{\Delta x}{2}(s-l+2)} (g_1(x) - f_h(x))\phi_{j+1,s-2l+2}(x)dx| \leq C(\Delta x)^p \|f_1^{(p)}\| 2^{\frac{-(j+1)}{2}}. \quad (2.38)$$

Thus, to bound $|\bar{\alpha}_{j,-l+1} - \alpha_{j,-l+1}|$, the only remaining task is to estimate the last term in (2.37).

Considering that $g_1(x)$ is a $(p-1)$-th order polynomial, we obtain

$$\int f_h(x)\psi_{j,-l+1}(x)dx = 0 = \int g_1(x)\psi_{j,-l+1}(x)dx.$$

Substituting the wavelet equation (2.36) into the above equation, we have:

$$\sum_{s=0}^{l-1} h_s \int (f_h(x) - g_1(x))\phi_{j+1,s-2l+2}(x)dx = 0.$$

We can rewrite this equation in the following form:

$$\begin{aligned} h_{l-1} \int_{\frac{\Delta x}{2}(-l+1)}^{\frac{\Delta x}{2}} & (f_h(x) - g_1(x))\phi_{j+1,1-l}(x)dx \\ & = -\sum_{s=0}^{l-2} h_s \int_{\frac{\Delta x}{2}(s-2l+2)}^{\frac{\Delta x}{2}(s-l+2)} (f_h(x) - g_1(x))\phi_{j+1,s-2l+2}(x)dx. \end{aligned}$$

We note that the left hand side contains the term we need to estimate, whereas the right hand side only uses $f_h(x)$ at the left side of the origin and thus can be controlled by Taylor expansion. This means that we have:

$$\begin{aligned} | \int_{\frac{\Delta x}{2}(-l+1)}^{\frac{\Delta x}{2}} & (f_h(x) - g_1(x))\phi_{j+1,1-l}(x)dx| \\ & \leq |h_{l-1}^{-1}| \sum_{s=0}^{l-2} |h_s| \int_{\frac{\Delta x}{2}(s-2l+2)}^{\frac{\Delta x}{2}(s-l+2)} |f_1(x) - g_1(x)||\phi_{j+1,s-2l+2}(x)|dx \\ & \leq C(\Delta x)^p \|f_1^{(p)}\| 2^{\frac{-(j+1)}{2}}. \end{aligned} \quad (2.39)$$

Combining (2.37), (2.38) and (2.39), we have:

$$|\bar{\alpha}_{j,-l+1} - \alpha_{j,-l+1}| \leq C(\Delta x)^p \|f_1^{(p)}\| 2^{-\frac{(j+1)}{2}}.$$

Similarly, we can prove that for all $m, -l+1 < m \leq 0$,

$$|\bar{\alpha}_{j,m} - \alpha_{j,m}| \leq C(\Delta x)^p \|f_1^{(p)}\| 2^{-\frac{(j+1)}{2}}.$$

Substituting them into (2.32), we prove that (2.27) holds for the high frequency extension case.

The last case we need to consider is the coarse level extrapolation of low frequency coefficients. To prove (2.27), we use the result obtained for the direct function extrapolation.

We denote by $\alpha_{j,m}^d$ the low frequency coefficients for $f_d(x)$. The j-th level low frequency extrapolation approximation $f_{l,j}(x)$ is defined as:

$$f_{l,j}(x) = \sum_m \alpha_{j,m}\phi_{j,m}(x).$$

For any point $x_0 \in [q\Delta x, (q+1)\Delta x) \subset [-l\Delta x, 0]$, we have:

$$|f_1(x_0) - f_{l,j}(x_0)| \leq |f_1(x_0) - f_{d,j}(x_0)| + |f_{d,j}(x_0) - f_{l,j}(x_0)|. \tag{2.40}$$

Using (2.27) for the direct function extrapolation case, we know that:

$$|f_1(x_0) - f_{d,j}(x_0)| \leq C(\Delta x)^p \|f_1^{(p)}\|. \tag{2.41}$$

And the remaining term can be bounded by:

$$|f_{d,j}(x_0) - f_{l,j}(x_0)| \leq \sum_{q-l\leq m\leq q} |\alpha_{j,m}^d - \alpha_{j,m}||2^{\frac{j}{2}}\phi_(2^j x_0 - m)|. \tag{2.42}$$

Again, we need to estimate $|\alpha_{j,m}^d - \alpha_{j,m}|$.

Unlike the previous two cases where the low frequency coefficients $\alpha_{j,m}$ are computed by integration (2.29) or (2.34), in this case $\alpha_{j,m}$ are determined by the low frequency extrapolation on coarse level in wavelet space. So, to estimate

$|\alpha_{j,m}^d - \alpha_{j,m}|$, we need to consider them in wavelet space. We introduce the following operator notations first.

Define the continuous wavelet transform (WT) of any function $f(x)$ in space V_j by

$$WT(f)(s) = \int f(x)\phi_{j,s}(x)dx = 2^{\frac{j}{2}} \int f(x)\phi(2^j x - s)dx.$$

Also define the following Taylor extrapolation operator (EX) of $f(x)$:

$$EX(f)(x) = \begin{cases} f(x) & x \leq 0 \\ g(x) & x > 0 \end{cases}$$

where $g(x)$ is the $(p-1)$-th order Taylor polynomial of $f(x)$. Using these notations, we can represent the wavelet coefficients:

$$\alpha_{j,m} = EX_w(WT(f_1))(m), \quad for \quad -l+1 \leq m \leq 0,$$

and

$$\alpha_{j,m}^d = WT(EX_f(f_1))(m), \quad for \quad -l+1 \leq m \leq 0,$$

where EX_w and EX_f represent the extrapolation operator EX in the wavelet and physical space respectively.

Instead of estimating $|\alpha_{j,m}^d - \alpha_{j,m}|$ directly, we prove the following more general result.

Lemma 2 *Given a smooth function $g(x)$, let $g_{we}(s) = WT(EX_f(g))(s)$ and $g_{ew}(s) = EX_w(WT(g))(s)$, then*

$$|g_{we}(s) - g_{ew}(s)| \leq C(\Delta x)^p \|g^{(p)}\| 2^{-\frac{j}{2}}.$$

Using this lemma, we obtain the desired bounds for $|\alpha_{j,m}^d - \alpha_{j,m}|$ easily by taking $s = m$. Combining them with (2.41) and (2.42), we prove that (2.27) holds for the low frequency coefficient extrapolation case.

Proof of Lemma: Denote $\bar{g}(x) = EX_f(g)(x)$, then:

$$\begin{aligned} g_{we}(s) &= 2^{\frac{j}{2}} \int \bar{g}(x)\phi(2^j x - s)dx \\ &= 2^{-\frac{j}{2}} \int \bar{g}(2^{-j}(y-s)\phi(y)dy. \end{aligned}$$

By changing variable $z = 2^{-j}s$, and denoting:

$$e_j(z) = \int \bar{g}(2^{-j}y - z)\phi(y)dy,$$

we have:

$$g_{we}(s) = 2^{-\frac{j}{2}} e_j(2^{-j}s).$$

We also know that $e_j(z)$ is a smooth function, and by differentiating p times, we have:

$$\|e_j^{(p)}\| = \| \int (-1)^p \bar{g}^{(p)}(2^{-j}y - z)\phi(y)dy\| \leq C\|g^{(p)}\| \| \int \phi(y)dy\| \leq C\|g^{(p)}\|. \tag{2.43}$$

Taking the $(p-1)$-th order Taylor expansion of $e_j(z)$ at $z = -l\Delta x$, we have:

$$e_j(z) = \hat{e}_j(z) + e_j^{(p)}(\xi)\frac{(z + l\Delta x)^p}{p!},$$

where $\hat{e}_j(z)$ is the $(p-1)$-th order Taylor polynomial and $\xi \in [2l, 0]$. Since $g_{ew}(s)$ is the same as $g_{we}(s)$ if $s \leq -l$, it is defined as the Taylor polynomial for $s > -l$ according to the definition of EX, i.e. we have:

$$g_{ew}(s) = \begin{cases} 2^{-\frac{j}{2}} e_j(2^{-j}s) & s \leq -l \\ 2^{-\frac{j}{2}} \hat{e}_j(2^{-j}s) & s > -l \end{cases}$$

Therefore,

$$\begin{aligned} |g_{we}(s) - g_{ew}(s)| &\leq 2^{-\frac{j}{2}} |e_j(2^{-j}s) - \hat{e}_j(2^{-j}s)| \\ &\leq C(\Delta x)^p \|g^{(p)}\| 2^{-\frac{j}{2}}. \end{aligned}$$

This completes the proof of Lemma 2 and also completes the proof of Theorem 2.

2.5 Numerical Examples

In this section, we give some 1-D and 2-D numerical examples by using the ENO-wavelet transforms. In particular, we show results of the ENO-Haar, ENO-DB4 and ENO-DB6 wavelet transforms.

In all examples, for simplicity, we just consider functions with zero values at the boundary. For non-zero boundary functions, we can easily extend the function by zero and treat the boundaries as discontinuities.

To illustrate the performance of ENO-wavelet transforms, we show picture comparisons of the standard wavelet approximations and corresponding ENO-wavelet approximations. In addition, we compare the L_∞ and L_2 errors of the standard wavelet approximations and the ENO-wavelet approximations at different levels by measuring $E_{\infty,j} = \inf_x \|f(x) - f_j(x)\|$, which is computed by finding the largest difference on the finest grid, and $E_{2,j} = \|f(x) - f_j(x)\|_2$. Using them, we compute the orders of accuracy defined by:

$$Order_\infty = \log_2 \frac{E_{\infty,i}}{E_{\infty,i-1}},$$

and

$$Order_2 = \log_2 \frac{E_{2,i}}{E_{2,i-1}},$$

which indicates the order of accuracy of the approximation in the L_∞ norm and L_2 norm respectively.

For all noise free examples, we use the method described in section 2.3.2 to locate the exact positions of the discontinuities. And we select $a = 2$ and $\epsilon = 0.0001$ (as used in the algorithms in section 2.3.3) for all 1-D examples.

Firstly, we compare the approximations for smooth functions. Table 2.1 is the comparison of Haar and ENO-Haar approximations for the smooth function

level	Haar E_∞	ENO-Haar E_∞	$Order_\infty$
4	0.0919	0.0919	
3	0.0430	0.0430	1.070
2	0.0184	0.0184	1.202
1	0.0061	0.0061	1.585

Table 2.1: Comparison of maximum error of the standard Haar and the ENO-Haar wavelet approximation for the smooth function $\sin(x)$. We see that they have the same approximation error for the smooth functions.

$f(x) = \sin(x), 0 \leq x \leq 2\pi$ at different levels, and Table 2.2 the comparison of DB4 and ENO-DB4 approximations for the function $f(x) = \exp\left[-(\frac{1}{x} + \frac{1}{1-x})\right], 0 < x < 1$,

We see from these tables that for smooth functions, the ENO-wavelet transforms have exactly the same approximation error as the standard wavelet transforms. Both of them maintain the approximation order 1 and 2 for Haar and DB4 respectively, which agree with the results in Theorem 1.

Next, we consider a piecewise smooth function defined by

$$f(x) = \begin{cases} 0 & 0 \leq x < 0.2 \\ -50x - 5 & 0.2 \leq x < 0.4 \\ 10\sin(4\pi x + 0.8\pi) - 1 & 0.4 \leq x < 1.1 \\ 5e^{2x} - 100 & 1.1 \leq x < 1.6 \\ 0 & 1.6 \leq x \leq 2. \end{cases}$$

We apply Haar and ENO-Haar, DB4 and ENO-DB4, and DB6 and ENO-DB6 transforms to this function and compare the approximation error. Fig 2.5 shows the comparison of the order of accuracy in the L_∞ and L_2 norm. It is clear that both L_∞ and L_2 order of accuracy for ENO-wavelet transforms are of the order

level	DB4 E_∞	ENO-DB4 E_∞	$Order_\infty$
4	3.316e-5	3.316e-5	
3	7.650e-6	7.650e-6	2.104
2	1.590e-6	1.590e-6	2.232
1	2.972e-7	2.973e-7	2.406

Table 2.2: The Comparison of maximum error of the standard DB4 and the ENO-DB4 approximations for the smooth function $f(x) = \exp[-(\frac{1}{x} + \frac{1}{1-x})], 0 < x < 1$. They have the same error and both achieve second order of accuracy which agrees with the results in Theorem 1 for the smooth functions.

1, 2 and 3 for ENO-Haar, ENO-DB4 and ENO-DB6 respectively. And they agree with the results in Theorem 2. In contrast, standard wavelet transforms do not retain the corresponding order of accuracy for piecewise smooth functions.

To see the Gibbs' oscillations, we display the 4-level ENO-wavelet and standard wavelet approximations in Fig 2.6, 2.7, and 2.8 for ENO-Haar, ENO-DB4 and ENO-DB6 approximations respectively. In the left column, we show the original function (dotted line), the standard wavelet linear approximations (dash-dotted) and the ENO-wavelet approximations (solid line). The right pictures are zoom-in's of the left pictures near a discontinuity. We clearly see the Gibbs' oscillations in the standard approximations; in contrast, the ENO-wavelet approximations preserve the jump accurately.

In Fig 2.9, 2.10 and 2.11, we also present the standard Haar, DB4 and DB6 wavelet coefficients (dotted line) and the ENO-Haar, ENO-DB4 and ENO-DB6 wavelet coefficients (solid line) respectively. The left part corresponds to the low frequency coefficients and the right part the high frequency coefficients. We notice

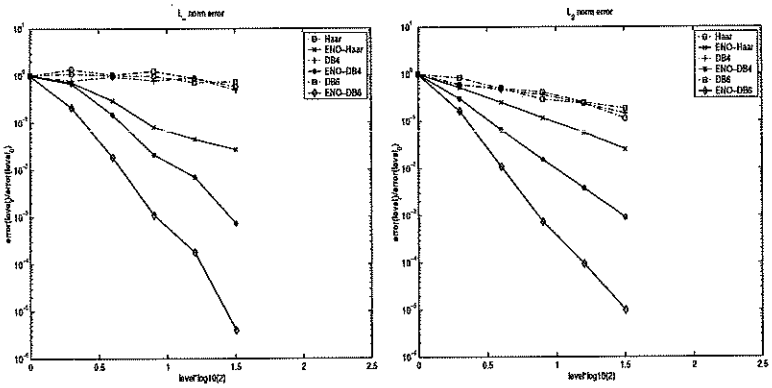

Figure 2.5: The approximation accuracy comparison of ENO-wavelet and wavelet transforms. Both L_∞ (left) and L_2 (right) order of accuracy show that ENO-wavelet transforms maintain the order 1, 2 and 3 for ENO-Haar, ENO-DB4 and ENO-DB6 respectively and they agree with the results of Theorem 2. In contrast, standard wavelet transforms do not retain the order of accuracy for piecewise smooth functions.

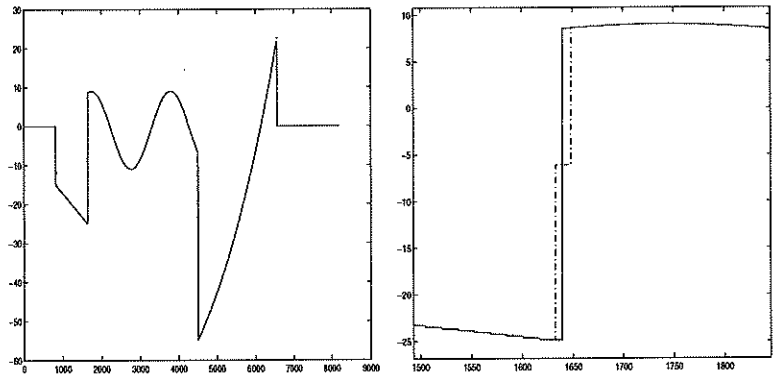

Figure 2.6: The 4-level ENO-Haar and Haar Approximation. The left picture shows the original function (dotted line), the standard Haar approximation (dash-dotted line) and the ENO-Haar approximation (solid line). The right picture is a zoom in near a discontinuity. We see the Gibbs' phenomenon (staircase) in the standard Haar approximation but not in the ENO-Haar approximation.

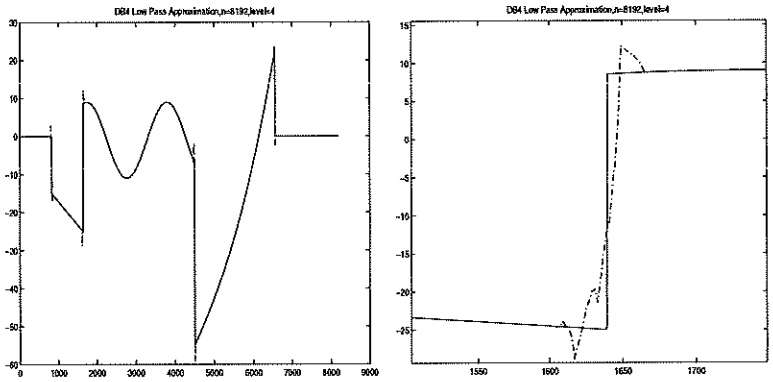

Figure 2.7: The 4-level ENO-DB4 and the standard DB4 Approximations. The original discontinuous function (dotted line), the standard DB4 approximation (dash-dotted line) and the ENO-DB4 approximation (solid line) are displayed. Gibbs' phenomenon is clearly seen for the standard DB4 approximation but not for the ENO-DB4 approximation.

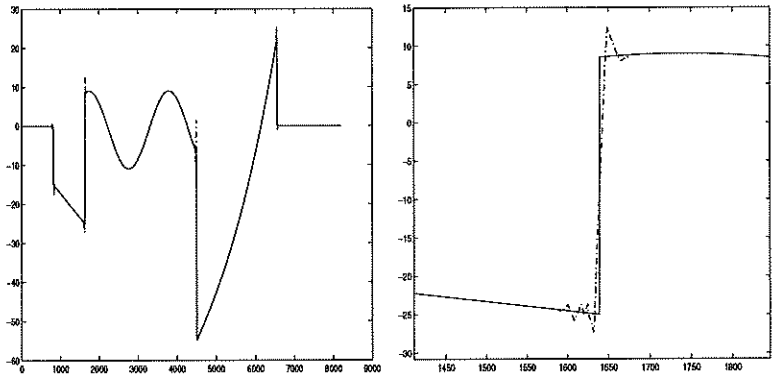

Figure 2.8: The 4-level ENO-DB6 (solid line) and the standard DB6 (dash-dotted line) Approximation. The standard DB6 generates oscillations near discontinuities, but the ENO-DB6 does not.

