UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

ENO-Wavelet Transforms and Some Applications

Tony F. Chan
Hao-Min Zhou

August 2002

CAM Report 02-50

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

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

ENO-wavelet Transforms and Some Applications

Tony F. Chan
Hao-Min Zhou

Abstract. Standard wavelet linear approximations generate oscillations (Gibbs' phenomenon) near singularities in piecewise smooth functions. Nonlinear and data dependent methods are often considered as the main strategies to avoid those oscillations. Using ideas from Essentially Non-Oscillatory (ENO) schemes for numerical shock capturing to standard wavelet transforms, we have designed an adaptive ENO-wavelet transform for approximating discontinuous functions without oscillations near the discontinuities. The crucial point is that the wavelet coefficients are computed without differencing function values across jumps. The ENO-wavelet transform retains the essential properties and advantages of standard wavelet transforms such as concentrating the energy to the low frequencies, obtaining arbitrary high order accuracy uniformly and having a multiresolution framework and fast algorithms, all without any edge artifacts. We have also shown the stability of the ENO-wavelet transforms and 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. We briefly discuss several applications of the ENO-wavelet transforms, including function approximation, image compression and signal denoising.

1 Introduction

In this chapter, we present wavelet algorithms designed to approximate piecewise continuous functions, for instance, piecewise smooth functions connected by large jumps. We begin with a summary of the basic idea of the design of such ENO-wavelet transforms, and then show some theoretical results illustrated by numercial examples. We will refer all implementation details and the proof of the theorems to [11] and [46]. Some of the results and experiments have already been published in our earlier papers, but many of them are new.

Wavelet theory is a very rich and well developed field in mathematics. It has many successful applications, such as in the digital image processing, computer graphics, numerical computations of partial differential equations (PDE's) and integral equations. There is a large literature on wavelet theory and its applications in the past two decades. Here we just list some mathematically oriented books on this subject, see [13], [21], [38], [42] and [33].

Beyond Wavelets
J. Stoeckler and G. V. Welland (Eds.), pp. 1–34.

ISBN 0-12-xxxxxx-x

There are several ways to introduce the wavelet theory. One is to view wavelets as orthogonal basis functions of the L^2 space, the space of square integralable functions. Every function $f(x)$ in L^2 can be decomposed into a sum of wavelet functions with coefficients called wavelet coefficients, which are computed by the L^2 projections of $f(x)$ onto the wavelet basis. This procedure is called a wavelet transform. According to their structures, wavelet coefficients are divided into two parts, namely low frequency coefficients and high frequency coefficients (in the literature, they are also called scaling coefficients and wavelet coefficients respectively). They represent different natures of the function $f(x)$. Low frequency coefficients describe the local averages of the function and high frequencies reflect the local smoothness. Obviously, high frequency coefficients corresponding to smooth regions have small magnitudes, while the high frequency coefficients depending on regions containing discontinuities have large magnitudes.

Most applications of wavelet deal with the wavelet coefficients of the functions. For instance, wavelet based function approximations usually use the multiresolution structure [36] of the wavelet coefficients, and the fact of having small high frequency coefficients in smooth regions to approximate the function by reconstructing it using only a small portion of its wavelet coefficients. For example, one can use only the low frequency coefficients to rebuild an approximation of the function, and this is called wavelet linear approximation.

It is well known that wavelet linear approximation 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. However, the 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, see [38]. The reason for it is that the jumps generate large high frequency wavelet coefficients, and the linear approximations do not use this information in their reconstruction, thus they cannot get the same high accuracy near the points of discontinuity as in the smooth region. In fact, the oscillations generated near the jumps cannot be removed by mesh refinement.

How to get rid of the Gibbs' phenomenon, or more generally speaking, how to better approximate the singularities in functions has become one of the very active research topics in wavelet studies in recent years. Many methods have been proposed by different authors. Most of them can be classified into two types. The first one is to still use standard wavelet transforms but to improve the approximations by post-processing the wavelet coefficients in different manners. For example, nonlinear data-dependent approximations are used which retain certain high frequency coefficients.

The most notable methods in these data-dependent methods are various thresholding techniques, including hard and soft thresholding, see [23], [27], [26],

[38] and corresponding references listed there. The main idea of this thresholding approximation is to truncate both low and high frequency wavelet coefficients according to their magnitudes, not frequencies. For instance, hard thresholding sets all coefficients whose magnitudes are less than a given tolerance to zero and retains the other coefficients unchanged. It has been shown through many research efforts that such non-linear processes can effectively reduce Gibbs oscillations. Consequently, they have being widely used in many applications such as image compression (e.g. Shapiro's embedded zerotree wavelet (EZW) coding scheme [40]) and denoising (e.g. Coifman and Donoho's translation invariant denoising [15]), numerical solutions of partial differential equations (PDE's) [31] and integral equations [7]. However, from a function approximation point of view, 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 unless all jump-related coefficients are preserved. Recently, an interesting approach, which combines wavelet thresholding techniques with PDE's derived from variational principles (e.g. total variation (TV)) to reduce the Gibbs' oscillations, has been proposed in many works for different purposes, please see [12], [14] and [22].

Unlike the first type of methods which still use the standard wavelet transforms, the second type of methods works more directly on the wavelet transforms. In many research studies, new wavelet-like transforms are introduced so that the singularities can be more efficiently represented. For instance, one approach is to construct new orthonormal (complete or over-complete) bases such as Donoho's wedgelets [24], Candes and Donoho's rigdelets [8], [25] and curvelets [9], and Le Pennec and Mallat's bandelets [35]. Another approach is to modify the standard wavelet transforms to avoid large high frequency wavelet coefficients near jumps. A few papers in the literature have discussed this approach. Claypoole, Davis, Sweldens and Baraniuk [19] proposed an adaptive lifting scheme which lowers the order of approximation near jumps, thus minimizing the Gibbs' effect. This scheme suffers from reduced approximation accuracy near jumps, and some residual Gibbs' phenomenon still exists.

Recently, we have proposed ENO-wavelet transforms for piecewise smooth functions as an alternative for this problem by borrowing the well developed Essentially Non-Oscillatory (ENO) technique for shock capturing in computational fluid dynamics (e.g. see [32] and [41]) to modify the standard wavelet transforms near discontinuities in order to overcome the Gibbs' oscillations.

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 difference across 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 constructing the approximations. In fact, it has been explored by Harten in his general framework of multiresolution [28], [29], [30] and [4] (Sweldens' lifting scheme [43] is similar to it). Recent studies of Harten's general framework and its application in data compression can be found in [1], [2], [5], and [18]. Harten's approach is to directly blend the two ideas, and to fully implement the ENO schemes at every point. This consists of using an adaptive ENO finite difference table to select the stencil and then computing the decomposition as well as the reconstruction process. However, Harten's method cannot be directly applied to the more interesting and generally used pyramidal filtering algorithms in which the standard wavelet transforms are implemented. This is because we have to work only with fixed size and fixed value filters in this context, and these rigid filters can not be directly used to compute the adaptive divided difference tables at each grid point.

The ENO method retains a fixed wavelet transform and locally modifies the function near discontinuities so that wavelet filters are applied to smooth data. By recording how the changes are made, 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. The extension idea in wavelet methods, such as extension obtained by spline wavelet methods, has been used in constructing wavelets for closed intervals [3], [16] and [17]. However, those approaches usually modify the wavelet basis at the boundary of the interval rather than the function.

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.

The resulting wavelet transform retains all the desirable properties of the standard transform: it is stable and can have uniformly arbitrarily high order of approximation (with a rigorous uniform order of the error bound), it concentrates the large 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 $l+1$ 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.

Besides, since the designed ENO-wavelet transforms play the same role as the standard wavelet transforms in the applications, it is natural and even more beneficial to use them in conjunction with the standard adaptive nonlinear techniques such as hard and soft thresholding in many applications such as image compression and denoising. We will discuss those applications briefly at the last part of this chapter.

The arrangement of the chapter is as follows. In section 2, we start by reviewing the standard wavelet transforms. Then we give a general idea to construct the ENO-wavelet transforms. In section 3, we state the stability results and 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 *away* from the discontinuities. Finally, in section 4, we discuss some possible applications of the ENO-wavelet transforms including function approximation, image compression and signal denoising, and we give some numerical examples.

2 The ENO-Wavelet Algorithm

In this section, we give the general idea to construct ENO-wavelet transforms for piecewise smooth functions.

