W-Matrices, Nonorthogonal Multiresolution Analysis, and Finite Signals of Arbitrary Length

Man Kam Kwong and P. T. Peter Tang*

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439-4844
E-mail: kwong@mcs.anl.gov tang@mcs.anl.gov

1 Introduction

Wavelet theory and discrete wavelet transforms have had great impact on the field of signal and image processing. For references to the extensive literature on the subject, we refer to the bibliographies found in [1,2]. In this paper we propose a new class of discrete transforms. It "includes" the classical Haar and Daubechies transforms. Our transforms treat the endpoints of a signal in a different manner from that of conventional techniques. This new approach allows us efficiently to handle signals of any length; thus, one is not restricted to work with signal or image sizes that are multiples of a power of 2. A different method for dealing with signals of arbitrary length was given earlier by Taswell and McGill [4]. Our method does not lengthen the output signal and does not require an additional bookkeeping vector.

*This is an abridged version of a forthcoming paper. This work was supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

An exciting result is the uncovering of a new and simple transform (presented in Section 3) that performs very well for compression purposes. It has compact support of length 4, and so is its inverse. The coefficients are symmetrical, and the associated scaling function is fairly smooth (it is the quadratic B-spline). The associated dual wavelet has vanishing moments up to order 2. Numerical results comparing the performance of our transform with that of the Daubechies D_4 transform are given in Section 4. The multiresolution decomposition, however, is not orthogonal. We will see why this apparent defect is not a real problem in practice. Furthermore, we will give in Section 5 a method to compute an orthogonal compensation that gives us the best approximation possible with the given scaling space.

Our transform can be described completely within the context of matrix theory and linear algebra. Thus, even without prior knowledge of wavelet theory, one can easily grasp the concrete algorithm and apply it to specific problems within a very short time, without having to master complex functional analysis. At the end of the paper, we shall make the connection to wavelet theory.

The experiments mentioned in this article were carried out in Matlab; most of the formulas, especially those involving matrix inverses, were derived by using Maple. We acknowledge the usefulness of these excellent packages. A description of our Matlab implementation can be found in [5]. The paper and the Matlab toolbox can be obtained through ftp at `info.mcs.anl.gov` under the directory `/pub/W-transform`.

2 Motivation – the Haar and Daubechies D_4 Transforms

Our goal is signal compression, and we look at two well-known wavelet transforms in the light of this objective. We permit lossy compression. The conventional strategy is to discard data that are small, because their contribution to the perception of the signal is not sufficiently significant, and to filter out high-frequency components of the signal, because the human ear and eye are not very sensitive to highly oscillatory signals.

In the method of transform coding, an invertible transform is first applied to the signal to produce an alternative but equivalent representation, before discarding small data and/or data that correspond to high-frequency components. A transform is well suited for compressing a given class of signals if it produces considerable near-zero data for most signals in the class in the equivalent representation.

2.1 The Haar Transform

We are given a finite discrete signal $\mathbf{x}$ of length $2n$. More precisely, $\mathbf{x} = [x_1, x_2, \cdots, x_{2n}]'$ is a column vector ($'$ denotes matrix transpose). We divide the vector into n ordered pairs of numbers, (x_1, x_2), $(x_3, x_4), \cdots, (x_{2n-1}, x_{2n})$. For each pair, we generate a new pair consisting of their sum and difference, for example, $(y_1, y_2) = (x_1 + x_2, x_1 - x_2)$. This is the 2×2 Haar transform.

By keeping the new numbers y_i instead of x_i, no information is lost, since each pair of x_i can be recovered from the corresponding pair of y_i by taking their sum and difference and then dividing by 2. Suppose that the original signal is slowly varying, with an occasional jump here and there. Then a majority of the y_i, for i even, will be small. The new representation $\mathbf{y}$ is thus more apt for our compression goal.

Since the odd and even components of $\mathbf{y}$ are computed in different ways, it makes sense to rearrange them into two smaller vectors $\mathbf{y}_1 = [\text{odd } y_i]'$ and $\mathbf{y}_2 = [\text{even } y_i]'$. The components of $\mathbf{y}_2$ are, one hopes, mostly small; so after discarding those below some threshold value, only a small portion remains. There is no need to further compress the survivors.

The components of $\mathbf{y}_1$, on the other hand, are not necessarily small. In fact, it is a low-pass-filtered output that retains the general shape of the original signal. This output carries the correlation between points that are twice as far apart in the original signal (two adjacent values in $\mathbf{y}_1$ represents four components in $\mathbf{x}$). It is natural to iterate the Haar transform on $\mathbf{y}_1$ to gain further compression, ad infinitum. The algorithm, consisting of a chain of transforms applied to the progressively shorter low-pass-filtered signals, was formerly called a pyramidal scheme and recently was recast into the concept of the renowned *multiresolution analysis*. One of the attractive features of multiresolution analysis is the self-similarity of the iterates. Exactly the same procedure is used to transform the principal component signal at each level, albeit the signal vector becomes shorter every time.

The Haar transform has an obvious matrix formulation. Let $\mathbf{H}$ be the $2n \times 2n$ block diagonal matrix

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & & & & & \\ 1 & -1 & & & & & \\ & & 1 & 1 & & & \\ & & 1 & -1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & 1 \\ & & & & & 1 & -1 \end{pmatrix}. \tag{2.1}$$

Then

$$\mathbf{y} = \mathbf{H}\mathbf{x} \tag{2.2}$$

and (for the inverse transform)

$$\mathbf{x} = \mathbf{H}\mathbf{y}/2 \tag{2.3}$$

The Haar matrix $\mathbf{H}$ is not orthogonal, but it can be easily converted into one by dividing by $\sqrt{2}$. For computational purposes, the current form of $\mathbf{H}$ is preferable.

