Image compression using wavelets and JPEG2000: a tutorial

by S. Lawson and J. Zhu

The demand for higher and higher quality images transmitted quickly over the Internet has led to a strong need to develop better algorithms for the filtering and coding of such images. The introduction of the JPEG2000 compression standard has meant that for the first time the discrete wavelet transform (DWT) is to be used for the decomposition and reconstruction of images together with an efficient coding scheme. The use of wavelets implies the use of subband coding in which the image is iteratively decomposed into high- and low-frequency bands. Thus there is a need for filter pairs at both the analysis and synthesis stages. This paper aims in tutorial form to introduce the DWT, to illustrate its link with filters and filterbanks and to illustrate how it may be used as part of an image coding algorithm. It concludes with a look at the qualitative differences between images coded using JPEG2000 and those coded using the existing JPEG standard.

1 Introduction

In the 7 years since the publication in this journal of a paper on wavelets by Bentley and McDonnell[1], there have been many developments in the area of wavelets, particularly in their applications. This paper presents a tutorial on the discrete wavelet transform (DWT) and introduces its application to the new JPEG2000* image compression standard.

We start by showing how, from a one-dimensional low-pass and high-pass filter pair, a two-dimensional transform can be developed that turns out to be a discrete wavelet transform. The article will look at how the four subbands generated by the DWT can be interpreted and will review the various ways of processing the image data at each stage of the transform. The next topic will be the use of compression algorithms that act on the DWT output (the 'wavelet coefficients'), e.g. SPIHT, EZW and EBCOT. The discussion will look particularly at the EZW algorithm and its essential features. Examples will be given showing the effects of decomposition, quantisation, coding and then reconstruction of images. Section 6 looks at the emerging international standard JPEG2000 and how it improves compression quality when compared with JPEG.

Bentley and McDonnell's paper discussed both continuous and discrete wavelets. There are applications for which continuous wavelets are the natural choice, e.g. in sonar or radar signal detection where we need information about range[2]. Here, however, we will only discuss discrete wavelets.

Our journey will begin with filters, in particular with a pair of finite-impulse response (FIR) filters, for with these we can construct a module that may be used any number of times to construct a filterbank.

Filterbanks were first introduced over 25 years ago for speech analysis and synthesis applications. However the concept goes back further than that. A discrete Fourier transform can also be viewed as a filterbank and interpreted as a bank of bandpass filters tuned to different frequency bands. Unfortunately its frequency selectivity is poor for a given number of points.

We are interested in filterbanks because of their application to subband coding, in which a signal is decomposed into frequency subbands. This allows us to perform different signal processing tasks on each subband, e.g. different quantisation strategies.

2 1-D FIR filter design

The choice of FIR (non-recursive) filters for the construction of filterbanks is appropriate for several reasons. Of these, the ability to realise an exact linear phase response is important because many applications, including image compression, require phase distortion to be reduced. In addition, FIR filters are always stable. There are some disadvantages in using FIR filters; for example, the filter orders are generally higher than those of IIR (infinite impulse response, recursive) filters

*A new image compression standard from the ITU's Joint Photographic Experts Group committee. See http://www.jpeg.org

meeting the same specification. However the industry uses these structures, so that they have become the standard. Furthermore, their physical structure is simple and well understood and it is arguably easier to develop VLSI hardware architectures for FIR filters if required[3].

The transfer function of a general Nth order FIR filter, $G(z)$, is given by:

$$G(z) = \sum_{n=0}^{N} g_n z^{-n} \quad (1)$$

where $z = \exp(j\omega T)$ and $\{g_0, g_1, \ldots, g_N\}$ are the filter coefficients. For a linear phase response, $g_i = g_{N-i}$ for $i = 0, 1, \ldots, N/2$. So the design problem is to find the values of the coefficients that will yield a filter that meets an appropriate specification. Many techniques are available and optimisation tends to be used most to design the filters in filterbanks.

3 1-D filter banks and subband coding

When two FIR filters are combined with decimators in the structure shown in Fig. 1, we have what is known as a two-channel analysis filter bank. To complement it, a synthesis filter bank is constructed with the aim that their cascade will yield at the output a good estimate of the original input signal. The analysis filterbank splits the signal into two equal frequency bands so that the filters $H(z)$ and $G(z)$ are lowpass and highpass, respectively*. After filtering, the signal's sampling frequency is too high and so half the samples can be discarded. The symbol $\downarrow 2$ used in Fig. 1 means decimation by 2, i.e. discarding every other sample. At this stage the amount of data at the two outputs is still equal to the amount of input data. As we will see, it is the partitioning of the signal into frequency bands that allows us to make savings in terms of coded data. The synthesis filter bank reconstructs the signal from the two filtered and decimated signals (Fig. 2). This involves expanding the signals in each branch by 2 (denoted by $\uparrow 2$) and filtering[4]. The expansion or interpolation is achieved by inserting zeros between successive samples.

The theory of filterbanks tells us that there are three principal errors, namely aliasing error, magnitude and phase distortion. Perfect reconstruction (PR) is achieved when we eliminate all three errors. The search for the right kind of filters to be used in filterbanks has led to many different filter classes. Amongst these are the orthogonal filters, so called because of an orthogonal relationship between the coefficients of the two filters. However an orthogonal filter can not have linear phase, except in the case of the simple Haar filter set[5]. Linear phase is an important requirement for image coding applications. A second class, biorthogonal filters, gives linear phase and in certain cases the coefficients offilters almost satisfy the orthogonality relationship. Orthogonality is an important property of functions (filters) used to analyse a signal. Fourier analysis, for example, uses sine and cosine functions that are known to be mutually orthogonal. As an example, the filter coefficients of the biorthogonal 9/7 filter pairs* used for the analysis and synthesis filters are shown in Table 1. There is a relationship between analysis and synthesis filters that stems from the PR requirement. The frequency

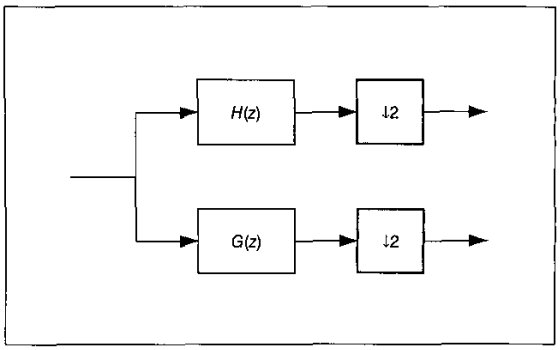


Fig. 1 Two-channel analysis filterbank

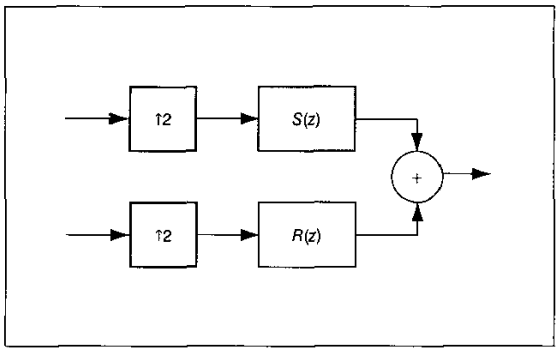


Fig. 2 Two-channel synthesis filterbank

*We use the terminology $H(z)$ and $G(z)$ to represent the transfer functions of the low- and high-pass filters, respectively, in agreement with many authors of texts on wavelets.

