Wavelets and Filterbanks
Image and Video Processing

Dr. Anil Kokaram* anil.kokaram@tcd.ie

This section develops the idea of the Haar Transform into an introduction to wavelet theory. Wavelet applications are given for compression and de-noising. The goals here are

- To introduce the 1D and 2D wavelet
- To understand the purpose and limitations of wavelet analysis
- To observe the use of wavelets for compression
- To introduce the use of wavelet analysis for de-noising.
- To understand that the wavelet decomposition is not necessarily shift-invariant.

*Abridged version of material from Nick Kingsbury, Signal Processing Group, Cambridge University

1 Filter Banks and Wavelets

Digital filter banks have been actively studied since the 1960s, whereas Wavelet theory is a new subject area that was developed in the 1980s, principally by French and Belgian mathematicians, notably Y Meyer, I Daubechies, and S Mallat. The two topics are now firmly linked and of great importance for signal analysis and compression.

1.1 The 2-band Filter Bank

Recall the 1-D Haar transform from earlier lectures.

$$\begin{bmatrix} y(1) \\ y(2) \end{bmatrix} = \mathbf{T} \begin{bmatrix} x(1) \\ x(2) \end{bmatrix} \qquad \text{where} \quad \mathbf{T} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1}$$

We can write this in expanded form as:

$$y(1) = \tfrac{1}{\sqrt{2}}x(1) + \tfrac{1}{\sqrt{2}}x(2) y(2) \qquad = \tfrac{1}{\sqrt{2}}x(1) - \tfrac{1}{\sqrt{2}}x(2)$$

More generally if $\mathbf{x}$ is a longer sequence and the results are placed in two separate sequences $\mathbf{y}_0$ and $\mathbf{y}_1$, we define the process as:

$$\begin{aligned} y_0(n) &= \tfrac{1}{\sqrt{2}}x(n-1) + \tfrac{1}{\sqrt{2}}x(n) \\ y_1(n) &= \tfrac{1}{\sqrt{2}}x(n-1) - \tfrac{1}{\sqrt{2}}x(n) \end{aligned} \tag{2}$$

These can be expressed as 2 FIR filters with tap vectors $\mathbf{h}_0 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$ and $\mathbf{h}_1 = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$.

Hence as z-transforms, equations (2) become:

$$\begin{aligned} Y_0(z) = H_0(z)\ X(z) \quad \text{where} \quad H_0(z) = \tfrac{1}{\sqrt{2}}(z^{-1}+1) \\ Y_1(z) = H_1(z)\ X(z) \quad \text{where} \quad H_1(z) = \tfrac{1}{\sqrt{2}}(z^{-1}-1) \end{aligned} \tag{3}$$

Recall that for use in compression, every pair of input samples ($x_n,\ x_{n-1}$ say) is transformed into a corresponding output pair ($y_n,\ y_{n-1}$). All such pairs e.g. (x_n, x_{n-1}), (x_{n-2}, x_{n-3}) are treated separately. Thus the total number of output samples is the same as the total number of input samples. In practice therefore, this is equivalent to calculating $y_0(n)$ and $y_1(n)$ at alternate (say even) values of n so that the total number of samples in $\mathbf{y}_0$ and $\mathbf{y}_1$ is the same as in $\mathbf{x}$.

We may thus represent the Haar transform operation by a pair of filters followed by downsampling by 2, as shown in fig 4.1a. This is known as a 2-band analysis filter bank.

In the first handout on the use of the Haar Transform for compression, to reconstruct $\mathbf{x}$ from $\mathbf{y}$ we calculated $\mathbf{x} = \mathbf{T}^T\mathbf{y}$. For long sequences this may be written:

$$\begin{aligned} x(n-1) &= \tfrac{1}{\sqrt{2}}y_0(n) + \tfrac{1}{\sqrt{2}}y_1(n) \\ x(n) &= \tfrac{1}{\sqrt{2}}y_0(n) - \tfrac{1}{\sqrt{2}}y_1(n) \end{aligned} \qquad \text{for } n \text{ even.} \tag{4}$$

Since $y_0(n)$ and $y_1(n)$ are only calculated at even values of n, we may assume that they are zero at odd values of n. We may then combine equations (4) into a single expression for $x(n)$, valid for all n:

$$x(n) = \tfrac{1}{\sqrt{2}}[y_0(n+1) + y_0(n)] + \tfrac{1}{\sqrt{2}}[y_1(n+1) - y_1(n)] \tag{5}$$

or as z-transforms:

$$X(z) = G_0(z)\ Y_0(z) + G_1(z)\ Y_1(z) \tag{6}$$

where

$$G_0(z) = \tfrac{1}{\sqrt{2}}(z+1) \quad \text{and} \quad G_1(z) = \tfrac{1}{\sqrt{2}}(z-1) \tag{7}$$

In (6) the signals $Y_0(z)$ and $Y_1(z)$ are not really the same as $Y_0(z)$ and $Y_1(z)$ in (3) because those in (3) have not had alternate samples set to zero. Also, in (6) $X(z)$ is the reconstructed output whereas in (3) it is the input signal.