2.1 ENO-wavelet at Discontinuities

Before we present the adaptive ENO-wavelet transforms, we want to briefly recall some basic knowledge in the standard wavelet transforms. In this section, we do not intend to cover all fundamentals in wavelet theory, we just want to use this opportunity to introduce some notations used in the standard wavelet transforms so that they can be used in our ENO-wavelet transforms. For readers who are interested in the standard wavelet theory, please see [13], [21], [38], [42] and many other relevant references that we do not list here. We also want to point out that in this chapter, we only discuss the design of ENO-wavelet transforms using Daubechies orthonormal wavelet frameworks. The idea can be easily extended to other types of wavelets such as biorthogonal wavelets, but that is not our focus here.

To simplify the discussion, 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:

$$\phi(x) = \sqrt{2}\sum_{s=0}^{l} c_s \phi(2x - s); \tag{2.1.1}$$

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

$$\psi(x) = \sqrt{2}\sum_{s=0}^{l} h_s \phi(2x - s); \tag{2.1.2}$$

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

Wavelet $\psi(x)$ having p vanishing moments means:

$$\int \psi(x)x^j dx = 0, \qquad for \quad j = 0, 1, \cdots, p-1. \tag{2.1.3}$$

We will use the following standard notations:

$$\phi_{j,i}(x) = 2^{\frac{j}{2}}\phi(2^j x - i), \tag{2.1.4}$$

and

$$\psi_{j,i}(x) = 2^{\frac{j}{2}}\phi(2^j x - i). \tag{2.1.5}$$

Consider the subspace V_j of L^2 defined by:

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

and the subspace W_j of L^2 defined by:

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

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:

$$L^2 = V_J \oplus \sum_{j=J}^{\infty} W_j. \tag{2.1.6}$$

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

$$f_j(x) = \sum_i \alpha_{j,i}\phi_{j,i}(x), \tag{2.1.7}$$

where

$$\alpha_{j,i} = \int f(x)\phi_{j,i}(x)dx, \quad i = \cdots, -1, 0, 1, \cdots, \tag{2.1.8}$$

which we call low frequency wavelet coefficients (they are often called scaling coefficients in the literature). Similarly, we can project $f(x)$ onto W_j by:

$$w_j(x) = \sum_i \beta_{j,i}\psi_{j,i}(x), \tag{2.1.9}$$

where

$$\beta_{j,i} = \int f(x)\psi_{j,i}(x)dx. \quad i = \cdots, -1, 0, 1, \cdots, \tag{2.1.10}$$

which we call high frequency wavelet coefficients (often called wavelet coefficients in the literature). In this paper, we use the term wavelet coefficients to denote both low and high frequency coefficients. Therefore, the function $f(x)$ can be decomposed by:

$$f(x) = f_j(x) + \sum_{t=j}^{\infty} w_t(x). \tag{2.1.11}$$

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

From (2.1.4) and (2.1.5), the projection coefficients $\alpha_{j,i}$ and $\beta_{j,i}$ of $f(x)$ in the subspaces V_j and W_j can be easily computed from the coefficients $\alpha_{j+1,i}$ by the so called fast wavelet transform:

$$\alpha_{j,i} = \sum_{s=0}^{l} c_s \alpha_{j+1,2i+s}; \tag{2.1.12}$$

and

$$\beta_{j,i} = \sum_{s=0}^{l} h_s \alpha_{j+1,2i+s}. \tag{2.1.13}$$

The standard linear wavelet approximation can achieve arbitrary high accuracy away from discontinuities, but it oscillates near the jumps. The intuitive 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 Figure 1, we display a piecewise continuous function (left) and its DB-6 wavelet coefficients (right) with low frequencies at the left end and high frequencies at the right end. From the right picture, we see that most of the high frequency coefficients are zeros, except for a few large coefficients which are computed near jumps. Figure 2 displays the linear approximation (dash-dotted 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 (also known as 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

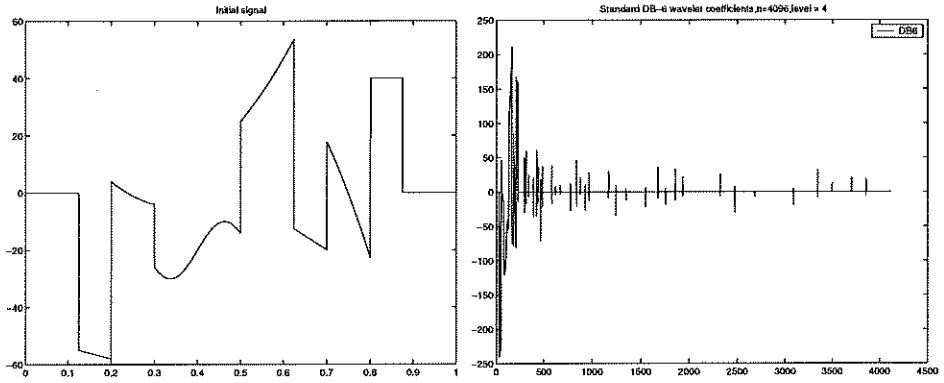

Figure 1. The initial function (left) and its DB6 coefficients (right). Most of the high frequency coefficients (right part) are zero except for a few large coefficients computed near the jumps..

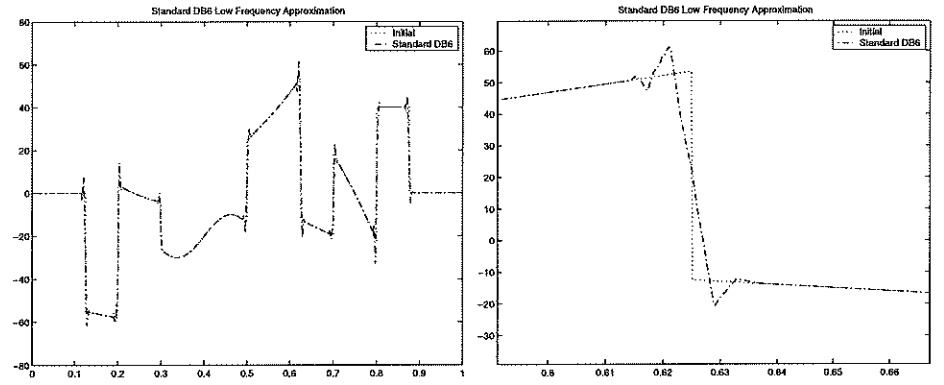

Figure 2. The approximation function (left) and its zoom in (right), Oscillations are generated near the discontinuities in the linear approximation..

rid of the oscillations, we want to avoid stencils crossing discontinuities. This motivates us to apply the ENO idea to avoid stencils crossing jumps.

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, to better explain the algorithm, we assume that the location of the jumps are known, and we give the ENO-wavelet approximations at the discontinuities by using one-sided information to avoid oscillations. Then, we give some methods to detect the location of the discontinuities.

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. In order to simplify the explanation, we also assume that the discontinuities are well separated so that the modification we will make at one jump will not interact the modification at another jump. Therefore, 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.

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 Figure 3 to show how to extend the function and compute the ENO-wavelet coefficients.

As shown in Figure 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 $i \leq m \leq i+k-2$, where $l = 2k-1$, 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 (2.1.12) and (2.1.13) to compute the coarser level coefficients.

There are many methods to extrapolate the extended values. For example, a straightforward way is to use p-point polynomial extrapolation such as Lagrange

polynomials or Taylor expansion polynomials. In our numerical experiments in this chapter, we use Lagrange polynomial extrapolation for noise free data, and least square extrapolation [45] for noisy data.

There is a storage problem for this direct function extrapolation or the extrapolation of the finer coefficients. Indeed, it doubles the number of the wavelet coefficients near every discontinuity. To retain the perfect invertible property, using the notation in Figure 3, 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.

Facing this challenge, we have proposed a better way, which we called coarse level extrapolation, to accomplish our goals. 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 Figure 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, and then compute the extended coarser level wavelet coefficients $\hat{\alpha}_{j,m}$ and $\hat{\beta}_{j,m}$, $i \le m \le (i+k-2)$ 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 (or some predefined values), then determine the extended finer level wavelet coefficients.

