A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000

Bryan E. Usevitch

The JPEG committee has recently released its new image coding standard, JPEG 2000, which will serve as a supplement for the original JPEG standard introduced in 1992. Rather than incrementally improving on the original standard, JPEG 2000 implements an entirely new way of compressing images based on the wavelet transform, in contrast to the discrete cosine transform (DCT) used in the original JPEG standard. The significant change in coding methods between the two standards leads one to ask: What prompted the JPEG committee to adopt such a dramatic change? The answer to this question comes from considering the state of image coding at the time the original JPEG standard was being formed. At that time wavelet analysis and wavelet coding were still very new technologies, whereas DCT-based transform techniques were well established. Early wavelet coders had performance that was at best comparable to transform coding using the DCT. The comparable performance between the two methods, coupled with the considerable momentum already behind DCT-based transform coding, led the JPEG committee to adopt DCT-based transform coding as the foundation of the lossy JPEG standard.

The state of wavelet-based coding has improved significantly since the introduction of the original JPEG standard. A notable breakthrough was the introduction of embedded zero-tree wavelet (EZW) coding by Shapiro [1]. The EZW algorithm was able to exploit the multiresolutional properties of the wavelet transform to give a computationally simple algorithm with outstanding performance. Improvements and enhancements to the EZW algorithm have resulted in modern wavelet coders which have improved performance relative to block transform coders. As a result, wavelet-based coding has been adopted as the underlying method to implement the JPEG 2000 standard.

Prior to JPEG 2000, wavelet-based coding was mainly of interest to a limited number of compression researchers. Since the new JPEG standard is wavelet based, a much larger audience including hardware designers, software programmers, and systems designers will be interested in wavelet-based coding. One of the purposes of this article is to give a general audience sufficient background into the details and techniques of wavelet coding to better understand the JPEG 2000 standard. The focus of this discussion is on the fundamental principles of wavelet coding and not the actual standard itself (more details on the standard can be found in [2]). Part of this discussion will try to explain some of the confusing design choices made in wavelet coders. For example, those familiar with wavelet analysis know that there are two types of filter choices: orthogonal and biorthogonal [3]-[5]. Orthogonal filters have the nice property that they are energy or norm preserving and in this aspect are similar to the DCT transform. Nevertheless, modern wavelet coders use biorthogonal filters which do not preserve energy. Another peculiarity of wavelet coders is that the wavelet transform can use essentially an infinite number of possible biorthogonal (or orthogonal) filters. Nevertheless, only a very small number of filter sets, often one or two, are used in practice. Reasons for these specific design choices will be explained.

1053-5888/01/$10.00©2001IEEE

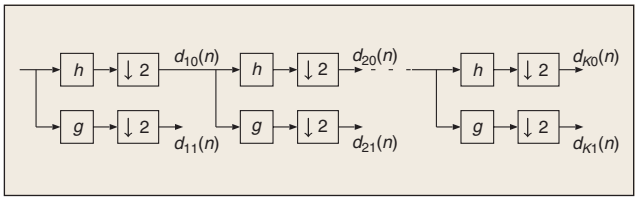
Another purpose of this article is to compare and contrast "early" wavelet coding with "modern" wavelet coding. Image coding was one of the first applications of the newly discovered wavelet theory. The reason for this was that wavelet analysis was very similar to the well-established subband analysis, which meant that the techniques of subband coding could be directly applied to wavelet coding. Modern wavelet coders use techniques which are significantly different from the techniques of subband coding and are based on ideas originating with EZW. This article will compare the techniques of the modern wavelet coders to the subband coding techniques so that the reader can appreciate how different modern wavelet coding is from early wavelet coding.

The remainder of the article proceeds as follows. The following section discusses basic properties of the wavelet transform which are pertinent to image compression. The material in this section builds on the background material in generic transform coding given in [6] (further background in data compression can be found in [7]-[10]). This section shows that boundary effects motivate the use of biorthogonal wavelets, and introduces the symmetric wavelet transform. The next section discusses the subband coding or "early" wavelet coding method followed by an explanation of the EZW coding algorithm. The last section describes other modern wavelet coders that extend the ideas found in the EZW algorithm and summarizes the article.

Wavelet Background

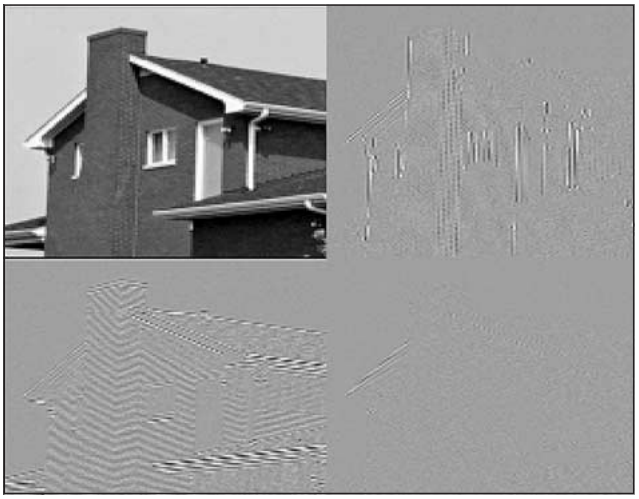
This section describes some of the properties of the discrete wavelet transform that are pertinent to image compression. The discussion here shows why current compression systems use biorthogonal instead of orthogonal wavelets and shows why some biorthogonal wavelets are better choices than others. In addition, this section discusses a particular form of the discrete wavelet transform, the symmetric wavelet transform, which has been specifically designed to handle boundary effects. Only those aspects of wavelet analysis which are important for understanding lossy image compression are described here. A full introduction to wavelets is beyond the scope here but can be found elsewhere [11]-[18].

The generic form for a one-dimensional (1-D) wavelet transform is shown in Fig. 1. Here a signal is passed through a lowpass and highpass filter, h and g, respectively, then down sampled by a factor of two, constituting



▲ *1. A K-level, 1-D wavelet decomposition. The coefficient notation $d_{ij}(n)$ refers to the jth frequency band (0 for low and 1 for high) of the ith level of the decomposition.*

▲ *2. Original image used for demonstrating the 2-D wavelet transform.*

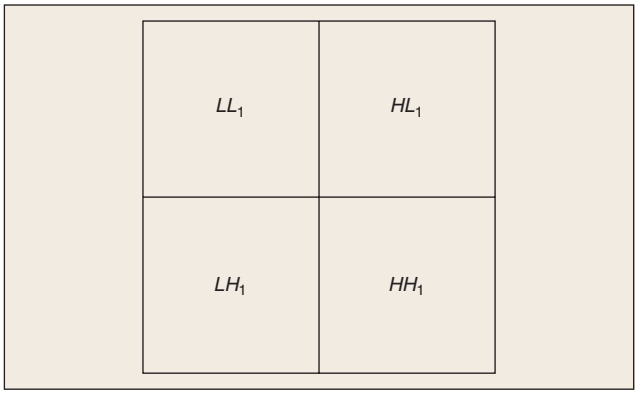


▲ *3. A one-level ($K = 1$), 2-D wavelet transform using the symmetric wavelet transform with the 9/7 Daubechies coefficients (the high-frequency bands have been enhanced to show detail).*

one level of transform. Multiple levels or "scales" of the wavelet transform are made by repeating the filtering and decimation process on the lowpass branch outputs only. The process is typically carried out for a finite number of levels K, and the resulting coefficients, $d_{i1}(n), i \in \{1,\ldots,K\}$ and $d_{K0}(n)$, and are called wavelet coefficients. When it is not necessary to know scale or frequency information, the entire set of wavelet coefficients is referred to as $\{w(n)\}$. This article uses only the maximally decimated form of the wavelet transform, where the downsampling factor in the decomposition and upsampling factor in the reconstruction equals the number of filters at each level (namely two).

