WAVELET COMPRESSION FOR IMAGE TRANSMISSION THROUGH BANDLIMITED CHANNELS

A. Langi, VE4ARM, and W. Kinsner, VE4WK

Department of Electrical and Computer Engineering,
and Telecommunications Research Laboratories
University of Manitoba
Winnipeg, Manitoba, Canada **R3T 5V6**
Tel.: (204) 474-6992; Fax: (204) 2750261
eMail: kinsner@ee.umanitoba.ca

ABSTRACT

This paper studies an image compression scheme using **wavelets** for image transmission through **bandlimited** channels. During encoding, the scheme first transforms the image to a **wavelet** domain and then compresses the **wavelet** representation into an image code. Conversely, during decoding, the scheme first decompresses the image code into **wavelet** representation and then transforms it back to the original domain. The **wavelet** transform is explained from a perspective of signal representation in $L^2(R)$ space. The transform is further computed using a pyramidal algorithm. The compression is possible because (i) the **wavelet** representation has many small values that can be coded using fewer bits, and (ii) the **wavelet** basis functions are localized in space and frequency domains such that an error in the **wavelet** representation only locally affects the image in those domains. An experiment has been performed to compress two **256×256** greyscale images on such a scheme through (i) a transformation using a simple **wavelet** called DAUB4, (ii) redundancy removal by truncation the small-valued **wavelet** representation, and (iii) a ZIP compression (based on Shannon-Fano and Lempel-Ziv-Welch techniques). The results show that highly truncated **wavelet** representation (2 90%) still provides good image quality (PSNR ≥ 30dB) at less than 2 bits per pixel (bpp). Severe truncation still preserves general features of the image.

1. INTRODUCTION

Image compression is a problem of considerable importance because it leads to efficient usage of channel bandwidth and storage during image transmission and storage, respectively. There are many applications that require image transmission, such as high-definition television (HDTV), videophone, video conferencing, interactive slide show, facsimile, and multimedia **[AnRA91], [Kins91], [Prag92], [JaJS93]**. However, images require many data bits, making it impossible for real-time transmission through bandlimited channels, such as 48 **kbit/s** and 112 **kbit/s** integrated services digital network (ISDN), as well as 9.6 **kbit/s** voice-grade telephone or radio channels. Non real-time transmission on such channels also takes a long time. For example, a **256×256** pixel greyscale still image with 8 bits per pixel (bpp) requires 65,536 bytes. Transmission of such an image over the voice-grade line would require at least 54.61 s. Furthermore, one HDTV format needs **60** frames of 1280x720 pixels per second. Using 24 bpp color pixels, this HDTV format would require a channel capacity of 1,440 Mbits/s.

Image compression can improve transmission performance by reducing the number of bits that must be transmitted through the channel. As shown in Fig. 1, an image compressor compactly represents the image prior to channel encoding. At the receiving end, the demodulation and signal decoding obtain the compressed image, and an image decompressor converts the compressed image back to a **viewable** image. The transmission now requires a shorter time because the number of bits that must be transmitted has been reduced. Image compression also reduces the storage space requirement for image storage.

Although there exists several image compression methods, we still need better ones because current methods are still inadequate, especially for image **transmis-**

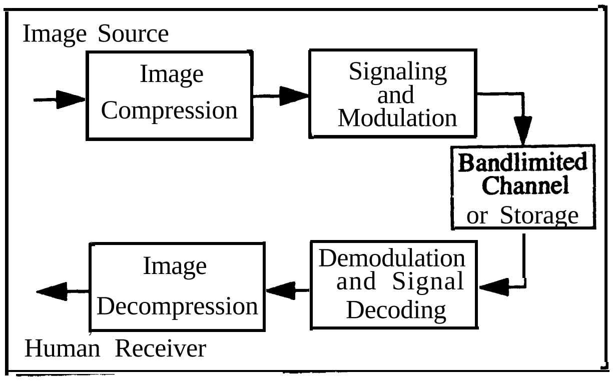


Fig. 1. Efficient image transmission system.

sion through bandlimited channels. For example, a sophisticated joint photograph expert group (JPEG) method needs 1.6 bpp for 24 bpp color images. This means that HDTV with 30 frames per second would still require a 4.423 Mbit/s channel capacity. Low bit rate compression methods have been relying on *lossy* type of compressing data, in which reducing the bit rate has caused a removal of some image information. Thus, there is a limit on how far a method can reduce the bit rate, while keeping information distortion sufficiently low.

The performance measurement of a compression technique is based on how successful it is to reduce the bit rate while maintaining image quality [Kins91]. The other performance aspects are algorithm complexity and communication delay [JaJS93]. This paper mainly focuses on the interplay between the first two aspects: reducing the bit-rate and maintaining quality.

The wavelet technique (or subband coding in general) has been emerging as an important class of image compression, in that it has an efficient representation with few visible distortions [Kins91], [Mall89], [JaJS93]. It is more promising than the JPEG standard, due to (i) a match between wavelet processing and the underlying structure of visual perception model, and (ii) a match between characters of wavelet analysis and natural images, i.e., higher frequency tends to have shorter spatial support [JaJS93]. They are also attractive because wavelet techniques are quite fast and easy to implement. Furthermore, wavelet techniques can provide special representation such as multi-resolution encoding [Mall89].

This paper shows the potential of using wavelet representation for image compression. The wavelet representation is introduced through a signal representation theory, and computed though a wavelet transform pair in Section 2. Section 3 shows a wavelet compression scheme based on the wavelet transform pair to convert spatial-domain images to and from the wavelet domain. The scheme then compresses the image in wavelet domain using a combination of lossless (no information loss) or lossy (some minor losses) techniques of data compression. Preliminary experiments on two 256x256 greyscale images (`lena. img` and `camera. img`) using a simple wavelet (DAUB4) and a simple compression (truncation and ZIP) show the potential of the scheme. Those images are used because they are standard and easily obtained by other researchers. At 4 bpp, the distortion is almost unnoticeable. At 2 bpp, the distortion is noticeable, but the errors are of salt and pepper noise type such that a simple filtering may be able to eliminate them. At highly compressed images, the details are lost but the general features are still preserved.