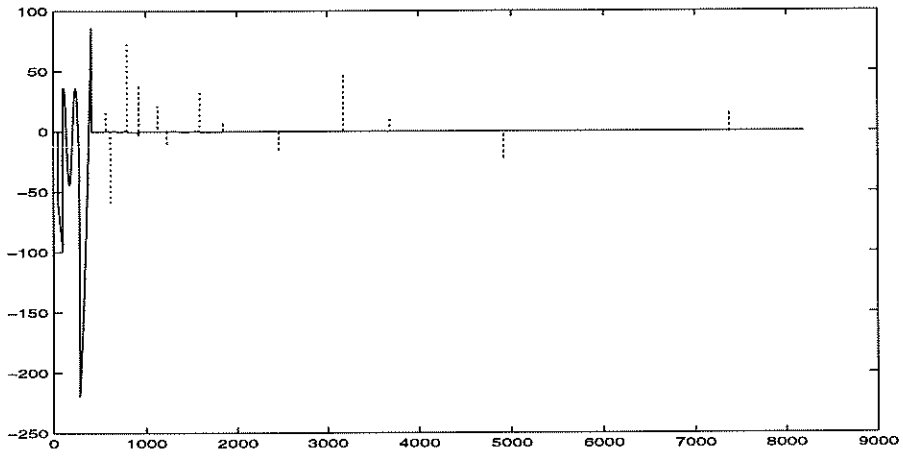

Figure 2.9: The 4-level ENO-Haar (solid line) and the standard Haar coefficients (dotted line). The left part corresponds to the low frequencies, the right part the high frequencies. In the standard Haar coefficients, large high frequency coefficients present near discontinuities while in the ENO-Haar case, there are no large high frequency coefficients.

that there are some large standard high frequency coefficients near the discontinuities. On the other hand, no large high frequency coefficients are present in the ENO-wavelet coefficients. This illustrates that the ENO-wavelet coefficients have better distribution than standard wavelet coefficients, i.e., no large coefficients in the high frequencies and the energy is concentrated in the low frequency end.

The next 1-D example we present here (Fig 2.12) is a comparison of the standard DB6 and the ENO-DB6 transforms to illustrate the performance at places where the DSP is not valid and also at jumps in the derivative. The original data (circles) has two discontinuities (the middle bump) which violate the DSP assumption, which requires that there are at least 9 data points between any pair of discontinuities, Although the ENO-DB6 approximation (solid line) does not preserve this pair of discontinuities exactly, its approximation error is still

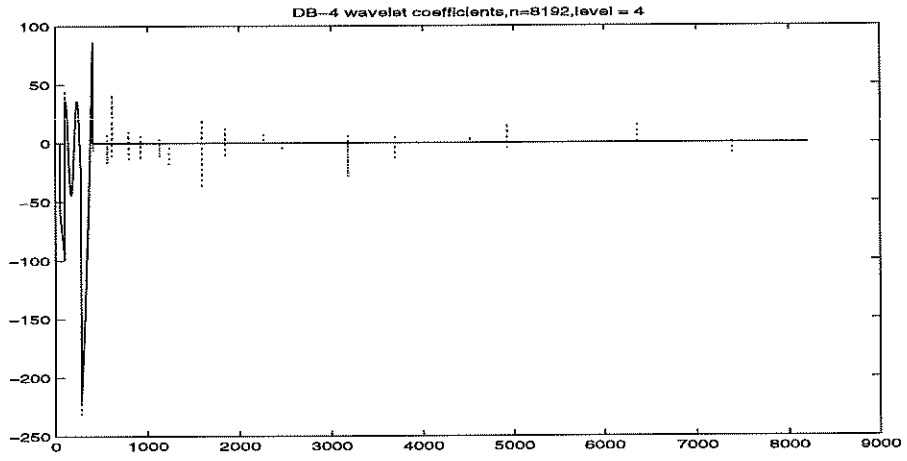

Figure 2.10: The 4-level ENO-DB4 coefficients (solid line) and the standard DB4 coefficients (dotted line). There are large high frequency coefficients (right part) near the discontinuities in the standard DB4 transform but not in the ENO-DB4 transform.

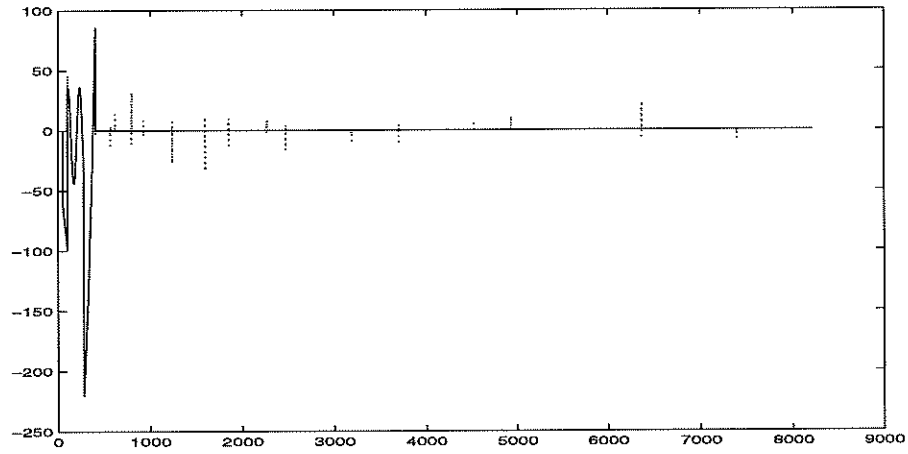

Figure 2.11: The 4-level ENO-DB6 coefficients (solid line) and the standard DB6 coefficients (dotted line). There are large high frequency coefficients near the discontinuities in the standard DB6 transform but not in the ENO-DB6 transform.

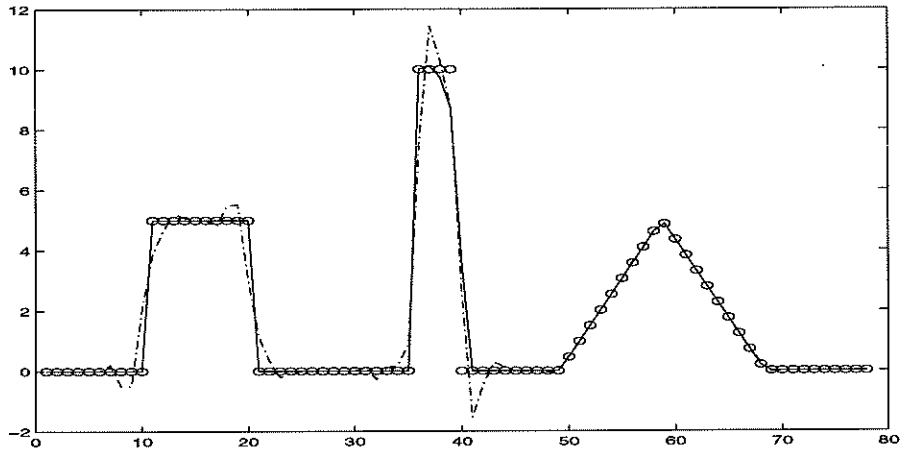

Figure 2.12: The level-1 approximation comparison of the ENO-DB6 and the standard DB6 wavelets at places where the DSP is invalid (the middle bump). The initial data (circles) has two close discontinuities. The ENO-DB6 approximation (solid line) error is comparable to that of the standard DB6 approximation (dotted line). The left bump satisfies the DSP and therefore the ENO-DB6 exactly recovers it. The right kink is a discontinuity in the first derivative, the standard DB6 still generates oscillations although their magnitudes are not significant. The ENO-DB6 restores it perfectly, We display a zoom-in picture of this kink in Fig. 2.13.

comparable (actually better in this case) to that of the standard DB6 approximation (dotted line). At the left bump where the DSP holds, the ENO-DB6 does preserve the exactly discontinuities as we expected. In the same example, we also display the comparison of the ENO-DB6 and the standard DB6 approximations at the right kink, which is not a discontinuity in function values but in its first order derivative. The standard DB6 approximation has oscillations although their magnitudes are small, but the ENO-DB6 restores it exactly (see Fig. 2.13).

The next 1-D example is also about the DSP for the ENO-DB6. As we defined

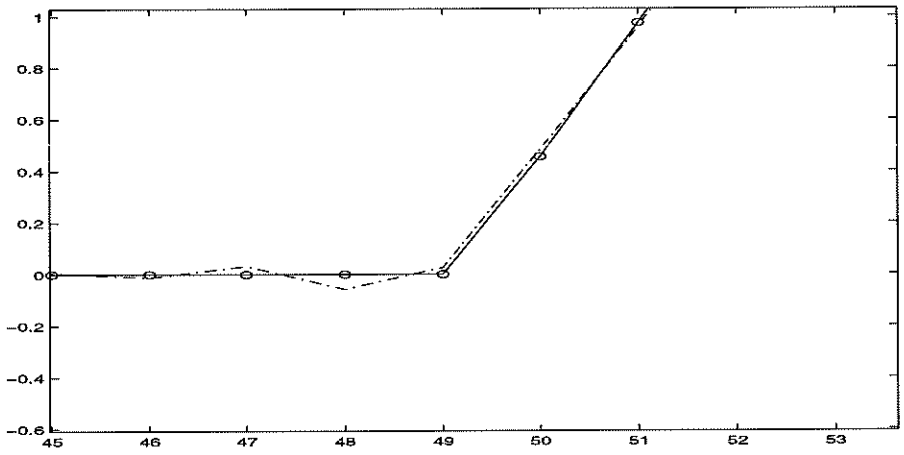

Figure 2.13: The zoom-in of the Fig 2.12 at the kink where there is a discontinuity in its derivative. The ENO-DB6 (solid line) can recover it perfectly but the standard DB6 (dash-dotted line) generates oscillations.

before, in this case, the DSP requires that any two discontinuities be separated by at least 9 data points. In Fig 2.14, we have a sequence of bumps satisfying the DSP and the ENO-DB6 exactly recovers them. But in Fig 2.15, there are only 8 data points between every pair of neighboring discontinuities (and therefore the DSP is invalid), and the reconstruction of the ENO-DB6 has oscillations but the error is comparable to that of the standard DB6 approximation.

The last 1-D example is applying the ENO-DB6 wavelet transform to a piecewise constant function polluted by Gaussian random noise (see Fig. 2.16). For this example, the jump detection method corresponding to Lemma 1 does not work. Instead, we use the simple method given in section 2.3.2, which detects jumps by looking for stencils with significant larger high frequency coefficients than their neighbors and then locates the exact jump locations by directly comparing the differences between two adjacent function values within the stencil. Despite the presence of noise in the initial data (circles), the level-3 ENO-DB6

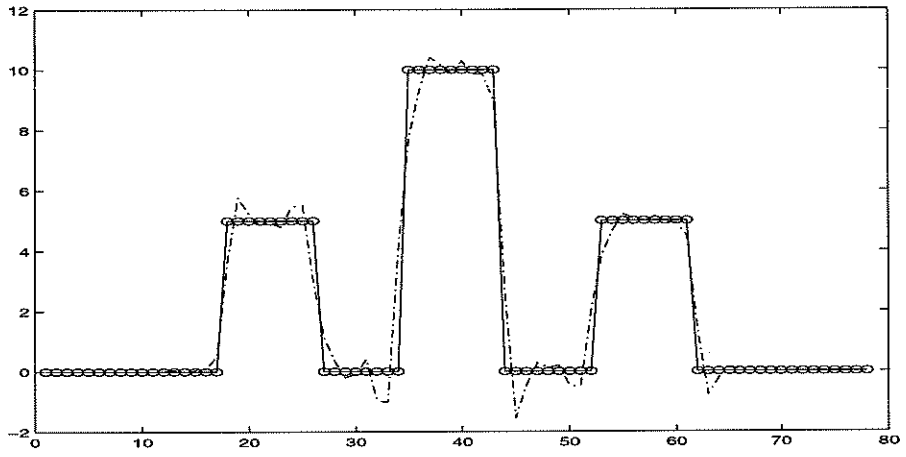

Figure 2.14: The number of data points between every two consecutive jumps is exactly 9, the DSP is satisfied in this case, the ENO-DB6 perform the exactly approximation of the original data.

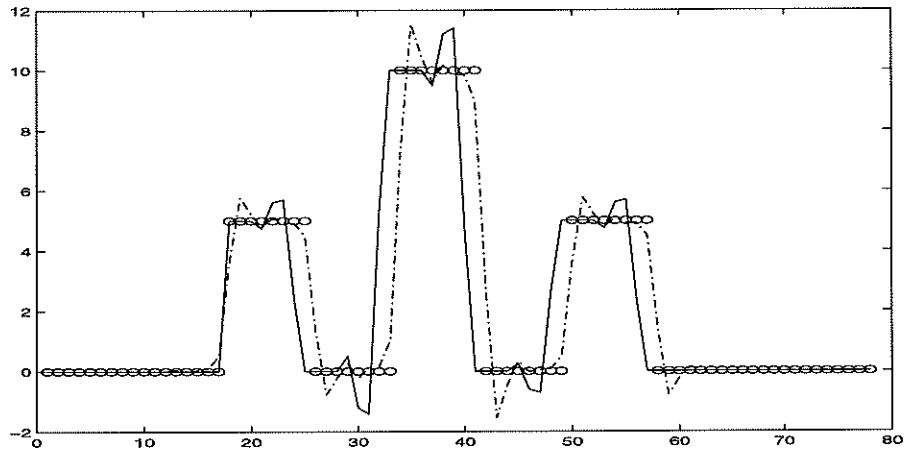

Figure 2.15: The number of data points between every two consecutive jumps is exactly 8, the DSP is invalid in this case, the ENO-DB6 approximation contains oscillations. But the error is comparable to that of the standard DB6.

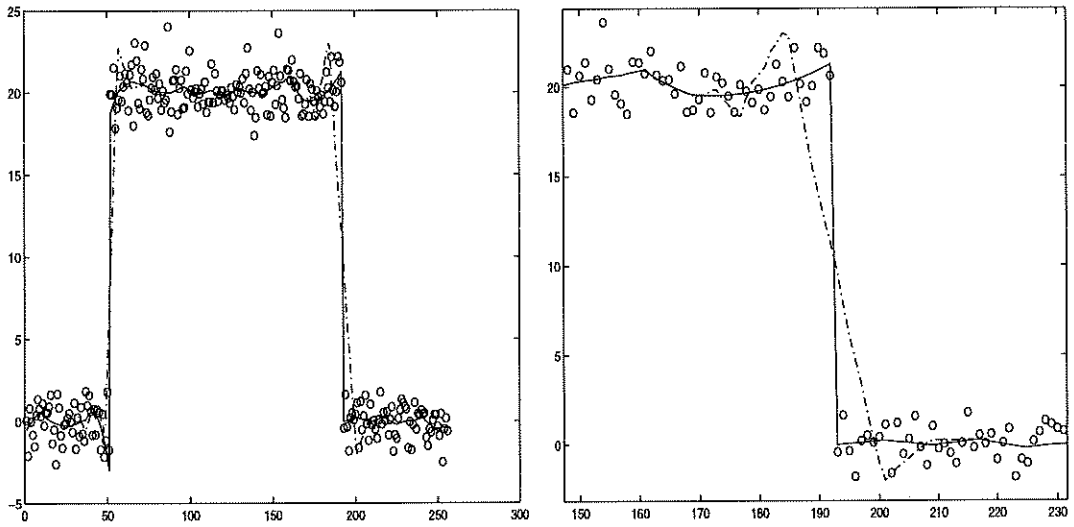

Figure 2.16: Left: The comparison of the 3-level ENO-DB6 approximation (solid line) with the standard DB6 approximation (dash-dotted line) for noisy initial data (circles). The ENO-DB6 approximation retains the sharp jumps but the standard DB6 approximation does not (right picture). Right: A zoom-in of the left example at a discontinuities.

approximation (solid line) still retains the sharp edges (see zoom-in in the right picture in Fig. 2.16) compared to the standard DB6 approximation (dash-dotted line) which not only has oscillations at the discontinuities but also smears them.

Finally, we give two 2-D image compression examples to compare the standard Haar and the ENO-Haar approximations. Here we use tensor products of 1-D transforms. In the first 2-D example, the original picture is shown in Fig 2.17. Fig 2.18 is the 3-level standard Haar approximation and Fig 2.19 is the 3-level ENO-Haar approximation. Both use low frequency approximation and store the same number of coefficients ($\frac{1}{64}$ of the original data). It is clear that in the standard Haar case, the function becomes fuzzier than the ENO-Haar case. This illustrates that the ENO-Haar approximation can reduce the edge oscillations for

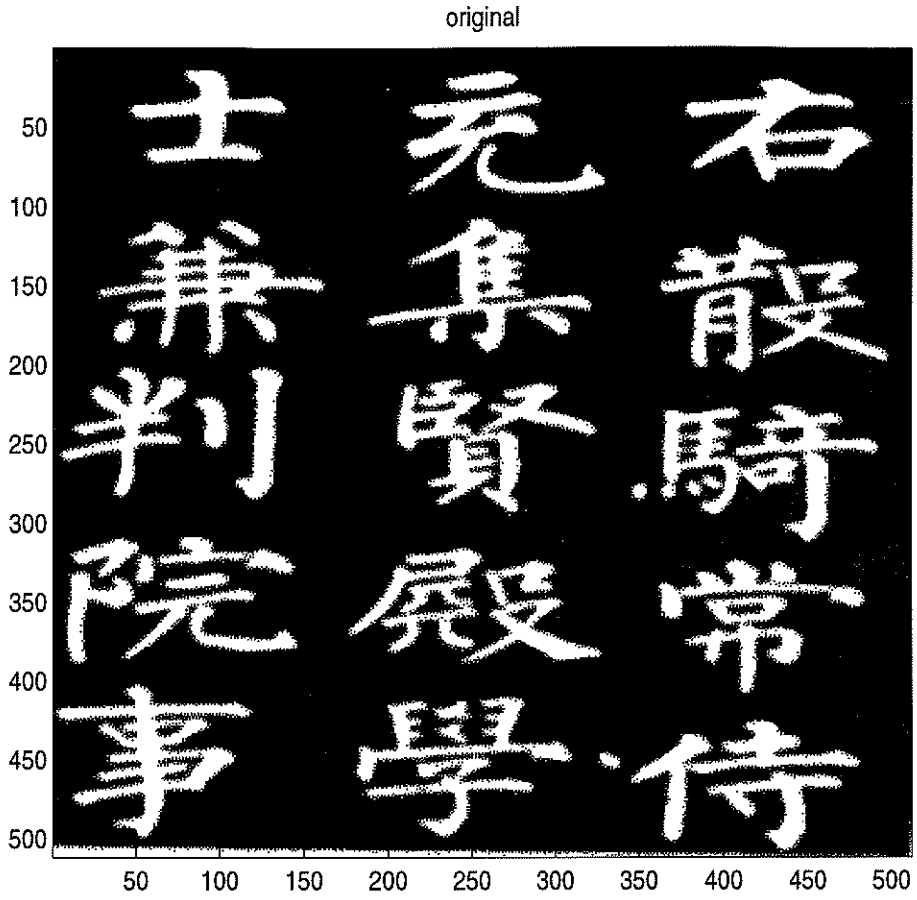

Figure 2.17: Original 2-d Function

2-D functions.