However, in Daubechies' orthonormal wavelet transforms, we cannot arbitrarily prescribe both $\hat{\alpha}_{j,m}$ and $\hat{\beta}_{j,m}$ simultaneously. This is because they are not linearly independent. Let's take $m = i$ as an example. Assume that we have prescribe both $\hat{\alpha}_{j,i}$ and $\hat{\beta}_{j,i}$ as given values, this means that we have implicitly extended the finer level values $\hat{\alpha}_{j+1,2i+l-1}$ and $\hat{\alpha}_{j+1,2i+l}$ satisfying:

$$\begin{pmatrix} \hat{\alpha}_{j,i} \\ \hat{\beta}_{j,i} \end{pmatrix} = \begin{pmatrix} \sum_{s=0}^{l-2} c_s \alpha_{j+1,2i+s} + c_{l-1}\hat{\alpha}_{j+1,2i+l-1} + c_l \hat{\alpha}_{j+1,2i+l} \\ \sum_{s=0}^{l-2} h_s \alpha_{j+1,2i+s} + h_{l-1}\hat{\alpha}_{j+1,2i+l-1} + h_l \hat{\alpha}_{j+1,2i+l} \end{pmatrix}.$$

As we have $\frac{h_{l-1}}{c_{l-1}} = \frac{h_l}{c_l}$, this implies that we can only prescribe one of the coarse level coefficients $\hat{\alpha}_{j,i}$ and $\hat{\beta}_{j,i}$ and determine the other one by the above relationship. 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 $\hat{\beta}_{j,m}$

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

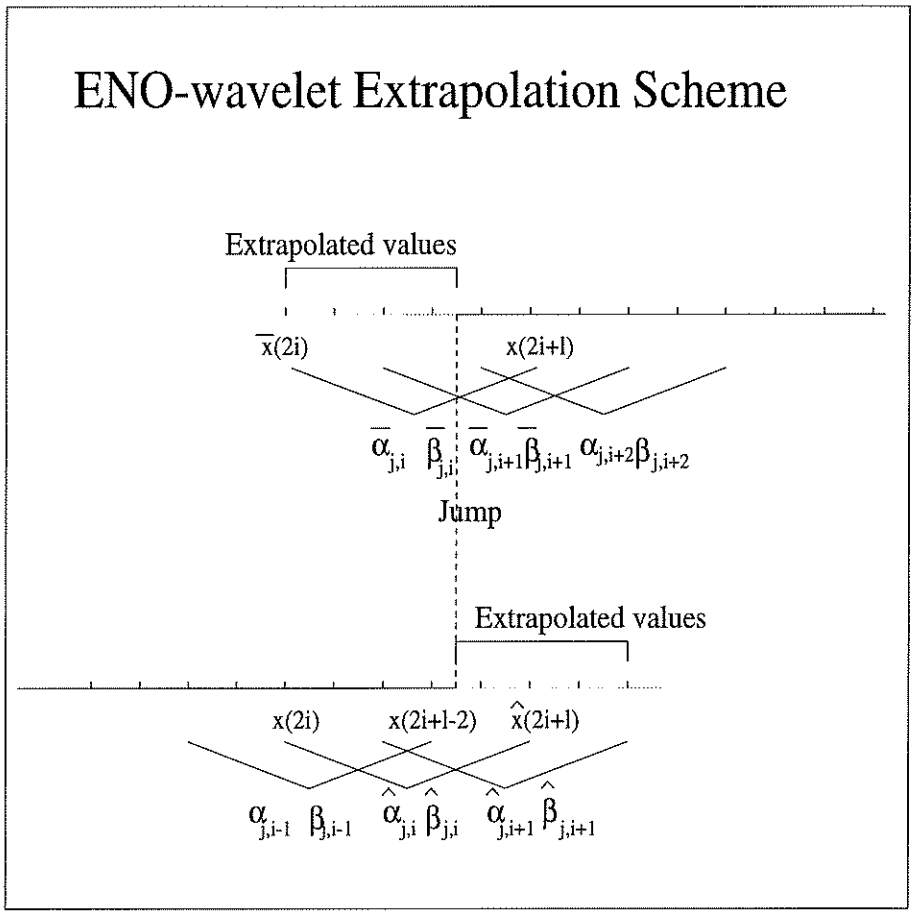

Figure 3. Coarse Level Extrapolation Illustration. From the left side of the discontinuity, we extrapolate the low frequency coefficients $\hat{\alpha}_{j,m}$ to determine corresponding high frequency coefficients $\hat{\beta}_{j,m}$ and store them. From the right side of the discontinuity, we extend the high frequency coefficients $\bar{\beta}_{j,m}$ to determine and store the low frequency coefficients $\bar{\alpha}_{j,m}$..

Again 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 low frequency and one high frequency coefficients for every stencil.

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.

Of course, in the ENO-wavelet transforms, to retain the perfect invertibility property, we need to store all adaptive information, i.e. the locations of the discontinuities. In our implementation in this chapter, we just use one extra bit for each stencil near the discontinuities to indicate it contains a discontinuity. In the application of compression, which aims to reduce the total storage of representing an image, these extra bits need to be taken into account carefully, we will discuss it in the last section of this chapter.

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, 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 large.

2.2 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.

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

nuity if for some $m < p$, we have:

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

We define the intensity of a jump in the m-th derivative at x as

$$[f^{(m)}(x)] = |f^{(m)}(x+) - f^{(m)}(x-)|.$$

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, it has been shown in [6], [34] and [44] that if a function $f(x)$ is Lipschitz $\mu \leq p$ at x, i.e. $|f(x+\delta) - f(x)| \leq \delta^{\mu}$ for any small δ, the corresponding high frequency wavelet coefficients are of the order of $O(\Delta x^{\mu})$. From this, it is easy to obtain that at smooth regions, the magnitudes of high frequency coefficients $|\beta_{j,i}|$ have the order of $|f^{(p)}(x)|O(\Delta x^{p})$. By Taylor expansion, if both $|\beta_{j,i-1}|$ and $|\beta_{j,i}|$ are in a smooth region, we have

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

where the constant in the term $O(\Delta x)$ depends on the high order (larger than p) derivatives of $f(x)$. On the other hand, if a stencil contains a discontinuity, no matter a discontinuity in function value ($m = 0$) or in its m-th derivative, the magnitude of the corresponding high frequency coefficient $|\beta_{j,i}|$ is of the order of $O(\Delta x^{(m)})$, i.e.

$$|\beta_{j,i}| = |[f^{(m)}(x_0)]|O(\Delta x^{m}),$$

which is at least one order lower than that at the smooth regions.

Therefore, 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 $|\beta_{j,i}| \leq \tau|\beta_{j,i-1}|$, where $\tau > 1$ is a given constant, then we treat the current stencil as a smooth stencil. Otherwise, we conclude that there are discontinuities contained in it.

The choice of constant τ depends on the grid size Δx, and also the intensity of the jumps. In fact, the ratio between a high frequency coefficient at the rough regions and that at the smooth regions is of the order of $|[f^{(m)}(x)]|O(\Delta x^{(m-p)})$. When Δx becomes small, this ratio is large. We can choose τ as any number such that

$$(1 + O(\Delta x)) \leq \tau \leq \min_{x}\{|[f^{(m)}(x)]|O(\Delta x^{(m-p)})\}, \tag{2.2.1}$$

provided the above minimal number is larger than $1 + O(\Delta x)$. This is always true for piecewise smooth functions with small enough grid size Δx. Obviously, the jump detection procedure can capture all jumps in the m-th derivative with intensity larger than $O(\Delta x^{(p-m)})$. On the other hand, when a jump in the m-th derivative has small intensity, which is even less than $O(\Delta x^{(p-m)})$, this jump can not be detected by the above described method. However, the error caused

by missing this jump is also very small, which is at the same order of the error bound we will give in section 3. In practice, especially when we just care about the jumps in function values, we have a large range to select τ.

The extra cost introduced by this comparison jump identification method over the standard wavelet transforms is just the comparison $|\beta_{j,i}| > \tau|\beta_{j,i-1}|$ for each stencil.

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 [11].

2.3 A Simple Example

In the last part of this section, to better illustrate the idea of the construction of ENO-wavelet transforms, we give a simple example in the ENO-Haar case. We consider computing the transform coefficients of the following initial data containing two discontinuities at $[0, 1]$ and $[2, 10]$ respectively:

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

The standard Haar produces the low and high frequency coefficients:

$$\alpha = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{2}} & \frac{12}{\sqrt{2}} & \frac{23}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{8}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

We notice that comparing to their neighbors, there are two relatively large high frequency coefficients corresponding to the two jumps. The corresponding linear approximation by setting $\beta = 0$ is:

$$\begin{pmatrix} 0 & 0 & 0.5 & 0.5 & 1 & 1 & 6 & 6 & 11.5 & 11.5 \end{pmatrix},$$

which does not recover the discontinuities correctly.