2. Wavelet Transform of Signals

The purpose of this section is to construct a discrete *wavelet transform* (DWT) for discrete signal representation. This construction is explained through a theory of signal representation in a special space, called $L^2(R)$, which is a space for all signals with bounded energy (square integrable) [Mall89]. This restriction does not really affect the application generality because most of the signals that are of interest to us are in that space. Although we use a one-dimensional signal as an example, we can extend the results to two-dimensional signals such as images.

2.1. Signal Representation in $L^2(R)$

In $L^*(R)$, a signal $x(t)$ can be represented in various domains using various sets of basis functions. Let a domain be spanned by basis functions $\varphi_i(t)$, having reciprocal signals $\theta_i(t)$, where $i \in Z$ (integer numbers). The representation of $x(t)$ in this domain is a set of scalars α_i [Fran69], satisfying

$$\alpha_i = \langle x, \theta_i \rangle \tag{1}$$

where $\langle \bullet,\bullet \rangle$ is an inner product, defined in $L^*(R)$ as

$$\langle x, \theta_i \rangle = \int_R x(t)\, \theta_i(t) dt \tag{2}$$

between signals $x(t)$ and $\theta_i(t)$. The $x(t)$ can be reconstructed back through

$$x(t) = \sum_i \alpha_i \varphi_i(t) \tag{3}$$

Thus $x(t)$ is a linear combination of $\varphi_i(t)$

It may happen that the basis are not countable. In other word, we have $\varphi(\tau,t)$ and $\theta(\tau,t)$, called *basis kernel*, instead of $\varphi_i(t)$ and $\theta_i(t)$. In such a continuous case, the new representation of $x(t)$, which is called $u(\tau)$, becomes

$$u(\tau) = \int_R x(t)\, \theta(\tau, t) dt \qquad t, \tau \in R \tag{4}$$

and the reconstruction becomes

$$X(t) = \int_R u(t)\, \varphi(t, \tau)\, d\tau \tag{5}$$

In many applications, it is desirable to have the same set for basis and the reciprocal, or equivalently $\varphi_i(t)$ is equal to $\theta_i(t)$, because of the difficulties obtaining and dealing with certain $\theta_i(t)$. In this case, the basis functions are ***self-reciprocal,*** or ***orthogonal.*** **Let** ***the nom*** of a **signal** $f(t)$ be defined as

$$\|f(t)\| = \sqrt{\langle f(t), f(t)\rangle} \qquad (6)$$

If the norm of each basis function is one, the basis **functions** are ***orthonormal.***

There are many basis functions available. In fact, there is a very large number of them. However, only a few of them are useful, including wavelets.

2.2. Wavelets as Basis Functions

As basis functions, **wavelets** have several special properties. First, in the frequency domain, a **wavelet** can be seen as a special **bandpass** signal. A **wavelet** operating at a higher frequency has a wider bandwidth. The bandwidth is proportional to the centre frequency of the signal. This is useful for short-time signal analysis because **wavelets** can provide enough analysis resolution in both original and frequency domains. Second, the **wavelets** in a set of basis functions are scaled (by a factor *a)* and translated (by a time-shift parameter *b*) versions of a single **wavelet** prototype $\psi(t)$ **[RiVe91].** In a mathematical notation, each **wavelet** is in the form of

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \qquad (7)$$

where *a* is the scaling factor and *b* is the translation parameter. Thus, the **wavelets** have self similarity, and useful in revealing self-similarity aspect of signals. This also leads to fast algorithms. Third, in contrast with Fourier representation, there are more than one **wavelet** prototypes. In Fourier representation, the prototype is e^{jwt} only. Thus the same **wavelet** tools can be applied to many different sets of **wavelet** basis.

The selection of *a* and *b* leads to a different class of **wavelet** representation. In general, the $\psi_{a,b}$ are not orthogonal because *a* and *b* can be any real, continuous value. In this case, applying Eq. (2) through (6) is more difficult because we must find the reciprocal basis for each **wavelet** set. However with some restrictions, we can use those equations as if the **wavelets** were orthogonal basis functions. The restrictions are that the **wavelet** prototype $\psi(t)$ must be (i) finite energy and (ii) **bandpass** (no DC component) **[RiVe91].** In this situation, if we use **Eq.** (5) and **(6),** ***we*** have a ***continuous wavelet transform*** (CWT) that operates on continuous signal using **wavelet** basis kernel. Notice that some modification is needed, because we have two parameters *a* and *b* instead of just one τ as in the Eq. (5) and Eq. (6). We then have

$$u(a,b) = \int_R x(t)\,\psi_{a,b}(t)\,dt \qquad t, a, b \in R \qquad (8)$$

and

$$x(t) = \int_R\left(\int_R u(a,b)\,\psi_{a,b}(t)\,db\right)da \qquad t, a, b \in R \quad (9)$$

It is important to notice that by changing the parameter *a* and ***b, we*** can position a **wavelet** in any location in the time-frequency plane (phase plane). The parameter *a* is a scaling parameter which changes the frequency positions of the **wavelet.** A larger *a* results in a lower center frequency, smaller bandwidth, and larger time support of the **wavelet** However, in a logarithmic **frequency** scale, different values of *a* result in different **bandpass** signals of the same bandwidth. Furthermore, parameter *b* is a time shift parameter. It changes the time position of the **wavelet.** Thus **wavelet** analysis reveals both time and frequency characteristics of the signal.

2.3. Wavelet Series

We **may** want to use discrete *a* and *b* because **continuous** *a* and *b* lead to redundant representations. One optimal selection is a dyadic sequence of $a = 2^j$ and $b = 2^j k$ **[Mall89],** resulting in orthogonal **wavelets**

$$\psi_{j,k}(t) = \sqrt{2^{-j}}\,\psi(2^{-j}t-k) \qquad (10)$$

We can now operate using Eq. (2) and (4). As before, **Eq.** (2) and (4) must be modified to use these basis signals because they have two integer indices, *j* and k, instead of one ***i.*** This modification results in a **wavelet** series representation.