The Haar transform is easy to understand, but it lacks sophistication. Nonconstant signals do not lead to zero $\mathbf{y}_2$. The transform also does not have sufficient smoothness (this can be adequately explained only by referring to the scaling function associated with the transform; see Section 7).

2.2 The Daubechies D_4 Transform

Daubechies made a significant contribution when she constructed higher-order orthogonal wavelets of compact support that led to discrete wavelet transforms generalizing the classical Haar transform [2]. To fully understand the beauty of her wavelets and the motivation for imposing certain properties requires a fair amount of reading. However, her wavelet transforms can still be appreciated and applied without delving too much into the technical details of wavelet theory. What follows is our attempt to explain her D_4 discrete transform succinctly.

Daubechies has found two magic sets of four numbers each:

$$\left[\ g_1,\quad g_2,\quad g_3,\quad g_4\ \right] = \frac{\sqrt{3}+1}{4\sqrt{2}}\left[\ 1,\quad \sqrt{3},\quad 2\sqrt{3}-3,\quad \sqrt{3}-2\ \right] \tag{2.4}$$

and

$$\left[\ h_1,\quad h_2,\quad h_3,\quad h_4\ \right] = \frac{\sqrt{3}-1}{4\sqrt{2}}\left[\ 1,\quad \sqrt{3},\quad -3-2\sqrt{3},\quad 2+\sqrt{3}\ \right]. \tag{2.5}$$

These numbers satisfy some remarkable yet easily verifiable orthogonal properties, which can be stated in the following form.

Stack n copies of the sets of numbers to form the $2n \times 2n$ matrix

$$\mathbf{D} = \begin{pmatrix} g_1 & g_2 & g_3 & g_4 & & & & & & \\ h_1 & h_2 & h_3 & h_4 & & & & & & \\ & & g_1 & g_2 & g_3 & g_4 & & & & \\ & & h_1 & h_2 & h_3 & h_4 & & & & \\ & & & & \ddots & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & & & g_1 & g_2 \\ & & & & & & & & h_1 & h_2 \end{pmatrix}. \tag{2.6}$$

In the first two rows, the numbers are flushed to the left. In subsequent pairs of rows, the numbers are shifted successively by two positions to the right. Note that the last two rows can hold only the first two numbers of each set. Then

$$\mathbf{D}\mathbf{D}' = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & \times & \times \\ & & & & & \times & \times \end{pmatrix}, \tag{2.7}$$

which is almost the $2n \times 2n$ identity matrix (if not for the 2×2 submatrix at the lower right-hand corner).

Analogous to the Haar transform (2.2) discussed in Subsection 2.1, the Daubechies matrix transforms a given signal $\mathbf{x}$ into a new signal $\mathbf{y}$:

$$\mathbf{y} = \mathbf{D}\mathbf{x}. \tag{2.8}$$

The original signal (or at least the first $2n-2$ components of) $\mathbf{x}$ can be recovered as

$$\mathbf{x} \doteq \mathbf{D}'\mathbf{y}, \tag{2.9}$$

where $\doteq$ indicates that equality holds only for the first $2n-2$ components.

Many techniques have been devised to deal with the problem caused by the last two components of the signal — by using periodic or even extension, zero padding, etc. We present a new method in this paper.

Also analogous to the Haar transform, the odd components of $\mathbf{y}$ are a low-pass-filtered output which retains the general shape of $\mathbf{x}$, while the even components of $\mathbf{y}$ are mostly very small. Indeed, Daubechies has, in her search for the h_i, imposed the condition that they will transform any linear function to zero (in the language of wavelet theory, the wavelet has vanishing moments of orders 0 and 1).

The multiresolution analysis defined by D_4 is the algorithm consisting of multiplying $\mathbf{x}$ by the matrix $\mathbf{D}$, separating out the odd and even components of the output to form two vectors $\mathbf{y}_1$ and $\mathbf{y}_2$, and then repeating the process on $\mathbf{y}_1$. Each level of transform can be easily reversed (if one ignores the problem with the endpoints, for the time being) — merge $\mathbf{y}_1$ and $\mathbf{y}_2$ into a longer vector by interlacing their components and then multiply by D'.

Although both the Haar transform and the Daubechies D_4 transform define invertible multiresolution analysis, the latter is better for at least two reasons. It produces near-zero $\mathbf{y}_2$ for more signals, and its scaling function is continuous. The Daubechies D_4 wavelet is still not perfect because it is not symmetrical, not smooth (the graphs contain an infinite number of sharp cusps), and not intuitive. Higher-order Daubechies wavelets have progressively better smoothness properties.

3 W-Matrices and Some New Transforms

In the preceding section, we noted that the crucial element in the multiresolution algorithm is a suitable transform "matrix" such as **H** or **D**. This "matrix" is really a family of matrices of sizes $2n \times 2n$ (for all positive integers n), all having the same structure. Three properties are essential for constructing the multiresolution analysis algorithm.

1. With the exception of two rows (more rows may be needed for higher order transforms) at the top or bottom of the matrix, the other rows come in pairs (let us call them the g- and h-vectors). Each pair is obtained from the previous pair by a shift of two positions to the right.

2. Each row has only a small number of nonzero elements.

3. Each matrix is near-orthogonal, in the sense that the product of each matrix with its transpose is almost identical to the identity matrix.

Two additional properties are desirable for compression and other purposes.

- The g-vector is associated with a sufficiently smooth scaling function.

- The h-vector has vanishing moments up to some high order.