*The term 9/7 refers to the number of filter coefficients used, respectively, in $H(z)$ and $G(z)$.

Table 1: Coefficients of the biorthogonal 9/7 filter set where the transfer functions are $H(z)=\sum_{n=-4}^{4} h_n z^{-n}$, $G(z)=\sum_{n=-3}^{3} g_n z^{-n}$, $S(z)=\sum_{n=-34}^{4} s_n z^{-n}$ and $R(z)=\sum_{n=-4}^{4} r_n z^{-n}$

	Analysis filters		Synthesis filters	
n	Lowpass, h_n	Highpass, g_n	Lowpass, s_n	Highpass, r_n
0	0·8527	0·7885	0·7885	0·8527
±1	0·3774	−0·4181	0·4181	−0·3774
±2	−0·1106	−0·0407	−0·0407	−0·1106
±3	−0·0238	0·0645	−0·0645	0·0238
±4	0·0378			0·0378

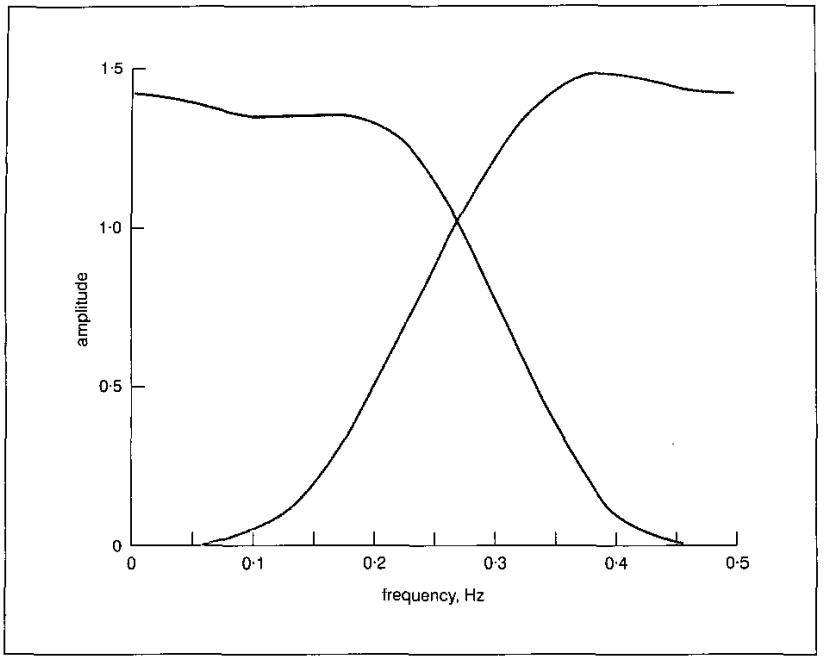


Fig. 3 Frequency responses of low- and high-pass filters used in two-channel 9/7 analysis bank. The sampling frequency is normalised to unity.

responses of the analysis filters are shown in Fig. 3.

The two-channel bank can be used as a building block to create different topologies, known as tree structures. By this means we can decompose an input signal into any desired number of bands with particular emphasis on one or a group of subbands. Fig. 4 shows the arrangement for a four channel uniform bank. The input signal, $x(k)$, is sampled at rate f_s. The four output signals—$u(k)$, $v(k)$, $w(k)$, $y(k)$—are each sampled at $f_s/4$ and each occupies about one quarter of the input signal's bandwidth. In Fig. 5, the decomposition of a signal is non-uniform: only the lowpass band is split into subbands. Here, the input signal is again sampled at rate f_s but the outputs have different bandwidths and sample rates. The bandwidth of $y(k)$ is about twice that of $w(k)$, which in turn is about twice that of $v(k)$ and $u(k)$. This type of tree is known as an octave decomposition and is also the same as the discrete wavelet transform (DWT). However the DWT is more general as the filters $H(z)$, $G(z)$ are selectable from a large range of FIR and IIR types[5]. The wavelet decomposition itself can be generalised, leading to many different structures. Space does not permit discussion here.

4 2-D filterbanks and subband coding

2-D filters have transfer functions in 2 variables, i.e. $B(z_1, z_2)$ where z_1 and z_2 relate, in the case of images, to the two spatial dimensions. The mathematics for two-variable polynomials is much more difficult than for one; for example, factorisation is not possible in general. However the problem can be simplified by allowing separable transfer functions, i.e. $B(z_1, z_2) = B_1(z_1)B_2(z_2)$. This eases computation too, but restricts the range of filter responses. This is the solution often taken and one that we will adopt here. For a 2-D signal, we will use the notation $x(m, n)$, where m and n are the two spatial directions. The 2-D separable analysis filterbank has essentially four stages, filtering and decimation along the rows and then along the columns. Fig. 6 illustrates these stages by considering an image as an $M \times N$ array of pixel values called X and charting its progress through the filter bank. At point A in Fig. 6, we have $X = \{x_{ij}: i = 1,2,3,...M, j = 1,2,3,...N\}$. The filter $G(z_1)$ operates on each row of X to produce a new array $Y = \{y_{ij}: i = 1,2,3,...M, j = 1,2,3,...N\}$ at point B. Decimation will delete every other column to produce at C, $Y' = \{y_{ij}: i = 1,2,3,...M, j = 1,3,5,...N\}$. This matrix is then filtered by columns with $G(z_2)$ and at D we have $Z = \{z_{ij}: i = 1,2,3,...M, j = 1,3,5,...N\}$. Finally the filtered output is decimated by deleting every other row to finally produce at E, $Z' = \{z_{ij}: i =$