2.4. Discrete Wavelet Transform

It is natural to extend the **wavelet** series for representing discrete signals using countable, discrete basis, that leads to DWT. Here, a certain selection of **wavelet** prototype and time-scale parameters leads to orthonormal **wavelets. Mallat** uses multi-resolution signal decomposition **[Mall89]** to obtain DWT, outlined here. The previous equations are still useful with slight modifications. Since we deal with countable basis functions, we use **Eq. (2) and (4)** as our basic transform pair, with Eq. (10) as the source of the basis. Furthermore, for digital signals, $x(t)$ is represented by $x[n]$, where $n \in Z$ (integer numbers).

The DWT can be developed through a multiresolution signal decomposition. Let us introduce a space V_0 which is a **subspace** of $L^2(R)$. ***The*** signal $x(t)$ lies in this space. The signal can be decomposed into several **sig-**

nals, which later called **wavelet** decomposition of $x(t)$ (analogy to Fourier decomposition of a signal into its frequency components). Conversely, we can use the decomposed signals to reconstruct $x(t)$ without any loss. The decomposed signals exist in decomposed subspaces created in a ***pyramidal*** structure. To explain this space structure, we first decompose V_0 into two orthogonal complement spaces called V_1 and O_1 denoted as

$$V_1 \oplus O_1 = V_0 \tag{11}$$

Orthogonal complement means

$$\begin{aligned} V_1 \cup O_1 &= V_0 \\ V_1 \cap O_1 &= \{\mathbf{0}\} \\ \text{VE } V_1, \mathbf{o} \in O_1 &\Rightarrow \langle \mathbf{v}, \mathbf{o} \rangle = 0 \end{aligned} \tag{12}$$

where $\{\mathbf{0}\}$ is a null set. Second similar decomposition is from V_1 into V_2 and O_2. The decomposition continues down to Jth decomposition, where V_{J-1} is decomposed into V_J and O_J, where $J \in Z$, $J > 0$. ***Thus,*** for every j, **where $j \in Z$, $J \le j \le 1$, and $J > 0$, we have**

$$V_j \oplus O_j = V_{j-1} \tag{13}$$

Mallat shows that there exists $\psi(t)$ such that ψ_j, k(f) as defined in Eq. (10) are basis functions of O_j. The basis functions are orthonormal in both O_j and $L^2(R)$ **[Mall89]**. Furthermore, **Mallat** showed that there exists ϕ (t) in the space $L^2(R)$ such that

$$\phi_{j,k}(t) = \sqrt{2^{-j}}\phi(2^{-j}t-k) \tag{14}$$

are orthonormal basis of V_j.

We are now ready to define the DWT. Let $x[n]$ be a digital signal of limited energy, i.e.,

$$\sum_n |x[n]|^2 < \infty \tag{15}$$

(This implies x [n] $\in l^2(Z)$). Let also continuous orthonormal signals $\{\psi_{j,\ k}(t)\}$ **and** $\{\phi_{j,k}(t)\}$ ($\in R$ and j, $k \in Z$) span $L^2(R)$. Finally, associate $x[n]$ with f(t) $\in L^2$ (R) according to

$$f(t) = \sum_k x[k]\phi_{0,k}(t) \tag{16}$$

The **DWT** of $x[n]$ is then a mapping from l^2 (Z) to l^2 (Z^2) **resulting** in a set of real numbers ($c_{j,k}$,$d_{J,k}$) (called **wavelet** coefficients), according to

$$c_{j,k} = \langle f(t), \psi_{j,k}(t)\rangle \equiv \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t)dt \tag{17a}$$

$$d_{J,k} = \langle f(t), \phi_{J,k}(t)\rangle \equiv \int_{-\infty}^{\infty} f(t)\phi_{J,k}(t)dt \tag{17b}$$

as in Eq. (1).

In practice, we restrict $x[n]$ to be of finite length, with N elements. In this case, J is any integer between 1 and $\log_2 N$. (In this work, we set J to $(\log_2 N)-1$, thus the description of DWT in **[Pres91a]** is directly applicable). Indexj is called ***scale,*** ranging from 1, 2, ..., to J, while k is 0, 1, ... to $(2^{-j}N)-1$. Although DWT can be defined for complex signals, we have limited the Eq. (2) and Eq. (3) to real input and basis signals only.

Orthonormality **of** signals $\{\psi_{j,k}(t)\}$ and $\{\phi_{j,k}(t)\}$ implies that the inverse DWT gives back $x[n]$ from $\{c_{j,k}\, d_{J,k}\}$ **through**

$$\begin{aligned} f(t) = \sum_{j=1}^{J}\sum_{k=0}^{2^{-j}N-1} c_{j,k}\psi_{j,k}(t) \ + \\ \sum_{k=O}^{2^{-J}N-1} d_{J,k}\phi_{J,k}(t); \end{aligned} \tag{18}$$

as in **Eq. (3),** and then

$$x[n] = \langle f(t), \phi_{0,n}(t)\rangle \tag{19}$$

2.5. Fast Pyramidal Algorithm for DWT

Given a discrete signal $x[n]$, one can actually compute the basis inner products in Eq. (17) using a matrix **multiplication,** according to

$$\begin{vmatrix} c_{0,0} \\ \cdots \\ c_{j,k} \\ \cdots \\ d_{J,k} \end{vmatrix} = \begin{bmatrix} \psi_{0,0}[0] & .** & \psi_{0,0}[N-1] \\ \cdots & & \cdots \\ \psi_{j\ k}[0] & ::: & \psi_{j,k}[N-1] \\ \cdots & \cdots & \cdots \\ \phi_{J,k}[0] & \cdots & \phi_{J,k}[N-1] \end{bmatrix} \begin{bmatrix} x[0] \\ \cdots \\ x[N-1] \end{bmatrix} \tag{20}$$

Here, the matrix elements are the sampled version of signals $\psi_{j,k}(t)$ and $\phi_{j,k}(t)$, with each signal becomes a row of the matrix. If the length of the samples is N, the matrix is *N*×*N*, and the complexity becomes $O(N^2)$.