How important is the near-orthogonality of the matrix? Our work is based on a relaxation of this requirement. Orthogonality makes it easy to find the inverse of the transform — simply multiply $\mathbf{y}$ by the transpose of the matrix. We can, however, achieve the same goal by requiring that the inverse of the matrix is easy to find and that it has a small number of nonzero elements (so that the inverse transform can be efficiently implemented).

We define a W-matrix as one for which both it and its inverse satisfy conditions 1 and 2. The first question that comes to mind is: Are there any simple W-matrices besides the orthogonal ones? Examples are given below.

A more subtle point concerning orthogonality is the "stability" of the inverse transform. With nonorthogonal matrices, small data in the transformed output may not correspond to small input data if the transform matrix has a large condition number. The W-matrices of our example do have a reasonable condition number.

3.1 The Quadratic Spline W-matrices

We call our first example of W-matrix the quadratic spline W-matrix because, as we will see in Section 7, its associated scaling function is the well-known quadratic B-spline.

The W-matrices of even order in the family have the form

$$\mathbf{K} = \begin{pmatrix} 2 & 3 & -1 & & & & & & & \\ 2 & -3 & 1 & & & & & & & \\ & -1 & 3 & 3 & -1 & & & & & \\ & -1 & 3 & -3 & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & -1 & 3 & 3 & -1 & \\ & & & & & -1 & 3 & -3 & 1 & \\ & & & & & & & -1 & 3 & 2 \\ & & & & & & & -1 & 3 & -2 \end{pmatrix}. \tag{3.1}$$

The building blocks are the g- and h-vectors $[-1, 3, 3, -1]$ and $[-1, 3, -3, 1]$.

The top two rows and bottom two rows are obtained from the basic vectors by adding the number(s) that has (have) been cut off to the nearest neighborhood that is retained. The inverse of $\mathbf{K}$ is

$$\mathbf{K}^{-1} = \frac{1}{16} \begin{pmatrix} 4 & 4 & & & & & & & & \\ 3 & -3 & 1 & 1 & & & & & & \\ 1 & -1 & 3 & 3 & & & & & & \\ & & 3 & -3 & 1 & 1 & & & & \\ & & 1 & -1 & 3 & 3 & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 3 & -3 & 1 & 1 \\ & & & & & & 1 & -1 & 3 & 3 \\ & & & & & & & & 4 & -4 \end{pmatrix}. \tag{3.2}$$

Table 1 gives a simple example of a two-level decomposition. The first column is the input signal. The second and third columns are the first-level transformed signals using the matrix $\mathbf{K}$ in (3.1). The last two columns are the second-level transformed signals using the odd-sized matrix $\mathbf{K}$ given by (3.4) in the next subsection.

Table 1. Two-level multiresolution analysis of $\mathbf{x}$ using the quadratic spline transform

$\mathbf{x}$	$\mathbf{y}$	$\mathbf{y}_1, \mathbf{y}_2$	$\mathbf{y}$	$\mathbf{y}_1, \mathbf{y}_2$
1	-1	-1	-198	-198
8	5	140	194	2893
27	140	616	2893	16755
64	6	1620	523	194
125	616	3675	16755	523
216	6	-1		
343	1620	6		
512	6	6		
729	3675	6		
1000	-325	-325		

This algorithm is very easy to implement. We will see in the next section that this example is only a special case of a general class of W-matrices — *given any set of four numbers* $[h_1, h_2, h_3, h_4]$ *with* $h_1 h_4 \neq h_2 h_3$*, one can supplement it with an* g*-vector to form a* W*-matrix.* Our Matlab implementation encompasses the general W-matrix transform.

The quadratic spline W-matrix given in this subsection is particularly good for compression because its h-vector has vanishing moments up to order two. Furthermore, the exact inverse of $\mathbf{K}$ is used in the restoration step; hence, the endpoints of the signal $\mathbf{x}$ will be recovered exactly, without any need to extend $\mathbf{x}$ either periodically or with zero padding. This strategy avoids the possible introduction of an artificial discontinuity.

Although the inverse matrix $\mathbf{K}^{-1}$ has a structure similar to that of $\mathbf{K}$, it is more appropriate to think of $\mathbf{K}^{-1}$ as being built by columns instead of rows. The building blocks of $\mathbf{K}^{-1}$ are thus the $\overline{g}$-vector and $\overline{h}$-vector

$$[1, 3, 3, 1] \quad \text{and} \quad [1, 3, -3, -1]. \tag{3.3}$$

They are dual to the g- and h-vectors of $\mathbf{K}$, respectively. The columns of the matrix come in pairs, made up of the transpose of the above two basic vectors; successive pairs of columns are shifted two positions downwards.

3.2 Odd-sized W-Matrices

Classical discrete wavelet transforms require the length of the input signal to be an even number. As a consequence, the length of signals that can have J levels of multiresolution analysis must have a factor of 2^J. Our approach to treat signals of odd length is to transform them with a W-matrix of odd size. For the family of quadratic spline W-matrices, we let (there is more than one possible choice) the odd-sized matrices be

$$\mathbf{K} = \begin{pmatrix} 2 & 3 & -1 & & & & & & \\ 2 & -3 & 1 & & & & & & \\ & -1 & 3 & 3 & -1 & & & & \\ & -1 & 3 & -3 & 1 & & & & \\ & & & & \ddots & & & & \\ & & & & & -1 & 3 & 3 & -1 \\ & & & & & -1 & 3 & -3 & 1 \\ & & & & & & & -1 & 5 \end{pmatrix}. \tag{3.4}$$

Its inverse can be obtained from (3.2) by first deleting the last row and last column and then replacing the lower right 2×3 submatrix by

$$\frac{1}{28}\begin{bmatrix} 5 & -5 & 2 \\ 1 & -1 & 6 \end{bmatrix} \tag{3.5}$$

The transformed signal $\mathbf{y} = \mathbf{K}\mathbf{x}$ has the same length as $\mathbf{x}$. After separating the odd and even components, the vector $\mathbf{y}_1$ has one component more than the vector $\mathbf{y}_2$. Since the total number of components of the output signal is always the same as the length of $\mathbf{x}$, the latter can be restored without additional information (other than the number of multiresolution analysis levels applied).