The Final 2-D exmaple is to apply the ENO-Haar wavelet transform to a standard designed testing image for compression algorithms shown in Fig 2.20. Similar to the previous example, we use 3-level standard Haar and ENO-Haar wavelet transforms, and keep the 64×64 coefficients. Fig 2.21 shows the standard Haar linear approximation which loses all details. Fig 2.22 displays the same setting ENO-Haar linear approximation. It retains much more details and keeps sharper edges than the standard Haar linear approximation. In addition, it is well known that the wavelet hard thresholding (nonlinear) approximation can

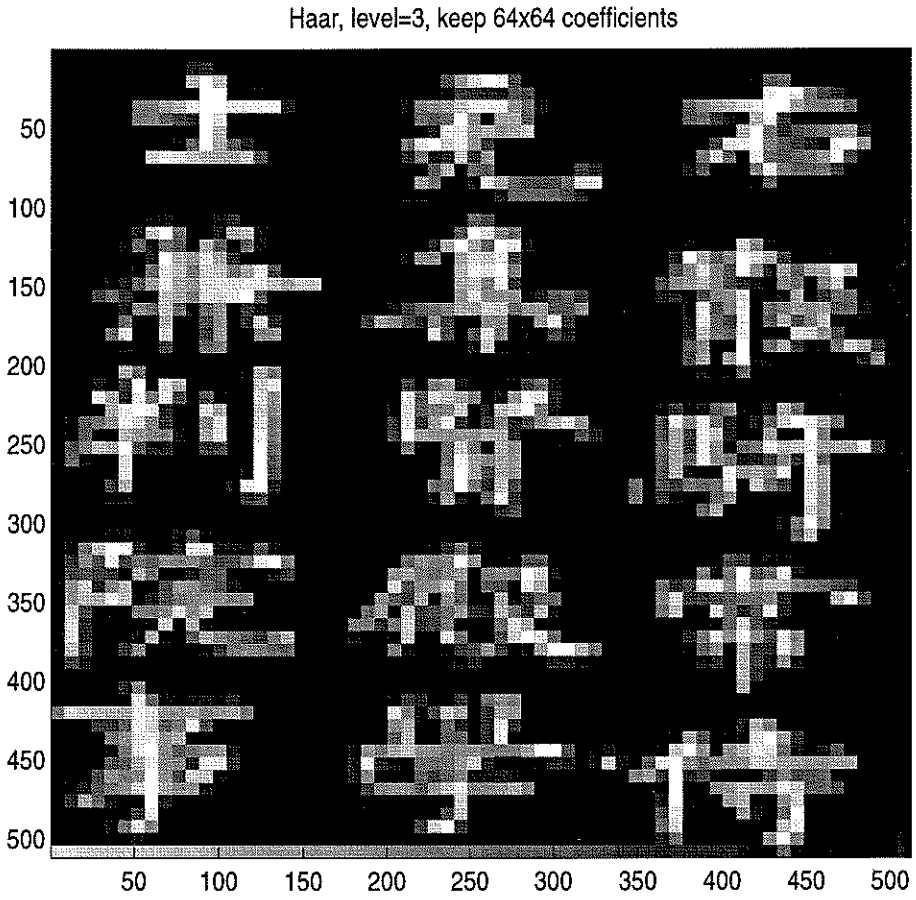

Figure 2.18: The 3-level standard Haar Approximation, the edges are fuzzier than that in the next picture.

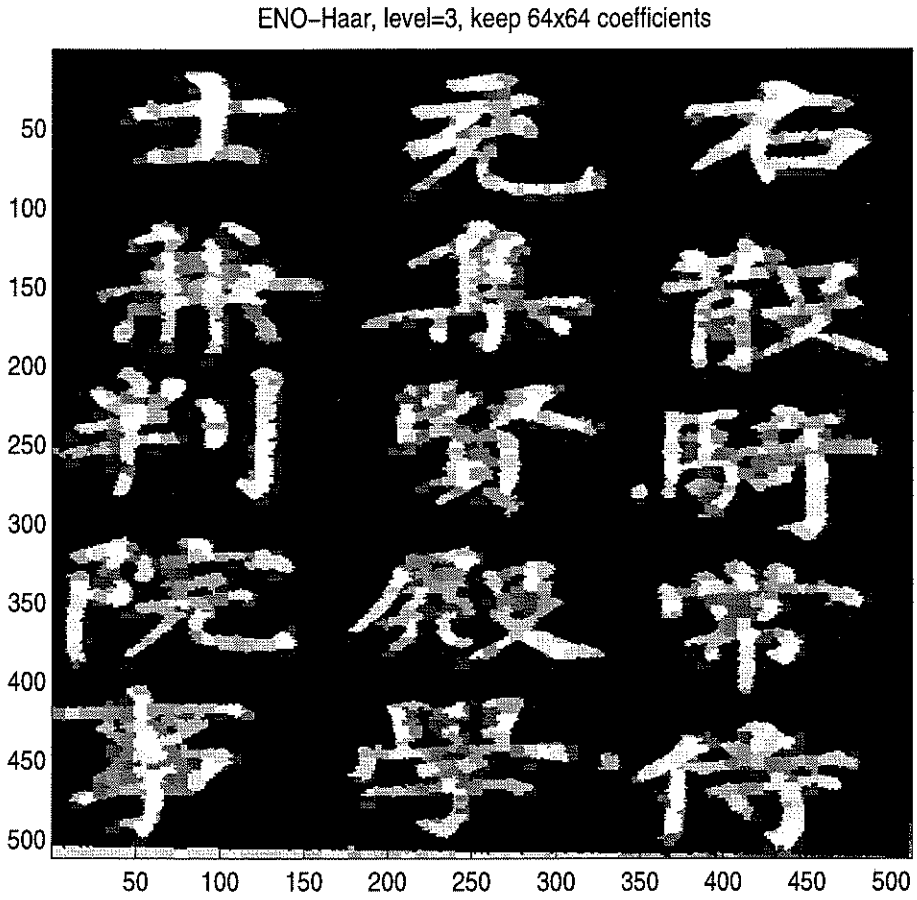

Figure 2.19: The 3-level ENO-Haar Approximation, both the edges and the interior of the characters are clearer than that in the standard Haar approximation.

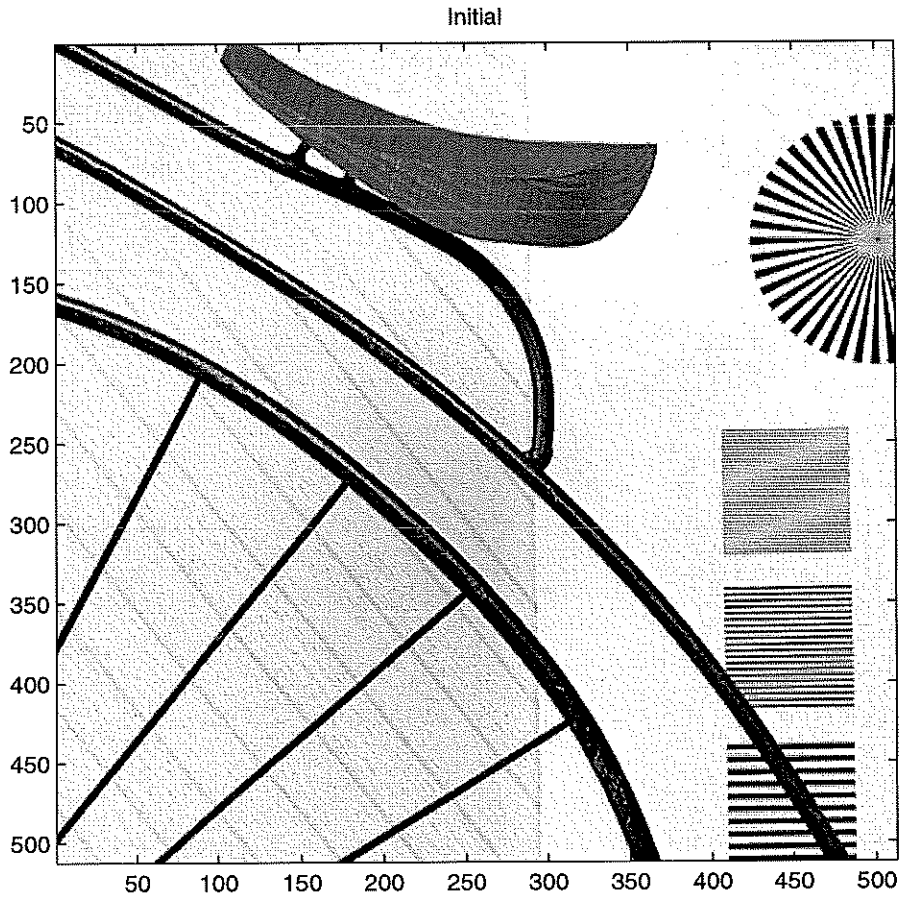

Figure 2.20: The original 2-D testing image

reconstruct more details and better edges, we show the standard hard thresholding approximation image by retaining the largest 48×48 coefficients in Fig 2.23. Similarly, we can apply the same thresholding techniques to the ENO-wavelet transforms. In this example, we give the compressed image by using ENO-Haar hard thresholding technique in Fig 2.24. In this image, more details have been recovered and the edge artifacts are much less severe.

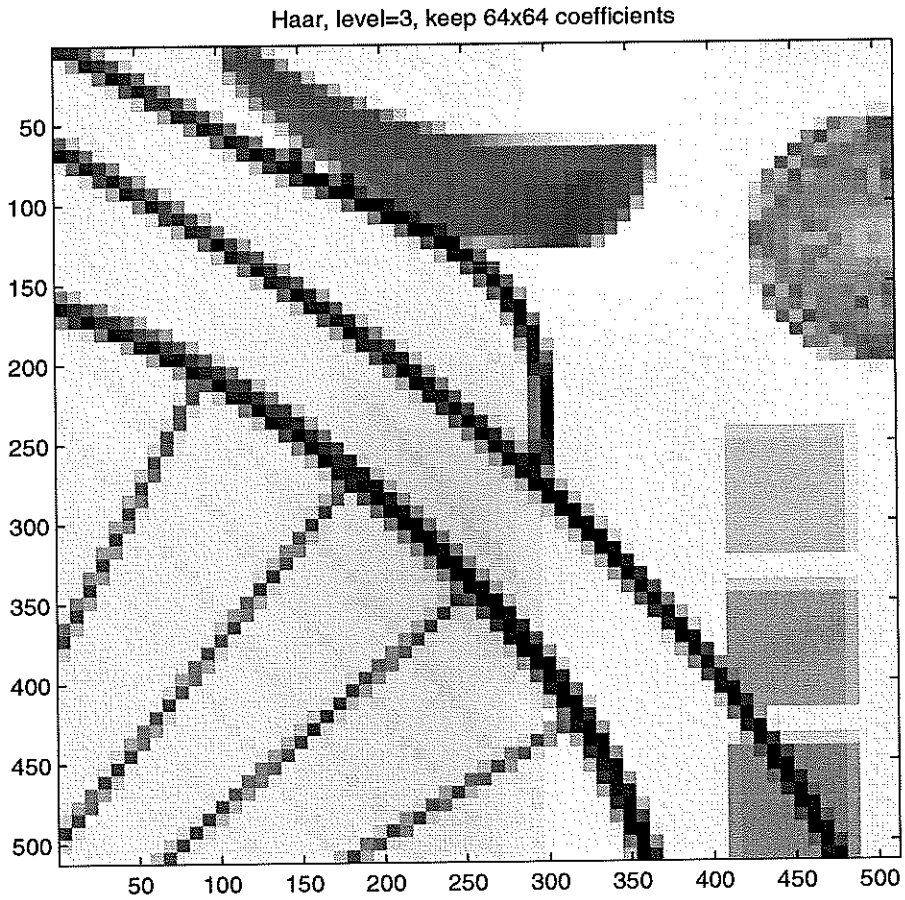

Figure 2.21: The 3-level standard Haar approximation, the edges are fuzzier than that in the next picture. Most of the details are lost.

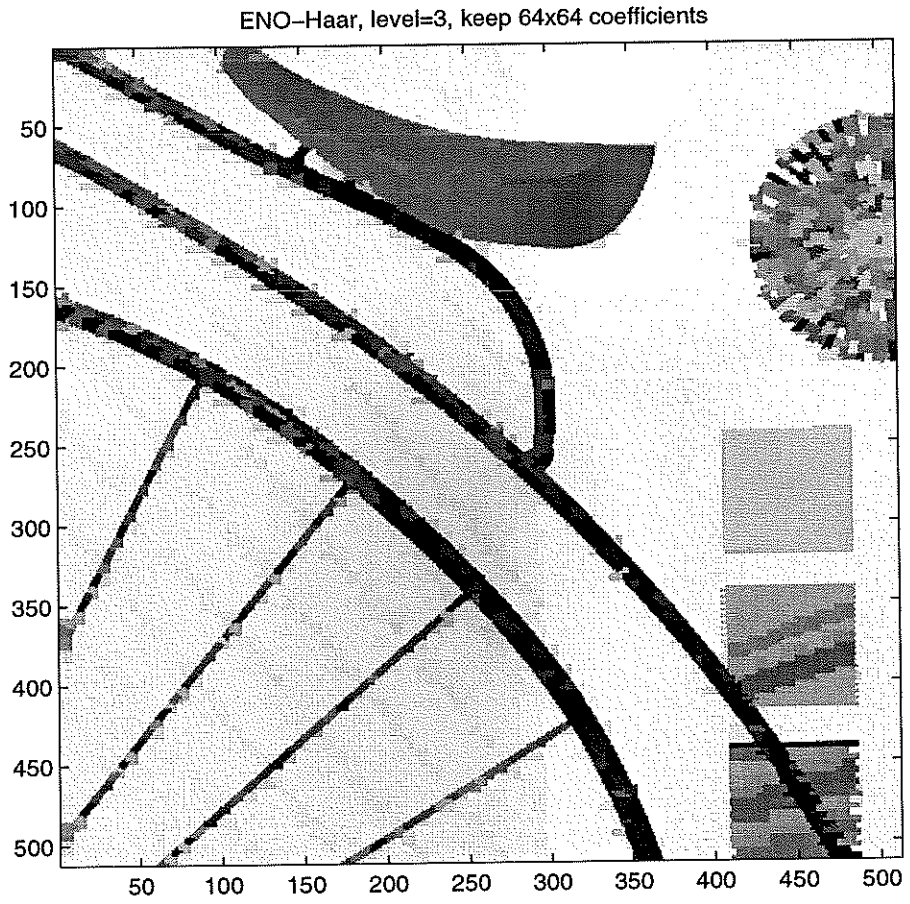

Figure 2.22: The 3-level ENO-Haar approximation, both the edges and the interior of the features are clearer than that in the standard Haar approximation. More details have been shown.

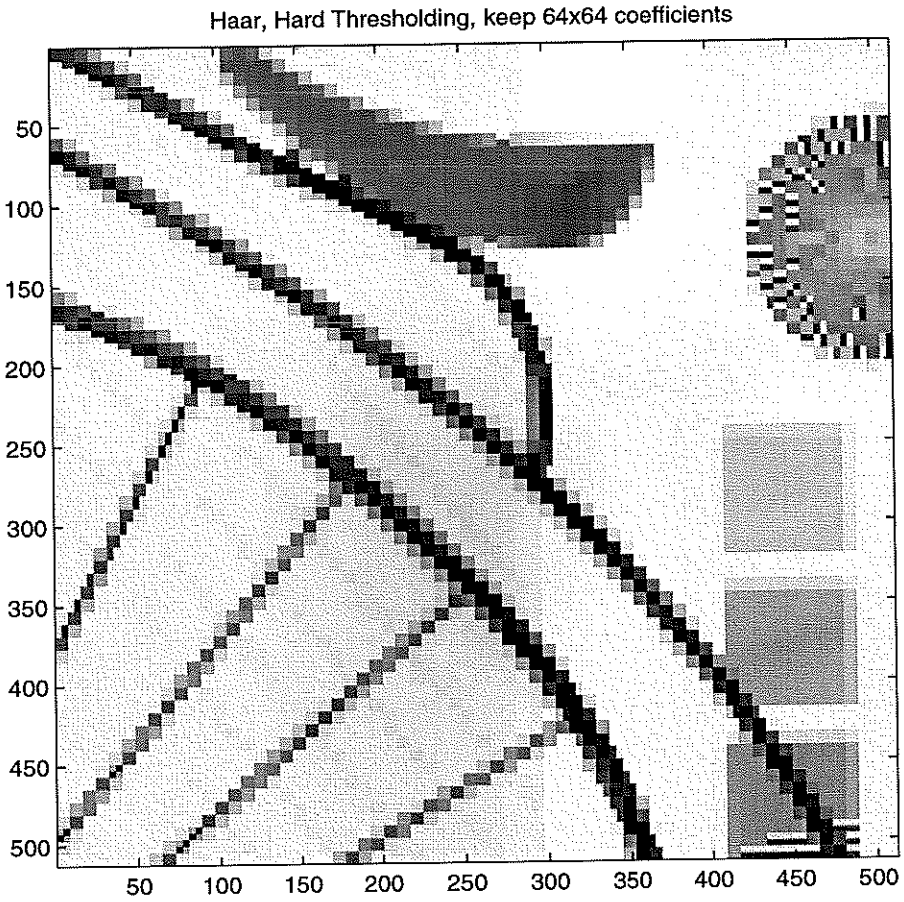

Figure 2.23: The 3-level standard Haar hard thresholding approximation, More details are preserved than the standard linear approximation. Also the edge artifacts are less severe than the standard linear approximation. On the other hard, the picture is just comparable to the ENO-Haar linear approximation.