To reduce the DWT complexity, a ***pyramidal algorithm*** can be used based on the multiresolution decomposition explained before. In the space V_j, $f(t)$ can be represented by

$$d_{j,k} = \langle f(t), \phi_{j,k}(t)\rangle \equiv \int_{-\infty}^{\infty} f(t)\,\phi_{j,k}(t)dt \qquad (21)$$

Since each $\phi_{j,k}(t)$ is a member of both V_j and $V_{j\quad 1}$, while the set of $\phi_{j\quad 1,k}(t)$ is an orthonormal basis of $Vj_{\quad 1}$, *we can express any* $\phi_{j,k}(t)$ **as**

$$\phi_{j,k}(t) = \sum_l \langle \phi_{j,k}(t), \phi_{j-1,l}(t)\rangle \phi_{j-1,l}(t) \qquad (22)$$

By changing the integration variable in the **inner-product**, it is easy to show that there exists $h[n]$ defined as

$$h[n] \equiv \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} \phi(\tfrac{1}{2}t)\phi(t-n)dt \qquad (23)$$

such that

$$\langle \phi_{j,k}(t), \phi_{j-1,l}(t)\rangle = h[l-2k] \qquad (24)$$

Thus **Eq.** (20) becomes

$$\phi_{j,k}(t) = \sum_l h[l-2k]\,\phi_{j-1,l}(t) \qquad (25)$$

Combining Eq. (21) and Eq. **(25),** and applying **inner-product** properties **[Fran69]**, we obtain

$$d_{j,k} = \sum_l h[l-2k]\,d_{j\quad 1,l} \qquad (26)$$

This relationship is very important, because one can efficiently compute $d_{j,k}$ from the previous $d_{j-1,l}$. Thus, by defining $x[n]$ as $d_{0,n}$, one can iteratively **calculate** all subsequent $d_{j,k}$ for j = 1, 2, ... using Eq. (26). Observe that Eq. (26) is essentially a filtering process of $d_{j-1,l}$ using a non-recursive filter of impulse responses $h[n]$, followed by subsampling by two. If we call this operation as H. Then

$$d_{j,k} = H\cdot d_{j\quad 1,l} \equiv \sum_l h[l-2k]\,d_{j-1,l} \qquad (27)$$

We can arrive with a similar relation for $c_{j,k}$, since $\psi_{j,k}(t)$ **is** a member Of both W_j and $V_{j\quad 1}$, while the set of $\phi_{j\quad 1,k}(t)$ is an orthonormal basis of $V_{j\quad 1}$. Using **similar** derivation yields

$$g[n] \equiv \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} \psi(\tfrac{1}{2}t)\phi(t-n)dt \qquad (28)$$

and then, after defining an operator G,

$$c_{j,k} = G\cdot d_{j\quad 1,l} \equiv \sum_l g[l-2k]\,d_{j\quad 1,l} \qquad (29)$$

Both Eqs. (27) and (29) replace Eqs. (17b) and **(17a),** respectively.

Thus in this pyramidal algorithm, the DWT becomes a processing of input signal $x[n]$ through a pyramid of operators H and G for j = 1 to *J.* In each stage of j, the outputs of the operator G are collected as $c_{j,k}$, while the outputs of the operator H are reapplied as inputs for the next stage.

Although the above derivation is for the forward DWT, similar approach can be used to derive **'upside-down'** pyramidal algorithm for the inverse of DWT. In fact, the derivation results in a simple relation between the lower stage to its next upper stage as follows

$$d_{j-1,l} = H^{*}d_{j,k} + G^{*}c_{j,k} \qquad (30)$$

where H^* and G^* are the **adjoints** of H and G **[Daub88].**

To show that this algorithm is efficient, consider a DWT of $x[n]$ having N samples. Suppose that the total of computational cost of one input sample in a stage is a constant c. At the first stage, the cost is clearly NC. At the second stage, there are only N/2 samples to be processed due to the subsampling process, resulting in a cost of *Nc*/2. Similarly, the third stage requires *Nc*/4. This process can continue until there is no input available for the next stage. Thus total complexity is

$$Nc\left(1 + \frac{1}{2} + \frac{1}{4} + \dots\right) \le 2Nc \qquad (31)$$

which is proportional to NC instead of N^2. We can estimate the c if we know the length of *h[n]* and $g[n]$. Suppose the lengths of *h[n]* and $g[n]$ ***are the same,*** which is **2*L*,** for an $L \in Z$. Then NC must be the number of multiply-and-accumulate (MAC) processes to complete both convolutions in Eq. (17) and Eq. (19) for all input samples at the first stage, which must be equal to *2LN.* Thus, total MAC processes to complete the DWT is **4*LN*,** as opposed to N^2. Table 1 shows the complexity comparison with and without pyramidal algorithm.

Table 1. Comparison of complexity of DWT and pyramidal DWT, for N input sample and filters of length *2L*.

Algorithm	$O()$	Complexity N=64, L=2
Basis Inner Product	N^2	*4096*
Pyramidal Algorithm	$4LN$	512

2.6. Daubechies Wavelets

Ingrid Daubechies has derived a class of compactly supported wavelets that are compatible with Mallat's multiresolution analysis and pyramidal algorithm. Forcing the wavelets to be compactly supported while maintaining the compatibility, she obtained a method of constructing such wavelets, as well as some properties [Daub88], such as

$$\sum_n h[n] = \sqrt{2}$$
$$\sum_n g[n] = 0 \qquad (32)$$
$$8[n] = (-1)^n h[2m+1-n] \qquad m \in Z$$

The simplest Daubechies wavelet is the DAUB4, with *L* in Table 1 is two. The filter part of *H* has four impulse responses *h*[0], h[l], *h*[2], and *h*[3], with values

$$\begin{aligned} h[0] &= 0.48296291 \\ h[1] &= 0.8365163 \\ h[2] &= 0.22414387 \\ h[3] &= -0.12940952 \end{aligned} \qquad (33)$$

The filter part of G also has impulse responses *g*[0], g[l], *g*[2], and *g*[3] which are related to *h[n]* according to Eq. (32). Here, we select *m*=1, such that *h*[*n*] and *g[n]* have a similar filter structure (down to tap positions), simplifying the implementation. This option of *m* also results in wavelet and scaling prototypes of the same supports.