See Taswell [4] for a different technique in dealing with signals of odd length. Our method is more efficient because the transformed signal does not increase in length and there is no need to use an additional bookkeeping vector.

3.3 Other Examples of W-Matrices

Note that each row of $\mathbf{K}$ has at most four nonzero elements. We say that $\mathbf{K}$ is of order 4. The Daubechies and Haar transforms have order 4 and 2, respectively. W-matrices of a higher order lead to computationally more extensive multiresolution analyses. One can obtain smoother transforms, however, using suitable choices of g- and h-vectors.

The order 6 generalization of the quadratic spline W-matrix is built from the vectors

$$\left[\; 1, \quad -5, \quad \tfrac{20}{3}, \quad \tfrac{20}{3}, \quad -5, \quad 1 \;\right], \tag{3.6}$$

and

$$\left[\; -1, \quad 5, \quad -10, \quad 10, \quad -5, \quad 1 \;\right]. \tag{3.7}$$

For even-sized matrices, the first and last two rows of the matrix have only four nonzero elements. For odd-sized matrices, the last three rows have to be modified. For each of these rows, the numbers that are cut off are added to the next surviving number.

The inverse matrix is built (by columns) by using the dual vectors

$$\left[\; 1, \quad 5, \quad 10, \quad 10, \quad 5, \quad 1 \;\right] \text{ and } \left[\; 1, \quad -5, \quad \tfrac{20}{3}, \quad -\tfrac{20}{3}, \quad 5, \quad -1 \;\right]. \tag{3.8}$$

The exact form of $\mathbf{K}^{-1}$ can be found by using either Matlab or Maple.

W-matrices of odd orders also exist. For example, the analogous spline matrix of order 3 is generated by the basic vectors

$$\left[\; -1, \quad 2, \quad 1 \;\right] \text{ and } \left[\; -1, \quad 2, \quad -1 \;\right] \tag{3.9}$$

(note that one of them is not symmetric) and the inverse by the dual vectors

$$\left[\; 1, \quad 2, \quad 1 \;\right] \text{ and } \left[\; -1, \quad -2, \quad -1 \;\right]. \tag{3.10}$$

Generalized W-matrices can be easily constructed with a set of more than two basic vectors, or with more than one set of basic vectors. Examples are given in the full paper.

4 Numerical Results

We have carried out experiments with W-matrix transforms, in particular, comparing the quadratic spline (QS) and D_4 transforms in compressing various types of signals. Our conclusion is that for reasonably smooth signals, with occasional jumps, QS outperforms D_4. The situation is reversed for signals having high levels of high-frequency components or random noise. This fact does not indicate weakness of the QS transform, however. It means only that the transform tends to filter out high-frequency components and noise, to which most human eyes and ears are insensitive. More experiments are needed to test the physiological aspects on observers.

4.1 One-dimensional Signals

In the experiments reported below, an input signal $\mathbf{x}$ is decomposed into $\mathbf{y}_1$ and $\mathbf{y}_2$ using, in turn, the QS and D_4 transforms. A certain number N of the largest (in absolute value) components of $\mathbf{y}_2$ are retained, and an approximation to the original signal is obtained by using $\mathbf{y}_1$ and the compressed $\mathbf{y}_2$. The relative errors between the original and the restored approximation, measured using the Euclidean l^2 norm, are given.

Our first example uses the signal $\mathbf{x} = \sin((1{:}100)/10) + 0.2\sin((1{:}100)/2)$, where 1:100 denotes the vector $[1, 2, 3, \cdots, 100]$. In Table 2, the first column gives the number of components in $\mathbf{y}_2$ that are retained. The third and fourth columns give the relative l^2 error of the restored signal, for the QS and D_4 transforms, respectively. The second column gives the error of the restored signal using the QS transform with orthogonal compensation, an additional step (explained in Section 5) that can be applied to reduce the error.

The original signal is plotted in Figure 1 alongside the two approximate signals, restored by using the largest three components of $\mathbf{y}_2$. To display the graphs better, we displaced the restored signals by ± 0.2, respectively — the middle graph is the original signal, the upper graph is from the QS transform, and the lower graph is from the D_4 transform. The figure clearly shows that the D_4 restored signal has some sharp corners, due to the non-smoothness of the D_4 scaling function. The sharp corners are even more pronounced when the multiresolution analysis is continued to higher levels.