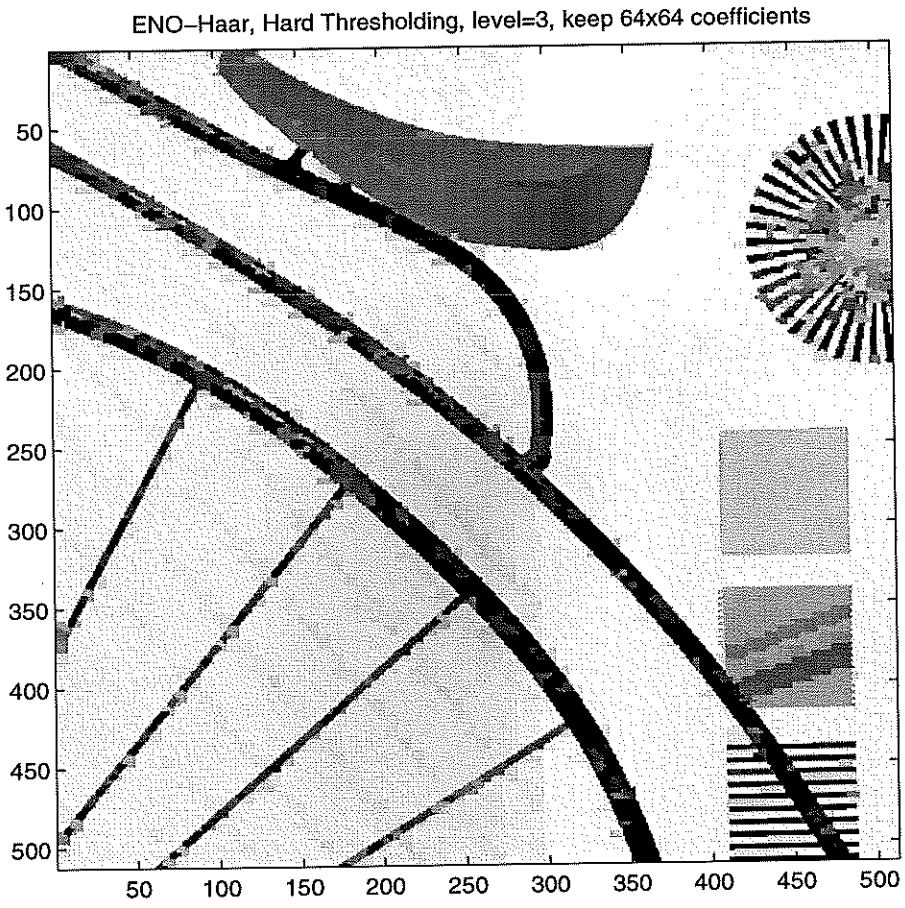

Figure 2.24: The 3-level ENO-Haar hard thresholding approximation, Much more detailed features are retained and less severe edge artifacts are generated comparing to the previous images.

CHAPTER 3

Total Variation Improved Wavelet Thresholding and Its Application in Image Processing

3.1 Introduction

In this chapter, we are concerned with the suppression of edge artifacts caused by wavelet thresholding in digital image denoising and compression. As we have described in the previous chapter, that wavelet thresholding, including linear (i.e. truncating the high frequencies) and nonlinear thresholding (i.e. retaining large coefficients,) may generate oscillations near discontinuities, especially when the functions or the images contain high level noise. This Gibbs' phenomenon is the primary reason for edge artifacts in digital image processing.

Many methods have been proposed to overcome this problem. Donoho's soft thresholding truncates wavelet coefficients on different scale levels subject to different thresholds [19]. Another type of approach also due to Donoho is to a construct special basis for discontinuities; such as wedgelets [20], ridgelets [21], and curvelets [14]. A different approach is to modify the wavelet transforms so that fewer large high frequency coefficients are generated near discontinuities, resulting in fewer large coefficients truncated in the thresholding process. Along this direction, Claypoole, Davis, Sweldens and Baraniuk [16] proposed an adaptive lifting scheme which lowers the order of approximation near jumps, thus minimizing

the Gibbs' effects. We have proposed ENO-wavelet transforms which apply the one-side approximation idea of constructing Essentially Non-Oscillatory (ENO) schemes in numerical shock capturing to design adaptive wavelet transforms such that no large high frequency coefficients are generated through differencing across discontinuities [11], essentially eliminating oscillations in the reconstructed images.

In this chapter, we propose an alternative method, which uses other Partial Differential Equation (PDE) techniques, especially PDE's from variational principles, to reduce the oscillations in wavelet thresholding approximations. Our goal is to use a variational framework, in particular, the techniques for minimizing total variation (TV), to select and modify the retained standard wavelet coefficients so that the reconstructed images have fewer oscillations near the edges.

It has been shown through many simulations in the literature that the TV model can effectively suppress noise while retaining sharp edges in images ([39], [10], and [5]). Chambolle, DeVore, Lee and Lucier [6] attempted to use wavelet based variational forms to accomplish compression and denoising. Using wavelet coefficients, they compute the best fitting of the observed images subject to minimizing certain norms in Besov spaces, which are close to the Bounded Variation (BV) space corresponding to the TV norm. An essential difference between the Besov spaces and the BV space is that Besov spaces do not admit the discontinuous functions. Therefore, sharp edges are unaviodably smoothed out in the reconstructed images.

Our first attempt is combining TV norm denoising model with the standard wavelet compression algorithms [12]. we demonstrate that compressing TV denoised images may produce higher ratio compression and better quality than denoising and compressing the images by directly using wavelets.

On the other hand, we notice that edge oscillations caused by standard wavelet thresholding significantly increase the TV norm of the reconstructed images. All this motivates us to select and modify the nonzero wavelet coefficients in the thresholding procedure subject to minimizing the TV norm of the reconstructed images so that they can produce fewer edge artifacts while retaining sharp edges.

In general, minimizers of such variational problems can be found by solving their associated Euler-Lagrangian equations, which are PDE's. In particular, the PDE produced by the TV minimization problem is highly nonlinear and usually degenerate at flat regions. Many works have been advocated to speed up the solvers in physical space, for instance, see [45], [8] and [37]. In the present work, we deduce the corresponding PDE's in wavelet space and solve them in analogous ways. We will discuss some aspects of the numerics in this paper as well.

The above described method can be easily embedded into a image compression framework by simply replacing the standard wavelet thresholding step by TV regularized wavelet thresholding . The produced non-zero wavelet coefficients can then be forwarded for quantizing and coding in the standard ways. In this situation, at the reconstruction end, the standard wavelet procedure will automatically restore the images with fewer edge artifacts. We will concentrate on selecting and modifying the non-zero wavelet coefficients subject to minimizing the TV norm of the reconstructed images, and we will not consider the quantization and coding steps. In addition, the ideas introduced here can also be used as a post-processing technique for the reconstructed images so that it can suppress the edge oscillations generated in the compression process.

The chapter is arranged in the following way. In section 3.2, we discribe the procedure of combining TV norm denoising with the standard wavelet compression algorithms [12]. In section 3.3, we give the general TV regularized wavelet

compression model for wavelet thresholding. In section 3.4, we study some relaxations of the general TV regularized model and their associated PDE's. In section 3.5, we discuss some numerical aspects of solving these PDE's. And in section 3.6, we show some examples to illustrate the results of the models.

3.2 TV Denoising Followed by Standard Wavelet Compression

In this section, we describe our first attempt in combining TV denoising model with the standard wavelet compression algorithms.

3.2.1 Wavelet Thresholding

Suppose we are given an observed image $z(x) = u_0(x) + n(x)$, where $u_0(x)$ is the original noise free image and $n(x)$ the Gaussian white noise with $\|n(x)\|_2 = \sigma$. Let us denote the standard orthonormal wavelet transform of $z(x)$ by:

$$z(\vec{\alpha}, x) = \sum_{j,k} \alpha_{j,k}\phi_{j,k}(x), \tag{3.1}$$

where $\phi_{j,k}(x)$ are wavelet basis functions and $\vec{\alpha} = \{\alpha_{j,k}\}$ the corresponding coefficients defined by

$$\alpha_{j,k} = \int z(x)\phi_{j,k}(x)dx. \tag{3.2}$$

One way to describe the wavelet thresholding technique is to prescribe a wavelet coefficient index set I, then retain all coefficients with indices belong to I and truncate the other coefficients to zero:

$$\alpha^*_{j,k} = \begin{cases} \alpha_{j,k} & (j,k) \in I \\ 0 & otherwise \end{cases}$$

For example, in linear thresholding, I is taken as the set of low frequencies; and in hard thresholding, I is defined as the set of all coefficients whose magnitudes are larger than a given tolerance, otherwise, it is smaller than the tolerance. Since orthonormal wavelets form an orthonormal basis of the L^2 space, it is obvious that the hard thresholding selection of I minimizes the L^2 error between the compressed image $u(x)$ and the observed image $z(x)$.

The hard thresholding approximations introduce oscillations at the edges, although they are optimal in the L^2 space. This is due to the fact that the L^2 norm minimization does not penalize oscillations. In Fig 3.1, we show a 4-level Daubechies-6 (DB6) wavelet hard thresholding approximation (dash-dotted) to a discontinuous function (dotted). The approximation is reconstructed by retaining the largest 64 non-zero coefficients and truncating the other coefficients to zero. It is obvious that it generates severe oscillations at each jump. Fig 3.2 is a 2-D image containing four noisy squares with different sizes and intensities. We show its 4-level DB6 wavelet hard thresholding approximation in Fig 3.3. The approximation contains edge artifacts along the boundaries of the objects, while in the observed image, these objects have sharp edges.

3.2.2 TV Denoising Model

The objective of TV denoising is to compute a denoised image by solving the following constrained minimization problem proposed by Rudin, Osher and Fatemi [39]:

$$\min_u TV(u) \equiv \int |\nabla u| dx, \tag{3.3}$$

subject to

$$||u - u_0||^2 = \sigma^2. \tag{3.4}$$

Here σ is the standard deviation of the noise $\eta(x, y)$.

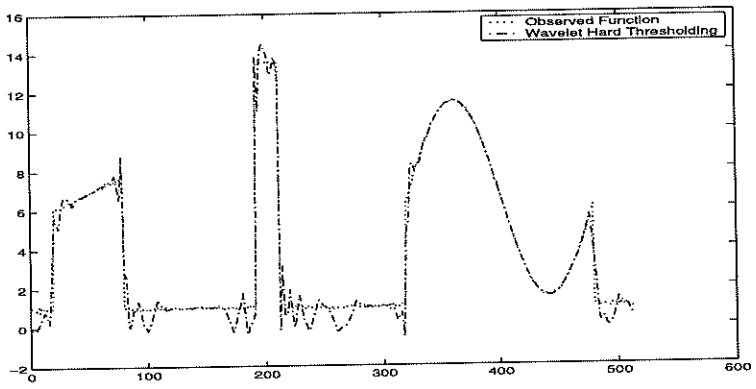

Figure 3.1: The observed function (dotted) has large jumps. The 4-level DB6 wavelet hard thresholding approximation (dash-dotted) is reconstructed by retaining the largest 64 coefficients. It generates oscillations at each jump.

Intuitively speaking, the TV norm measures the jumps of the images, no matter whether they are continuous or discontinuous. An important feature of TV norm denoising is that it does not penalize discontinuities in the image. For example, in Fig 3.4, the four functions have the same TV norm. Therefore, by solving the minimization problem (3.3) and (3.4), the data automatically determines whether a smooth function or a discontinuous function best models the restored image. It is important to note that it doesn't artificially over sharpen the edges. Hence, the most significant advantage of TV denoising is that while smoothing out the high frequency oscillations (corresponding to noise) in the image, it preserves the edges systematically [8], [43] and [12].

3.2.3 Wavelet Compression of TV Denoised Images

For very noisy images, denoising and compression using wavelet thresholding may not work well since the features of the images may be lost while removing the noise. The reason is that wavelet decomposition cannot efficiently distinguish

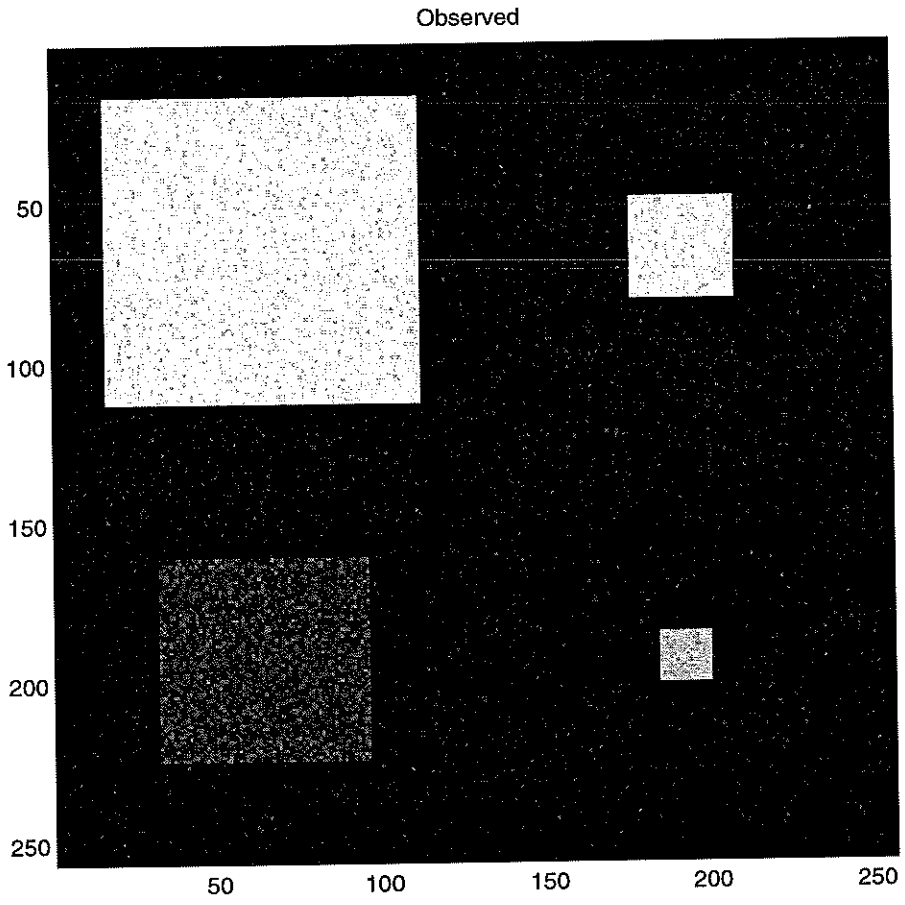

Figure 3.2: The observed image has features with sharp edges despite of the present of noise.

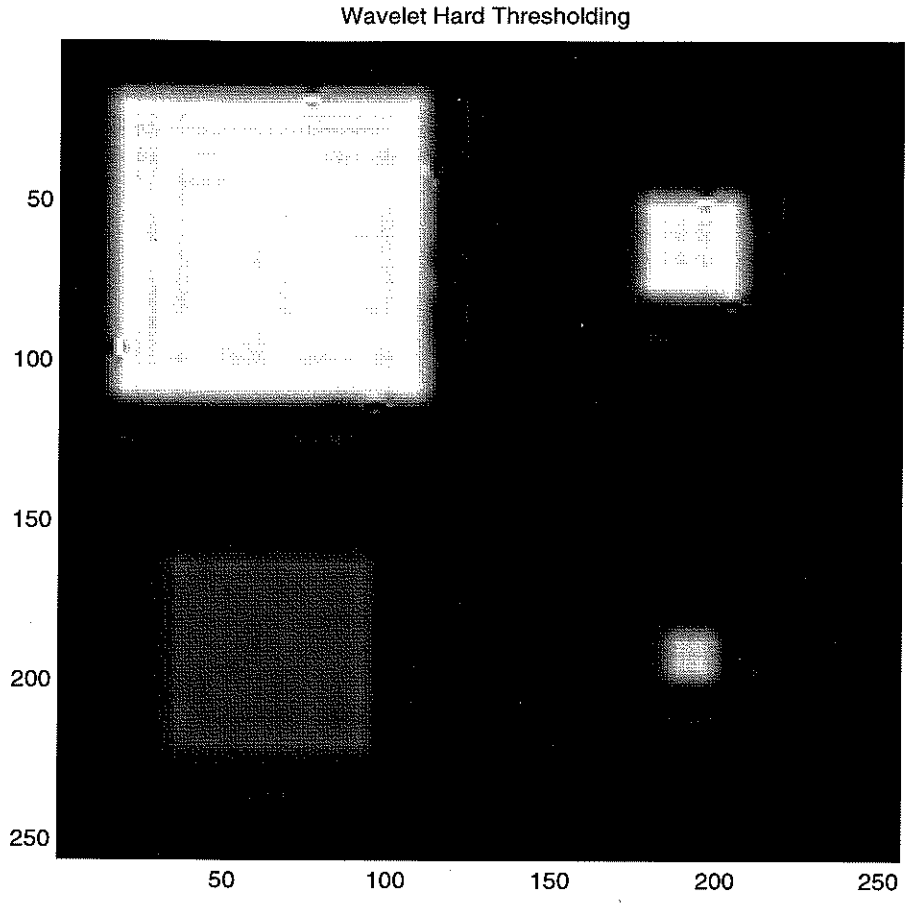

Figure 3.3: The 4-level DB6 wavelet hard thresholding reconstruction which retains the largest $16x16$ coefficients. Edge artifacts are clearly seen along the boundaries.