2.7. Extending to Two-Dimensional Signals

A scheme to extend the DAUB4 DWT for two-dimensional signals is similar to that of the Fourier transform. Here, the computation is in two steps [Pres91a]. First, we apply a one-dimensional DWT sequentially on the columns and replace each column of the image with its DWT results. Second, we redo the similar transformation, but now on the rows, resulting in two-dimensional wavelet representation. Since the transform is orthogonal, the number of representation data is the same with the number of pixel in the original image.

3. Wavelet Compression

3.1. Basic Scheme

Lossless compression works by identifying redundant data and removing them [Kins91]. The redundancy may come from non uniform distribution of data as well as data correlation. If more compression is still necessary, the scheme further looks for irrelevant data, converts them into redundant data, and then removes them. The compression then becomes lossy because the irrelevant data cannot be recovered.

The irrelevancy can be defined according to different criteria, ranging from subjective fidelity criteria to objective fidelity criteria [GoWi87]. Irrelevant data may be those with little contribution to the fidelity. They may also be those containing little information in the Shannon information theory sense. Clearly, one can associate a measure on irrelevancy, reflected in a quality measure.

Compression methods must then concentrate on redundancy removal and irrelevancy reduction [JaJS93]. Redundancy removal methods include predictors and transforms, while irrelevancy reduction methods include quantization and data elimination. They are often combined. For example, quantization may take place in the transform domain and/or predictor error.

In wavelet approach, it is possible to employ both predictors and transforms. The wavelet transform results carry both time and frequency samples that may have sample correlation, thus predictors can be used. Furthermore, the transform itself may have removed the redundancy and made the irrelevant data more visible in the wavelet domain. We thus study some of the possibilities through experimentation. Especially, we try to find parameters that govern the irrelevancy and redundancy.

3.2. Experimental Results

In this experiment, we study the role of the wavelet coefficient *magnitude* in governing irrelevancy. We setup the experiment to study its effect on quality (related to irrelevancy) as well as bit rate (related to redundancy). Based on a scheme shown in Fig. 2., the scheme first takes the DWT of the original image. As the primary test data, we have selected `lena . img.` A less intensive experiment was performed also on `camera. img` for confirmation (see Fig. 3(a) and Fig. 4(a)). The inverse DWT can convert the image back from wavelet domain to the spatial. domain for viewing. We can then perform various processes on the transformed data. By observing the processing effects on the quality

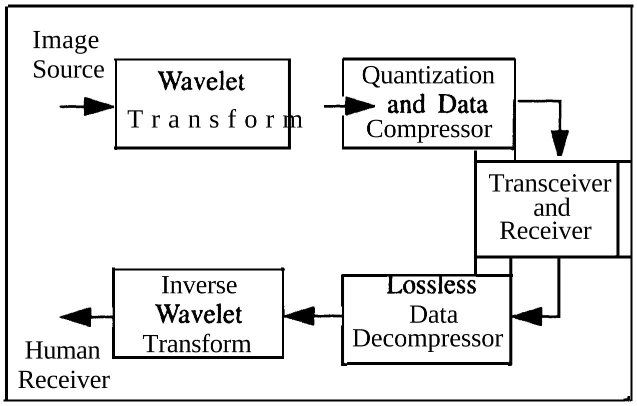


Fig. 2. A compression scheme using wavelets.

as well as the bit rate, we can design a compression scheme.

To facilitate the quality measurement, a routine computes the peak signal-to-noise ratio (PSNR) according to (in dB)

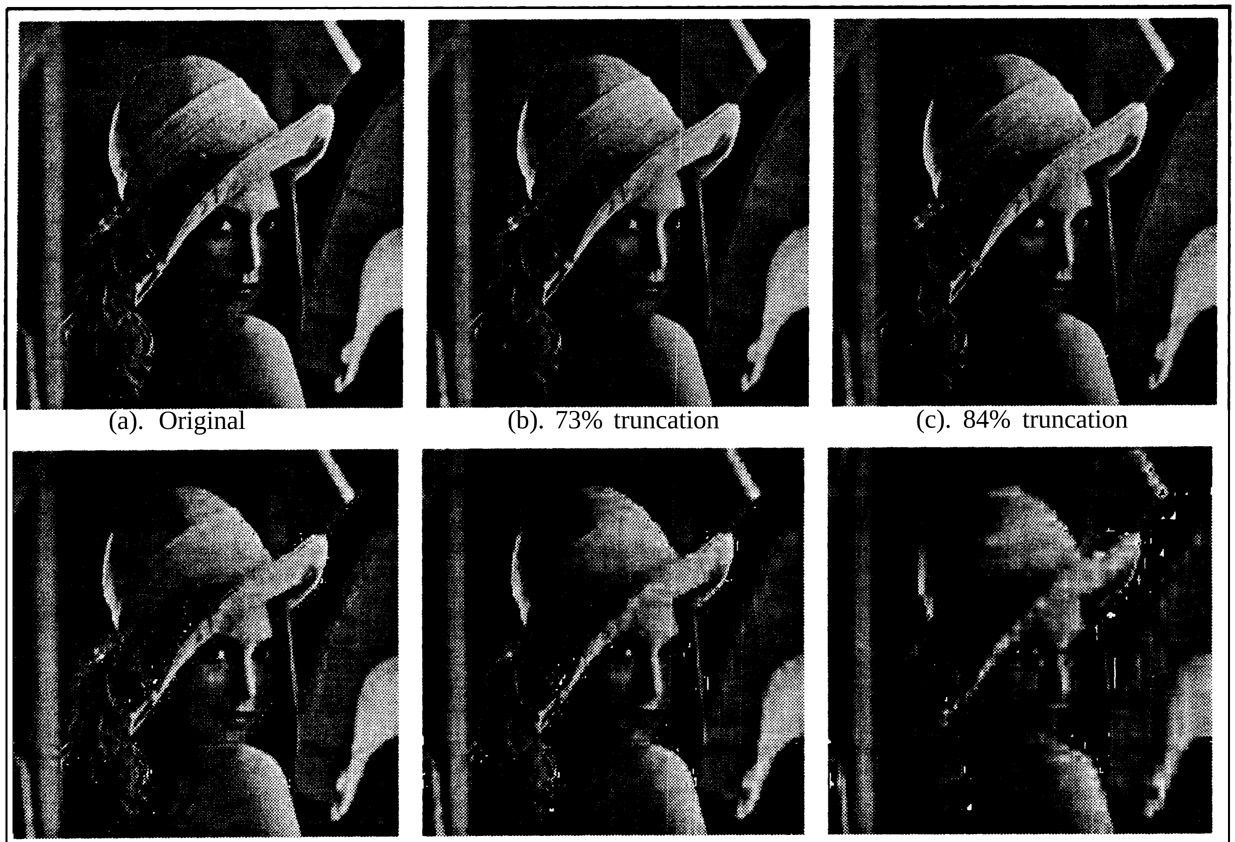
$$PSNR = 10\log\frac{255^2}{MSE} \qquad (34)$$