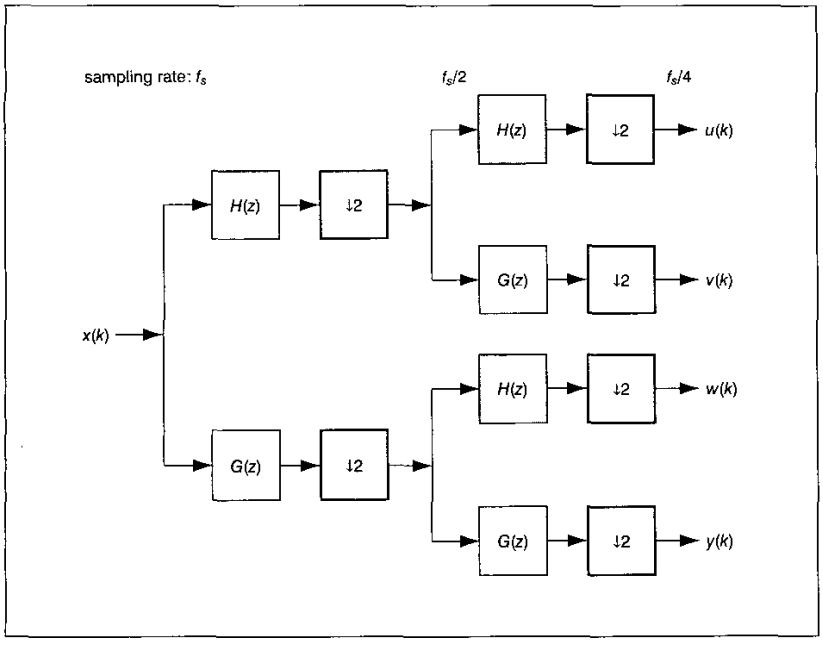


Fig. 4 Four-channel analysis filter bank. f_s is the sampling rate.

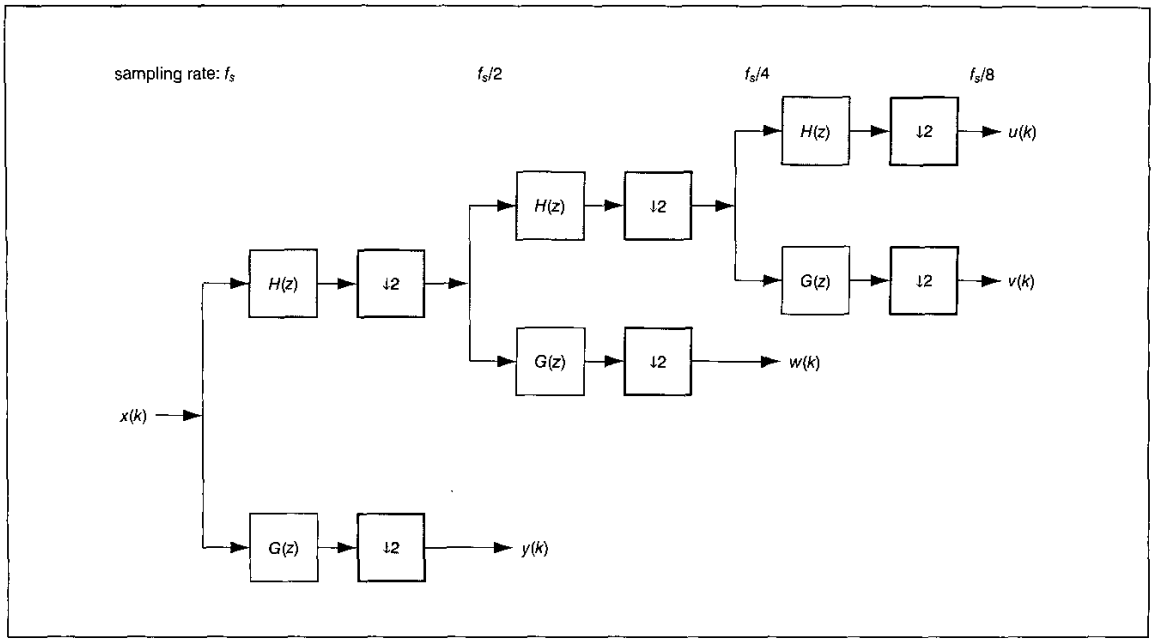


Fig. 5 Three-scale analysis filter bank (wavelet). f_s is the sampling rate.

1,3,5,...M, j = 1,3,5,...N}. If, for example, the input image has dimensions of 512 × 512 pixels then each of the four outputs in Fig. 6 will have dimensions of 256 × 256 pixels.

One level of decomposition as defined by Fig. 6 is not very helpful and we can develop many different tree structures as was done in the 1-D case. In the case of the octave or wavelet decomposition, after three such levels or scales, the number of subbands has reached 10 (Fig. 7). The wavelet transform is an example of a time–scale transform rather than the time–frequency transform associated with the Fourier transform.

The processing along rows and columns does lead to problems at the image edges. Filtering a row, for example, involves the data in the row being convolved with the filter coefficients. Unlike conventional filtering where the data is a continuous stream of samples, the row length is fixed. The solution to this problem is to periodically extend the image. The number of samples after filtering will be greater than the row length so truncation together with an appropriate time shift will be necessary (Fig. 8*a*). Unfortunately, as the value of $x(N)$ can be very different from that of $x(1)$, edge artefacts may be introduced. To alleviate this problem, the image should be extended in a symmetric way (Fig. 8*b*). This ensures that there is a smoother transition between sample values at the image boundaries. Two other types of symmetric extension are possible and their uses are explained in a recent paper by Li[6].

It is also possible to avoid the need for periodic extension of any kind by storing the filters' states for each row and column. This is especially useful if IIR filters are used, where symmetric expansion is not possible. The drawback of course is the additional amount of data