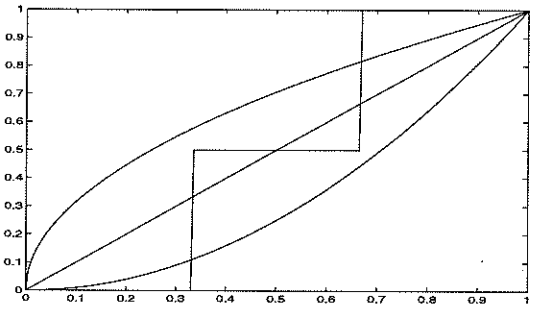

Figure 3.4: The four functions in graph have the same TV norm.

edges from noise, because they both generate high frequencies. We propose using the TV norm denoising method, followed by wavelet compression so that we can obtain high ratio compression which still keeps the features. Numerical experiments in [12] show that TV-denoising can bring more wavelet coefficients closer to zero than that of the original noisy image, thus making the compression more efficient, i.e., compared to the standard wavelet compression, this approach can get better quality images with the same compression ratio, or it can obtain higher compression ratio for similar image quality.

This approach cannot remove the edge artifacts generated by wavelet thresholding as shown in Fig 3.1 and 3.2, although it improves the compression results. It is simply because the thresholding is done after the TV denoising. This leads us to embed the TV norm into the wavelet compression algorithms so that it can suppress the oscillations generated by the thresholding as well as the noise while retaining the sharp edges in the compressed images.

3.3 General TV Regularized Wavelet Compression Model for Wavelet Thresholding

In this section, we give our TV regularized wavelet compression model for suppressing the oscillations generated by wavelet thresholding.

As we have shown in the previous section, wavelet thresholding can cause oscillations near edges, consequently increasing the TV norm of the reconstructed image. To suppress these oscillations, we propose the following model to select the index set I, and modify the values of the retained wavelet coefficients $\beta_{j,k}$ such that the reconstructed image $u(\vec{\beta}, x)$ form a less severe oscillatory approximation:

$$\min_{\beta_{j,k},(j,k)\in I} F(u,z) = \lambda \int |\nabla_x u(\vec{\beta}, x)| dx + \frac{1}{2}\|u - z\|_2^2 \tag{3.5}$$

$s$ubject to

$$|I| = m, \tag{3.6}$$

where $u(\vec{\beta}, x)$ has wavelet transform:

$$u(\vec{\beta}, x) = \sum_{j,k} \beta_{j,k} \phi_{j,k}(x).$$

Here we have $\beta_{j,k} = 0$ if $(j,k) \notin I$, $|I|$ represents the number of elements in I, m is a given integer, and λ the regularization parameter.

Compared to the standard TV denoising model, the two crucial points we introduce to the TV regularized wavelet compression model are the integer constraint and the minimization in terms of coefficients in wavelet spaces. They are essential for accomplishing the compression task.

The first term in the objective functional reduces the oscillations of $u(x)$ by diminishing its TV norm. The second term is the standard L^2 fitting term which controls the difference between $u(x)$ and the observed image $z(x)$. The regularization parameter λ is used to balance the trade-off between the suppression of oscillations and the fitting term. When λ tends to zero, $u(x)$ goes to the standard hard thresholding approximation. On the other hand, when λ tends to infinity, the suppression term dominates the objective functional, and therefore $u(x)$ tends to a constant. As a TV regularization parameter, λ also controls the smallest scale of features which are preserved [43], i.e. for a given value of λ, there exists a size of feature such that the model treats all features smaller than this size as oscillations and eliminates them, while preserving features which are bigger than this critical scale. In practice, λ can be determined in many ways, for instance, using the L-curve technique [30] to select the best λ, or determining it by using a set of training images. In this paper, we do not discuss these approaches in detail, though we use the latter choice to select λ in our numerical experiments.

Compared to the approach discussed in the previous section, which uses the TV denoising method followed by standard wavelet thresholding to obtain high ratio compression for noisy data, the advantage of the proposed TV regularized wavelet compression model is that the TV regularized model can reduce the oscillations generated by wavelet thresholding as well as the noise, while TV denoising followed by standard thresholding may generate new oscillations after denoising. Also, the TV regularized wavelet compression model can directly work on wavelet coefficients, making it easier to be fit into practical compression schemes, especially for images given in a wavelet coefficient format (e.g. the upcoming wavelet based JPEG 2000 compression standard). In addition, the TV regularized model operates on a smaller number of coefficients (in the hard thresholding case). Potentially, it could be faster than TV denoising followed by standard thresholding.

Remark: The TV regularization term in the model can be replaced by the H_1 regularization term $\|\nabla u\|_2^2$, or other regularization terms. Compared to the TV term, these other norms usually smooth out sharp edges in the reconstructed images. We will show a comparison in our numerical experiments in section 3.6.

The TV regularized model for wavelet thresholding is a nonlinear integer optimization problem which in general cannot be solved efficiently. There are two crucial tasks in finding the global minimizer of (3.5): selecting the index set I; and modifying the retained coefficients $\beta_{j,k}$, $(j,k) \in I$. The major difficulty is in the selection of the index set I, because there are too many combinations for possible I. In fact, for each selected I, which forms a subspace of the L^2 space, there is a local minimizer in this subspace. And the global minimizer is among these local minimizers. On the other hand, both the magnitude and the location of wavelet coefficients reflect the significance of corresponding features,

although they may not necessarily determine the index set I which contains the global solution. So, we can use them to approximate the optimal set I. We will consider several relaxations of such approximations in the next section. After the set I has been determined, finding the minimizer in such a subspace becomes a convex unconstrainted optimization problem. We will address some numerical methods to solve these problems in section 3.5.

3.4 Relaxations of the TV Regularized Wavelet Compression Model

In this section, we consider several relaxations of the TV regularized wavelet compression model and give their Euler-Lagrangian equations. The purpose is to reduce the difficulty in determining the index set in the integer constraint (3.6).

3.4.1 The Standard Hard Thresholding

A simple way to select the index set I is to choose it according to the magnitude of the coefficients. In other words, we can simply use the standard hard thresholding nonzero coefficient index set I_H to approximate I. Then the TV model is simplified as:

$$\min_{\beta_{j,k},(j,k)\in I_H} F(u,z) = \lambda \int |\nabla_x u(\vec{\beta},x)|dx + \|u-z\|_2^2. \tag{3.7}$$

As we mentioned before, once we restrict the index set to I_H, the minimization problem becomes convex and unconstrained, and has an unique solution $u(x)$ in this subspace. The solution $u(x)$ satisfies the Euler-Lagrangian equation in wavelet space:

$$-\lambda \int \nabla_x \left(\frac{\nabla_x u}{|\nabla_x u|} \right) \phi_{j,k}(x)dx + 2(\beta_{j,k} - \alpha_{j,k}) = 0, \quad (j,k) \in I_H. \tag{3.8}$$

3.4.2 Smooth Approximations of the Constraint

Another way to relax the integer constraint $|I| = m$ is to approximate $|I|$ by smooth functions so that we can apply standard techniques for continuous optimizations. Notice that we have the fact:

$$|I| = ||\vec{\beta}||_0,$$

where $||\cdot||_0$ is defined as the number of non-zero elements in the vector. Olshausen and Field [36] proposed using $\sum_{j,k} \log(1 + \beta_{j,k}^2)$ to approximate the 0-norm in controlling the number of non-zero patches in sparse images. Donoho [22] showed that the p-norm function $\|\cdot\|_p^p$ forms a more accurate approximation to the integer constraint.

Using these smooth approximations, we can relax the integer constraint to continuous constraints:

$$(\sum_{j,k} \log(1 + \beta_{j,k}^2) - m)^2 \leq \gamma^2, \tag{3.9}$$

or

$$(\sum_{j,k} |\beta_{j,k}|^p - m)^2 \leq \gamma^2, \tag{3.10}$$

where γ is a given small positive number used for controlling the number of non-zero components in the coefficients.

Since both approximations are smooth, we can easily convert the constrained problems to unconstrained problems by introducing the Lagrangian multiplier τ:

$$\min_{\beta_{j,k},(j,k)\in I_H} F(u,z) = \lambda \int |\nabla_x u(\vec{\beta}, x)| dx + \|u - z\|_2^2 + \tau(\sum_{j,k} \log(1+\beta_{j,k}^2) - m)^2, \tag{3.11}$$

or

$$\min_{\beta_{j,k},(j,k)\in I_H} F(u,z) = \lambda \int |\nabla_x u(\vec{\beta}, x)| dx + \|u - z\|_2^2 + \tau(\sum_{j,k} |\beta_{j,k}|^p - m)^2. \tag{3.12}$$

Then it is easy to obtain the corresponding Euler-Lagrangian equations in wavelet space as:

$$-\lambda \int \nabla_x \left(\frac{\nabla_x u}{|\nabla_x u|} \right) \phi_{j,k}(x) dx + 2(\beta_{j,k} - \alpha_{j,k}) + 2\tau (\sum_{j,k} \log(1+\beta_{j,k}^2) - m) \frac{\beta_{j,k}}{1+\beta_{j,k}^2} = 0, \tag{3.13}$$

or

$$-\lambda \int \nabla_x \left(\frac{\nabla_x u}{|\nabla_x u|} \right) \phi_{j,k}(x) dx + 2(\beta_{j,k} - \alpha_{j,k}) + 2\tau (\sum_{j,k} |\beta_{j,k}|^p - m) \frac{\beta_{j,k}}{|\beta_{j,k}|^{2-p}} = 0. \tag{3.14}$$

Remark: In these two approximations, since the log-function $\sum_{j,k} \log(1+\beta_{j,k}^2)$ and p-norm function $\sum_{j,k} |\beta_{j,k}|^p$ $(p \leq 1)$ are not convex functions, there may exist many local minimizers which are also solutions to the Euler-Lagrangian equations.

3.5 Numerics

To find the solutions for the relaxations of the TV regularized wavelet compression model, we want to solve the associated Euler-Lagrangian equations (3.8), (3.13), and (3.14). In fact, many numerical methods for similar equations in physical space have been proposed in literature, such as Rudin, Osher and Fatemi's time marching scheme [39], Vogel and Omen's fixed-point iterative method [45], and Chan, Golub and Mulet's primal-dual method [8]. All these methods can be adapted to the wavelet space. Here, we use the simple fixed-point iterative method as an example to show some numerical aspects involved in the computation.

The fixed-point iterative method linearizes the nonlinear terms in the Euler-Lagrangian equations with approximations from the previous iteration. We denote $D_{x,+}$ $(D_{x,-})$ as the forward (backward) finite differences in physical space. We discretize the $\nabla_x u$ in (3.7), (3.11) and (3.12) by the forward finite differences

$D_{x,+}$, then the fixed-point schemes to the three relaxations are:

$$-\lambda \int D_{x,-} \left(\frac{D_{x,+} u^{n+1}}{\sqrt{|D_{x,+} u^n|^2 + \epsilon_1}} \right) \phi_{j,k}(x) dx + 2(\beta_{j,k}^{n+1} - \alpha_{j,k}) = 0, \quad (j,k) \in I_H, \tag{3.15}$$

and

$$-\lambda \int D_{x,-} \left(\frac{D_{x,+} u^{n+1}}{\sqrt{|D_{x,+} u^n|^2 + \epsilon_1}} \right) \phi_{j,k}(x) dx \tag{3.16}$$
$$+2(\beta_{j,k}^{n+1} - \alpha_{j,k}) + 2\tau(\sum_{j,k} \log(1 + (\beta_{j,k}^n)^2) - m) \frac{\beta_{j,k}^{n+1}}{1 + (\beta_{j,k}^n)^2} = 0,$$

and

$$-\lambda \int D_{x,-} \left(\frac{D_{x,+} u^{n+1}}{\sqrt{|D_{x,+} u^n|^2 + \epsilon_1}} \right) \phi_{j,k}(x) dx \tag{3.17}$$
$$+2(\beta_{j,k}^{n+1} - \alpha_{j,k}) + \tau(\sum_{j,k} |\beta_{j,k}^n|^p - m) \frac{\beta_{j,k}^{n+1}}{(|\beta_{j,k}^n| + \epsilon_2)^{2-p}} = 0,$$

respectively, where $u^n = u(x, \vec{\beta}^n)$, ϵ_1 and ϵ_2 are small positive numbers which are used to prevent blow-up in regions where $\nabla u = 0$, or $\beta_{j,k} = 0$. Equations (3.15), (3.16) and (3.17) are linear equations in the unknowns $\beta_{j,k}$, which we solve by Conjugate Gradient (CG) without preconditioning.

We note that the unknowns $\beta_{j,k}^{n+1}$ are in wavelet space but the finite difference operators $D_{x,+}$ and $D_{x,-}$ are defined in physical space. To compute them, we need to transform the data from wavelet space back to physical space. Then after calculating the finite differences, we transform the data back to wavelet space.

Remark: On the other hand, since wavelet transforms are local, we can in fact directly compute the finite difference terms locally in wavelet space so that it is not necessary to transform the data back and forth between the two spaces. In our numerical examples, we do not use this method, so we will not discuss it in detail.

3.6 Examples

In this section, we will show some 1-D and 2-D examples to demonstrate the improvement in images of the TV regularized wavelet compression models for wavelet thresholding. In all computations, we use the fixed-point schemes (3.15), (3.16) and (3.17) introduced in section 3.5. We choose the parameters ϵ_1 and ϵ_2 as 10^{-8}, and $\tau = 10^{-5}$ in all situations.

In the first 1-D example, we show a sequence of images (Fig 3.6 to 3.9) which are computed by hard thresholding relaxation (3.15) of the observed function shown in Fig 3.5. We use the 4-level Daubechies-4 (DB4) wavelet transform, and take $\lambda = 0.001, 0.01, 0.1$, and 1 respectively. In each picture, we show the standard wavelet hard thresholding approximations (dotted line) restored by retaining the largest $m = 50$ coefficients (with respect to the original 519 coefficients), and the TV regularized wavelet compression approximations (solid) which are also reconstructed by these 50 non-zero coefficients with perturbed values. We notice that when λ tends to 0, the TV regularized model results are getting closer to the standard wavelet hard thresholding approximation which has more oscillations at the discontinuities. As λ increases, the TV norm of the reconstructed images decrease, specifically the oscillations at discontinuities are suppressed. When $\lambda = 1$, the reconstructed images are almost flat. All features of the observed image have been lost. This reflects the domination of the TV norm regularization over the fitting term.

More importantly, from these figures, we notice that when $\lambda = 0.01$, the TV regularized approximation has fewer oscillations than that of standard hard thresholding but still retains the silent features of the image and keep the sharp edges. For all images generated using larger λ, some features have been altered, while on the other end, for all images generated using smaller λ, the edge artifacts

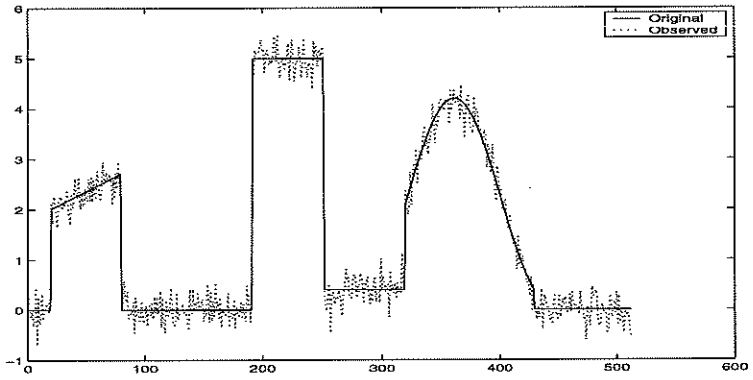

Figure 3.5: The observed function (dotted) and the original noise free function (solid).

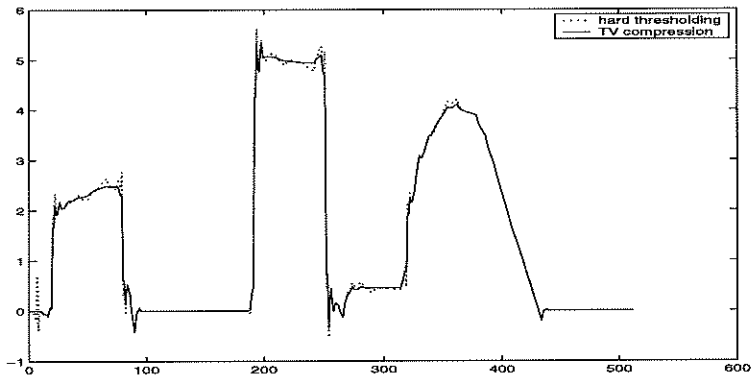

Figure 3.6: Left: $\lambda = 0.001$, As λ becomes smaller, the TV norm approximation (solid) tends to the hard threshold approximation.

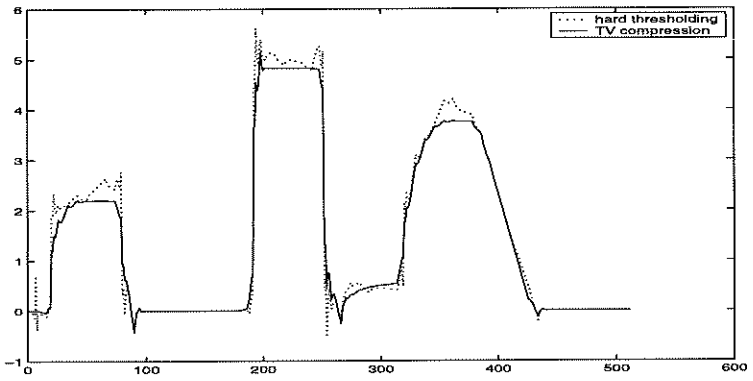

Figure 3.7: $\lambda = 0.01$, The TV norm reconstructed approximation (solid) has fewer oscillations at the discontinuities than that of the wavelet hard thresholding approximation (dotted).

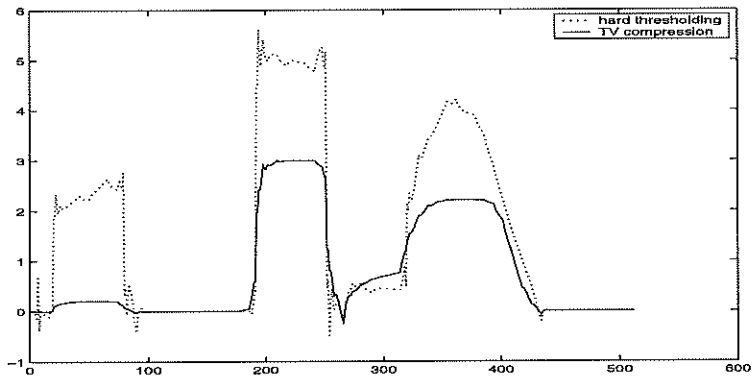

Figure 3.8: $\lambda = 0.1$, The TV norm hard thresholding approximation (solid) smooths the oscillations out but also alters the features.