The mean square error (MSE) is the average of the energy of the difference between the original and the reconstructed images. We loosely define good quality as having a PSNR of greater than 30 dB.

One way to study the magnitude effect is through a coefficient truncation scheme. Here, we use a truncation threshold, such that coefficients with absolute values smaller than the threshold are considered irrelevant, and converted into zero to become redundant data. Severe truncation leads to data losses, but resulting in an efficient compression. Thus, we study the loss in PSNR and subjective quality of the image, while at the same time observe the reduction in the size of the image code file.

Our observation reveals that the smaller the magnitude of the coefficients, the more irrelevant the coefficients are. Increasing the threshold results in more coefficients being truncated. Figures 3(b) to 3(f) and Table 2 show the effects of various degrees of truncation to the image quality. The truncation results in good quality at up to 90% truncation. As shown in Fig. 5, there are only few coefficients that are responsible for total fidelity. Those coefficients must have high coefficient values, because they survive the truncation.

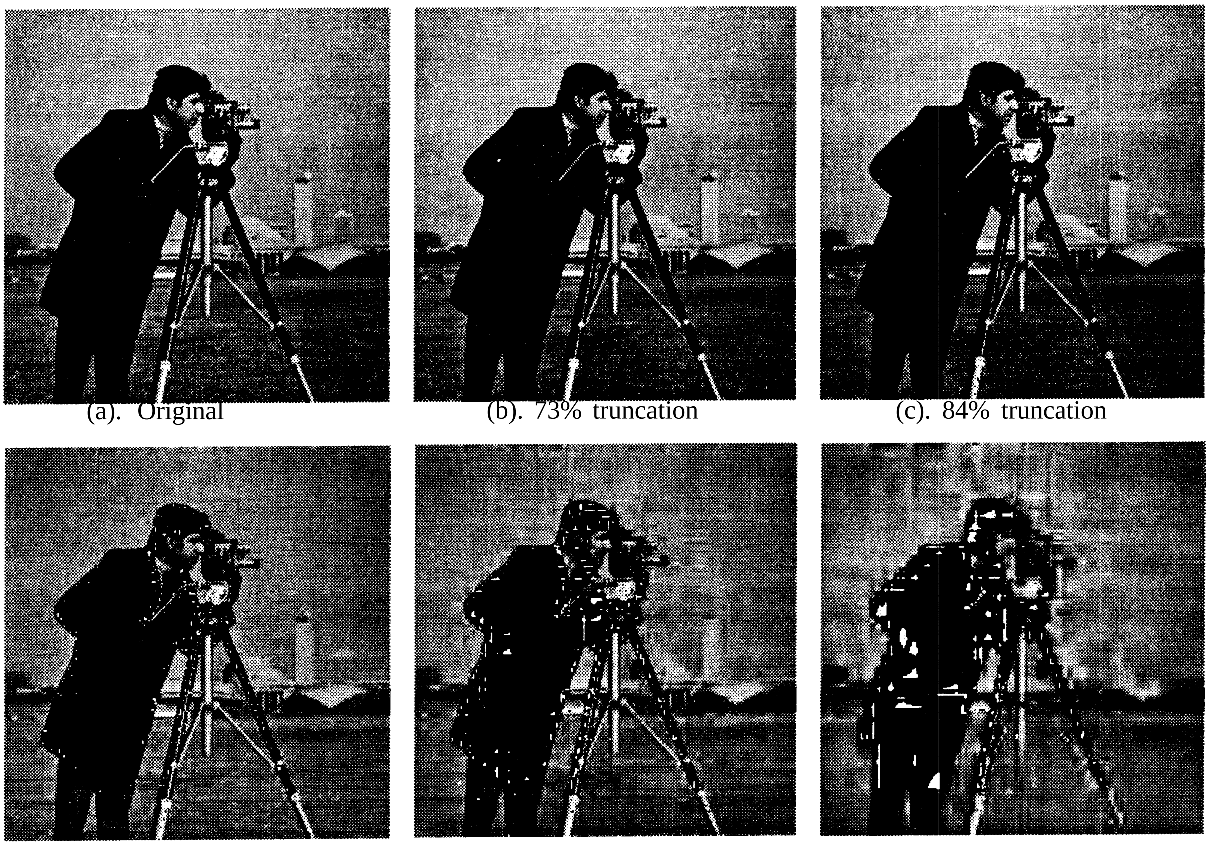
One explanation is that the wavelet transform is an orthonormal transform following the principles of Eq. (1) and (3). Such a transform satisfies the Parseval's



(a). Original (b). 73% truncation (c). 84% truncation

(d). 94% truncation (e). 97% truncation (f). 99% truncation

Fig. 3. Effects of various degrees of truncation on the image quality. See also Table 2.