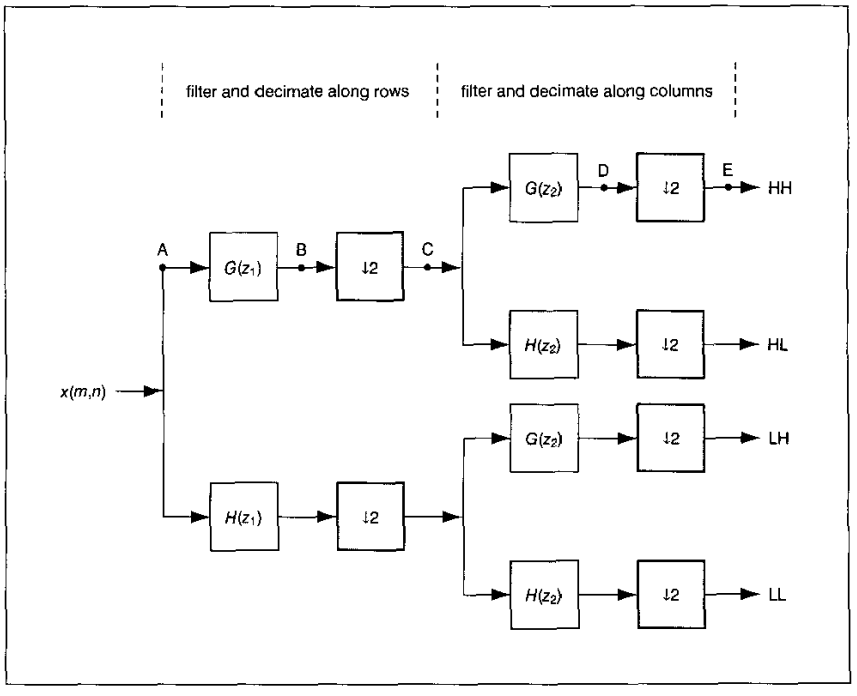


Fig. 6 2-D separable analysis filter bank

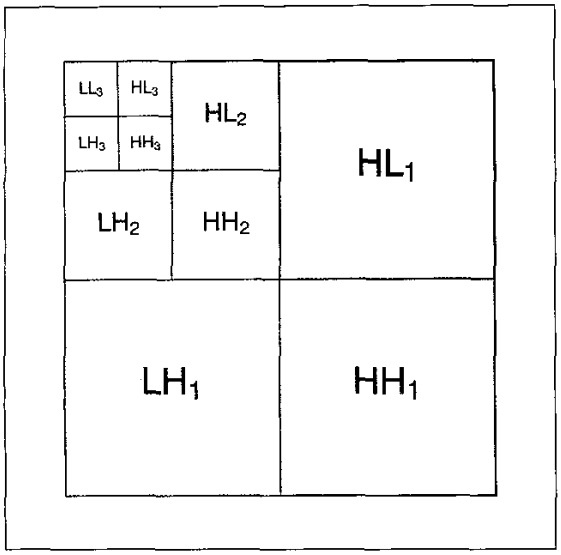


Fig. 7 3-scale wavelet subband decomposition. The letters denote whether lowpass (L) or highpass (H) filters were used in row and column filtering and the numbers denote the scale. Thus LH_2 means the LH output according to Fig. 6 at scale 2.

required that would need to be coded for transmission purposes[7].

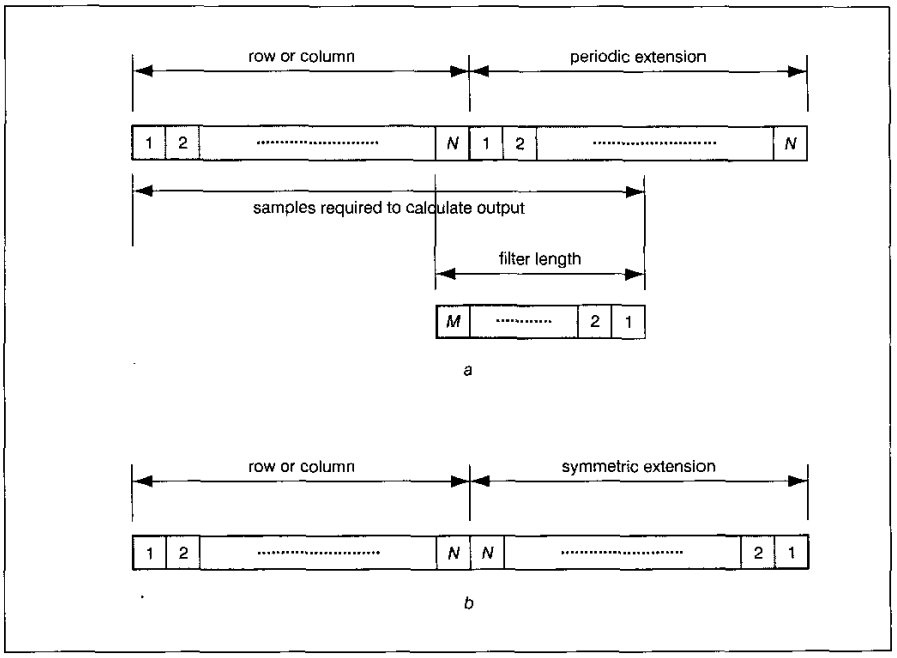
The discussion so far has concentrated on the analysis filter bank. The equivalent synthesis bank is illustrated in Fig. 9. For perfect reconstruction, there will be a relationship between the synthesis filters $R(z)$, $S(z)$ and those in the analysis bank.

5 Image compression and coding

The previous discussion has shown how we can take an image and split it into subbands using a wavelet decomposition. The wavelet decomposition is useful because with many images most of the energy lies in the low-frequency subband, although there is still some correlation between bands. This can be appreciated by looking at Figs. 10 and 11, which show an original image and the decomposed image using 3 scales, the latter having been brightened in order to see the high-frequency bands. We are predominantly interested here in compressing natural images. As a consequence if there is little energy in a specific location of a low-frequency band, such as HL_3, then it is likely that the energy in a high-frequency band, such as HL_2 or HL_1, at the same spatial location, will also be small. We can use this property to develop a hierarchical tree that starts at a coefficient in any subband except HL_1, LH_1 or HH_1, which we call the *root*, and branches to the same spatial locations in other higher frequency subbands, called *descendants*. In particular we are interested in zero roots, since this will allow us to radically reduce the amount of data that needs to be transmitted. Fig. 12 illustrates this by showing several trees starting from different roots, indicated by circles. As we move through the subbands, the number of descendants increases by a factor of 4 in each subband.

The amount of data after performing a 3-scale wavelet decomposition is still the same as that contained in the original image. However we are now in a position to look at the quantisation and coding of this output data (wavelet coefficients). A typical wavelet transform-based image coding system is illustrated in Fig. 13. A wavelet transform using a desired number of scales is applied to the image pixels. The wavelet coefficients (transform output) are organised in a certain way, quantised and entropy encoded, resulting in a bit stream. For decoding, the reverse is done. The encoded bit stream is entropy decoded and inverse quantisation is carried out to recover the wavelet coefficients, organised in the same way as for

Fig. 8 (*a*) Periodic expansion of row or column of an image; (*b*) symmetric expansion of row or column of an image



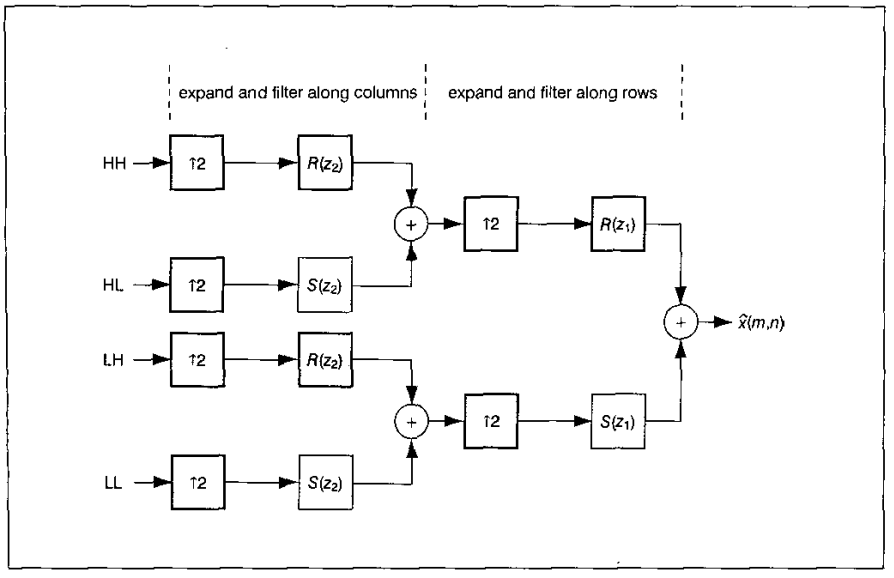


Fig. 9 2-D separable synthesis filter bank.