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

$$\begin{pmatrix} & & y & 1 & 1 & 1 & 1 & w & & \\ 0 & 0 & 0 & x & & & z & 10 & 11 & 12 \end{pmatrix},$$

where x, y, z and w 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

(2.3.1)) based on the stencil $(0, x)$ is the same as the previous α_1, which is based on the stencil $(0, 0)$ giving $x = 0$. Similarly, we extend y in a way such that the high frequency coefficient $\bar{\beta}_2$ (boxed in (2.3.1)) is zero giving $y = 1$. Therefore we compute the high frequency coefficients $\hat{\beta}_2$ based on stencil $(0, x)$ and the low frequency coefficients $\bar{\alpha}_2$ based on stencil $(y, 1)$ by using the corresponding standard filters giving $\hat{\beta}_2 = 0$ and $\bar{\alpha}_2 = \frac{2}{\sqrt{2}}$. Similarly, we determine $w = 0$ according to $\hat{\alpha}_4 = \alpha_3$ (boxed in (2.3.1)), then compute $\hat{\beta}_4 = \frac{2}{\sqrt{2}}$, and $z = 10$ by $\bar{\beta}_4 = 0$ and then $\bar{\alpha}_4 = \frac{20}{\sqrt{2}}$. Thus we have the coefficients:

$$\alpha = \begin{pmatrix} & \frac{2}{\sqrt{2}} & \frac{2}{\sqrt{2}} & \boxed{\frac{2}{\sqrt{2}}} & \\ 0 & \boxed{0} & & \frac{20}{\sqrt{2}} & \frac{23}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} & \boxed{0} & 0 & \frac{2}{\sqrt{2}} & \\ 0 & 0 & & \boxed{0} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \quad (2.3.1)$$

Since we know how we extended $\hat{\alpha}_2$, $\bar{\beta}_2$, $\hat{\alpha}_4$ and $\bar{\beta}_4$, we do not need to store them. In fact, we just need to store the low and high frequency coefficients as:

$$\alpha = \begin{pmatrix} 0 & \frac{2}{\sqrt{2}} & \frac{2}{\sqrt{2}} & \frac{20}{\sqrt{2}} & \frac{23}{\sqrt{2}} \end{pmatrix}, \beta = \begin{pmatrix} 0 & 0 & 0 & \frac{2}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix},$$

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

When we reconstruct the linear approximation, we can first recover $\hat{\alpha}_2$, $\bar{\beta}_2$, $\hat{\alpha}_4$ and $\bar{\beta}_4$ 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

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 10 & 11.5 & 11.5 \end{pmatrix}.$$

We notice that the first discontinuity is perfectly retained, and the second one is more accurate than that of the standard transform, although it is not exactly recovered. More importantly, this approximation preserves the discontinuities sharply in contrast to the standard Haar wavelet which takes the average at the discontinuity.

We would like to close this section by making the following two 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) 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.

3 Theory: Error Bound and Stability

In this section, we present the ENO-wavelets approximation error bound for piecewise continuous functions and the stability of the algorithm. We do not give proof. They can be found in [11] and [46].

Given a function $f(x)$ in L^2, in standard wavelet theory [38] [21] [42], it can be linearly approximated by its projection $f_j(x)$ in V_j as in (2.1.7) and (2.1.8). 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 $f(x)$, which has bounded p-th order derivative, in V_j with basis $\phi_{j,k}(x)$, then,*

$$\|f(x) - f_j(x)\| \leq C(\Delta x)^p \|f^{(p)}(x)\|, \tag{3.1}$$

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 its wavelet 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 error bound does not hold if the function does not have finite p-th derivative. 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, the approximation error cannot be controlled as for smooth functions, 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 if they are well separated. In order to assure such error bound, we need to introduce the following definition.

Given a function $f(x)$ which has discontinuous set D, i.e.

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

Denote t by 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 $l+1$, we say a projection of $f(x)$ in space V_j with spatial step $\Delta x = 2^{-j}$ satisfies the* **Discontinuity 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 plus two data points away from other discontinuities. In other words, there are no two consecutive stencils containing two discontinuities. This property will avoid the modifications near one discontinuity interacting with the modifications near other discontinuities.

In fact, 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, see [11].

With the definition of DSP, we are ready to state the error estimate in the following theorem.

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 an interval $[a, b]$ with bounded p-th derivatives in each piece of smooth regions, and $f_j(x)$ is its j-th level ENO-wavelet projection obtained by using the extrapolation methods given in section 2 with choice of τ satisfying (2.2.1). If the projection $f_{j+1}(x)$ satisfies the DSP, then*

$$\|f(x) - f_j(x)\| \leq C(\Delta x)^p \|f^{(p)}(x)\|_{(a,b)\setminus D}, \tag{3.2}$$

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

This theorem implies that the error is uniformly bounded in the smooth regions as well as the discontinuous regions if the jump can be correctly detected. In fact, as we mentioned in section 2.2, with the choice of τ satisfying (2.2.1), all jumps in the m-th derivative with intensity larger than $O(\Delta x^{(p-m)})$ can be captured by the jump detection mechanism for smooth data. On the other hand, although the detection algorithm fails at the small jumps with intensity less than $O(\Delta x^{(p-m)})$, the error caused by missing these jumps is small, and is of the same order of the error generated in smooth regions and can be absorbed into the right hand side of (3.2). Therefore, comparing to the standard error estimation, ENO-wavelet approximation achieves the bound as if there is no discontinuity. In other words, this is the best possible error bound we can hope for. Considering the ENO-wavelet algorithms, which essentially consists of two steps: locating discontinuities and extrapolation using one-sided information in addition to the standard wavelet transforms, this is not a surprising result, because basically at each side of a discontinuity, the ENO-wavelet transforms perform the standard wavelet transforms to the extended function which is smooth. Therefore, they can approximate each side of the jump as accurate as if the function is smooth.

For the same reason, it is also not surprising that if the discontinuities can be correctly identified, both forward and inverse ENO-wavelet transforms for piecewise discontinuous functions are stable with respect to small perturbations. Here, we summarize the stability results into the following theorems.

Theorem 3. *Given a piecewise smooth function $f(x)$, and a function $\tilde{f}(x)$ which is a perturbation of $f(x)$ satisfying*

$$\|f(x) - \tilde{f}(x)\| \leq \epsilon, \tag{3.3}$$

where ϵ is a smooth positive number. Suppose one uses an ENO-wavelet transform with a detection algorithm which can capture the correct discontinuous points in both $f(x)$ and $\tilde{f}(x)$, and denote their ENO-wavelet coefficients (including both low and high frequency coefficients) as α and $\tilde{\alpha}$ respectively, then

$$\|\alpha - \tilde{\alpha}\| \leq C\epsilon, \tag{3.4}$$

where C is a constant independent of $f(x)$ and ϵ.

Theorem 4. *Given $f(x)$ a piecewise smooth function, denote D as the jump set detected by a ENO-wavelet transform, and α the ENO-wavelet coefficients. Assume $\tilde{\alpha}$ is a perturbation of α with*

$$\|\alpha - \tilde{\alpha}\| \leq \epsilon, \tag{3.5}$$

where ϵ is a small positive number. If $\tilde{f}(x)$ is the ENO-wavelet reconstruction with jump set D from coefficients $\tilde{\alpha}$, then

$$\|f(x) - \tilde{f}(x)\| \leq C\epsilon, \tag{3.6}$$

where constant C is independent of $f(x)$ and ϵ.

We note that due to Theorem 4, the inverse transforms are always stable. This is because the inverse transforms depend only on the extrapolation schemes and the standard inverse wavelet transforms, and there is no detection process involved. Therefore, the stability of the extrapolation schemes and the standard wavelet transforms guarantees the stability of inverse ENO-wavelet transforms.

4 Applications

In this section, we briefly discuss some applications of the ENO-wavelet transforms.

4.1 Function Approximation

Constructing approximations to the piecewise continuous functions is a very natural application of the designed ENO-wavelet transform. One simple way is to use the low frequencies $f_j(x)$ to approximate $f(x)$ directly. Here, we use some 1-D numerical examples to illustrate the approximation abilities of the ENO-wavelet transforms. We will demonstrate the error bound 3.2 given in section 3. In particular, we show results for 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 graphical 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.

Since we consider noise free examples in this part, we use the method for noise free data described in section 2.2 to detect the positions of the discontinuities. And we select $a = 2$ (as used in the algorithms in section 2) for all 1-D examples.

Firstly, we compare the approximations for smooth functions. Table 1 shows the results of comparison of DB4 with ENO-DB4 approximations for the function $f(x) = \exp[-(\frac{1}{x} + \frac{1}{1-x})], 0 < x < 1$,

We see from the table 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 2, which agree with the results in Theorem 1. In fact, we notice that in this situation, no singularity is detected, the ENO-wavelet algorithms perform the standard transforms for completely smooth functions as we expected.

Next, we apply Haar and ENO-Haar, DB4 and ENO-DB4, and DB6 and ENO-DB6 transforms to a piecewise smooth function and compare the approximation error. Figure 4 shows the comparison of the order of accuracy in the

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 1. 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..