Table 2. Errors in the restored signal from the QS and D_4 transforms

N	QSOC	QS	D_4	N	QSOC	Q.S.	D_4
3	0.003203	0.003762	0.019268	15	0.003030	0.003255	0.013061
5	0.003163	0.003676	0.018245	18	0.002901	0.003067	0.011690
8	0.003102	0.003554	0.016651	20	0.002717	0.002848	0.010788
10	0.003073	0.003469	0.015579	23	0.002394	0.002496	0.009458
13	0.003055	0.003336	0.013983	25	0.002157	0.002248	0.008495

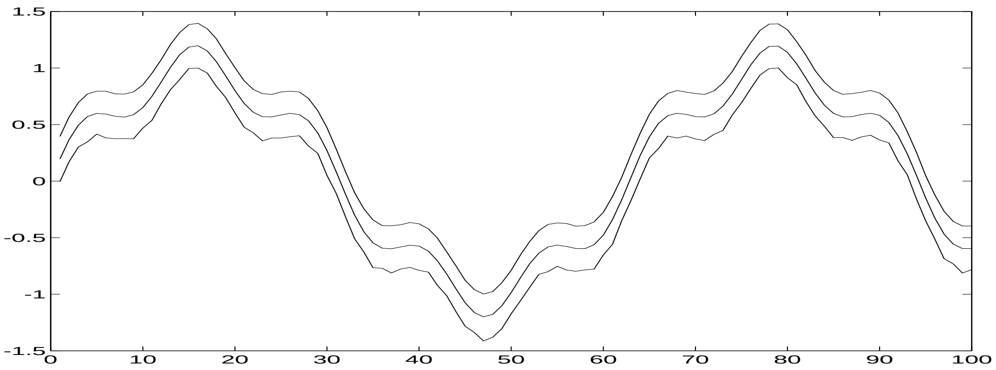

Figure 1. Example 1, $N = 3$, middle curve – original signal

The second example uses $\mathbf{x} = \sin((1{:}100)/10) + 0.2\sin((1{:}100))$; see Table 3. Experiments with signals having occasional jumps produce similar results; see Figure 2.

Table 3. Errors in the restored signal from the QS and D_4 transforms

N	QSOC	QS	D_4	N	QSOC	Q.S.	D_4
3	0.031252	0.037637	0.067649	15	0.021298	0.024515	0.048248
5	0.029834	0.035672	0.064475	18	0.018048	0.020580	0.043103
8	0.027535	0.032539	0.059749	20	0.015440	0.017590	0.039697
10	0.025854	0.030324	0.056537	23	0.013596	0.015300	0.034925
13	0.023110	0.026946	0.051601	25	0.012358	0.013816	0.031947

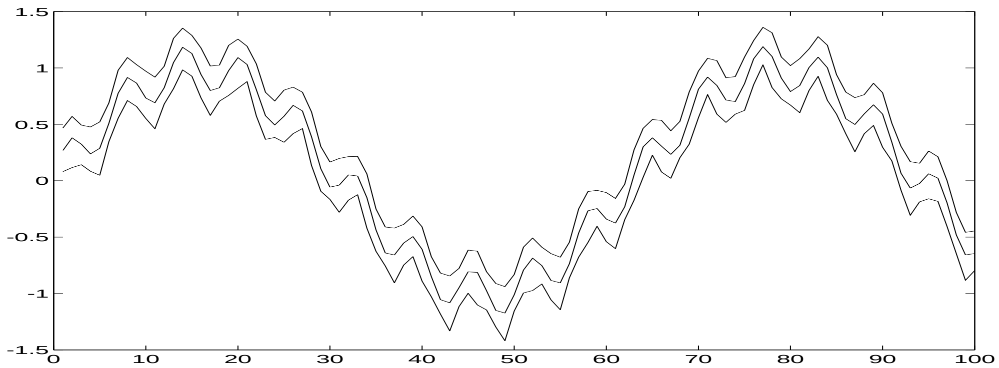

Figure 2. Example 2, $N = 3$, middle curve – original signal

Figures 3 shows the results of a three-level compression of the signal in the first example.

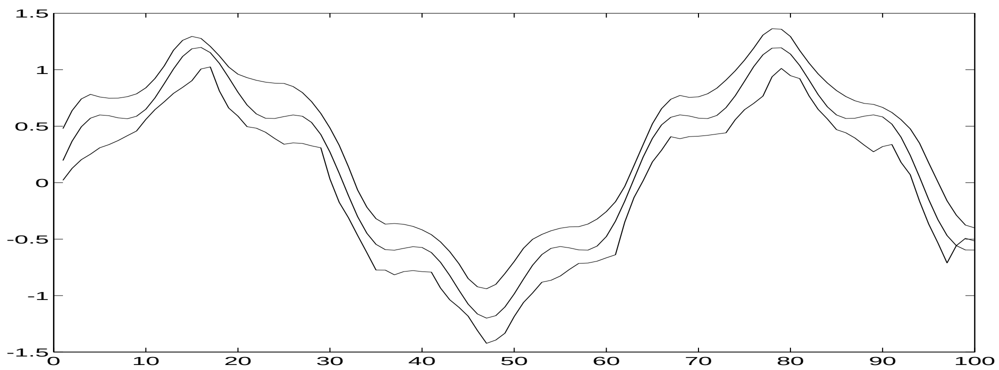

Figure 3. Example 5, three-level compression, fraction of $\mathbf{y}_2$ retained at the three levels are $0.05, 0.2, 0.4$, respectively, middle curve – original signal

4.2 Image Compression

The following figures show how our transform performs in compressing the ubiquitous Lena image, using a three-level analysis constructed from the 2D version of the QS transform.

Figure 4. Original Lena image

At each level, four half-sized matrices are produced. The first carries the general shape of the original picture, while the others contain the details. Compression is achieved by quantization. In the particular experiment given here, each of the first three submatrices at each level is quantized with the same quantization level (which is 800, 5000, and 20000 for the three levels, respectively) and the fourth is quantized at 4 times of that level. The number of nonzero wavelet coefficients retained at the three levels are 624, 8138, and 8331, respectively. Taking into consideration the number of bits needed to store each number, we estimate that the compression ratio is about 13.5 times. Entropy coding may be applied to raised the ratio slightly.