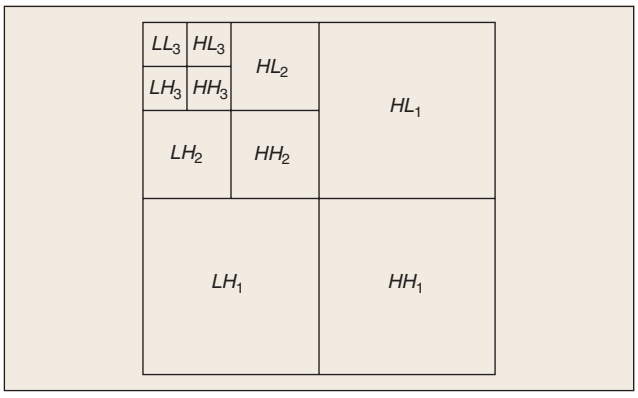
The 1-D wavelet transform can be extended to a two-dimensional (2-D) wavelet transform using separable wavelet filters [7], [19]. With separable filters the 2-D transform can be computed by applying a 1-D transform to all the rows of the input, and then repeating on all of the columns. Using the original image in Fig. 2, Fig. 3

shows an example of a one-level ($K=1$), 2-D wavelet transform, with corresponding notation given in Fig. 4. The example is repeated for a three-level ($K=3$) wavelet expansion in Figs. 5 and 6. In all of the discussion K represents the highest level of the decomposition of the wavelet transform. The focus of this article is on 2-D transforms. However, since the 2-D transform is readily obtained from separable extension of the 1-D transform and to simplify discussion, the remainder of this section illustrates concepts using only the 1-D transform.



▲ *4. The subband labeling scheme for a one-level, 2-D wavelet transform.*

▲ *5. A three-level (K = 3), 2-D wavelet transform using the symmetric wavelet transform with the 9/7 Daubechies coefficients (the high-frequency bands have been enhanced to show detail).*



▲ *6. The subband labeling scheme for a three-level, 2-D wavelet transform.*

The original wavelet transforms was implemented using orthogonal wavelets, which are wavelet filters that satisfy orthogonality constraints

$$\begin{aligned}\sum_n h(n-2i)h(n-2j) &= \delta(i-j)\\ \sum_n g(n-2i)g(n-2j) &= \delta(i-j)\\ \sum_n h(n-2i)g(n-2j) &= 0.\end{aligned} \tag{1}$$

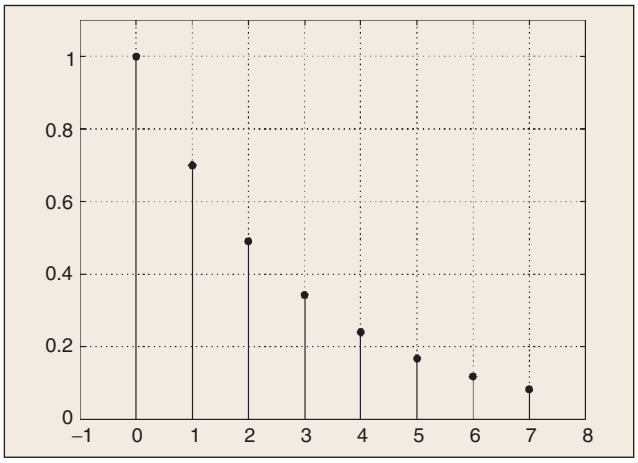
With orthogonal filters, the wavelet transform can be viewed as projecting the input signal onto a set of orthogonal basis functions. If the filters are also normalized, as they are in (1), the resulting wavelet transform is energy preserving. For an input signal $x(n)$ of length N, this energy preserving property, which is analogous to the Parseval property in Fourier analysis, can be written

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

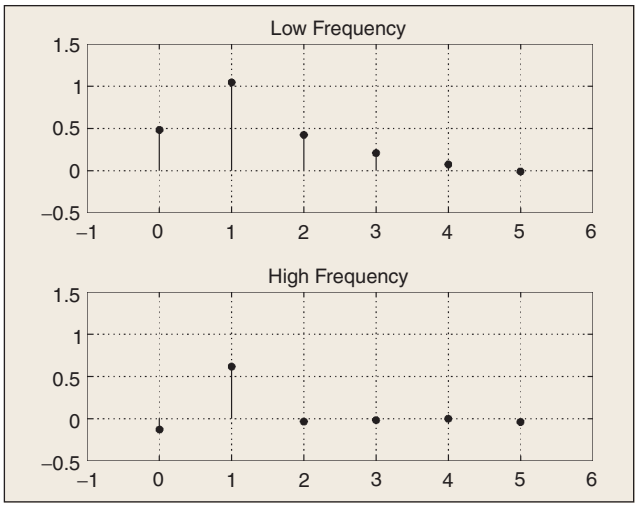
The energy conservation property is convenient for coding system design since the mean squared distortion introduced by quantizing the transformed coefficients equals the mean squared distortion in the reconstructed signal. Thus the energy conservation property simplifies designing the coder since the quantizer design can be carried out completely in the transform domain.

The standard orthogonal wavelet transform has some shortcomings that make it less than ideal for use in a coding system. One shortcoming is highlighted in (2), where it is shown that the total number of input coefficients, N, does not equal the total number of wavelet coefficients, L, using the maximally decimated wavelet transform. In general L is greater than N and the wavelet transform results in "coefficient expansion." This expansion is illustrated with a simple example. Consider a length N (even) input and length M (even) wavelet filters. The outputs of the filters h and g will be length $N+M-1$, and the outputs of the decimators will be length $(N+M)/2$. Thus, the N original input samples result in a total of $N+M$ wavelet coefficients after one level of transform. More levels of wavelet analysis only makes the problem worse, since more levels result in more than $N+M$ samples. Figs. 7 and 8 illustrate the coefficient expansion problem for a length 8 input and length 4 wavelet filters, and show that the wavelet transform outputs $8+4=12$ coefficients after one level of transform.

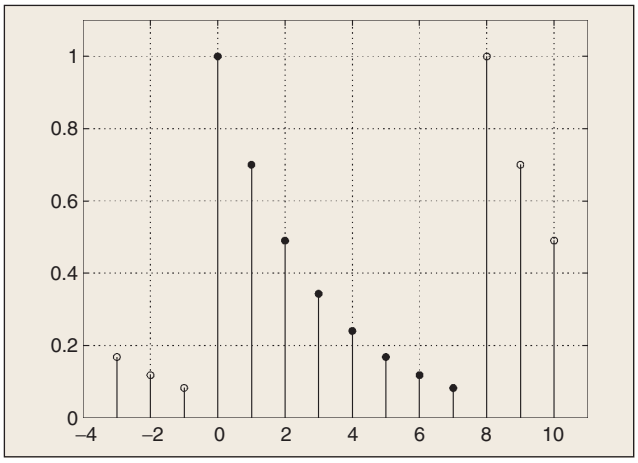
Coefficient expansion is a problem for coding systems where the aim is to reduce, not increase, the amount of information to be coded. One simple way to eliminate coefficient expansion is to use circular convolution, rather than linear convolution, on the finite length input $x(n)$. While solving the coefficient expansion problem, circular convolution leads to the introduction of artifacts. Con-



▲ *7. Length 8 example input sequence.*



▲ *8. The $d_{10}(n)$ and $d_{11}(n)$ outputs of the wavelet transform for the input given in Fig. 7, using the length 4 Daubechies orthogonal filters. Note that the number of output coefficients after one level of analysis is 12, illustrating the coefficient expansion problem.*