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 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-DB6 and standard DB6 approximations to a piecewise smooth function in Figure 5. In the left picture, we show the original function (dotted line), the standard wavelet linear approximations (dash-dotted) and the ENO-wavelet approximations (solid line). The right pictures is the zoom-in of the left picture near a discontinuity. We clearly see the Gibbs' oscillations in the standard approximations; in contrast, the ENO-wavelet approximations preserve the jump accurately.

In Figure 6, we also present the standard DB6 wavelet coefficients (dotted line) and the 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 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.

4.2 Image Compression

Digital image compression aims to reduce the storage requirement of digital images with (or without) losing information (they are called lossy (or lossless) compression). Wavelet based lossy image compression algorithms have been the leading methods in high ratio (the ratio of original file size over the compressed file size) compression. The most notable work in this area goes to Shapiro's EZW compression [40]. Many studies have been conducted along this direction, including the remarkable work of Said and Pearlman's coding algorithm based

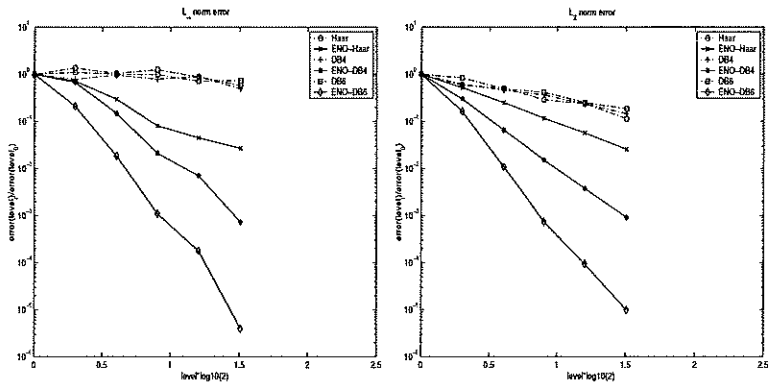

Figure 4. The approximation accuracy comparison of ENO-wavelet and wavelet transforms. Both L_∞ (left) and L_2 (right) order of accuracy show that ENO-wavelet transforms maintain the order 1, 2 and 3 for ENO-Haar, ENO-DB4 and ENO-DB6 respectively and they agree with the results of Theorem 2. In contrast, standard wavelet transforms do not retain the order of accuracy for piecewise smooth functions.

.

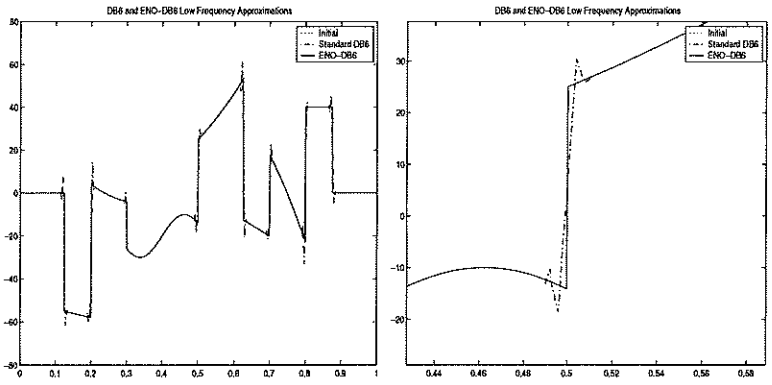

Figure 5. The 4-level ENO-DB6 (solid line) and the standard DB6 (dash-dotted line) Approximation. The standard DB6 generates oscillations near discontinuities, but the ENO-DB6 does not..

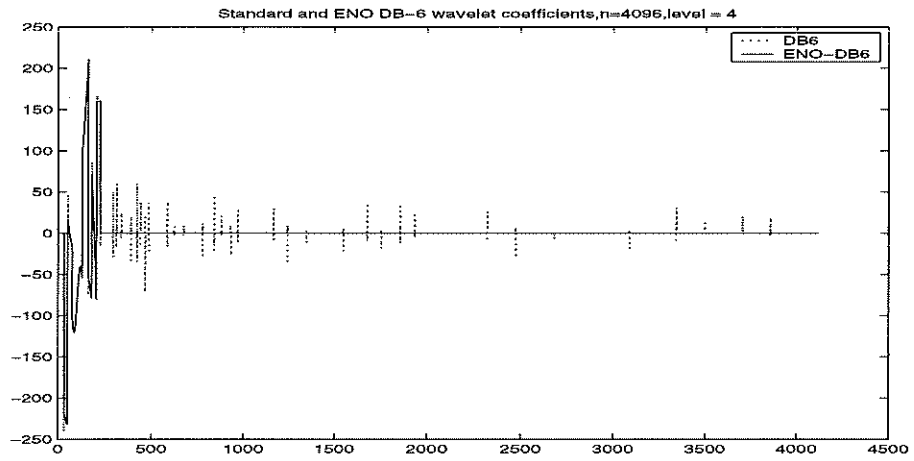

Figure 6. The 4-level ENO-DB6 coefficients (solid line) and the standard DB6 coefficients (dotted line). There are large high frequency coefficients near the discontinuities in the standard DB6 transform but not in the ENO-DB6 transform..

on set partitioning in hierarchical trees (SPIHT) [39].

A wavelet based image compression algorithm usually consists of three steps, namely transform, quantization and coding. Transform means that instead of working directly on the pixel values of the digital images themselves, one uses wavelets transforms to compute the wavelet coefficients so that the spatial correlations between pixels in the original images can be decoupled. In other words, in smooth regions where the pixel values are close, the generated high frequency coefficients have small magnitudes and eventually will not be retained. Quantization refers to truncating the real valued wavelet coefficients into a finite set of fixed values so that they can be used in the coding process. In this step, the small wavelet coefficients are usually quantized to zero. Therefore, the more small wavelet coefficients a transform generates the better compression it achieves. Obviously, this step is a lossy and non-invertible process. Coding converts finite quantized wavelet coefficients into binary bit streams for storage.

How to use ENO-wavelet transforms together with quantization and coding steps to form complete image compression algorithms is an ongoing research topic. Many open questions, especially those questions related to the quantization and coding steps, need to be answered. In this chapter, we do not intend to answer those questions. Instead, we focus on the transform step and discuss the potential of using ENO-wavelet transforms to obtain more efficient compression algorithms. We show pictures to illustrate such potential.

As we have explained, one of the most important reasons for the success of wavelets in image compression is their ability to approximate smooth functions efficiently. Many classes of digital images can be modeled as piecewise

smooth functions connected by large discontinuities (edges and boundaries of objects). Using a small number of coefficients one can obtain very high accuracy approximations to the smooth regions. On the other hand, jumps (edges) indicate important features and should be retained. Standard wavelet transforms require many coefficients to represent jumps (edges) and tend to generate a Gibbs phenomenon called jump (edge) artifacts when compression techniques are applied.

In addition, in order to reconstruct the edges correctly, not only does one need to store those large jump related high frequency coefficients, but also to record their positions (coordinates). Otherwise, the decompression process will not be able to put them in the correct places to rebuild the images. In fact, recording the position information often consumes more space than remembering those coefficient values themselves. How to code the positions of the large coefficients is the most important question answered by EZW and SPIHT algorithms, where the tree structures are used to code both the positions and coefficient values. Therefore, if one can reduce the number of large high frequency coefficients, a better compression may be obtained.

ENO-wavelet transforms give methods which can reduce the number of large high frequency coefficients. ENO-wavelet transforms maintain all advantages of the standard wavelet transforms, such as multiresolution data structures and concentrating energy to fewer large coefficients. Their performance in smooth regions is the same as that of the standard wavelet transforms. More importantly, ENO-wavelet transforms do not generate large high frequency coefficients near jumps because all filters are applied to smooth functions; thus more efficiently represent jumps (edges).

We give a 2-D testing image example to compare the standard Haar and the ENO-Haar reconstructions by keeping certain number of coefficients. Here we use tensor products of 1-D transforms. The original image is shown in Figure 7. Figure 8 is the 3-level standard Haar reconstruction and Figure 9 is the 3-level ENO-Haar reconstruction. Both use only the low frequencies 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 much fuzzier than the ENO-Haar case. This illustrates that the ENO-Haar reconstruction can reduce the edge oscillations for 2-D functions.