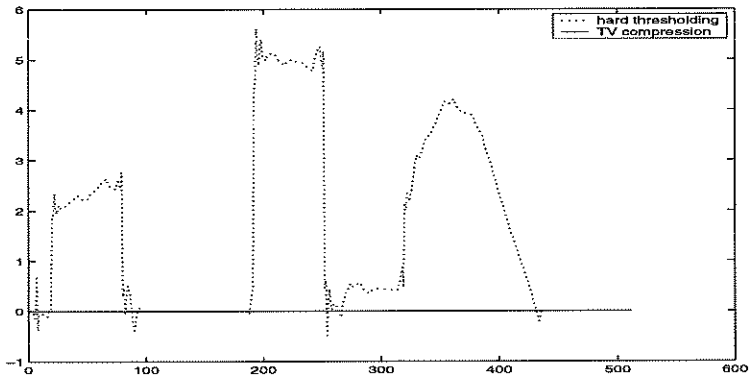

Figure 3.9: $\lambda = 1$, The TV reconstructed approximation (solid) is almost a straight line which indicates that the regularization term dominates the objective functional. All features have been eliminated.

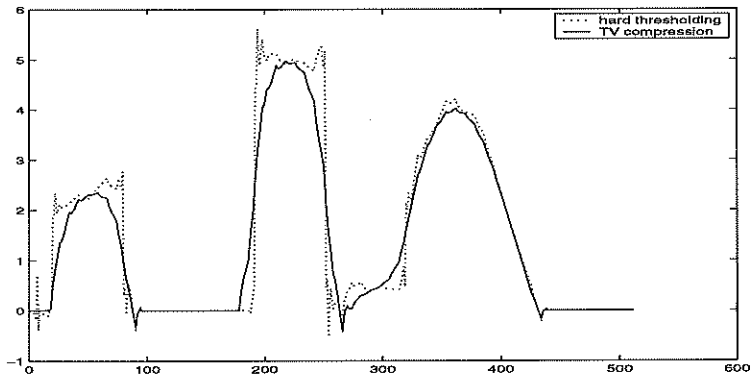

Figure 3.10: $\lambda = 0.0002$, the H-1 regularization approximation (solid) smooths all sharp edges in the reconstructed image.

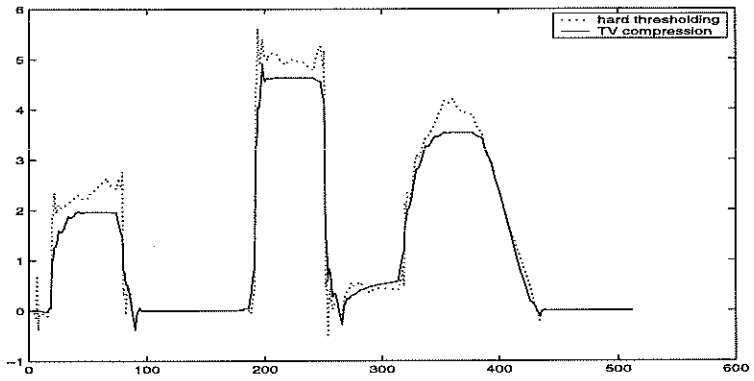

Figure 3.11: $\lambda = 0.02$, TV norm hard thresholding approximations (solid) have fewer oscillations at discontinuities. It keeps all features.

are still significant, although all features are preserved.

For the next 1-D example, we display in Fig 3.10 the reconstructed image using the H_1 regularization $||\nabla u||_2^2$ term instead of the TV term in (3.3) with $\lambda = 0.0002$. The approximation (solid) is also reconstructed by using $m = 50$ non-zero coefficients. Comparing to the previous examples, it is obvious that H_1 regularization smears all sharp edges because it doesn't allow the existence of discontinuities.

The next 1-D example shows another sequence of images (Fig 3.11 to 3.13) which correspond to the TV regularized wavelet compression models with the hard thresholding (3.7), the log function (3.11), and the p-norm (3.12) approximations of the noisy image in the previous example with $\lambda = 0.02$. They are computed by the fixed-point scheme with 10 iterations. All of them are restored by keeping $m = 50$ non-zero wavelet coefficients. The three approximations retain the features of the observed function and have less severe edge oscillations at the discontinuities. Among them, the p-norm approximation fits the original function better than the two others in this example, although all are very similar.

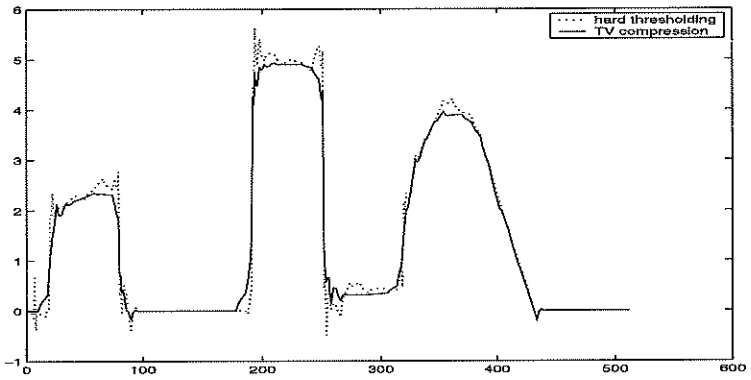

Figure 3.12: $\lambda = 0.02$, The TV norm log function reconstructed approximation (solid) has fewer oscillations at the discontinuities than that of the wavelet hard thresholding approximation (dotted). It keeps all features as well.

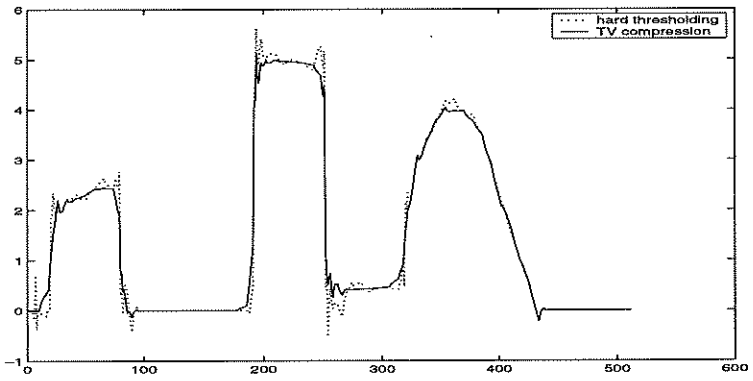

Figure 3.13: $\lambda = 0.02$, the p-norm approximation (solid) keeps all features as well as eliminating most of the edge oscillations.

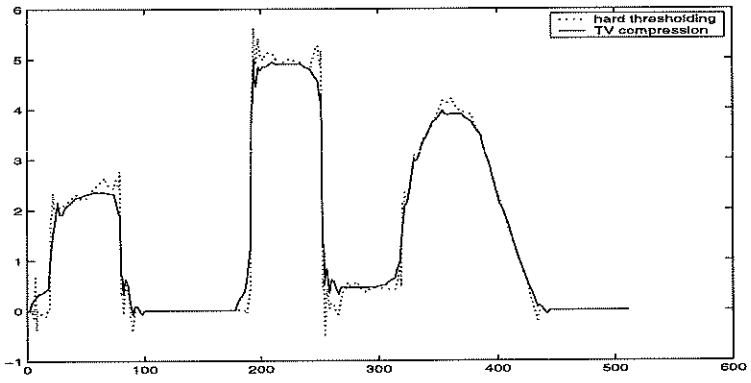

Figure 3.14: The approximation obtained by TV denoising followed by standard hard thresholding.

We also compare the results of this TV regularized wavelet compression models with that of the procedure we described in section 3.2 which is to denoise the image by the TV denoising model first, and followed by the standard hard thresholding compression (shown in Fig 3.14). The results are very similar to the approximations obtained in the previous example except that in the result of TV denoising followed by standard hard threshold, there exist more edge oscillations (at the right jump of the first bump) than the results of the TV regularized wavelet compression shown in the previous examples.

In the last 1-D example, we show another comparison of approximations obtained by different numbers of fixed-point iterations. The pictures in Fig 3.15 to refiteration3 are calculated using $5, 10$, and 20 fixed-point iterations of (3.15) with $\lambda = 0.01$ respectively. The approximations are very close. This illustrates that the fixed-point scheme converges fast in the first few iterations.

The next example is for a 2-D comparison of the standard hard thresholding and the TV regularized wavelet compression model images. As shown in Fig 3.3), the standard 4-level DB6 wavelet compression by retaining the largest $m = 16\times16$ coefficients (the ratio of compression is 256:1) has obvious edge artifacts along the

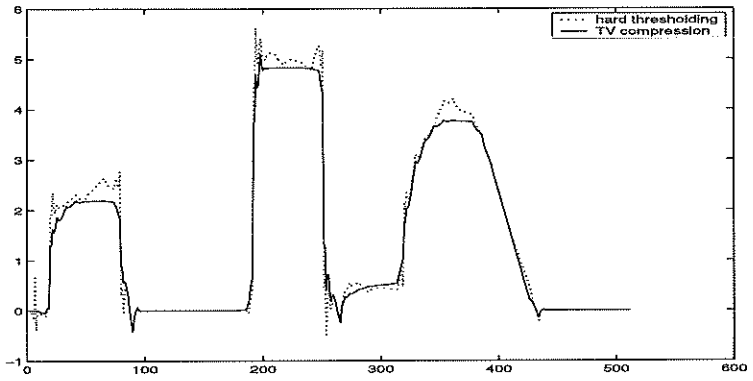

Figure 3.15: $\lambda = 0.01$, TV hard thresholding reconstruction after 5 fixed-point iterations.

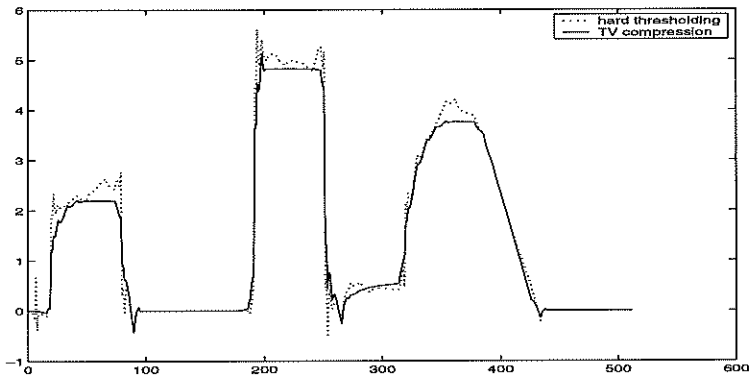

Figure 3.16: $\lambda = 0.01$,TV hard thresholding reconstruction after 10 fixed-point iterations. The difference between it and the result shown in Fig 3.15 are also invisible.

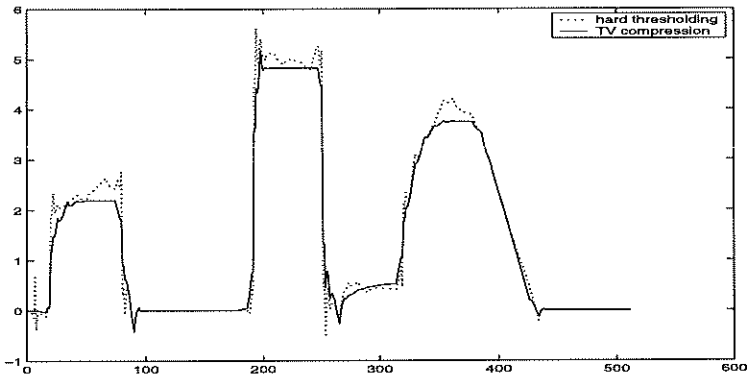

Figure 3.17: $\lambda = 0.01$, TV hard thresholding reconstruction after 20 fixed-point iterations. The difference between it and the results shown in Fig 3.15 and 3.16 is also invisible.

boundaries of the objects. In Fig 3.18, the image is obtained by solving the TV regularized hard thresholding approximation (3.8) with $\lambda = 0.05$. We perturb the values of the 16×16 non-zero coefficients retained in the standard hard thresholding. It is obvious that in this picture, the edge artifacts are less severe than in the standard case. Meanwhile, since the regularization parameter λ also controls the smallest size of features to preserve, in the TV regularized restored image, smaller features (such as the smallest square) are altered more than the large features, i.e. the intensities are lower than the standard approximation. In Fig 3.19, we show the cameraman image with Gaussian white noise. We display the 64×64 non-zero coefficient reconstruction calculated by standard hard thresholding in Fig 3.20, and the TV regularized wavelet compression model with hard thresholding in Fig 3.21. Compared to the standard hard thresholding image, the edge artifacts in the TV model approximations are much less severe.

We also use our TV regularized wavelet compression model on color images by applying it to the RGB channels independently. In Fig 3.22 and 3.23, we show the original color image and the noisy color image respectively. Fig 3.24

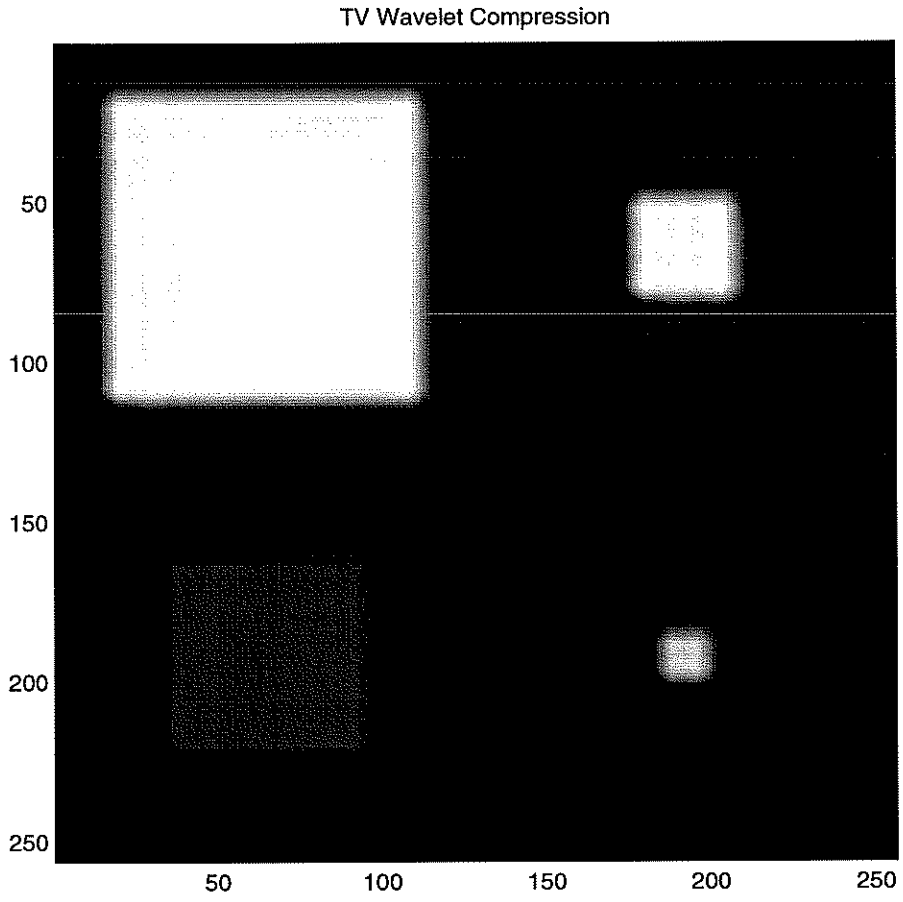

Figure 3.18: The TV norm hard thresholding. Also keep the same 16×16 nonzero coefficients but perturb their values. There are less edge artifacts in the compressed image.

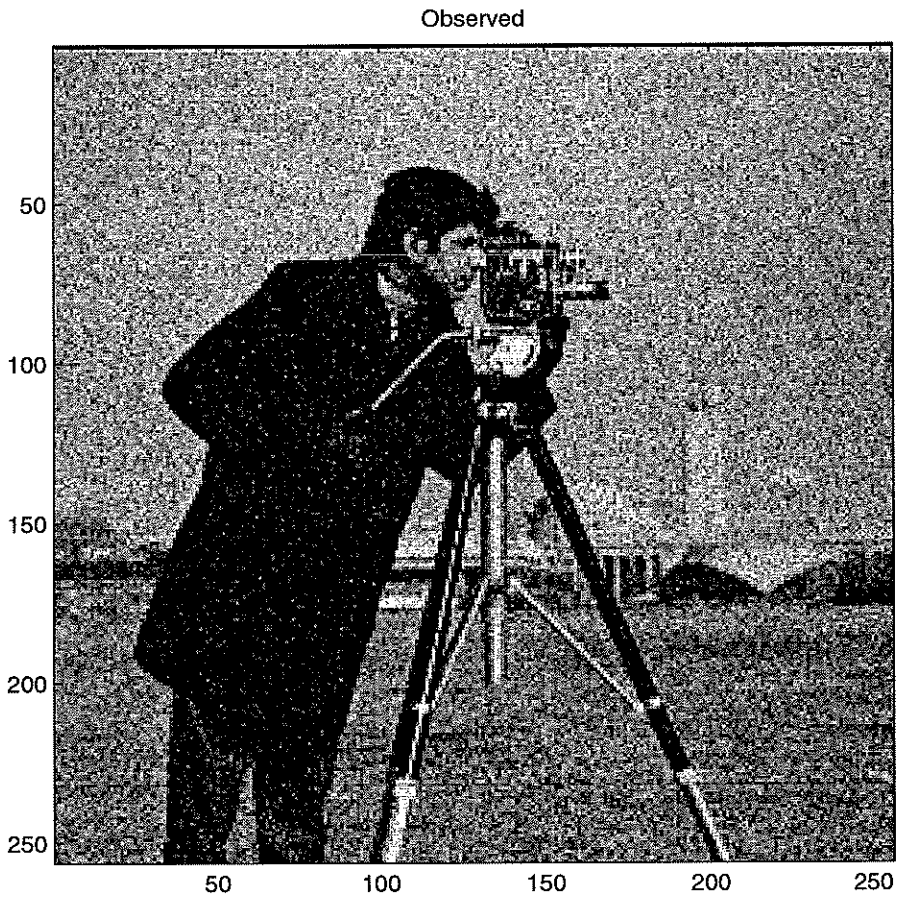

Figure 3.19: The noisy cameraman image.