encoding. Then the coefficients are inverse-transformed to reconstruct the image pixels. The process described results in losses and so is called *lossy coding* or *lossy compression*. The difference between the original and recovered images is measured and known as the PSNR (peak-signal-to-noise ratio). For an $M \times N$ image, it is defined as follows:

$$PSNR = \frac{MNP^2}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |y(m,n) - x(m,n)|^2} \quad (2)$$

where P is the maximum value for a pixel, i.e. 255 for 8-bit precision, and $x(m, n)$, $y(m, n)$ are, respectively, the original and recovered pixel values at the mth row and nth column. PSNR is normally quoted in decibels. We are often interested in the way that PSNR varies with compression rate, measured in bits/pixel (bpp), producing what is known as a rate–distortion curve. Some applications, such as medical imaging, require that any compression is lossless.

Various techniques have been developed to perform the quantisation and coding given that the data has been transformed by the DWT. The first of these was the embedded zerotree wavelet (EZW) coding method[8], which was improved by Said and Pearlman in their SPIHT (set partitioning in hierarchical trees) algorithm[9], and more recently by the embedded block coding with optimised truncation (EBCOT) technique[10]. All of these techniques exploit the correlation between neighbouring coefficients and/or inter-subband correlation, and use bit-plane coding, from the most significant bits (MSB) to the least significant bits (LSB) (Fig. 14). In addition, these encoding algorithms are iterative in the sense that after an initial transmission of encoded bits, further refinement bits can be generated, encoded and transmitted. The refinement can continue until the desired compression rate has been reached.

To understand the principles governing the EZW algorithm and its derivatives, it is helpful to look at the development of a significance map. This map contains

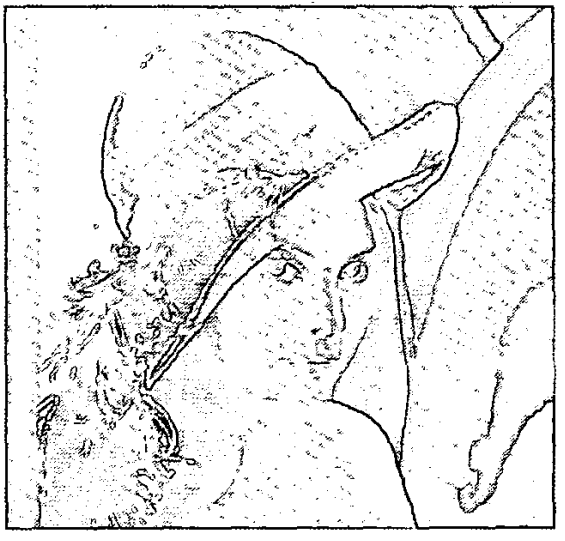


Fig. 10 The standard 512 × 512 × 8 Lena image

Fig. 11 The result of applying a 3-scale DWT to the Lena image. The high-frequency subbands have been brightened so as to see them.

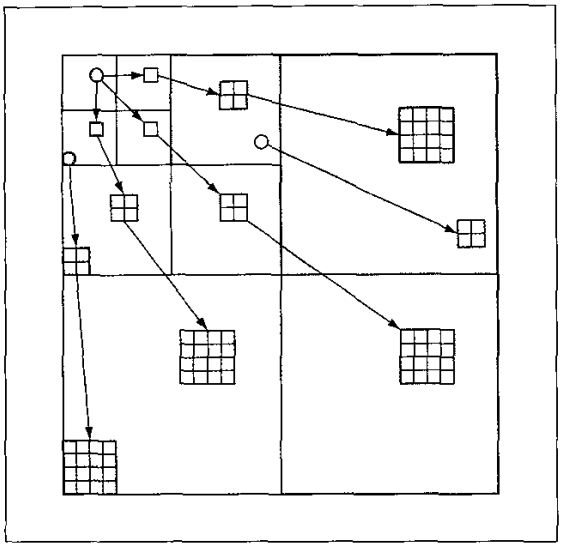


Fig. 12 Hierarchical trees for EZW and derivatives. Circles represent roots, and each small square represents a group of four adjacent wavelet coefficients.

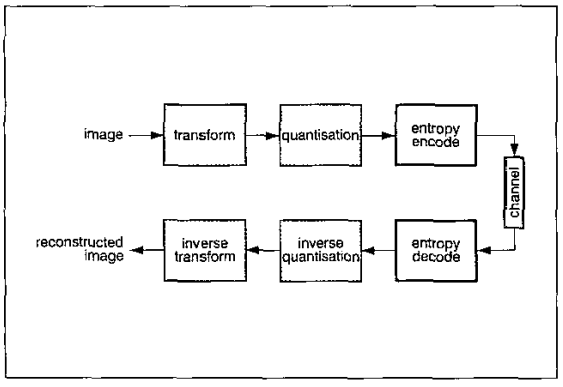


Fig. 13 Typical wavelet transform-based image coding system

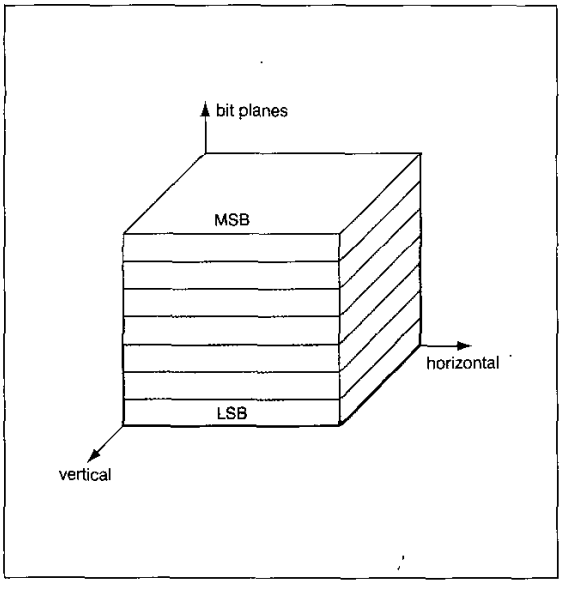


Fig. 14 Bit planes of wavelet coefficients

information about wavelet coefficients and their relationships across subbands. If we begin with coefficients in the coarsest scale, e.g. LL_3 in the 3-scale example of Fig. 7, then we may build a tree by noting that a coefficient will have descendants at finer scales with the same spatial location (Fig. 12). We can go further by considering coefficients that are insignificant with respect to some threshold value, T, i.e. that are smaller in magnitude than T. In addition we hypothesise that all their descendants will also be insignificant. For natural images, this is a reasonable assumption (see Fig. 11). So part of the coding algorithm is the identification of zerotrees. Each zerotree begins at a *root* coefficient that is insignificant and is not itself a descendant. These zerotrees allow a large reduction in the amount of data to be transmitted since, by knowing the position of the root, we can discover all of its descendants[8].