In practical compression algorithms, it is rare to use only the low frequency coefficients to build reconstructions. In fact, it is very common to use thresholding techniques. As we mentioned in the introduction, we can also combine ENO-wavelets with thresholding techniques. We show the standard hard thresholding reconstructed image by retaining the largest 64×64 coefficients in Figure 10. We note that more details are recovered, but the edges are still quite fuzzy. Similarly, we can apply the same thresholding techniques to the ENO-wavelet transforms. In Figure 11, we give the reconstructed image by using ENO-Haar hard thresholding technique by keeping the largest 64×64 ENO-Haar coefficients. In this image, edges and more details are preserved.

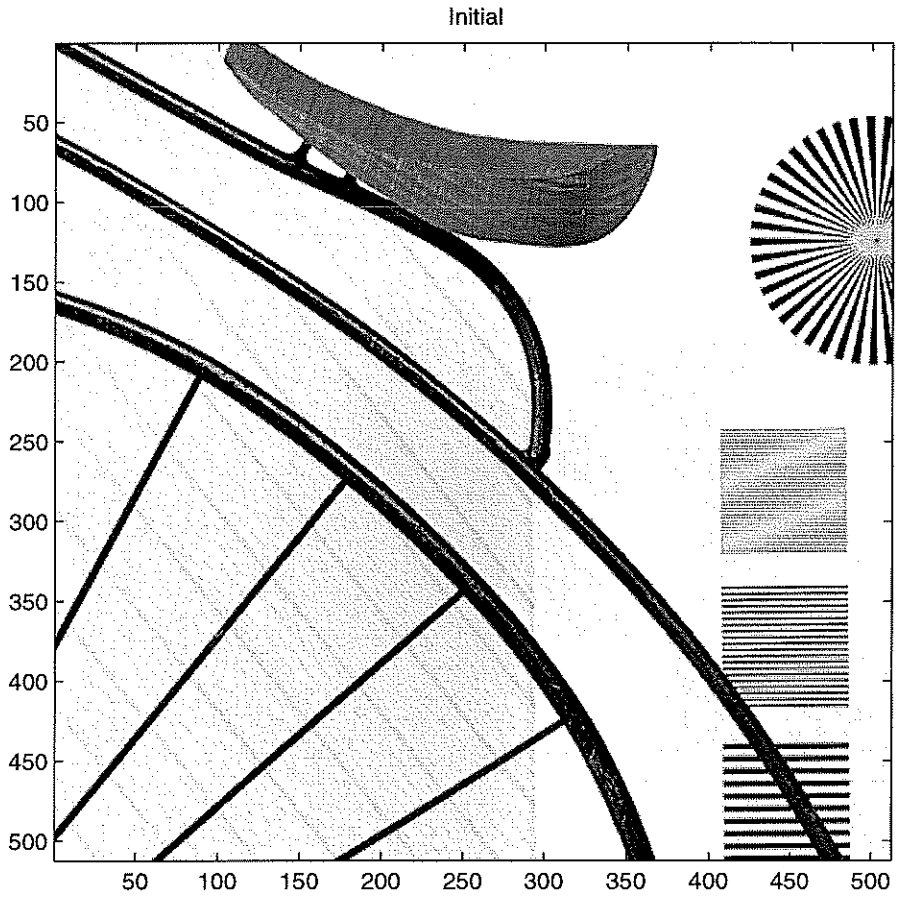

Figure 7. Original 2-d Image.

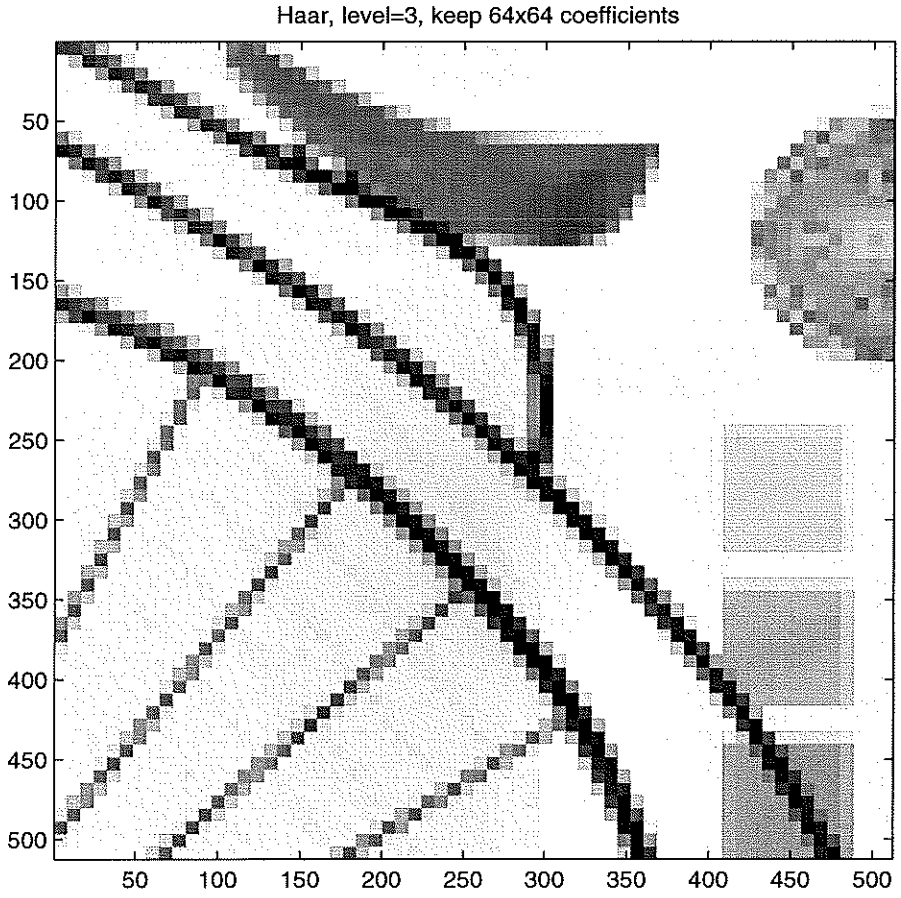

Figure 8. The 3-level standard Haar Approximation, the edges are fuzzier than that in the next picture. Most detail information is lost..

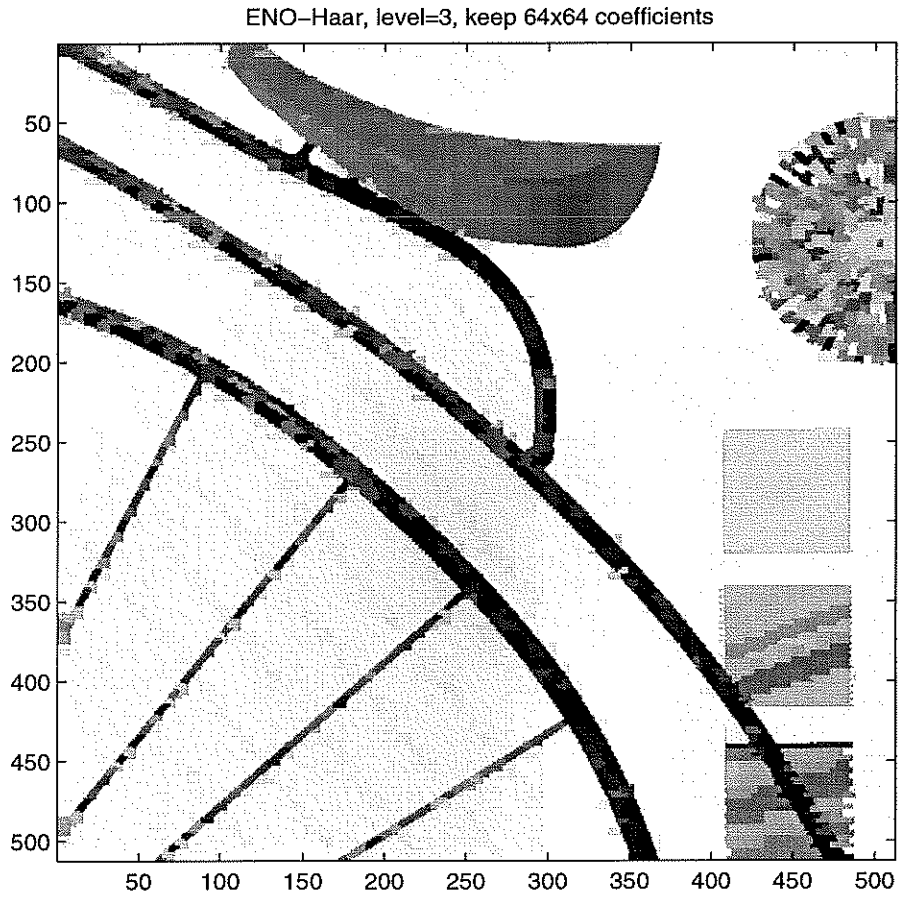

Figure 9. The 3-level ENO-Haar Approximation. Edges are preserved sharply, more detail information is retained. .