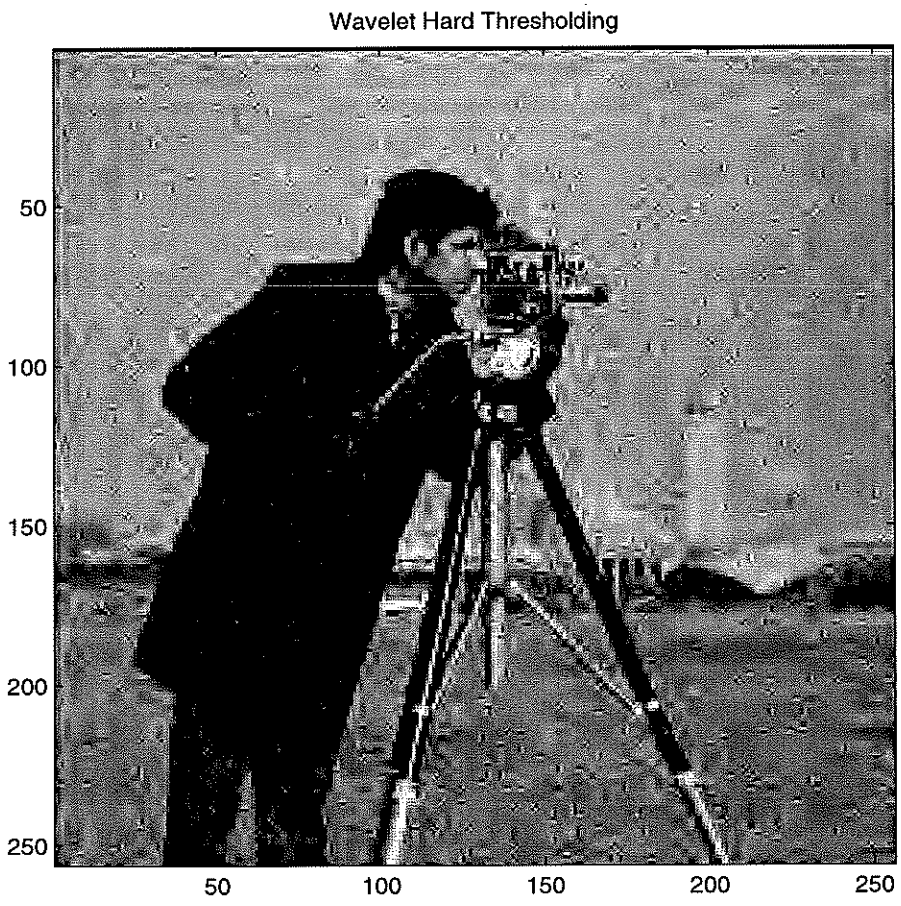

Figure 3.20: The standard hard thresholding approximation, keep the largest 64×64 coefficients. Severe edge artifacts present in the compressed image.

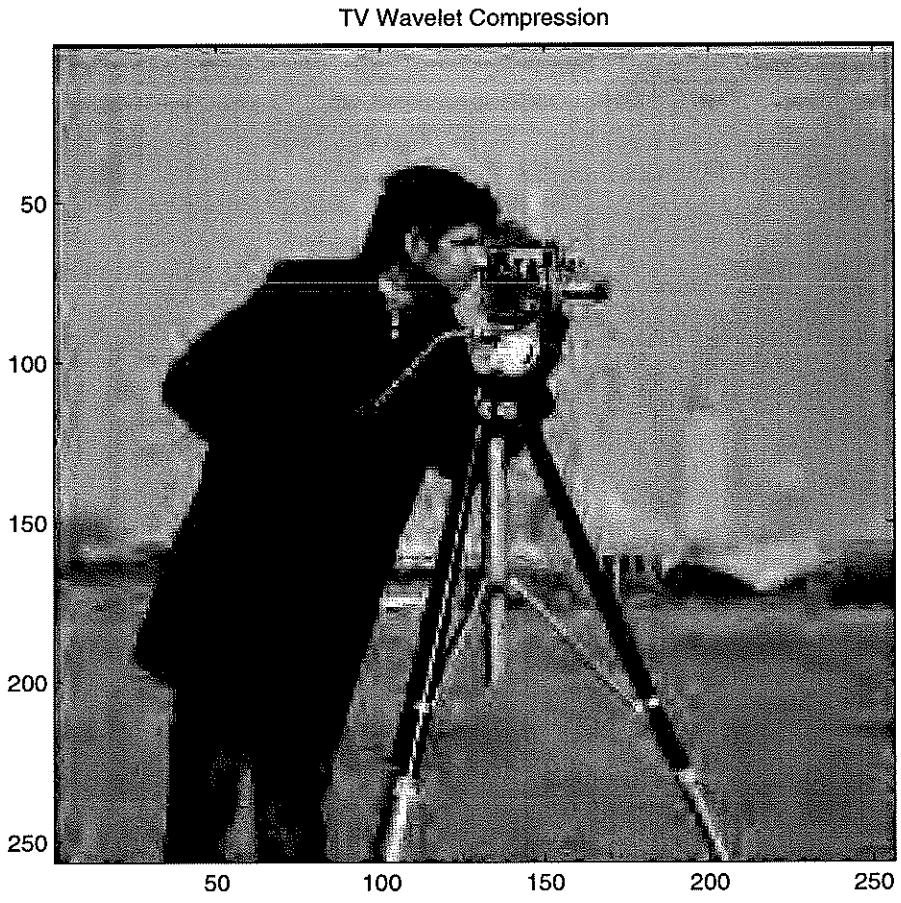

Figure 3.21: The TV norm hard thresholding. It keeps 64×64 nonzero coefficients. There are much less edge artifacts in the compressed images.

and 3.25 correspond to compressed images obtained by using the standard 4-level DB6 hard thresholding and the TV regularized wavelet compression model. Both of them keep 64×64 non-zero wavelet coefficients. We also display a similar pair of images with more compression in Fig 3.26 and 3.27. This pair of images retain only 32×32 non-zero coefficients. Both compressed images obtained by TV regularized model have much less severe edge artifacts than that of the standard hard thresholding images. This tells us that the TV regularized wavelet compression model improves the quality of the compressed images. In the last comparison, we show in Fig 3.28 the standard hard thresholding compression image of the noise free image (Fig 3.22), and in Fig 3.29 the TV regularized model compression image. In the Fig 3.28, the severe edge artifacts are caused purely by wavelet thresholding. Compared to it, the image in Fig 3.29 has less severe edge artifacts which is reduced by perturbing the retained wavelet coefficient according to TV regularization.

3.7 Conclusion

, We have used the TV regularized model to select and modify the non-zero wavelet coefficients in the thresholding procedure. The resulting compressed images contain less severe edge artifacts than those in the standard thresholding images, especially when large noise is present in the image. The model can directly operate on the wavelet coefficients, and therefore, can easily be embedded into practical compression schemes. More work needs to be done to improve the speed of convergence and to make the method more practical.

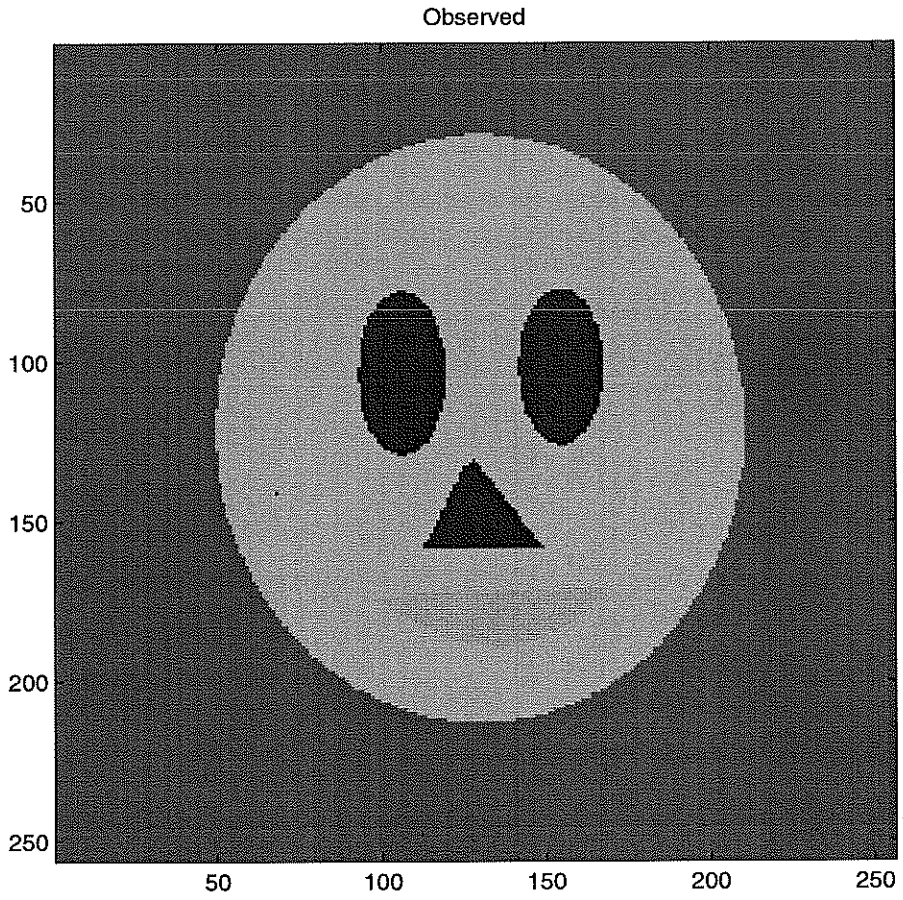

Figure 3.22: The original color image. This image and the subsequent color images can be found at the website:

http://www.math.ucla.edu/applied/cam/index.html .

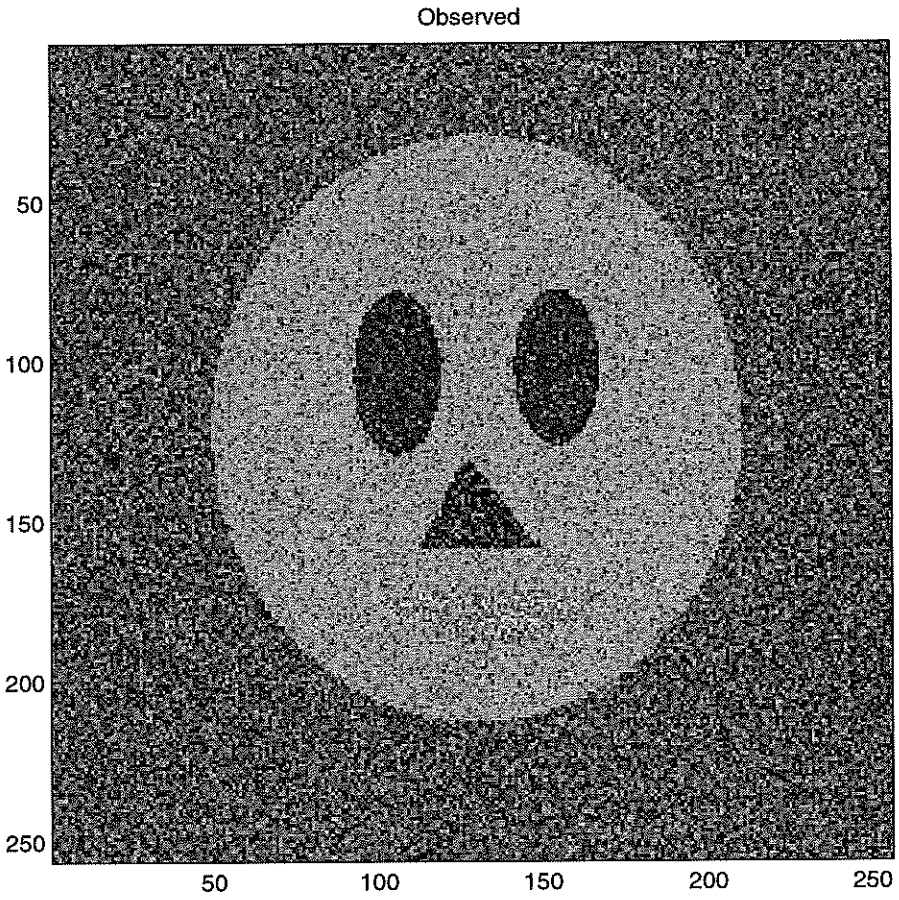

Figure 3.23: The noisy color image.

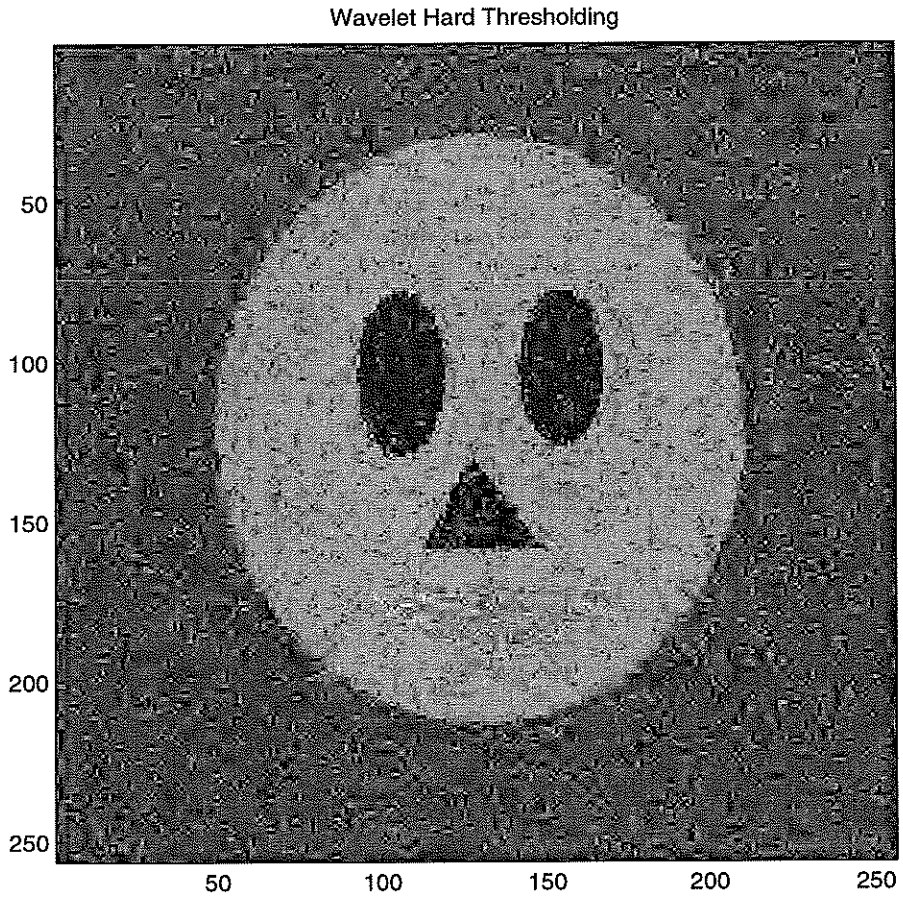

Figure 3.24: The standard hard thresholding compressed image, keep the largest 64×64 coefficients. Severe edge artifacts present in the compressed image.

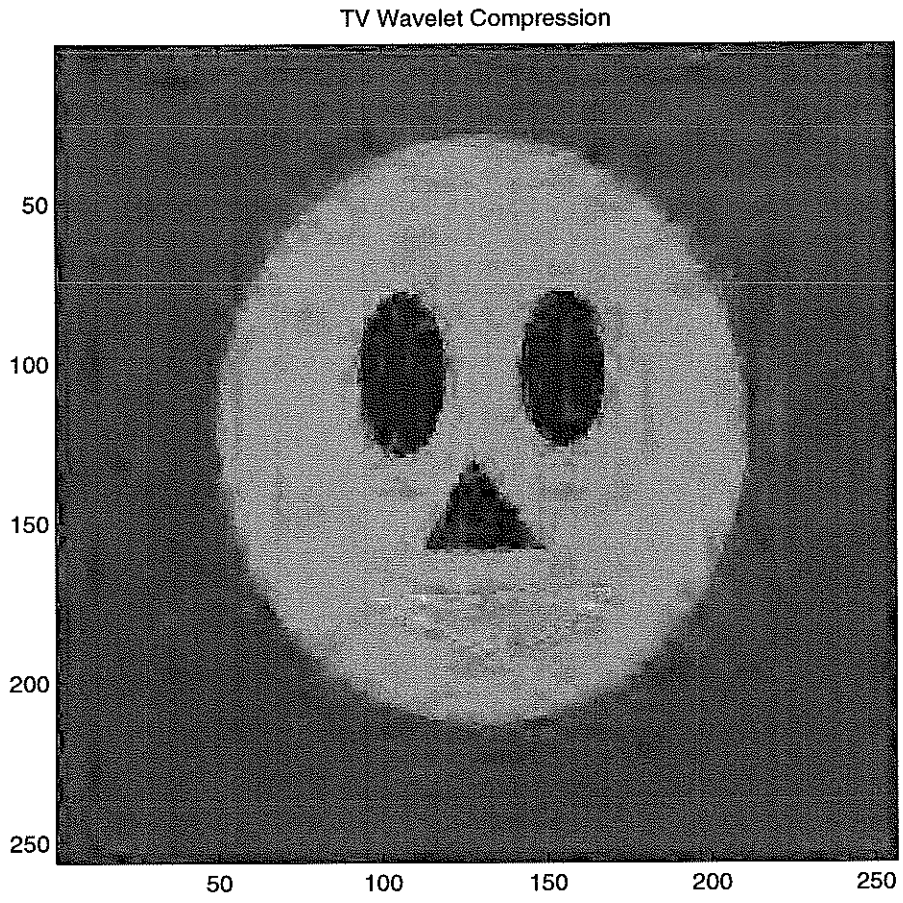

Figure 3.25: The TV regularized model compressed image, also keep 64×64 non-zero coefficients. Less severe edge artifacts present in the compressed image.