▲ *9. The periodic extension of the input in Fig. 7.*

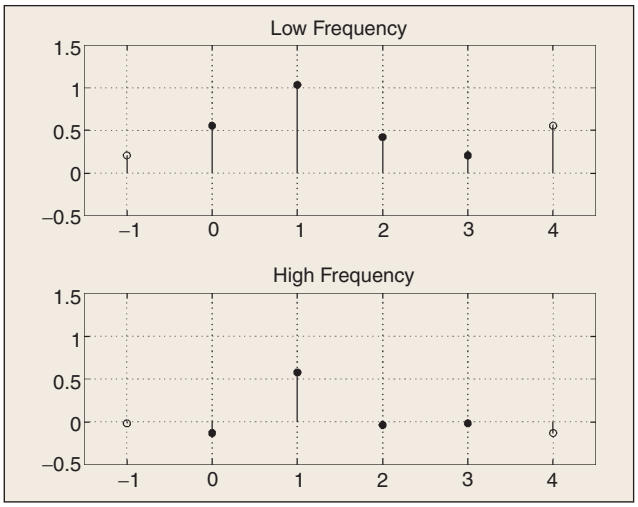
sider the example input signal of Fig. 7, which when periodically extended as shown in Fig. 9 has a large discontinuity ($|x(N-1)| << |x(0)|$). Circular convolution of the input results in large wavelet coefficients in the highpass band at the location of the discontinuity as shown in Fig. 10. These large wavelet coefficients due to border discontinuities are undesirable because:

▲ They require more bits to code so that the reconstructed signal accurately represents the input;

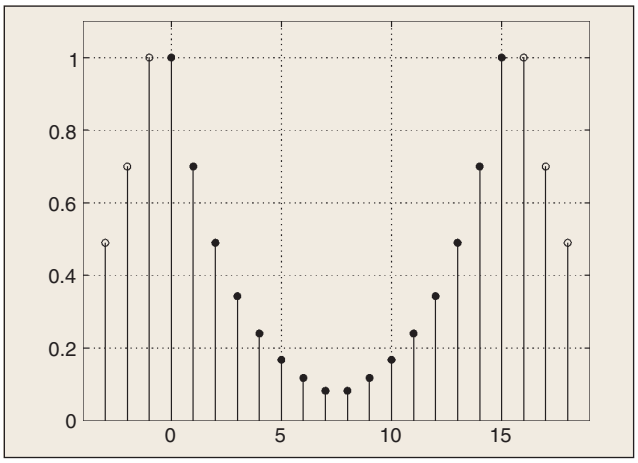
▲ They do not represent any information present in the original signal, but rather are an artifact of the method used to perform the transform.

Bits used to code these artificially introduced artifacts could be better used to code the original data.

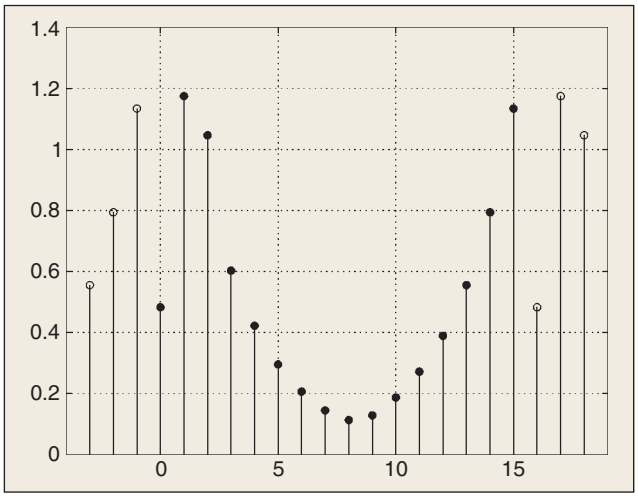
The border or edge artifacts can be eliminated by performing a symmetric periodic extension of the input in place of a periodic extension (see Fig. 11). This symmet-



▲ *10. The $d_{10}(n)$ and $d_{11}(n)$ outputs of the wavelet transform of the periodically extended input of Fig. 9. Note that the large discontinuity at the boundaries of the input result in large high-frequency coefficients in the wavelet output.*



▲ *11. Symmetric periodic extension of the original input shown in Fig. 7.*



▲ *12. The output resulting from filtering the symmetric periodic signal of Fig. 11 with the length 4 Daubechies lowpass filter. Since the filter is not symmetric, the filtered output no longer has symmetry.*

ric extension guarantees continuity across replicas of the input and eliminates the large wavelet coefficients caused by border discontinuities. Note that symmetric extension doubles the number of input samples. This is not a problem initially since half the samples are redundant because of symmetry. However, when the input is filtered and decimated, it results in outputs that are not necessarily symmetric periodic (see Fig. 12). As a result, half the coefficients cannot be eliminated by symmetry, and there is a doubling of the number of coefficients required to represent the input. Thus this case is even worse than the linear convolution case where the number of coefficients only increased by the filter length M. Fortunately, symmetry can be preserved across scales of the wavelet transform by imposing an additional constraint on the wavelet filters h and g: they must be either symmetric or antisymmetric (also known as linear phase in signal processing terminology). For this special case, periodic symmetric inputs give periodic symmetric outputs and the result is no coefficient expansion [20], [15].

The use of symmetric extensions and linear phase wavelet filters would seem to solve the problem of border effects in the wavelet transform. However, there is still one technical difficulty to overcome, which is illustrated by the following:

Fact: For real valued, compactly supported orthogonal wavelets, there is only one set of linear phase filters, and that set is the trivial Haar filters, $h = (1,1)$, $g = (1,-1)$ [3].

The lack of linear phase filters in orthogonal wavelets led to research in extending wavelet analysis to more general forms, which would allow for linear phase filters. The research resulted in a more general form of wavelets known as "biorthogonal wavelets" [4], [5]. As the name implies, biorthogonal wavelets have some orthogonality relationships between their filters. But biorthogonal wavelets differ from orthogonal in that the forward wavelet transform is equivalent to projecting the input signal on to nonorthogonal basis functions. The orthogonal and biorthogonal wavelets transforms are analogous to orthogonal and nonsingular matrix transforms, respectively. Both the orthogonal and nonsingular matrix transforms are invertible, but only the orthogonal matrix transform is energy preserving. The main advantage in using the biorthogonal wavelet transform is that it permits the use of a much broader class of filters, and this class includes includes symmetric filters.

When the wavelet transform uses linear phase filters, it gives symmetric outputs when presented with symmetric inputs. This particular form of the wavelet transform (linear filters with symmetric inputs and outputs) is called the symmetric wavelet transform (SWT). The SWT solves the problems of coefficient expansion and border discontinuities and its use has been shown to improve the performance of image coding applications [21], [20]. Efficient practical implementation of the SWT involves many tricky details due to the lengths of the input and filters (even or odd) and decimation of the symmetric extensions. These details are fully explained in [15] and [22]. Those desiring to implement the SWT will also find the paper by Vetterli and Herley [5] helpful since it characterizes the complete class of linear phase filters which can be obtained under the biorthogonal filter constraints.

In summary, the biorthogonal wavelet transform has the advantage that it can use linear phase filters, but the disadvantage that it is not energy preserving. The fact that biorthogonal wavelets are not energy preserving does not turn out to be a big problem, since there are linear phase biorthogonal filter coefficients which are "close" to being orthogonal. One example of such a wavelet filter set is the 9/7 filter given in Table 1. This filter set can be plugged into the orthogonality constraints of (1) to show that they are nearly orthogonal. Another way of showing the approximate orthogonality of these filters is to consider the weighting introduced by nonorthogonality [23],

Table 1. Two Sets of Linear Phase, Biorthogonal Wavelet Filter Coefficients.

9/7 Filter Coefficients		5/3 Filter Coefficients		Filter Index
h_0	g_0	h_0	g_0	
0.852699	0.788486	1.060660	0.707107	0
0.377402	0.418092	0.353553	0.353553	–1, 1
–0.110624	–0.040689	–0.176777		–2, 2
–0.023849	–0.064539			–3, 3
0.037828				–4, 4

The 9/7 coefficients have the nice property that, although they are biorthogonal, they are very close to being orthogonal as shown in Table 2.