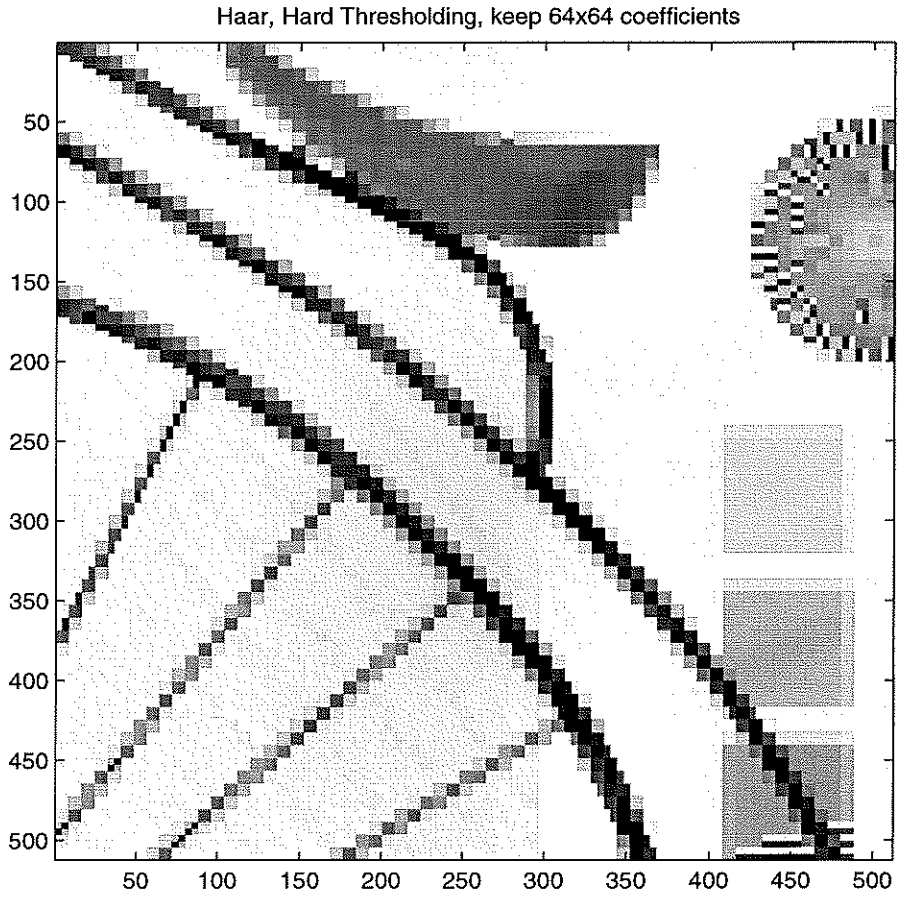

Figure 10. The 3-level standard Haar hard thresholding approximation. More details are preserved. Edges are still fuzzy..

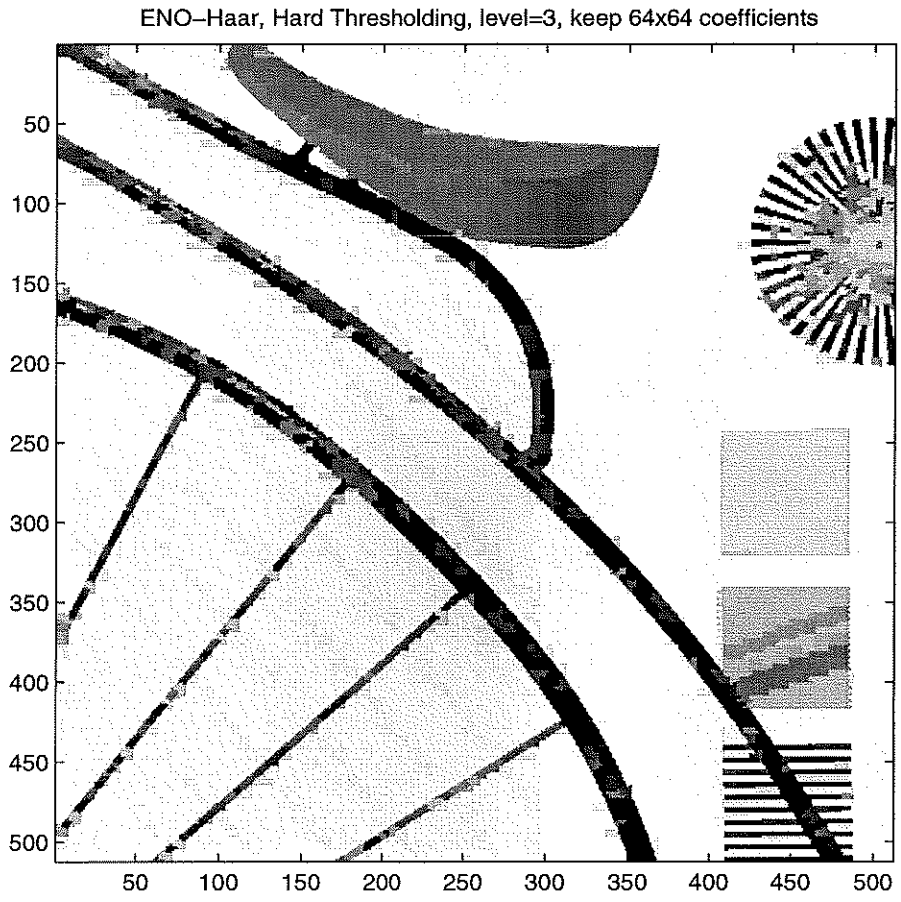

Figure 11. The 3-level ENO-Haar hard thresholding approximation, Most of the edges and interior information are retained and less severe edge artifacts are generated comparing to the previous images..

Of course, in the context of compression, we should not forget that in order to obtain the perfect reconstruction property, we need one extra bit to record the location of each jump in the ENO-wavelet transforms. This extra bit is the price we pay to reduce the magnitudes of all high frequency coefficients related to jumps (the number of jump related large high frequency coefficients could be as many as $\frac{l+1}{2}$ at one jump, where $l+1$ is the length of the high pass filter). In principle, storing those extra bits should be cheaper than storing those large high frequency coefficients and their positions. Furthermore, there is a strong correlation among the spatial positions of those extra bits because they are usually along the edges of the images. Therefore, the extra bits corresponding to the same spatial location in different level of decompositions can also form a tree structure between different levels. All of these suggest that we may further compress those extra bits. However, the best way to accomplish this is under investigation.

4.3 Signal Denoising

Similar to the standard wavelet transforms, it is also very natural to use ENO-wavelet transforms in signal denoising. In this case, the extra bits used to indicate discontinuities in the signal are no longer a problem. Therefore, the design of the algorithms becomes more straight forward. As we said in the introduction, one can modify a signal denoising method based on the standard wavelet transforms to a method based on the ENO-wavelet transforms, simply by replacing the standard wavelet transforms by the ENO-wavelet transforms, for instance, the translation invariant wavelet denoising method [15] can be used to together with the ENO-wavelet transforms.

The major advantage of the ENO-wavelet based denoising over the standard wavelet denoising is that ENO-wavelet transform denoising can remove the oscillation without smearing edges, while the smearing problem is a drawback for standard wavelet denoising algorithms. ENO-wavelet transforms use filtered (less noisy) values to form extensions at jumps which makes the method more stable and more effectively denoises near discontinuities. However, as we explained in section 2.2, the presence noise makes it harder to detect jumps. One often needs to use some heuristic to find the correct locations. Edge detection methods can be used together with the ENO-wavelet transforms.