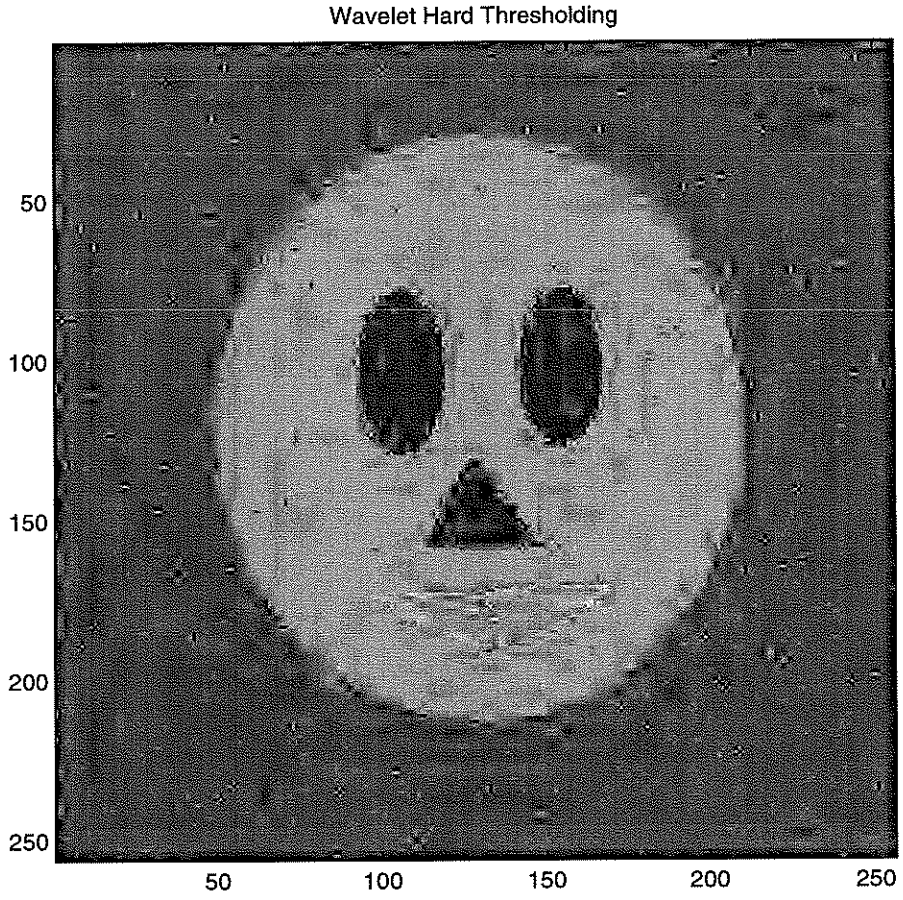

Figure 3.26: The standard hard thresholding compressed image with more compression, keep the largest 32×32 coefficients. Severe edge artifacts present in the compressed image.

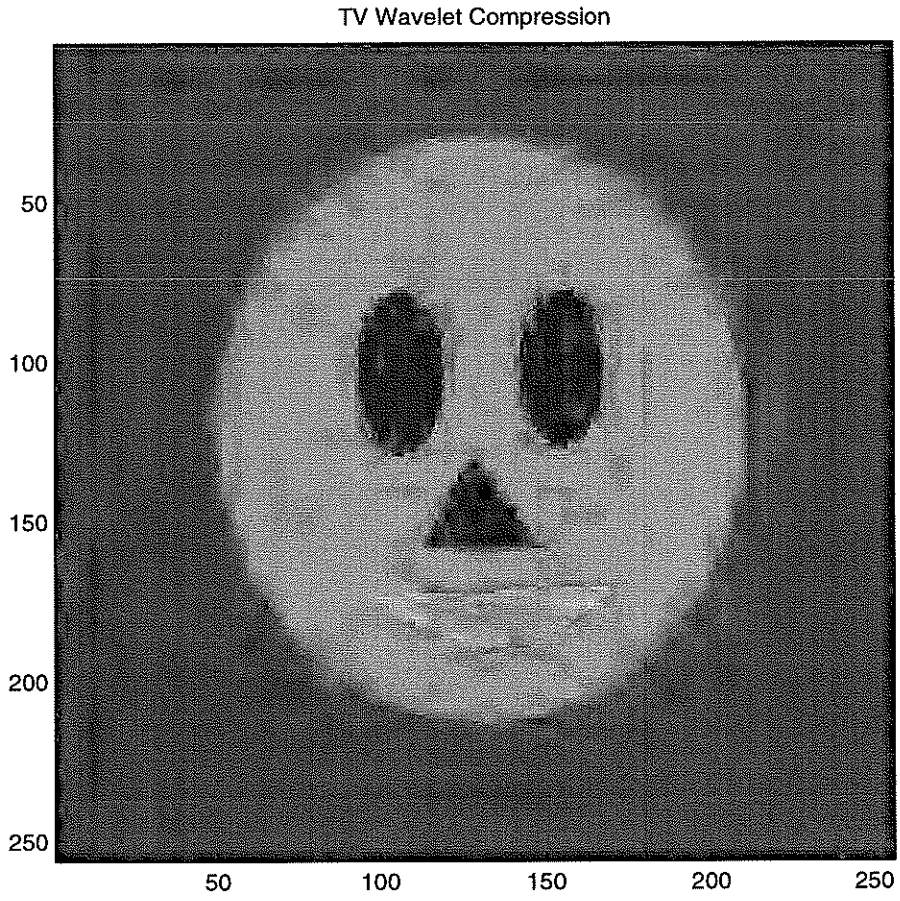

Figure 3.27: The TV regularized model compressed image, also keep 32×32 non-zero coefficients. Less severe edge artifacts present in the compressed image.

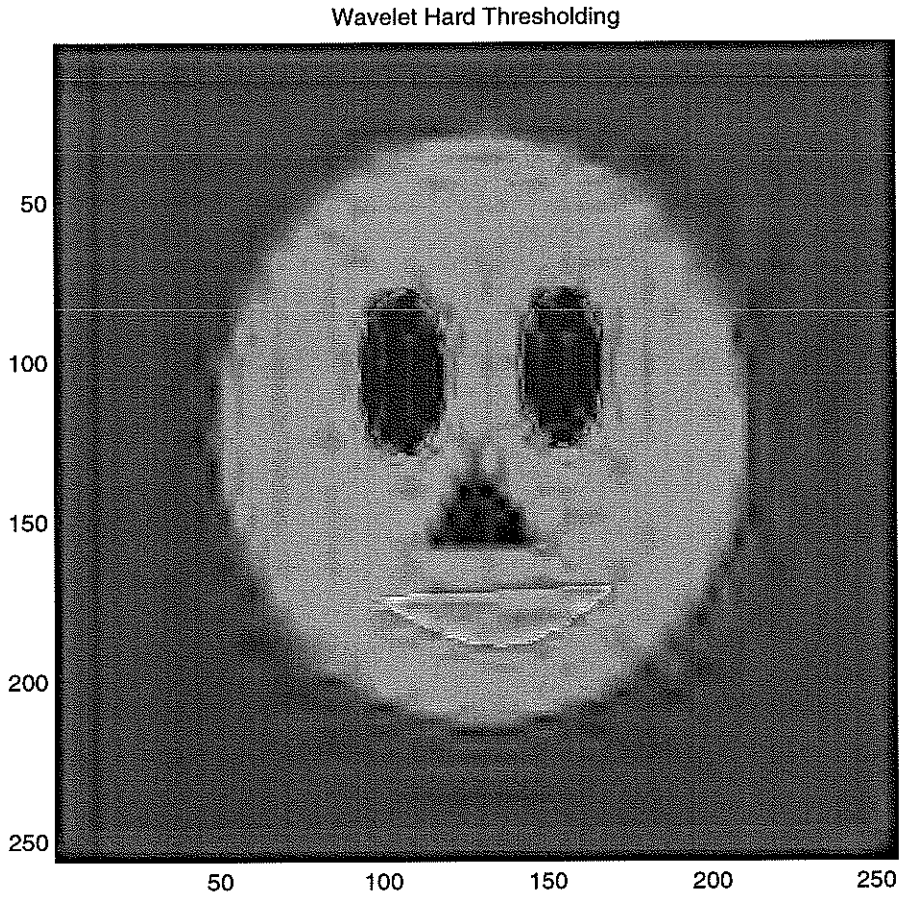

Figure 3.28: The standard hard thresholding compressed image of the noise free image, keep the largest 24×24 coefficients. Severe edge artifacts caused purely by wavelet thresholding present in the compressed image.

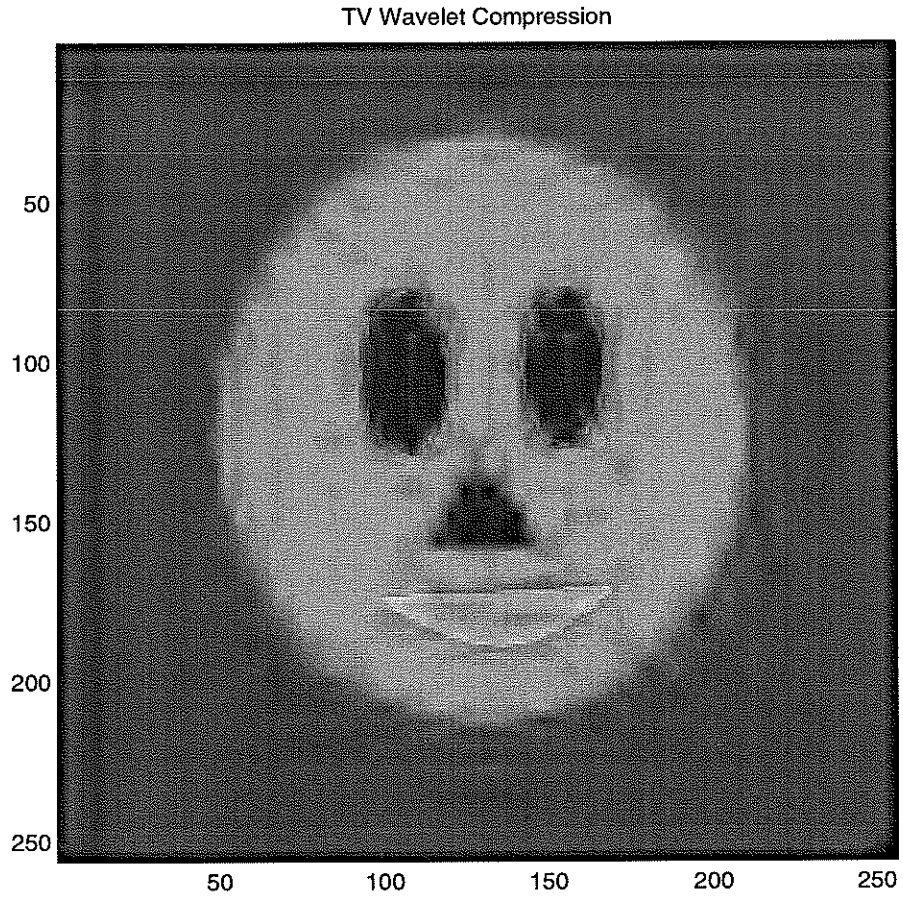

Figure 3.29: The TV regularized model compressed image of the noise free image, also keep 24×24 non-zero coefficients. Less severe edge artifacts present in the compressed image.

REFERENCES

[1] L. Alveraz, F. Guichard, P. L. Lions and J. M. Morel, *Axioms and Fundamental Equations of Image Processing*, Arch. Rational Mechanics and Anal., 16 (1993), pp200-257.

[2] L. Alveraz, J. M. Morel, *Formalization and Computational Aspects of Image Analysis*, Acta Numerica (1994), pp1-59.

[3] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, *Image Inpainting,* Tech. Report, ECE-University of Minnesota, 1999.

[4] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast Wavelet Transforms and Numerical Algorithms*, Comm. Pure Appl. Math, 44(1991), pp141-183.

[5] P. Blomgren and T. F. Chan, *Color TV: Total Variation Methods for Restoration of Vector Valued Images*, CAM Report, No. 96-5, Dept. of Math., UCLA, 1996.

[6] A. Chambolle, R. DeVore, N. Lee and B. Lucier, *Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal Through Wavelet Shrinkage*, IEEE Tran. Image Proc., Vol. 7, No. 3, Mar. 1998, pp319-333.

[7] A. Chambolle, P. L. Lions, *Image Recovery via Total Variational Minimization and Related Prolems,* Numer. Math., 76, 1997, pp167-188.

[8] T. F. Chan, G. H. Golub, and P. Mulet, *A Nonlinear Primal-Dual Method for Total Variation-Based Image Restoration*, in ICAOS'96, 12th International Conference on Analysis and Optimization of Systems: Images, Wavelets, and PDEs, Paris, June 26-28, 1996, number 219 in Lecture Notes in Control and Information Sciences, pp. 241-252.

[9] T. F. Chan and L. Vese, *Active Contour Without Edges* Submit to IEEE Tran. on Image Proc., 1998.

[10] T. F. Chan and C. K. Wong, *Total Variation Blind Deconvolution*, CAM Report, No. 96-44, Dept. of Math., UCLA, 1996.

[11] T. F. Chan and H. M. Zhou, *Adaptive ENO-Wavelet Transforms for Discontinuous functions*, CAM Report, No. 99-21, Dept. of Math., UCLA, Submit to SIAM Numer. Anal., 1999.

[12] T. F. Chan and H. M. Zhou, *Feature Preserving Lossy Image Compression Using Nonlinear PDE's*, SPIE Proceedings on Advanced Signal Processing Algorithms, Architectures, and Implementations VIII, Vol. 3461, F. T. Luk, ed., San Diego, California, July 1998, pp 316-327.

[13] E. Candes and D. Donoho, *Ridgelets: a Key to Higher-dimensional Intermittency?*, Phil. Trans. R. Soc. Lond. A(1999).

[14] E. Candes and D. Donoho, *Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges*, Tech. Report, Dept. of Stat., Stanford Univ., 1999.

[15] C. K. Chui, *Wavelet: A Mathematical Tool for Signal Analysis*, SIAM, 1997.

[16] P. Claypoole, G. Davis, W. Sweldens and R. Baraniuk, *Nonlinear Wavelet Transforms for Image Coding*, Correspond. Author: Baraniuk, Dept. of Elec. and Comp. Sci., also Submitt to IEEE Tran. on Image Proc., Preprint, 1999.

[17] I. Daubechies, *Orthonormal Bases of Compactly Supported Wavelets*, Comm. Pure Appl. Math. 41(1988), pp909-996.

[18] I. Daubechies, *Ten Lectures on Wavelets*, SIAM 1992.

[19] D. Donoho, *De-noising by Soft Thresholding*, IEEE Trans. Inf. Th. 41(1995), pp613-627.

[20] D. Donoho, *Wedgelets: Nearly-Minimax Estimation of Edges*, Tech. Report, Dept. of Stat., Stanford Univ., 1997.

[21] D. Donoho, *Orthonormal Ridgelets and Linear Singularities* Tech. Report, Dept. of Stat., Stanford Univ., 1998.

[22] D. Donoho, *Sparse Components of Images and Optimal Atomic Decompositions*, Tech. Report, Dept. of Stat., Stanford Univ., 1999.

[23] D. Donoho, I. Daubechies, R. DeVore, M. Vetterli, *Data Compression and Harmonic Analysis*, Dept. of Stat., Stanford Univ., Preprint, 1998.

[24] D. Donoho, I. Johnstone, *Adapting to Unknown Smoothness via Wavelet Shrinkage*, J. Amer. Stat. Assoc., Vol. 90, 1995, pp1200-1224.

[25] *Special Issue on Partial Differential Equations and Geometry-Driven Diffusion in Image Processing and Analysis*, IEEE Tran. on Image Proc., Vol. 7, No. 3, Mar. 1998.

[26] A. Harten, *Discrete Multi-resolution Analysis and Generalized Wavelet*, Appl. Numer. Math., Vol. 12, 1993, pp153-192.

[27] A. Harten, *Multiresolution Representation of Data, II. General Framework*, Dept. of Math., UCLA, CAM Report 94-10, April 1994.

[28] A. Harten, *Multiresolution Representation of Cell-Averaged Data*, Dept. of Math., UCLA, CAM Report 94-21, July 1994.

[29] A. Harten, B. Engquist, S. Osher and S. Chakravarthy, *Uniformly High Order Essentially Non-Oscillatory Schemes, III*, Journal of Computational Physics, v71 (1987), pp.231-303.

[30] P. C. Hensen, *The L-curve and Its Use in the Numerical Treatment of Inverse Problems*, Tech. Report, IMM-REP 99-15, Dept. of Math. Model., Tech. Univ. of Denmark, 1999.

[31] S. Mallat, *Multiresolution Approximation and Wavelet Orthonormal Bases of $L^2(R)$*, Tran. Amer. Math. Soc. 315(1989), pp.69-87.

[32] S. Mallat, *A Theory of Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Trans. PAMI 11 (1989), pp. 674-693.

[33] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

[34] J. M. Morel and S. Solimini, *Variational Methods in Image Segmentation*, Birkhauser, 1994.

[35] D. Mumford and J. Shah, *Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems*, Comm, Pure Appl. Math. 42, 1989, pp577-685.

[36] B. A. Olshausen and D. J. Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, Nature 281, pp607-609.

[37] A. Marquina and S. Osher, *Explicit Algorithms for a New Time Dependent Model Based on Level Set Motion for Nonlinear Deblurring and Noise Removal*, CAM Report, No. 99-5, Dept. of Math., UCLA, 1999.

[38] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE T PATTERN ANAL. 12: (7), July, 1990, pp629-639.

[39] L. Rudin, S. Osher and E. Fatemi, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D, Vol 60(1992), pp. 259-268.

[40] G. Sapiro and A. Tannenbaum, *Affine Invariant Scale-Space*, Internet. J. Comput. Vision, 11 (1993), pp. 25-44.

[41] C.W. Shu, *High Order ENO and WENO Schemes for Computational Fluid Dynamics*, Lecture Notes in Computational Science and Engineering 9: High-Order Methods for Computational Physics, T. Barth and H. Deconinck (Eds.), Springer, 1999, pp439-582.

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

[43] David M. Strong, Peter Blomgren, and Tony F. Chan, *spatially Adaptive Local Feature-Driven Total Variation Minimizing Image Restoration*, Proceedings of SPIE, vol. 3167, 1997.

[44] W. Sweldens, *The Lifting Scheme: A Construction of Second Generation Wavelets*, SIAM J. Math. Anal., Vol. 29, No. 2, 1997, pp511-546.

[45] C. Vogel and M. Oman, *Iterative Methods for Total Variation Denoising*, SIAM J. Sci. Comput., Vol. 17, 227-238, 1996..

[46] J. R. Williams and K. Amaratunga, *A Discrete Wavelet Transform without Edge Effects* MIT, IESL Tech. Report No. 95-02, 1995.