[24]. Table 2 shows the weighting from subband to reconstructed output caused by the reconstruction wavelet filters for some example filter sets (the relation between subband variances σ_{ij}^2, weights w_{ij}, and output variance σ^2 is given by

$$\sigma^2 = \frac{1}{2^K}\sigma_{L0}^2 + \sum_{i=1}^{K}\frac{1}{2^i}w_{ij}\sigma_{i1}^2,$$

where the $1/2^i$ factors arise because of subband size). These weights can be computed by taking the l_2 norm of the single equivalent filter which takes the subband coefficients directly to the reconstructed output. Note that for orthogonal filters the weights are all one (indicating that energy is preserved between transform coefficients and reconstructed output). Also note that the 9/7 filter set has weights that deviate by only a few percent from one, showing that the filter set is reasonably close to being orthogonal. Besides being nearly orthogonal, or perhaps because of being nearly orthogonal, the 9/7 set has been shown experimentally to give very good compression performance and has been used extensively in image compression applications [25], [26].

Using biorthogonal wavelets in conjunction with symmetric extensions is not the only solution to the border effects problem. Orthogonal wavelets which employ boundary filters [15] are another solution to this problem. So why not just use orthogonal wavelets with an alternate method? It turns out that given all things equal, such as orthogonal and biorthogonal wavelets using circular convolution, biorthogonal wavelets still give better performance than orthogonal wavelets [21].

Table 2. The Implicit Weighting Introduced on the Wavelet Coefficient Energy as It Is Transformed to the Reconstructed Output.

Weight	Orthogonal	9/7	5/3
w_{10}	1.00000	0.98295	0.75000
w_{11}	1.00000	1.04043	1.43750
w_{20}	1.00000	1.03060	0.68750
w_{21}	1.00000	0.96721	0.92187
w_{30}	1.00000	1.05209	0.67187
w_{31}	1.00000	1.03963	0.79297
w_{40}	1.00000	1.03963	0.66797
w_{41}	1.00000	1.07512	0.76074

Note that the 9/7 biorthogonal filter set deviates by only a few percent from the orthogonal filter weighting.

Wavelet coding techniques provide a very strong basis for the new JPEG 2000 coding standard.

Subband Coding

This section gives a brief overview of the subband coding method. Subband coding is a good example of a coding method which follows the generic transform coding model discussed in [6]. Subband coding is also used here to illustrate "early" wavelet coding, since early wavelet coders and subband coders use identical coding techniques, with the only possible distinction between the two being the choice of filters. As a result, the terms "early wavelet coding" and "subband coding" are used interchangeably here. Because early wavelet coding and subband coding are essentially identical, the coders discussed in this section will serve as a point of reference when discussing modern wavelet coders. A main goal of this section is to point out some of the weaknesses of subband coding that are later addressed by modern wavelet coders.

In subband coding, the transform block is implemented through filtering and decimating analogous to the wavelet transform (see Fig. 1). For image data, the filtering and decimation is applied recursively on the LL_i band to give an octave subband decomposition equivalent to those shown in Figs. 4 and 6. The main difference between subband and wavelet coding is the choice of filters to be used in the transform. The filters used in wavelet coding systems were typically designed to satisfy certain smoothness constraints [3]. In contrast, subband filters were designed to approximately satisfy the criteria of nonoverlapping frequency responses. To explain this nonoverlapping criteria, remember from [6] that the goal of the transform section of a coding system is to decorrelate coefficients. A well-known theorem states that random processes which have nonoverlapping frequency bands are uncorrelated [27]. It is this property that the subband filtering uses to try to achieve decorrelation. Adding the additional constraint that the transform be (nearly) lossless, subband filters are designed to be approximations to ideal frequency selective filters [28], where the combined response from all the filters covers the entire spectral band. Total decorrelation is not achieved since filters only approximate ideal filters.

The output of the transform stage for a K level octave decomposition on image data is $3K+1$ separate subbands of coefficients. By the design of the subband filters, the coefficients in each subband are (approximately) uncorrelated from coefficients in other subbands. As a result, the coefficients in each subband can be quantized independently of coefficients in other subbands with no

The biorthogonal wavelet transform has the advantage that it can use linear phase filters, but the disadvantage that it is not energy preserving.

significant loss in performance. The variance of the coefficients in each of the subbands is typically different, similar to the different variances of the DCT coefficients in the original JPEG standard, and thus each subband requires a different amount of bit resources to obtain best coding performance. The result is that each subband will have a different quantizer, with each quantizer having its own separate rate (bits/sample). The only issue to be resolved is that of bit allocation, or the number of bits to be assigned to each individual subband to give best performance.

To illustrate the process, we present an example solution to the bit allocation problem for the case of uniform scalar quantization in each of the subbands. To keep the discussion simple, assume that the subband decomposition is only one level, resulting in the four subbands LL_1, HL_1, LH_1 and HH_1, which are indexed as subbands 1 through 4 respectively. The goal is to assign each subband a bit rate, denoted as R_k bits/coefficient, such that

▲ 1) An overall bit rate

$$R = \frac{1}{4}\sum_{k=1}^{4} R_k \tag{3}$$

is satisfied and

▲ 2) The reconstruction distortion is minimized.

Since uniform scalar quantization is used, the distortion or error energy introduced by the quantizer in each subband can be modeled by [8], [9]

$$\sigma_{r_k}^2 = \alpha_k 2^{-2R_k}\sigma_{y_k}^2, \tag{4}$$

where $\sigma_{y_k}^2$ is the variance of coefficients in each subband, R_k is the subband bit rate, and α_k is a parameter which depends on the probability distribution in the subbands (Gaussian or Laplacian or uniform, etc.). Equation (4) makes intuitive sense since the more bits/sample allocated to the subband (R_k), the lower the resulting distortion from that subband. Using (4), the total reconstruction error of the wavelet coefficients, assuming the same α_k in each subband, is

$$\sigma_r^2 = \alpha\sum_{k=1}^{4} 2^{-2R_k}\sigma_{y_k}^2. \tag{5}$$

Equations (3) and (5) can be combined to form a constrained minimization problem which can be solved using Lagrange multipliers [29], where the Lagrangian to be minimized is

$$J = \alpha\sum_{k=1}^{4} 2^{-2R_k}\sigma_{y_k}^2 - \lambda\left(R - \frac{1}{4}\sum_{k=1}^{4} R_k\right).$$

Minimization of this function results in the best bit allocation of [8], [9]

$$R_k = R + \frac{1}{2}\log_2 \frac{\sigma_{yk}^2}{\prod_{k=1}^{M}(\sigma_{yk}^2)^{1/M}}. \tag{6}$$

Equation (6) is not valid in all cases, for example when it result in R_ks that are negative, but methods have been derived to handle these technicalities [10]. Using the optimal bit rates, the coefficients in each of the subbands are individually quantized with their respective quantizers and then entropy coded to give the final coded representation.

Some of the shortcomings of subband coding can now be pointed out by considering the previous discussion. The first shortcoming is that the quantizer model of (4) used to derive the optimal bit allocation is only valid for "high bit rates" of approximately 1 bit/sample or more. The approximation worsens as the bit rates are reduced below 1 bit/sample and the optimal bit allocation of (6) is no longer valid. Consequently, the subband coding method is not able to determine optimal coding systems for low bit rate applications.

A second problem with subband coding involves the coding of an image at multiple target bit rates. As an example, consider the case of coding an image to give the resulting bit rates of 0.5 and 1.0 bits/sample. Rather than solving two coding problems, it would be nice to code the image at one bit rate, say 0.5 bit/sample, and then use this information to simplify the coding at the 1.0 bit/sample rate. The idea of reusing information to simplify the derivation of extended results is a common theme in modern signal processing, with recursive least squares and lattice filters being common examples. Unfortunately, for subband coding, the optimal bit allocation changes as the overall bit rate changes, which requires that the coding process be repeated entirely for each new target bit rate desired.