(a). Original (b). 73% truncation (c). 84% truncation

(d). 93% truncation (e). 97% truncation (f). 99% truncation

Fig. 4. Effects of various degrees of truncation on the image quality. See also Table 3.

Table 2. The PSNR values and bit rates of the images shown in Fig. 3.

% Truncation	PSNR (dB)	Bit Rate (bpp)
77	39.16	3.8
86	33.23	2.5
94	27.16	1.4
97	23.76	0.7
99	21.08	0.36

relation which equates both energies in the original and wavelet domains [Fran69]. Since the wavelet basis is orthonormal, the greater the magnitude of one coefficient, the higher its contribution to the energy. Thus truncating the such coefficients would result in high MSE values, reducing the PSNR.

For more truncation, the distortion is localized and looks like salt-and-pepper noise, as shown in Fig.3(b) to 3(e). This is due to the fact that the coefficients responsible for the constructing the pixels have been eliminated. This also verifies one of wavelet properties of being localized in both spatial and frequency domain. Thus quantization error in the wavelet domain does not translate in distortion distrilbuted all over the image. Low pass or median filters should be able to reduce the effects of such a distortion. Severe truncations still pre-

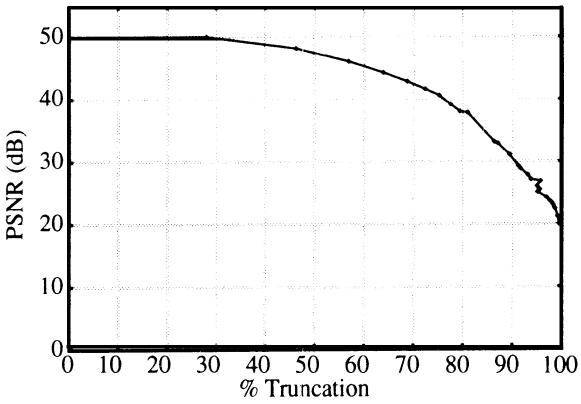


Fig. 5. Effects of % of truncation on the image quality for `lena.img`.

serve the general looks, as shown in Fig. 3(f). The images are very distorted, but the general feature is still observable.

The set of truncated coefficients has an unbalanced distribution of zeros now such that a **lossless** compression should be able to compress it. We can then use the **ZIP** compression to compress the coefficients. As shown in Fig. **6, 90%** of truncation translates to 2 bpp representation. The ZIP program was able to identify data repetition in the truncated coefficients, and encoded them compactly.

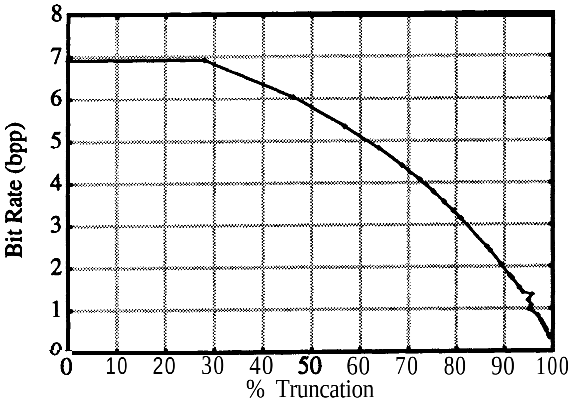


Fig. 6. Effects of the truncation to ZIP compression for `lena.img`.

The same scheme is also used on another image, `camera. img`, with almost similar results. Although the quality of the reconstructed image is not as good as that of `lena. img`, the results show similar trends (see Fig. 7 and 8). Also, Fig. 4 and Table 3 show similar effects of the truncation on the image quality.

Table 3. PSNR values and bit rates of images in Fig. 6.

% Truncation	PSNR **(dB)**	Bit Rate (bpp)
73	40.18	3.98
84	31.10	2.8
93	23.26	1.6
97	19.51	0.85
99	17.46	0.39

3.3. Compression Performance

We can then use the threshold scheme as a simple image compression scheme. The compression takes the forward **DWT** of the image, truncates the small magnitude coefficients according to a threshold, and losslessly compresses the truncated coefficients to become the compressed image. The decompression then starts with **lossless** decompression of the compressed image, and inverse transforms the results to obtain **viewable** image. Figures 9 and 10 show the compression performance measured for `lena. img` and `camera.img`, respectively. We observe that in 2 bpp the image still have good quality.

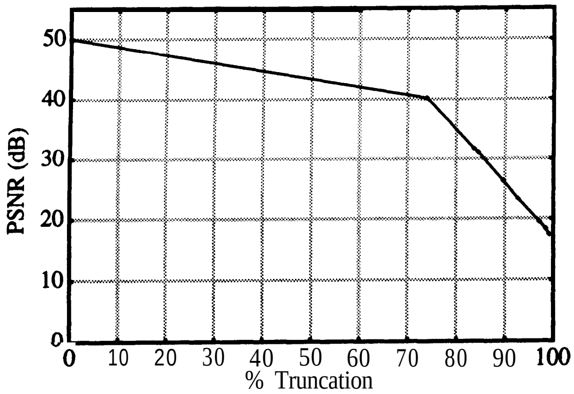


Fig. 7. Effects of % of truncation on the image quality for `camera. img`.

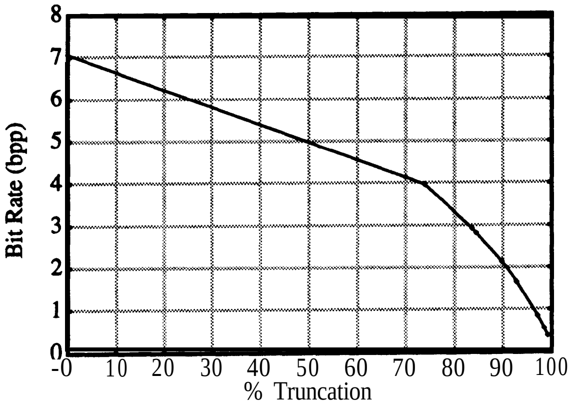


Fig. 8. Effect of the truncation to **ZIP** compression for `camera.img`.