To illustrate the idea, here we just give an 1-D signal denoising example by using simple linear thresholding (truncating all high frequency coefficients). We apply the ENO-DB6 wavelet transform to a piecewise constant signal polluted by Gaussian random noise (see Figure 12). Here, since we have noise in the data, we do not use 2-nd order polynomial extrapolation as we did to the smooth data, instead, we use least square extrapolation at jumps. Despite the presence of noise in the initial data, the level-6 ENO-DB6 reconstruction (solid line in Figure 13) still retains the sharp edges (see zoom-in picture in Figure 14) compared to the standard DB6 reconstruction (dotted line in Figure 13) which produces

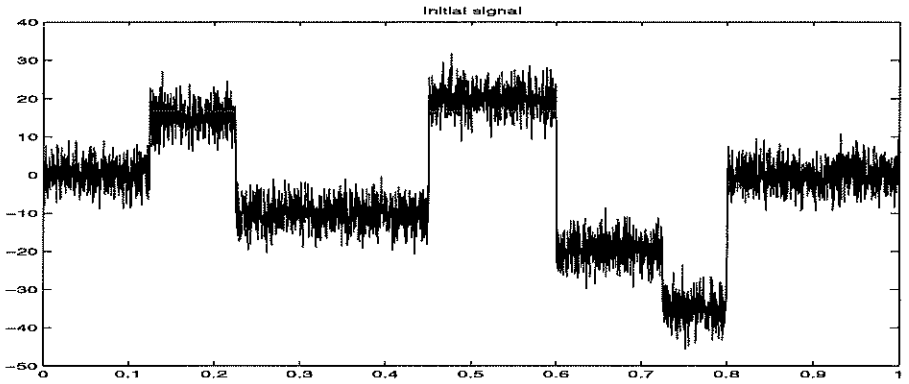

Figure 12. Noisy piecewise constant signal polluted by Gaussian white noise.

oscillations at the discontinuities and also smears them.

Acknowledgement. This work is supported in part by grants ONR-N00017-96-1-0277, NSF DMS-9973341, NSF DMS-0073916 and ARO DAAD19-99-1-0141.

References

[1] S. Amat, F. Arandiga, A. Cohen, R. Donat, G. garcia and M. von Oehsen, *Data Compresion with ENO Schemes: A Case Study*, ACHA 11, 273-288 (2001)

[2] S. Amat, F. Arandiga, A. Cohen and R. Donat, *Tensor product multiresolution analysis with error control for compact image representations*, to appear in Signal Processing, 2002.

[3] L. Anderson, N. Hall, B. Jawerth and G. Peters (1993). *Wavelets on closed subsets of the real line*, in Recent Advances in Wavelet Analysis, Schumaker L.L. and G. Webb (eds.), pp. 1-61, Academic Press, New York, 1993.

[4] F. Arandiga and R. Donat, *A Class of Nonlinear Multiscale Decompositions*, Preprint, 1999.

[5] F. Arandiga and R. Donat, *Nonlinear Multiscale Decompositions: The approach of A. Harten*, Numerical Algorithms 23 (2000) 175-216.

[6] A. Arneodo, *Wavelet Analysis of Fractals: From the Mathematical Concepts to Experimental Reality*, in Wavelets: Theory and Applications, Ed. G. Erlebacher, M. Hussaini, L. Jameson, Oxford Univ. Press, 1996.

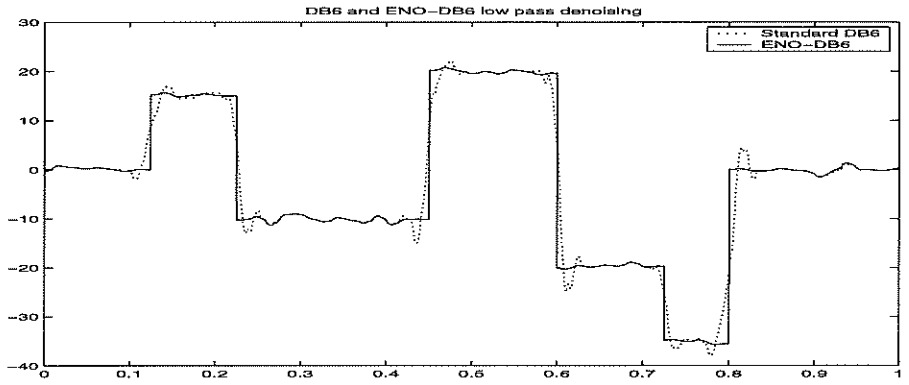

Figure 13. The comparison of the 6-level ENO-DB6 denoising (solid line) with the standard DB6 denoising (dotted line) by truncating high frequencies. The ENO-DB6 reconstruction retains the sharp jumps but the standard DB6 one does not, please see the zoom in picture. .

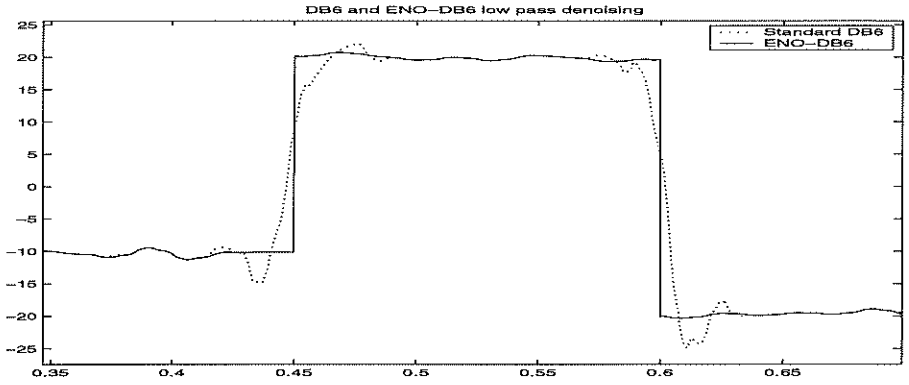

Figure 14. A zoom-in of the denoising example at a discontinuities. The ENO-DB6 reconstruction retains the sharp jumps but the standard DB6 one does not..

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

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

[9] E. Candes and D. Donoho, *Curvelets - A Surprisingly Effective Nonadaptive Representation for Objects with Edges*, Curves and Surfaces, L. L. Schumaker et al. Eds, Vanderbilt University Press, Nashville, TN. 1999.

[10] 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.

[11] T. Chan and H. M. Zhou, *Adaptive ENO-wavelet transforms for Discontinuous Functions*, UCLA CAM Reports, 99-21, June, 1999 (also submitted to SIAM, Numer. Anal.).

[12] T. Chan and H. M. Zhou, *Optimal Construction of Wavelet Coefficients Using Total Variation Regularization in Image Compression*, UCLA CAM Reports, 00-27, July, 2000.

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

[14] R. Coifman and A. Sowa, *New Type of Total Variation Diminishing Flows*, Preprint.

[15] R. Coifman and D. Donoho, *Translation Invariant De-Noising*, Wavelets and Statistics, A. Antoniadis and G. Oppenheim Eds, Springer-Verlag, 1995, pp125-150.

[16] A. Cohen, I. Daubechies, B. Jawerth, and P. Vial, *Multiresolution analysis, wavelets and fast algorithms on an interval*, Comptes Rendus Acad. Sci. Paris, 316 (Serie 1), pp. 417–421, 1993.

[17] A. Cohen, I. Daubechies, and P. Vial, *Wavelets on the interval and fast wavelet transforms*, Applied and Computational Harmonic Analysis 1, pp. 54–81, 1993.

[18] A. Cohen and B. Matei, *Compact representations of images by edge adapted multiscale transforms*, to appear in the proceedings of the IEEE ICIP conference, Tessaloniki, 2001.

[19] 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 Submitted to IEEE Tran. on Image Proc., Preprint, 1999.

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

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

[22] S. Durand and J. Froment, *Artifacts Free Signal Denoising with Wavelets*, in the Proceedings of ICASSP 2001 VI, Salt Lake City, Utah, May 7-11, 2001, pp3685-3689.

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

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

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

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

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

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

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

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

[31] A. Harten, *Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws*, Comm. Pure Appl. Math. 48(1995), pp1305-1342.

[32] 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.

[33] E. Hernandez and G. Weiss, *A First Course on Wavelets*, CRC Press, 1996.

[34] S. Jaffard, *C. R. Acad. Sci. Paris*, Serie I, 79, 1989, pp.308.

[35] E. Le Pennec and S. Mallat, *Image Compression with Geometrical Wavelets*, in IEEE Conference on Image Processing(ICIP), Vancouver, September, 2000.

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

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

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

[39] A. Said and W. Pearlman, *A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*, IEEE Trans. Circ. and Sys. for Video Tech., 6(3), 1996, pp243-250.

[40] J. Shapiro, *Embedded Image Coding Using Zerotrees of Wavelet Coefficients*, IEEE Trans. on Signal Processing, 41(12): pp.3445-3462, 1993.

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

[44] P. Tchamitchian, *Wavelets, Functions, and Operators*, in Wavelets: Theory and Applications, Ed. G. Erlebacher, M. Hussaini, L. Jameson, Oxford Univ. Press, 1996.

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

[46] H. M. Zhou and T. Chan, *The Stability of Adaptive ENO-wavelet Transforms for Piecewise Contiunous Functions*, Preprint.

Tony F. Chan
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90095-1555
chan@math.ucla.edu

Hao-Min Zhou
Department of Applied and Computational Mathematics
California Institute of Technology, Mail Code 217-50
Pasadena, CA 91125
hmzhou@acm.caltech.edu