In addition to finding zerotrees, the EZW algorithm uses successive approximation based on the threshold T. By scanning through the wavelet coefficients in a certain way (Fig. 15) it is possible to encode the data efficiently. The scanning procedure also ensures that coefficents are processed before their descendants. The algorithm repeatedly scans the data with a reducing threshold value, T_j, such that $T_j = T_{j-1}/2$ and $T_0 = 2^{\lfloor \log_2 k \rfloor}$, where k is the largest wavelet coefficient in magnitude and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. Each scan has two passes, known as dominant and subordinate, and corresponding lists. The dominant list keeps a note of the co-ordinates of those coefficients that have not yet been found significant whilst the subordinate list stores the magnitudes of significant coefficients. Ultimately, the algorithm produces a stream of symbols chosen from the alphabet of Table 2. The final part of the encoding process is to apply entropy coding to the symbol stream to produce a bit stream that may then be transmitted and/or stored.

The EZW algorithm can be used iteratively until a desired compression rate has been reached. In addition a transmitted image can be recovered with reduced quality even if the decoding process is terminated before all the received data has been processed because later bits only refine the image. For these reasons EZW is known as an embedded algorithm.

Although EZW and its derivatives outperform compression using subband coding and the discrete cosine transform (DCT), embedded coding does have some disadvantages, such as sensitivity to bit errors caused by noisy communication channels. Of the two well known developments of EZW, SPIHT achieves improved performance even without entropy coding[9]. This is accomplished by coding the significance map in a more efficient way using a partitioning algorithm that works on the sets of coefficients organised in hierarchical trees. We will use the SPIHT algorithm in the following example.

Recalling that the original $512 \times 512 \times 8$ bit Lena image is shown in Fig. 10, Fig. 16*a* shows the reconstructed image at 1·0 bpp with PSNR = 40·41 dB and Fig. 16*b* shows the reconstructed image at 0·2 bpp with PSNR = 33·15 dB. The differences between Fig. 16*a* and the original image are small and Fig. 16*b* retains the details of Lena very well.

Without entropy coding the performance is still good, as the rate–distortion curve in Fig. 17 shows, where the difference is less than 0·5dB.

Compared with the block-based JPEG image coding standard, which uses the discrete cosine transform, image coding schemes using the wavelet transform have better objective performance, and have fewer block artefacts.

For lossless compression, an integer wavelet transform can be defined that produces integer coefficients if the image is represented by an integer array. This process avoids quantisation. Such a transform is reversible so that the original image can be completely recovered[11]. Compression rates will be lower than for lossy compression systems such as the one described earlier.

6 JPEG2000

The existing JPEG standard has been with us for some years now but the high quality and size of modern digital images have required a change. That change is embodied in JPEG2000 Part 1. The new standard serves the following applications: Internet, colour facsimile, printing, scanning, digital photography, medical imaging, mobile communications, remote sensing and e-commerce[10]. A block diagram of the JPEG2000 system, showing the features relevant to the discussion in this paper, is shown in Fig. 13.

The key characteristics required in the new standard that relate directly to wavelet decomposition are as follows:

- better image quality than provided by the existing JPEG standard at compression rates of less than 0·25 bpp. This should be achieved without performance degradation in other parts of the rate–distortion curve.
- lossless and lossy compression.

Before any wavelet decom position is performed, the image is partitioned into non-overlapping tiles. These tiles are of equal size except possibly for those adjacent to the image boundary. Although tiling reduces PSNR for a given compression rate, this disadvantage is balanced by reduced memory and computational requirements as well as allowing selective encoding and decoding.

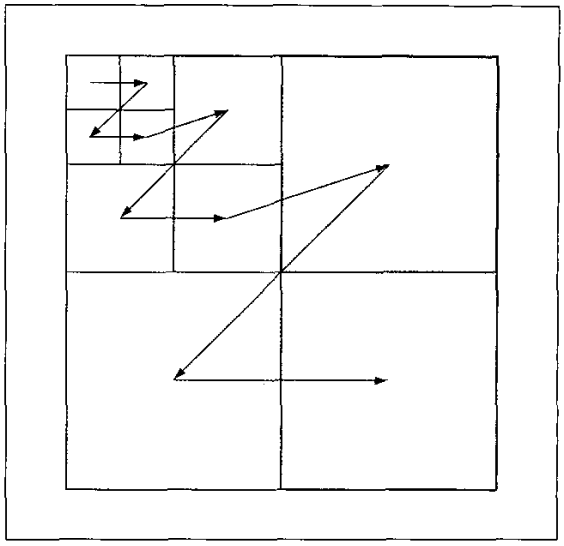


Fig. 15 Subband scanning order to ensure that a coefficient is processed before any of its descendants

Table 2: Symbol alphabet for the EZW algorithm

Symbol	Meaning
ZTR	Coefficient and all of its descendants are insignificant relative to the current threshold.
IZ	Coefficient is insignificant relative to the current threshold but one or more of its descendants are not.
POS	Coefficient is significant, i.e. greater than T_j, and positive relative to the current threshold.
NEG	Coefficient is significant and negative relative to the current threshold.