Figure 5. Lena image restored from data compressed 13.5 times

5 Orthogonal Compensation

The QS transform can be summarized in the equations

$$\mathbf{y} = \mathbf{K}\mathbf{x}, \quad \mathbf{x} = \mathbf{K}^{-1}\mathbf{y}. \tag{5.1}$$

The odd and even components of $\mathbf{y}$ form the pair of vectors

$$\mathbf{y}_1 = [y_{11}, y_{12}, \cdots]', \quad \mathbf{y}_2 = [y_{21}, y_{22}, \cdots]'. \tag{5.2}$$

Let us denote the columns of the matrix $\mathbf{K}^{-1}$ as

$$\left[\begin{array}{ccccc} \overline{\mathbf{g}}_1 & \overline{\mathbf{h}}_1 & \overline{\mathbf{g}}_2 & \overline{\mathbf{h}}_2 & \cdots \end{array}\right]. \tag{5.3}$$

Then the second equation in (5.1) has the equivalent form

$$\mathbf{x} = (y_{11}\overline{\mathbf{g}}_1 + y_{12}\overline{\mathbf{g}}_2 + \cdots) + (y_{21}\overline{\mathbf{h}}_1 + y_{22}\overline{\mathbf{h}}_2 + \cdots). \tag{5.4}$$

This equation suggests that the QS transform can be interpreted as the decomposition of $\mathbf{x}$ along the subspaces, G and H, spanned by translates of the dual basic $\overline{g}$- and $\overline{h}$-vectors (with appropriate modifications at the boundary), respectively.

In the analogous interpretation of the Haar and D_4 transforms, the linear subspaces G and H are orthogonal to each other. In addition, the one-dimensional subspaces generated by all the $\overline{\mathbf{g}}_i$ and $\overline{\mathbf{h}}_i$ are mutually orthogonal. When some of the components in $\mathbf{y}_2$ are discarded, the compressed vector is then the unique signal, in the space spanned by the remaining base vectors, that best approximates the original signal.

For the QS transform, G and H are not orthogonal. Neither are the one-dimensional subspaces generated by $\overline{\mathbf{g}}_i$ and $\overline{\mathbf{h}}_i$. Hence, the compression step will not give the optimal approximate signal, representable by the remaining base vectors. This fact seems to argue against the use of the QS transform. In practice, a reasonable signal (one that is not wildly oscillating or badly degraded by noise) has such small coefficients in the H subspace decomposition that even if we do not take additional steps to optimize the approximation, the error incurred in simply discarding them is still smaller than that incurred when using the D_4 transform. In other words, a reasonable signal is very likely to be closer to the G subspace associated with the QS than to the G subspace associated with the D_4 transform. This is supported by the numerical evidence given in the last section.

We give below the method of orthogonal compensation to enhance the approximation when discarding some of the components of $\mathbf{y}_2$. Let $\mathbf{d}$ be the vector to be discarded. It is likely to be a partial sum of the expression in the second pair of parentheses in (5.4). We decompose $\mathbf{d}$ into a linear combination of the vectors $\overline{\mathbf{g}}_i$ and an error vector that is orthogonal to G.

$$\mathbf{d} = (a_1\overline{\mathbf{g}}_1 + a_2\overline{\mathbf{g}}_2 + \cdots) + \mathbf{e}. \tag{5.5}$$

After determining a_i, they are added to the corresponding y_{1i}, so that the actual part that is discarded is $\mathbf{e}$, which is orthogonal to G. To this end, we take inner products of $\mathbf{d}$ with each of $\overline{\mathbf{g}}_i$. One can easily verify that a_i is the solution to the tridiagonal system of linear equations

$$\begin{pmatrix} 26 & 6 & & & & \\ 6 & 20 & 6 & & & \\ & 6 & 20 & 6 & & \\ & & & \ddots & & \\ & & & & 6 & 26 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} < \mathbf{d}, \overline{\mathbf{g}}_1 > \\ < \mathbf{d}, \overline{\mathbf{g}}_2 > \\ < \mathbf{d}, \overline{\mathbf{g}}_3 > \\ \vdots \\ \vdots \end{pmatrix} \tag{5.6}$$

for signals of even length. For signals of odd length, the corresponding tridiagonal matrix is just a bit more complicated — the last two rows are now

$$\frac{1}{49}\begin{bmatrix} 294 & 906 & 256 \\ & 256 & 640 \end{bmatrix}. \tag{5.7}$$

Tridiagonal systems can be solved by standard numerical linear algebra method with no more than $O(n)$ computational complexity.

6 General W-Matrices

In this section we give some properties of general W-matrices. Detailed proofs are omitted. As mentioned in the preceding section, there is more than one way of constructing the beginning and trailing rows of a W-matrix and all valid construction gives rise to multiresolution analysis with discrepancies that affect only a few boundary values. We therefore consider two W-matrices equivalent if they differ only in some boundary rows, in other words, if they are generated by the same g- and h-vectors.

Theorem 1 *If W_1 and W_2 are W-matrices of equal size, then their product W_1W_2 is again a W-matrix.*

One can use this theorem to generate new W-matrices from known ones, especially ones of higher order from those of lower order.

Theorem 2 *Let W be generated by a pair of vectors $(\mathbf{g}, \mathbf{h})$, and let a, b be two nonzero real numbers. Then the W-matrix generated by $(a\mathbf{g}, b\mathbf{h})$ is functionally the same as W in the sense that they correspond to the same multiresolution analysis. In a similar vein, the W-matrix generated by $(\mathbf{h}, \mathbf{g})$ differs from the first one only by a permutation, and we can regard them as being functionally equivalent.*

W-matrices of order 4 can be completely characterized by the following theorem. The inverse of the general W-matrix of order 4 have been computed explicitly with the help of Maple. We omit the formula.