Finally, with subband coding it is difficult to code an input to give an exact target bit rate (or predefined output size). This is due to the entropy coding of the quantizer outputs and the approximation inherent in the quantizer model of (4). One solution to this problem, if the coded output is too large, is to truncate the coded data. Unfortunately, truncation removes entire subband coefficients, which can result in visual artifacts (you can see in the image where the data was truncated). If the coded output data is too small, more overall bits can be allocated and the subband coding procedure repeated. The disadvantage to this is that the coding procedure must be repeated,

again with no guarantee that the target output size will be achieved.

To finish this section some examples of the subband coding method are mentioned. This list includes early work in subband coding done by Vetterli [30] and Woods and O'Neil [31]. Example coders using wavelet analysis filters include those using scalar quantization [19], [25] and vector quantization [25], [32] of the subband coefficients. Another very good example of an early wavelet coder is the wavelet scalar quantization standard (WSQ), which has been adopted by the Federal Bureau of Investigation as the method for compressing their entire fingerprint database [33], [22], [34].

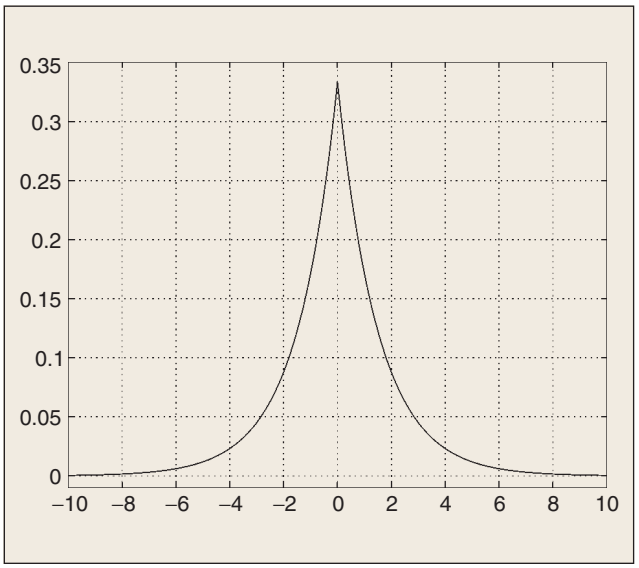
Embedded Zero-Tree Wavelet Coding

The original or "heritage" wavelet coders were based on the same basic ideas found in subband coding. The era of modern lossy wavelet coding began in 1993 when Jerry Shapiro introduced EZW coding [1]. EZW coding exploited the multiresolution nature of the wavelet decomposition to give a completely new way of doing image coding. The resulting algorithm had improved performance at low bit rates relative to the existing JPEG standard, as well as having other nice features such as a completely embedded bit representation. EZW marked the beginning of modern wavelet coding since improved wavelet coders proposed subsequent to EZW are based on fundamental concepts from EZW coding. This section fully describes the EZW algorithm by first discussing the statistical properties of wavelet coefficients resulting from image data. Next, the two key concepts of EZW, namely significance map coding using zero-trees and successive approximation quantization, are described. Finally, the EZW algorithm along with an illustrative example are given.

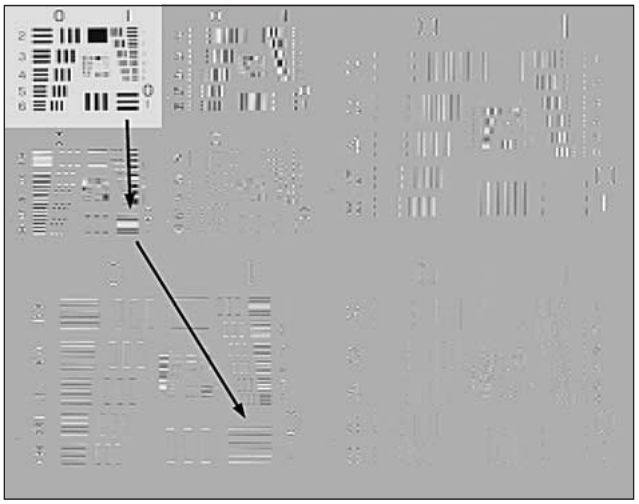
One of the beneficial properties of the wavelet transform, relative to data compression, is that it tends to compact the energy of the input into a relatively small number of wavelet coefficients (this property can be shown to be equivalent to reducing the correlation amongst wavelet coefficients) [35]. For example, in naturally occurring images, much of the energy in the wavelet transform is concentrated into the LL_K band. In addition, the energy in the high frequency bands (HL_i, LH_i, HH_i) is also concentrated into a relatively small number of coefficients. This energy compaction property can be observed in the probability distribution of wavelet coefficients in the high frequency subbands, which has been shown in previous studies to have a Laplacian-like density

$$f(x) = Ae^{(|x|/\sigma^2)^{\beta}}, \tag{7}$$

where σ^2 is the variance and β is the rolloff of the distribution [19], [31]. A sample density is shown in Fig. 13 which shows that the density is symmetric, peaked at zero, and has relatively long tails. The high peak around zero means that most coefficients in a frequency subband have small magnitudes and thus have small energy. Also, the long tails indicate that there are a few coefficients with large magnitudes, and it is these coefficients that have the highest concentration of energy in the subband. Previous designers of coding systems recognized that low bit rate, low mean squared error (MSE) coders can be achieved by coding only the relatively few high energy coefficients [36]. The only problem with this idea is that since only a select number of coefficients are now being coded, the encoder needs to send position information as well as magnitude information for each of the coefficients so that data can be decoded properly. Depending on the method used, the amount of resources required to code the position information can be a significant fraction of the total,



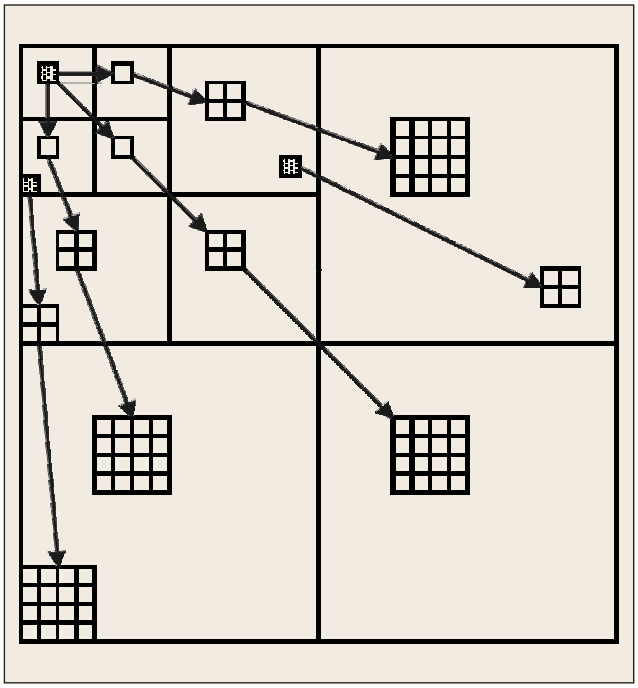
▲ *13. A plot of the probability density of (7) with $\sigma^2 = 1.5$, $\beta = 1$, and $A = 1/3$.*



▲ *14. A two-level, 2-D wavelet transform illustrating the propagation of significant coefficients across frequency bands. Note that the significant coefficients occur at the same relative spatial location in each of the subbands of the same frequency orientation.*

EZW has the desirable property, resulting from its successive approximation quantization, of generating an embedded code representation.

negating much of the benefit of the energy compaction. Various ways of lowering the cost of coding the position information associated with the significant coefficients have been proposed. Many methods are based on the idea of interband prediction (or prediction across frequency bands) which exploits the self-similar, hierarchal nature of the wavelet transform. Interband prediction can be explained by referring to the 2-D wavelet decompositions of Figs. 3, 5, and 14. Observation of these figures shows that the significant coefficients in the high frequency subband do not occur at random locations, but rather tend to cluster. Furthermore, these clusters tend to occur at the same relative spatial location in each of the high frequency subbands as illustrated in Fig. 14. Often these locations correspond to discontinuities or edges that occur in the original image [37]. The idea of interband prediction is to use the location of significant coefficients in one frequency band to predict the location and magnitude of significant coefficients in other frequency bands, thus reducing the cost of coding position information. As an example, the method proposed in [38] used the coarse approximation in LL_K to predict the location of significant coefficients in the higher frequency subbands, which resulted in improved overall coding efficiency.