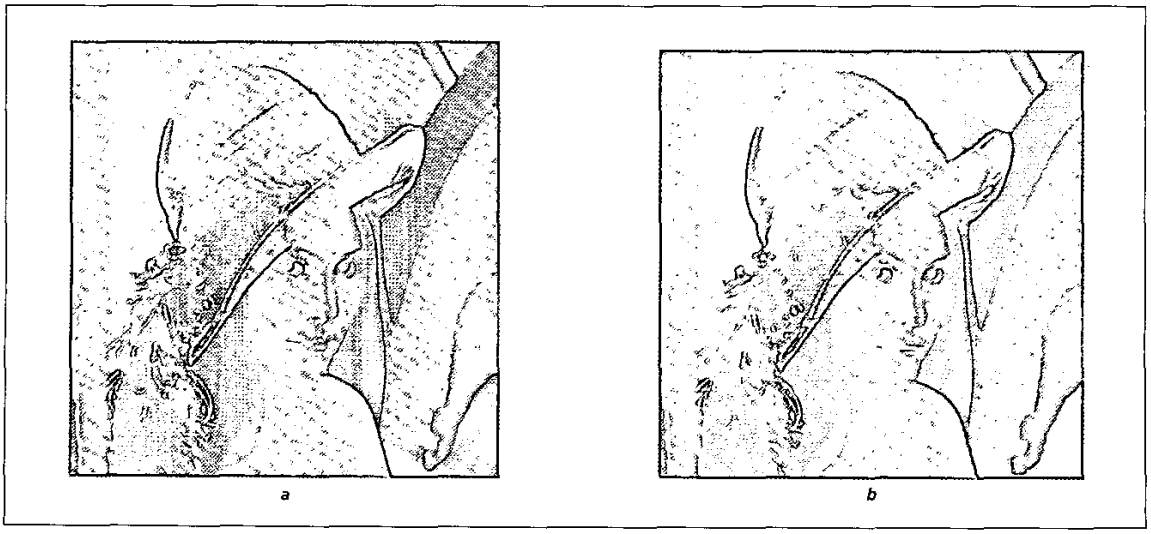


Fig. 16 Lena image coded with the SPIHT algorithm: (*a*) reconstructed image for coding at 1 bit per pixel, PSNR = 40·41dB; (*b*) reconstructed image for coding at 0·2 bits per pixel, PSNR = 33·15dB

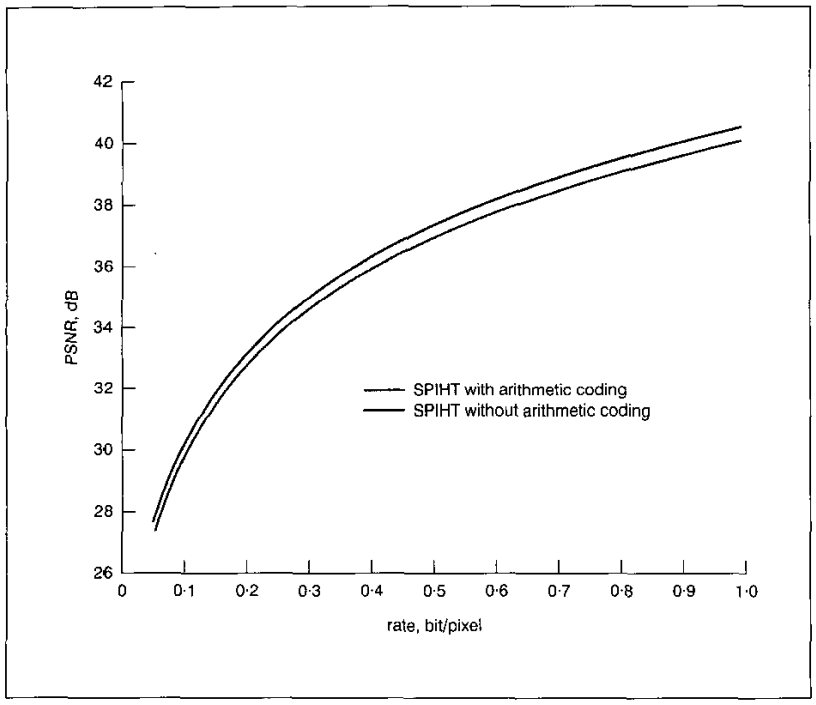


Fig. 17 Rate–distortion curve for Lena coded using SPIHT, with and without entropy coding

The DWT is applied to each tile using either the biorthogonal 9/7 filter set for lossy coding or an integer coefficient filter set for lossless coding[10,12]. The wavelet coefficients are then quantised using one step-size per subband. Entropy coding is then performed. Various tests have been performed with standard natural images and the results compared with existing coding systems. Lossy JPEG2000 has been shown to outperform all existing standards by around 2dB for compression ratios between 0·25 and 2bpp. In Fig. 18, using freely available software from the Web[13,14], we have compared lossy compression rates for JPEG2000 with those for the existing JPEG standard and have confirmed this observation for the Lena image. In addition, Fig. 19 shows qualitatively the difference in quality of the reconstructed image for a compression rate of 0·184bpp. Although all of these results are for the Lena image, other experiments have confirmed the superiority of JPEG2000[15].

Finally it is interesting to note that the content-based video coding standard MPEG-4 has adopted the wavelet transform for static texture coding[16].

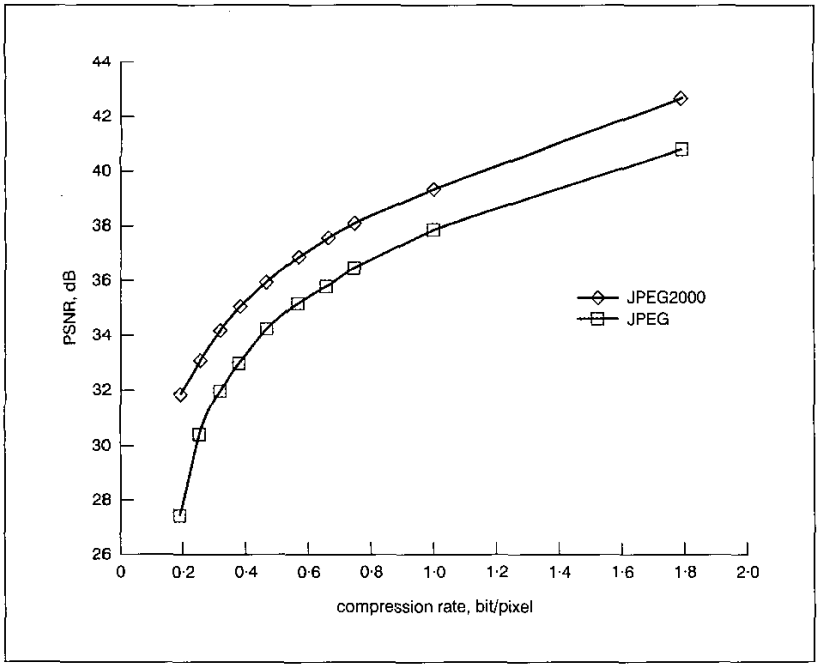


Fig. 18 Rate–distortion curve giving comparison between JPEG and JPEG2000 for the Lena image

7 Conclusions

In this short article we have aimed to show how filters are used as the basis for certain classes of wavelet decomposition methods used in modern image compression systems. We have briefly discussed these image compression methods and those elements of JPEG2000 that relate directly to the wavelet transform. At the time of writing some hardware and software products have been announced that meet the JPEG2000 Part 1 standard. Part 2 of the standard will use different quantisation strategies, allow user-defined wavelets, etc., whilst Part 3 will allow coding of motion for situations where, for example, both still and motion pictures are coded using the same hardware, e.g. in a digital camera.