Theorem 3 *Let $A, B, C, D, a,$ and b be any six given real numbers such that*

$$AD - BC \neq 0 \quad \text{and} \quad a \neq b. \tag{6.8}$$

Then the vectors

$$\left[\begin{array}{cccc} aA, & aB, & bC, & bD \end{array}\right] \tag{6.9}$$

and

$$\left[\begin{array}{cccc} A, & B, & C, & D \end{array}\right] \tag{6.10}$$

generate a W-matrix. Conversely, any W-matrix of order 4 can be generated this way.

Theorem 4 *In general, given any vector $\mathbf{h} = [h_1, h_2, \cdots, h_{2n}]$ of even length, one can supplement it with a vector $\mathbf{g}$ to form a W-matrix if the $(2n-1) \times 2n$ matrix*

$$\mathbf{A} = \begin{pmatrix} h_2 & -h_1 & & & & & \\ h_4 & -h_3 & h_2 & -h_1 & & & \\ & & & & \ddots & & \\ & & & & & -h_{2n-1} & h_{2n} \end{pmatrix} \tag{6.11}$$

has full rank. More precisely, let $\mathbf{B}$ be the matrix obtained from $\mathbf{A}$ by deleting the middle row of $\mathbf{A}$. The solution space of the matrix equation

$$\mathbf{B}\,[z_1, z_2, \cdots, z_{2n}]' = 0 \tag{6.12}$$

is a two-dimensional linear space that contains $\mathbf{h}$*. Any nonzero vector in the solution space other than a multiple of* $\mathbf{h}$ *can be used as* $\mathbf{g}$*.*

If the vector $\mathbf{h}$ *is of odd length, it can be considered to be a vector of even length with the last component being zero, and the above result can be applied.*

A simple corollary of Theorem 4 is the existence of W-matrices generated by symmetric basic vectors.

Theorem 5 *For any given* n *numbers* $h_1, h_2, \cdots, h_n$*, one forms the vector* $\mathbf{h} = [h_1, h_2, \cdots, h_n, -h_n, \cdots, -h_2, -h_1]$*. If* $\mathbf{h}$ *satisfies the conditions in Theorem 4, then there exists a symmetric* $\mathbf{g}$ *such that* $(h, \mathbf{g})$ *and* $\mathbf{g}$ *generates a* W*-matrix.*

Theorem 6 *All orthogonal* W*-matrices of order 4 are generated by the pair of basic vectors (after being normalized to be of unit length)*

$$\mathbf{g} = [1, \alpha, \alpha\beta, -\beta] \tag{6.13}$$

$$\mathbf{h} = [1, \alpha, -\alpha/\beta, 1/\beta] \tag{6.14}$$

for arbitrary real numbers α *and* β*. If, in addition, we require that* $\mathbf{h}$ *has vanishing zero-th moment, then*

$$\beta = \frac{\alpha - 1}{\alpha + 1}. \tag{6.15}$$

The particular choice of $\alpha = \sqrt{3}$ *leads to the* D_4 *matrix.*

7 Connection to Wavelet Theory

In this section we discuss how the well-known concepts in wavelet theory are related to our W-matrix transforms. Suppose that J levels of multiresolution analysis have been applied to a signal $\mathbf{x}$ of sufficiently long length N. For compression purposes, the ideal situation is such that we can throw away most of the components of each $\mathbf{y}_2$ at each level. The compressed data of the signal thus consists mainly of the last-level $\mathbf{y}_1$, supplemented by a few components from the $\mathbf{y}_2$ of earlier levels. The approximate signal restored from the compressed data is then the J times inverse transform of the last-level $\mathbf{y}_1$, plus some detail adjustments using the additional $\mathbf{y}_2$ data.

By linearity, the J-times-inverse transform of $\mathbf{y}_1$ is the sum of the J times inverse transform of each of the components of $\mathbf{y}_1$. By the shift invariance character of the W-matrix transform, the inverse transform of each component is simply a multiple of the translated inverse transform of another component (except for the boundary components). The signal restored from $\mathbf{y}_1$ is thus a linear combination of translates of some basic signal, which is the J times inverse transform of the ($\mathbf{y}_1$) vector $[0, 1, 0]'$— by this, we mean that we take the J-th level $\mathbf{y}_1$ vector to be $[0, 1, 0]'$, and $\mathbf{y}_2$ at all levels to be zero, and we compute the original signal that gives this decomposition. We normalize this signal by multiplying it with a constant so that the maximum of the signal is 1. This basic signal is called the J-th scaling signal.

What happens when we let $J \to \infty$? The length of the scaling signal increases as J increases, but we can consider it as the sampling of a continuous signal defined on a fixed interval, say $[0, a]$. If there exists a continuous signal on $[0, a]$ such that the J-th scaling signals converge to as $J \to \infty$, this continuous signal is defined to be the scaling function of the multiresolution analysis.

In a similar way, the J-times-inverse transform of the ($\mathbf{y}_2$) vector $[0, 1, 0]'$, after normalizing to have maximum 1, is the J-th wavelet signal. As $J \to \infty$, the J-th wavelet signal may converge to a continuous wavelet.

A signal restored from a J-level multiresolution analysis is then a linear combination of translates of the J-th scaling signal and translates of wavelet signals of various levels (lower than or equal to J). In practice, the continuous scaling function and continuous wavelet will never be used, but the discrete scaling signals and wavelet signals, for J large, closely approximate their continuous relatives. For instance, the D_4 scaling functions and wavelets are not smooth. Therefore, the high-level D_4 scaling and wavelet signals are also not smooth. This phenomenon explains why, when a signal is compressed using several levels of the D_4 transform and a majority of the $\mathbf{y}_2$ components are discarded, the restored signal has numerous sharp cusps. On the other hand, the scaling function of the QS transform is C^1 smooth, so that the corresponding discrete scaling signals are smooth and the restored signal will look smooth (except where adjustments are made with the $\mathbf{y}_2$ components retained in the compressed data).