▲ *15. Example trees that can be defined on the wavelet transform. The roots of the three trees, indicated by shading, originate in the LL_3, LH_3, and HL_2 subbands.*

Significance Map Coding Using Zero-Trees

The EZW algorithm recognized that a significant fraction of the total bits required to code an image were needed to code position information, or what the EZW algorithm called significance maps. A significance map was defined as an indication of whether a particular coefficient was zero or nonzero (i.e., significant) relative to a given quantization level (the reason for conditioning relative to a quantization level or threshold T will be explained in the next section on successive approximation quantization). The EZW algorithm determined a very efficient way to code significance maps not by coding the location of the significant coefficients, but rather by coding the location of the zeros. It was found experimentally that zeros could be predicted very accurately across different scales in the wavelet transform. Defining a wavelet coefficient as insignificant with respect to a threshold T if $|x|<T$, the EZW algorithm hypothesized that "if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then *all* wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T." Recognizing that coefficients of the same spatial location and frequency orientation in the wavelet decomposition can be compactly described using tree structures, the EZW called the set of insignificant coefficients, or coefficients that are quantized to zero using threshold T, zero-trees.

To make the discussion more precise, consider the tree structures on the wavelet transform shown in Fig. 15. In the wavelet decomposition, coefficients that are spatially related across scale (or frequency) can be compactly described using these tree structures. With the exception of the low resolution approximation (LL_K) and the highest frequency bands (HL_1, LH_1, and HH_1) each (parent) coefficient at level i of the decomposition spatially correlates to 4 (child) coefficients at level $i-1$ of the decomposition which are at the same frequency orientation. For the LL_K band, each parent coefficient spatially correlates with 3 child coefficients, one each in the HL_K, LH_K, and HH_K bands. The standard definitions of ancestors and descendants in the tree follow directly from these parent-child relationships. A coefficient is part of a zero-tree if it is zero and if all of its descendants are zero with respect to the threshold T. It is also a zero-tree root if is is not part of another zero-tree starting at a coarser scale. Zero-trees are very efficient for coding since by declaring only one coefficient a zero-tree root, a large number of descendant coefficients are automatically known to be zero. For example, a zero-tree root at level i in the wavelet decomposition determines the value of $(1/3)(4^i-1)$ total coefficients if the root is not in the LL_K band, and 4^i total coefficients if it is. The compact representation, coupled with the fact that zero-trees occur frequently, especially at

low bit rates, make zero-trees efficient for coding position information. Zero-trees are thus an elegant solution to both the interband prediction problem and the significant coefficient position coding problem.

Successive Approximation Quantization

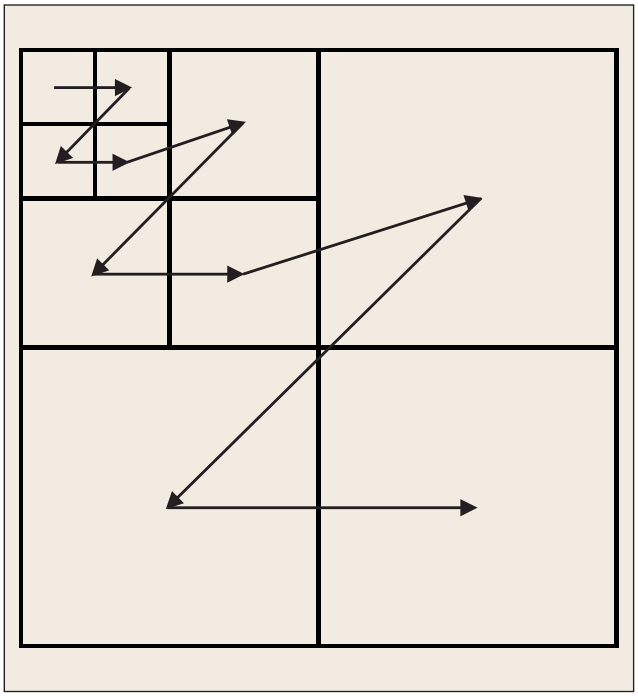
The next key to the EZW algorithm is the concept of successive approximation quantization. Successive approximation quantization serves two purposes in the EZW algorithm. First, it is used as a method to generate a large number of zero-trees, which is good since zero-trees are easily coded. Second, successive approximation quantization is used to sort the bit order of coded bits so that the most significant bits are sent first. An important end result of the most significant bits being sent first is that the coded bitstream is embedded. This means that bits needed to represent a higher fidelity image can be derived by simply adding extra refining bits to the lower fidelity image representation. Equivalently, a lower fidelity image can be derived by simply truncating the embedded bit stream, resulting in a lower overall bit rate.

To introduce the concept of successive approximation quantization, consider the case of an encoder wanting to transmit a coefficient, represented here by p, to a decoder. If the value of the coefficient p is represented in binary as 00100010_2, and if the the encoder can only send one bit, which bit of p should the encoder send? The answer from a MSE perspective is that the encoder should send the most significant bit first. To see this, consider the contribution that the reconstructed coefficient $\hat{p}$ would add to the overall reconstruction MSE. If the most significant bit were sent, the MSE error would increase by $(p - 00100000_2)^2 = (00000010_2)^2$. Alternatively, if the least significant bit were sent, the MSE would increase by the much larger amount of $(p - 00000010_2^2) = (00100000_2)^2$. The basic idea of successive approximation quantization is that the next bit sent will always be the one which reduces the resulting MSE by the maximum possible amount.

EZW implements successive approximation quantization through a multipass scanning of the wavelet coefficients using successively decreasing thresholds $T_0, T_1, T_2, \ldots$. The initial threshold is set to the value of $T_0 = 2^{\lfloor \log_2 x_{\max} \rfloor}$, where $x_{\max}$ is the largest wavelet coefficient. Each scan of wavelet coefficients is divided into two passes: dominant and subordinate. The dominant pass establishes a significance map of the coefficients relative to the current threshold T_i. Thus, coefficients which are significant on the first dominant pass are known to lie in the interval $[T_0, 2T_0)$, and can be represented with the reconstruction value of $(3T_0/2)$. The dominant pass essentially establishes the most significant bit of binary representation of the wavelet coefficient, with the binary weights being relative to the thresholds T_i.

After the first dominant pass the algorithm must track two classes of coefficients, where the classes depend on whether the coefficients were found to be significant on previous passes. EZW tracks these two classes by maintaining two lists: a dominant list consisting of the location of all coefficients not found to be significant in previous passes, and a subordinate list consisting of the location of all the coefficients found to be significant in previous passes. On the subordinate pass, or refinement pass, the successive approximation quantization determines the value of the next most significant bit of the binary representation of coefficients on the subordinate list. This is equivalent to finding the quantized value of these coefficients relative to the quantization step size $T_i/2$. For example, after the first dominant pass the coefficients in the subordinate list are known to lie in the interval $[T_0, 2T_0)$. The subordinate pass outputs 1 if a coefficient lies in the upper half of this interval $[(3/2)T_0, 2T_0)$, or zero if it is in the lower half $[T_0, (3/2)T_0)$.