4. Discussion

Localization characteristic and sparsity of **wavelet** representation are interesting features for image compression. The localized property of the **wavelet** coefficients reduces the effect of a quantization **error** to the total image. This is an advantage over Fourier representation. The **sparsity** means that most of the image energy is distributed among a few basis functions only. Thus a scheme that finely quantizes the high coefficients while coarsely quantizes the small-valued coefficients leads to efficient compression with high quality.

From a practical perspective, the compression scheme is interesting due to the simple and fast computation structure. Since the implementation of the **wavelet**

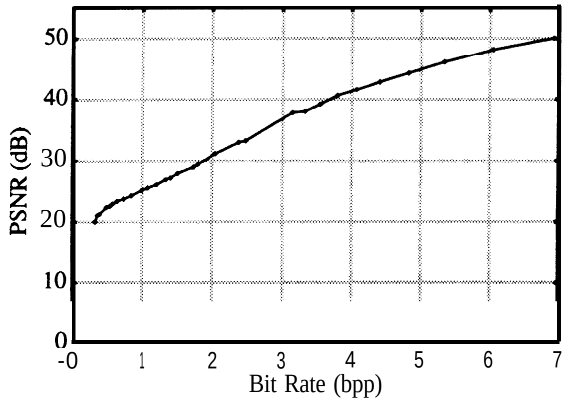


Fig. 9. Quality of the image for different compression rates for `lena . img`.

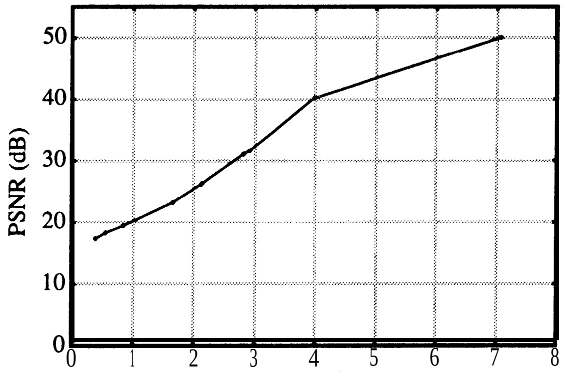


Fig. 10. Quality of the image for different compression rates for `camera.img`.

transform pair consists of filtering only, fast dedicated hardware is possible.

In this scheme, two aspects are open for improvements: (i) selection of the **wavelet** prototype, and (ii) compression of the coefficients. In our experiment, we selected DAUB4 because of its simplicity, its localized property, and its immediate availability. Thus, the selection has not been made through a study of characteristics of different wavelets. A more elaborate choice of **wavelet** may lead to much better results. For example, **[Mall89]** reports the use of image distribution characteristics into consideration results in 1.5 bit/pixel, with only a few noticeable distortions. Furthermore, in our experiment we apply brute force truncation to eliminate redundancy. A more thoughtful method should produce better results. A program which optimally embeds **Huffman** coding in the compression scheme results in 3: 1 **lossless** compression and 50: 1 lossy compression with some degradation **[Pres9** 1 b].

We conclude that **wavelet** coding is an important method for image compression. Characteristics of the **wavelet** representation such as locality and sparsity can be exploited for compact representation. The magnitude of a **wavelet** coefficient determines the coefficient's significance with respect to the image fidelity. This can lead to very promising compression schemes as demonstrated in this paper.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Telecommunications Research Laboratories **(TRLabs).** One of the authors (AL) wishes to thank ILK-Microelectronics ITB, Laboratory of Signals and Systems ITB, and PT INTI Persero, **Bandung,** Indonesia, for their support in this research work.

REFERIENCES

[AnRA91] P. H. Ang, T. A. Ruetz, and D. Auld, "Video compression makes big: gains", *IEEE Spectrum Magazine,* IEEE Cat. **0018-9235/91,** pp. 16-19, Oct. 1991.

[Daub88] I. Daubechies, **"Orthonormal** bases of compactly supported wavelets," *Comm. Pure. App. Math., v.* XLI, 1988, pp. **909-996.**

[Fran69] L. E. Franks, ***Signal*** *Theory. New* Jersey: Prentice-Hall Inc., 317 pp., 1969.

[GoWi87] R. C. Gonzales and P. **Wintz,** *Digital Image Processing.* Reading MS: Addison-Wesley, 502 pp., 1987

[JaJS93] N. **Jayant, J.** Johnston, and R. **Safranek,** "Signal compression based on models of human perception," *Proceeding IEEE,* 1993, v. 8 1, no 10, pp. 1385-1421.

[Kins91] W. Kinsner, "Review of data compression methods, including Shannon-Fano, **Huffman,** arithmetic, Storer, Lempel-Ziv-Welch, fractal, neural network, and **wavelet** algorithms", *Technical Report,* **DEL91-1,** University of Manitoba, 157 pp., Jan. 1991.

[Mall89] S. **Mallat,** "A theory for multiresolution signal decomposition: the **wavelet** representation," *IEEE Trans. Patt. Anal. Machine* ***Intell.,*** vol. 11, no 7, pp. 84-95, July 1989.

[Prag92] D. S. Prague, "How will multimedia change system storage **?",** *BYTE Magazine,* McGraw-Hill, Peterborough NH, pp. 164-165, Mar. 1992.

[Pres91a] W. H. Press, **Wavelet** Transforms, ***Harvard-*** *Smithsonian Centre for Astrophysics Preprint No.* 3184, 1991. (Available through anonymous ftp at 128.103.40.79, directory `/pub`).

[Pres9 1 b] W. H. Press, FITSPRESS, ***Harvard-Smithsonian*** *Centre for Astrophysics,* Beta Version 0.8, 199 1. (Available through anonymous ftp at **128.103.40.79,** directory `/pub` and file `fitspress08.tar.z`).

[RiVe91] 0. Rioul and M. Vetterly, "Wavelets and signal processing", *IEEE* ***Signal*** *Processing* ***Magz.,*** IEEE CT **1053-5888/91** pp. 14-38, Oct. 1991.