References

1 BENTLEY, P. M., and MCDONNELL, J. T. E.: 'Wavelet transforms: an introduction', *Electron. Commun. Eng. J.*, August 1994, **6**, (4), pp.175–186
2 WEISS, L. G.: 'Wavelets and wideband correlation processing', *IEEE Signal Process. Mag.*, 1994, **11**, pp.13–32
3 GRANT, P. M.: 'Digital signal processing. Part 1: Digital filters and the DFT', *Electron. Commun. Eng. J.*, February 1993, **5**, (1), pp.13–21

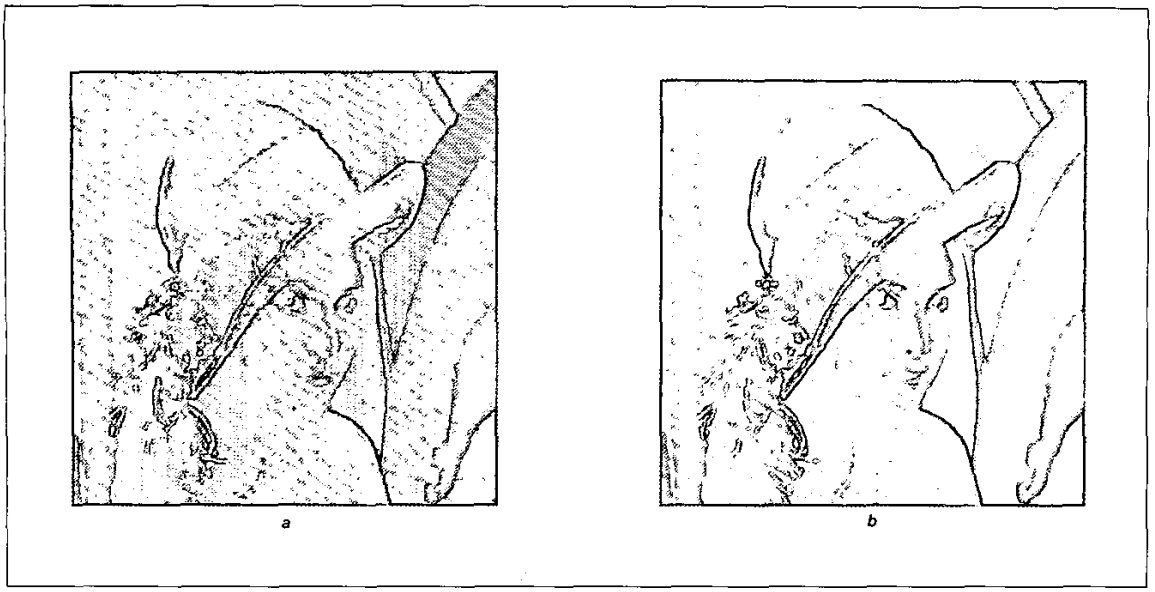


Fig. 19 Qualitative comparison between reconstructed Lena images using (*a*) JPEG and (*b*) JPEG2000. Compression rate = 0·184 bits per pixel.

4 GRANT, P. M.: 'Multirate signal processing', *Electron. Commun. Eng. J.*, February 1996, **8,** (1), pp.4–12

5 VETTERLI, M., and KOVACEVIC, J.: 'Wavelets and subband coding' (Prentice-Hall, Upper Saddle River, NJ, 1995)

6 LI, S., and LI, W.: 'Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding', *IEEE Trans. Circuits Syst. Video Technol.*, 2000, **10,** pp.725–743

7 CREUSERE, C. D., and MITRA, S. K.: 'Image coding using wavelets based on perfect reconstruction IIR filter banks', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6,** pp.447–458

8 SAHPIRO, J. M.: 'Embedded image coding using zerotrees of wavelet coefficients', *IEEE Trans. Signal Process.*, 1993, **41,** pp.3445–3462

9 SAID, W., and PEARLMAN, W. A.: 'A new, fast and efficient image codec based on set partitioning in hierarchical trees', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6,** pp.243–250

10 SKODRAS, A., CHRISTOPOULOS, C., and EBRAHIMI, T.: 'The JPEG2000 still image compression standard', *IEEE Signal Process. Mag.*, 2001, **18,** pp.36–58

11 CALDERBANK, A. R., DAUBECHIES, I., SWELDENS, W., and YEO, B. L.: 'Wavelet transforms that map integers to integers', *Appl. Comput. Harmon. Anal.*, 1998, **5,** pp.332–369

12 TAUBMAN, D.: 'High performance scalable image compression with EBCOT', *IEEE Trans. Image Process.*, 2000, **9,** pp.1158–1170

13 LANE, T. G., *et al.*: 'JPEG software implementation, ver. 6b'. Independent JPEG Group, 1998. See http://www.ijg.org

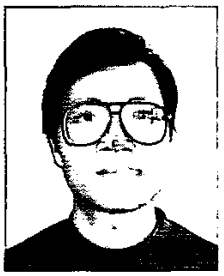
14 ADAMS, M. D.: 'JasPer JPEG2000 software, ver. 1.500.4'. University of British Columbia, Vancouver, 2001. See http://www.ece.ubc.ca/mdadams/jasper

15 XIONG, Z., RAMCHANDRAN, K., ORCHARD, M. T., and ZHANG, Y-Q.: 'A comparative study of DCT- and wavelet-based image coding', *IEEE Trans. Circuits Syst. Video Technol.*, 1999, **9,** pp.692–695

16 EBRAHIMI, T., and HORNE, C.: 'MPEG-4 natural video coding—an overview', *Signal Process., Image Commun.*, 2000, **15,** pp.365–385

Stuart Lawson received a 1st Class Honours BSc degree in Mathematics in 1971 from Woolwich Polytechnic and the DIC and PhD degree in Electrical Engineering in 1975 from Imperial College, University of London. From 1971 to 1978 he was with Plessey Telecommunications as a research engineer. He taught and researched at City University, London from 1978 to 1987. Since 1988 he has been at the University of Warwick, where he is now a Reader in the School of Engineering. His current research interests include digital filters and filter banks, image compression, modelling physical systems using DSP techniques and sonar signal processing. Dr. Lawson is a Fellow of IEE and a member of the Advisory Panel for the IEE Signal Processing Professional Network.

Address: School of Engineering, University of Warwick, Coventry, CV4 7AL, UK.
E-mail: ssl@eng.warwick.ac.uk



Jian Zhu received a BSc and an MSc degree in Electronic Engineering from Tsinghua University (China) in 1989 and 1991, respectively. After graduation, he worked in Tsinghua University on wireless and mobile communication systems, protocols and software implementation. He is now working at Warwick University towards a PhD on digital signal processing and image coding. He has worked for a short period at BT Labs on semantic video indexing and retrieval. His current research interests include multimedia services, source coding and transmission over Internet and mobile communication networks.

First received 5th November 2001 and in revised form 8th April 2002.