For the second and subsequent scans the threshold T_i is decreased by powers of two, $T_i = T_{i-1}/2$, resulting in a binary representation of the coefficients. Dominant passes are only made on those coefficients on the dominant list (those coefficients not found significant on previous passes), The scanning of the dominant coefficients follows the subband ordering shown in Fig. 16, which guarantees that a coefficient is always scanned before any of its descendants, relative to the tree structure of Fig. 15. Those coefficients which can be deduced to be zero from a zero-tree root are not coded, which guarantees the maximum benefit from any zero-trees which may occur. Coefficients found to be significant during the dominant pass are moved to the subordinate list and are not coded in subsequent dominant passes. However, the locations of the coefficients on the original wavelet mapping are set



▲ *16. The scanning order for dominant passes of the EZW algorithm. The order guarantees that coefficients are scanned prior to their respective descendants, which increases the coding effectiveness of any zero-trees which may occur.*

53	–22	21	–9	–1	8	–7	6
14	–12	13	–11	–1	0	2	–3
15	–8	9	7	2	–3	1	–2
34	–2	–6	10	6	–4	4	–5
–6	5	–1	1	1	3	–1	5
6	1	3	0	–2	2	6	0
4	2	1	–4	–1	0	–1	4
0	–2	7	5	–3	2	–2	3

▲ *17. An example three-level wavelet decomposition used to demonstrate the EZW algorithm.*

*	–22	21	–9	–1	8	–7	6
14	–12	13	–11	–1	0	2	–3
15	–8	9	7	2	–3	1	–2
*	–2	–6	10	6	–4	4	–5
–6	5	–1	1	1	3	–1	5
6	1	3	0	–2	2	6	0
4	2	1	–4	–1	0	–1	4
0	–2	7	5	–3	2	–2	3

▲ *18. The example wavelet transform after the first dominant pass. The symbol * is used to represent symbols found to be significant on a previous pass.*

to zero to increase the likelihood that the ancestors of the coefficients will be coded as zero-tree roots on future dominant passes.

EZW Coding Algorithm

Having discussed zero-trees and successive approximation quantization, the EZW coding algorithm can now be summarized as follows.

▲ *1) Initialization:* Place all wavelet coefficients on the dominant list. Set the initial threshold to $T_0 = 2^{\lfloor \log_2 x_{\max} \rfloor}$.

▲ *2) Dominant Pass:* Scan the coefficients on the dominant list using the current threshold T_i and subband ordering shown in Fig. 16. Assign each coefficient one of four symbols:

- ▲ positive significant (*ps*)—meaning that the coefficient is significant relative to the current threshold T_i and positive,
- ▲ negative significant (*ns*)—meaning that the coefficient is significant relative to the current threshold T_i and negative,
- ▲ isolated zero (*iz*)—meaning the coefficient is insignificant relative to the threshold T_i and one or more of its descendants are significant,
- ▲ zero-tree root (*ztr*)—meaning the current coefficient and all of its descendants are insignificant relative to the current threshold T_i.

Any coefficient that is the descendant of a coefficient that has already been coded as a zero-tree root is not coded, since the decoder can deduce that it has a zero value. Coefficients found to be significant are moved to the subordinate list and their values in the original wavelet map are set to zero. The resulting symbol sequence is entropy coded.

▲ *3) Subordinate Pass:* Output a 1 or a 0 for all coefficients on the subordinate list depending on whether the coefficient is in the upper or lower half of the quantization interval.

▲ *4) Loop:* Reduce the current threshold by two, $T_i = T_i / 2$. Repeat the Steps 2) through 4) until the target fidelity or bit rate is achieved.

Table 3. Resulting Output of the First Dominant Pass ($T_0 = 32$).

Subband	Coefficient Value	Symbol	Reconstruction Value	Comment (See Text)
LL_3	53	*ps*	48	1)
HL_3	–22	*ztr*	0	2)
LH_3	14	*iz*	0	3)
HH_3	–12	*ztr*	0	
LH_2	15	*ztr*	0	
LH_2	–8	*ztr*	0	
LH_2	34	*ps*	48	
LH_2	–2	*ztr*	0	
LH_1	4	*iz*	0	
LH_1	2	*iz*	0	
LH_1	0	*iz*	0	
LH_1	–2	*iz*	0	

EZW Example

This section demonstrates the details of the EZW algorithm using a simple example. The coefficients to be coded are from a three-level wavelet transform of an 8×8 image and are shown in Fig. 17. For clarity, the entropy coding is not shown, which means the coder output will be a sequence of symbols for the dominant pass. The largest coefficient in the transform is 53 which results in an initial threshold of $T_0 = 32$. The results for the first dominant pass are shown in Table 3, with corresponding comments given below.

▲ 1) The coefficient has a magnitude greater than or equal to the threshold 32 and is positive. The resulting symbol is positive significant (*ps*), and the decoder knows that this symbol lies in the interval [32,64) and that its reconstruction value is 48.

▲ 2) This coefficient and all of its descendants (comprising all of subbands HL_2 and HL_1) are less than the threshold of 32, which causes this symbol to be coded as a zero-tree root (*ztr*). As a result, the remaining coefficients in subbands HL_2 and HL_1 are not coded in this dominant pass.

▲ 3) This coefficient is less than the threshold 32, but one of its descendants, coefficient 34 in subband LH_2, is significant relative to the threshold, preventing this symbol to be coded as a zero-tree root (*ztr*). As a result, this coefficient is coded as isolated zero (*iz*).

In the first subordinate pass the encoder sends a 0 or 1 to indicate if the significant coefficients are in the intervals [32,48) or [48,64) respectively. Thus the encoder outputs are 1 and 0 corresponding to the reconstruction values of (48+64)/2 = 56 and (32+48)/2 = 40. The results of the first subordinate pass are summarized in Table 4.

For the second dominant pass the coefficients 53 and 34 do not have to be coded again. The wavelet transform thus appears as in Fig. 18, where the * indicates a previously significant coefficient.

The threshold for the second dominant pass is $T_1 = 16$ and the results of this pass are summarized in Table 5, with corresponding comment given below.

▲ 4) Since the coefficient 34 of subband LH_2 was found to be significant on a previous pass, its value can be considered zero for purposes of computing zero-trees. As a result, the coefficient 14 becomes a zero-tree root on the second dominant pass.

The process continues alternating between dominant and subordinate passes until a desired fidelity or bit rate is achieved.

EZW Performance

Having presented the EZW algorithm, the main question to be asked is: How well does it perform? The answer is that the performance is quite good. When EZW was first introduced it gave compression performance as good or better than other algorithms that existed at that time. Figs. 19 and 20 show examples of EZW performance comparing it with the results from the original JPEG standard (all images were coded using the freely available VcDemo software [39]). It is notable that EZW is able to achieve its good performance with a relatively simple algorithm. EZW does not require complicated bit allocation procedures like subband coding does, it doesn't require training or codebook storage like vector quantization does, and it doesn't require prior knowledge of the image source like JPEG does (to optimize quantization tables).

EZW also has the desirable property, resulting from its successive approximation quantization, of generating an embedded code representation. What this means is that given an image coded at one rate, that this code can be used to generate the code for the same image at higher or lower rates. To generate a higher rate or more detailed representation, simply continue the coding where the original representation left off and concatenate these bits

Table 4. Resulting Output of the Subordinate Pass.

Coefficient Magnitude	Symbol	Reconstruction Magnitude
53	1	56
34	0	40

Table 5. Resulting Output of the Second Dominant Pass (T_0 =16).

Subband	Coefficient Value	Symbol	Reconstruction Value	Comment (See Text)
HL_3	–22	*ns*	–24	
LH_3	14	*ztr*	0	4)
HH_3	–12	*ztr*	0	
HL_2	21	*ps*	24	
HL_2	–9	*ztr*	0	
HL_2	13	*ztr*	0	
HL_2	–11	*ztr*	0	
HL_1	–1	*iz*	0	
HL_1	8	*iz*	0	
HL_1	–1	*iz*	0	
HL_1	0	*iz*	0	

▲ *19. The "Barbara" image coded at 0.3 bits/pixel with JPEG (a) and EZW (b) using a three-level decomposition. The resulting PSNRs for the coded images are 25.1 dB (JPEG) and 26.8 dB (EZW).*

▲ *20. The "Barbara" image coded at 0.2 bits/pixel with JPEG (a) and EZW (b) using a three-level decomposition. The resulting PSNRs for the coded images are 23.3 dB (JPEG) and 24.4 dB (EZW).*

to the original code. To generate a lower rate or less detailed code, just truncate bits off from the original code to get the desired lower code rate. The resulting codes, at either higher or lower rate, would be exactly the same as those generated from scratch using the EZW algorithm. One desirable consequence of an embedded bit stream is that it is very easy to generate coded outputs with the exact desired size. Truncation of the coded output stream does not produce visual artifacts since the truncation only eliminates the least significant refinement bits of coefficients rather than eliminating entire coefficients as is done in subband coding.