We can see that the scaling function we define here coincides with the classical scaling function defined as in [3] by the following theorem.

Theorem 7 *Let ϕ be the scaling function, if it exists, corresponding to a W-matrix. Then ϕ satisfies the dilation equation*

$$\phi(x) = \frac{2}{\sum \overline{g}_i} \sum \overline{g}_i \phi(2x - i + 1), \tag{7.1}$$

where $\overline{\mathbf{g}} = [\overline{g}_1, \cdots, \overline{g}_n]$ is the first basic vector of the inverse W-matrix.

In particular, the QS scaling function satisfies

$$\phi(x) = \frac{1}{4}(\phi(2x) + 3\phi(2x-1) + 3\phi(2x-2) + \phi(2x-3)). \tag{7.2}$$

It is well known (see, for example, [1]) that ϕ is the classical quadratic spline.

The wavelet defined here coincides with the classical wavelet in the case where the multiresolution analysis is orthogonal. In the contrary case, we can consider our wavelet as a generalization of the classical wavelet. Our wavelet has the following property:

Theorem 8

$$\psi(x) = C \sum \overline{h}_i \phi(2x - i + 1), \tag{7.3}$$

where $\overline{\mathbf{h}} = [\overline{h}_1, \cdots, \overline{h}_n]$ is the second basic vector of of the inverse W-matrix and C is some scaling constant.

Figures 6 and 7 show the QS scaling function and wavelet, respectively.

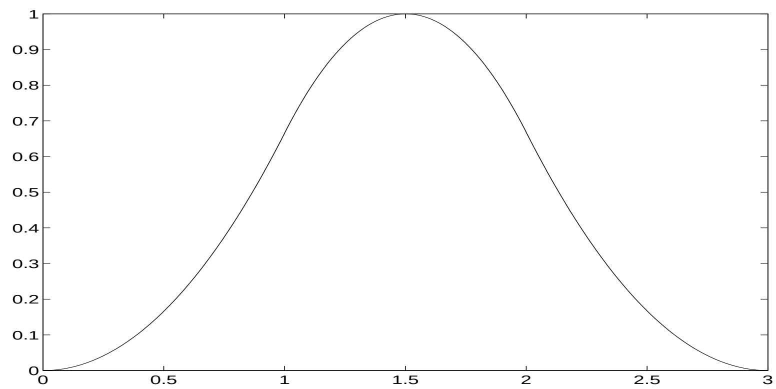

Figure 6. The QS scaling function

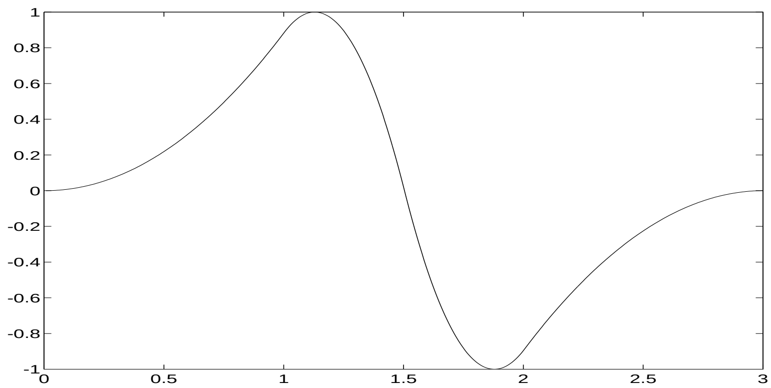

Figure 7. The QS wavelet

8 Conclusion

We have introduced the W-matrices, shown how they are constructed, and discussed how each W-matrix leads to a transform and a multiresolution analysis for signals of arbitrary length. We have demonstrated that the new QS (quadratic spline) transform is suitable for compression purposes and that, for reasonable signals and images, it performs better than the D_4 transform.

The generality of our W-matrices opens up some possibilities that are worth further investigation. By varying the parameters (the h- and g-vectors that define the W-matrix), one can choose an optimal transform according to the input signal. Using generalized W-matrices (that are generated by more than one set of basic vectors), one can transform different parts of the signal differently to enhance the compression performance. The optimal W-matrix for a subregion of an image may also be used to characterize the texture. One can also employ different transforms at different levels of the multiresolution analysis.

That the QS transform does not work as well as the D_4 transform in compressing signals that are wildly oscillating or that are degraded by noise really means that the QS transform does not retain much of the high-frequency part of the signal in the $\mathbf{y}_1$ vector. This fact can be turned into an advantage for the QS transform if it is used for denoising purposes. Further study is planned for this topic.

References

[1] Chui, C. K., *An Introduction to Wavelets*, Academic Press, 1992.

[2] Daubechies, I, *Ten Lectures on Wavelets*, CBMS-NSF Series Appl. Math., SIAM, 1991.

[3] Mallat, S. G., *A theory for multiresolution signal decomposition: The wavelet representation,* IEEE Trans. on Pattern Analysis and Machine Intelligence 11 (1989), 674–693.

[4] Taswell, C. and McGill, K. C., *Wavelet transform algorithms for finite-duration discrete-time signals,* Numerical Analysis Project Manuscript NA-91-07, Department of Computer Science, Stanford University, 1991.

[5] Kwong, Man Kam, *MATLAB Implementation of W-Matrix Multiresolution Analyses*, Argonne National Laboratory Preprint Series MCS-P462-0894, 1994.