Given all the advantages, are there any disadvantages to the EZW algorithm? One problem with EZW is that it performs poorly when errors are introduced into the coded data. This is because the embedded nature of the coding causes errors to propagate from the point that they are introduced to the end of the data. This is not a problem in low noise environments but does pose a problem in the modern wireless world where error rates in data communication can be quite high. Modifications to the original EZW algorithm have addressed this issue [40]. Another concern is that the original EZW data structure is not very flexible. For example, some applications may want to selectively decode an image to increase resolution only in certain portions of the image. Such selective spatial decoding requires modifications to the original EZW algorithm. Other new techniques, such as the EBCOT algorithm which is used in JPEG 2000 [41], address some of these shortcomings of EZW.

Beyond EZW

A number of wavelet coding methods have been proposed since the introduction of the EZW algorithm [41]-[43]. A common characteristic of these methods is that they use fundamental ideas found in the EZW algorithm. As a result, the "look and feel" of these modern wavelet coders is much closer to the EZW algorithm than to subband coding. One of the listed methods is the set partitioning in hierarchal trees (SPIHT) algorithm. SPIHT became very popular since it was able to achieve equal or better performance than EZW without having to use an arithmetic encoder. The reduction in complexity from eliminating the arithmetic encoder is significant (and perhaps best appreciated by anyone who has surveyed the arithmetic encoding literature, or who has attempted to build an arithmetic encoder in hardware). Another of the techniques listed, called the EBCOT algorithm, has been chosen as the basis of the JPEG 2000 standard and is discussed further in [2]. Not included as a successor to EZW is a technique called stack run coding (SRC) [44], which has low complexity and good coding performance. SRC is somewhat of a hybrid between early and modern wavelet coders since it resembles EZW in some aspects (same uniform quantizer for all subbands) and subband coding in others (no interband prediction). A comparison of the performance of these and other wavelet-based image coding algorithms can be found on the World Wide Web at [45].

One important point that can be seen by comparing the EZW algorithm and its successors to subband coding is that lossy wavelet image coding techniques have matured significantly over the past decade. The result is that wavelet coding techniques provide a very strong basis for the new JPEG 2000 coding standard. Hopefully this tutorial has helped the reader to better understand and appreciate wavelet-based image coding and has given the reader the background to better understand the JPEG 2000 coding standard.

Acknowledgments

The author would like to thank Charles Creusere and Martin Vetterli for their careful review of this article and their many constructive comments. The author also thanks Olga Kosheleva for pointing out the VcDemo software and generating the image coding examples. Por-

tions of this work were supported by NASA FAR awards 961119 and 1218191.

Bryan E. Usevitch is an Associate Professor at the University of Texas at El Paso's Department of Electrical and Computer Engineering.

References

[1] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.

[2] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Processing Mag.*, vol. 18, pp. 36-58, Sept. 2001.

[3] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Mathematics*, vol. XLI, pp. 909-996, 1988.

[4] A. Cohen, I. Daubechies, and J. Feauveau, "Biorthogonal bases of compactly supported wavelets," AT&T Bell Labs., Tech. Rep. 20878, May 1990.

[5] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, pp. 2207-2231, Sept. 1992.

[6] V. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Mag.*, vol. 18, pp. 9-21, Sept. 2001.

[7] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[8] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[9] K. Sayood, *Introduction to Data Compression*. San Mateo, CA: Morgan Kaufmann, 2000.

[10] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[11] G. Strang, "Wavelets and dilation equations: A brief introduction," *SIAM Rev.*, vol. 31, pp. 614-627, Dec. 1989.

[12] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Mag.*, vol. 8, pp. 14-38, Oct. 1991.

[13] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[14] S. Mallat, *A Wavelet Tour of Signal Processing*. Boston, MA: Academic, 1998.

[15] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Cambridge, MA: Wellesley-Cambridge, 1996.

[16] S. Burrus, R. Gopinath, and G. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[17] A. Akansu and R. Haddad, *Multiresolution Signal Decomposition: Transforms, Subband, and Wavelets*. Boston, MA: Academic, 1992.

[18] M. Vetterli and J. Kovačevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[19] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674-693, July 1989.

[20] M. Smith and S. Eddins, "Analysis/synthesis techniques for subband image coding," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, pp. 1446-1456, Aug. 1990.

[21] M. Lightstone, E. Majani, and S. Mitra, "Lowbit-rate design considerations for wavelet-based image coding," *Multidimensional Syst. Signal Processing*, vol. 8, pp. 111-128, Jan. 1997.

[22] T. Hopper, C. Brislawn, and J. Bradley, "WSQ gray-scale fingerprint image compression specification," FBI, Tech. Rep. IAFIS-IC-0110v2, Feb. 1993.

[23] J. Woods and T. Naveen, "A filter based bit allocation scheme," *IEEE Trans. Image Processing*, vol. 1, pp. 436-440, July 1992.

[24] B. Usevitch, "Optimal bit allocation for biorthogonal wavelet coding," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1996, pp. 387-395.

[25] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205-220, Apr. 1992.

[26] J. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Trans. Image Processing*, vol. 2, pp. 1053-1060, Aug. 1995.

[27] H.V. Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968.

[28] A. Oppenheim and A. Wilsky, *Signals and Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[29] D. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1989.

[30] M. Vetterli, "Multidimensional subband coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97-112, Feb. 1984.

[31] J. Woods and S. O'Neill, "Sub-band coding of images," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 34, pp. 1278-1288, Oct. 1986.

[32] P. Cosman, R. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Trans. Image Processing*, vol. 5, pp. 202-225, Feb. 1996.

[33] Federal Bureau of Investigation, "WSQ gray-scale fingerprint image compression specification," IAFIS-IC-0110v, 2nd ed., Feb. 1993. Drafted by T. Hopper, C. Brislawn, and J. Bradley.

[34] J. Bradley and C. Brislawn, "The wavelet/scalar quantization compression standard for digital fingerprint images," in *Proc. IEEE ISCAS*, London, U.K., 1994, pp. 205-208.

[35] B. Usevitch and M. Orchard, "Smooth wavelets, transform coding, and Markov-1 processes," *IEEE Trans. Signal Processing*, vol. 43, pp. 2561-2569, Nov. 1995.

[36] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Trans. Signal Processing*, vol. 46, pp. 1027-1042, Apr. 1998.

[37] S. Mallat and W. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inform. Theory*, vol. 38, pp. 617-643, Mar. 1992.

[38] O. Johnsen, O. Shentov, and S. Mitra, "A technique for the efficient coding of the upper bands in subband coding of images," in *Proc. IEEE ICASSP*, 1990, pp. 2097-2100.

[39] Information and Communication Theory Group, VcDemo: Image and Video Compression Learning Tool. TU-Delft. Available http: //www-ict.its.tudelft .nl/~inald/vcdemo.

[40] C. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. Image Processing*, vol. 6, pp. 1436-1442, Oct. 1997.

[41] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158-1170, July 2000.

[42] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243-250, June 1996.

[43] Z. Xiong, K. Ramchandran, and M. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Processing*, vol. 6, pp. 677-693, May 1997.

[44] M. Tsai, J. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 519-521, Oct. 1996.

[45] Image Communications Lab, Wavelet Image Coding: PSNR Results. UCLA School of Engineering and Applied Sciences. Available http: //www.icsl.ucla.edu/~ipl/psnr_results.html.