To avoid confusion we shall use $\hat{X}$, $\hat{Y}_0$ and $\hat{Y}_1$ for the signals in (6) so it becomes:

$$\hat{X}(z) = G_0(z)\ \hat{Y}_0(z) + G_1(z)\ \hat{Y}_1(z) \tag{8}$$

We may show this reconstruction operation as upsampling followed by 2 filters, as in fig 4.1b.

If $\hat{Y}_0$ and $\hat{Y}_1$ are not the same as Y_0 and Y_1, how do they relate to each other?

Now

$$\hat{y}_0(n) = y_0(n) \ \text{ for } n \text{ even}, \qquad \hat{y}_0(n) = 0 \ \text{ for } n \text{ odd}. \tag{9}$$

Therefore $\hat{Y}_0(z)$ is a polynomial in z, comprising *only* the terms in even powers of z from $Y_0(z)$. This may be written as:

$$\hat{Y}_0(z) = \sum_{\text{even } n} y_0(n)\ z^{-n} = \sum_{\text{all } n} \tfrac{1}{2}[y_0(n)\ z^{-n} + y_0(n)\ (-z)^{-n}] = \tfrac{1}{2}[Y_0(z) + Y_0(-z)] \tag{10}$$

Similarly

$$\hat{Y}_1(z) = \tfrac{1}{2}[Y_1(z) + Y_1(-z)] \tag{11}$$

This is our general model for downsampling by 2, followed by upsampling by 2 as defined in equation (9).

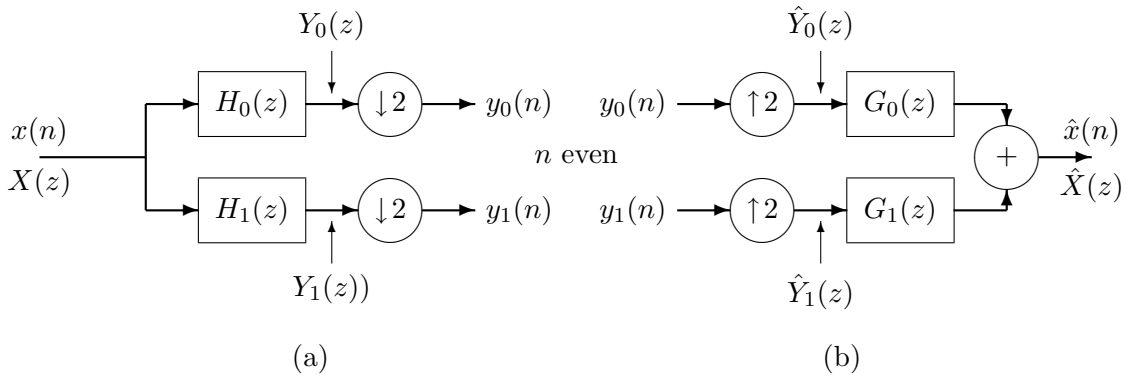


Fig 4.1: Two-band filter banks for analysis (a) and reconstruction (b).

1.2 Perfect Reconstruction (PR)

We are now able to generalise our analysis for arbitrary filters H_0, H_1, G_0 and G_1. Substituting (10) and (11) into (8) and then using (3), we get:

$$\begin{aligned}
\hat{X}(z) &= \tfrac{1}{2}G_0(z)[Y_0(z)+Y_0(-z)] + \tfrac{1}{2}G_1(z)[Y_1(z)+Y_1(-z)] \\
&= \tfrac{1}{2}G_0(z)H_0(z)X(z) + \tfrac{1}{2}G_0(z)H_0(-z)X(-z) \\
&\quad + \tfrac{1}{2}G_1(z)H_1(z)X(z) + \tfrac{1}{2}G_1(z)H_1(-z)X(-z) \\
&= \tfrac{1}{2}X(z)[G_0(z)H_0(z) + G_1(z)H_1(z)] \\
&\quad + \tfrac{1}{2}X(-z)[G_0(z)H_0(-z) + G_1(z)H_1(-z)] \qquad (12)
\end{aligned}$$

If we require $\hat{X}(z) \equiv X(z)$ — the Perfect Reconstruction (PR) condition — then:

$$G_0(z)H_0(z) + G_1(z)H_1(z) \equiv 2 \qquad (13)$$

and

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) \equiv 0 \qquad (14)$$

Identity (14) is known as the anti-aliasing condition because the term in $X(-z)$ in (12) is the unwanted aliasing term caused by down-sampling $\mathbf{y}_0$ and $\mathbf{y}_1$ by 2.

It is straightforward to show that the expressions for H_0, H_1, G_0 and G_1, given in (3) and (7) for the filters based on the Haar transform, satisfy (13) and (14). They are the simplest set of filters which do.

Before we look at more complicated PR filters, we examine how the filter structures of fig 4.1 may be extended to form a binary filter tree (and the discrete wavelet transform).

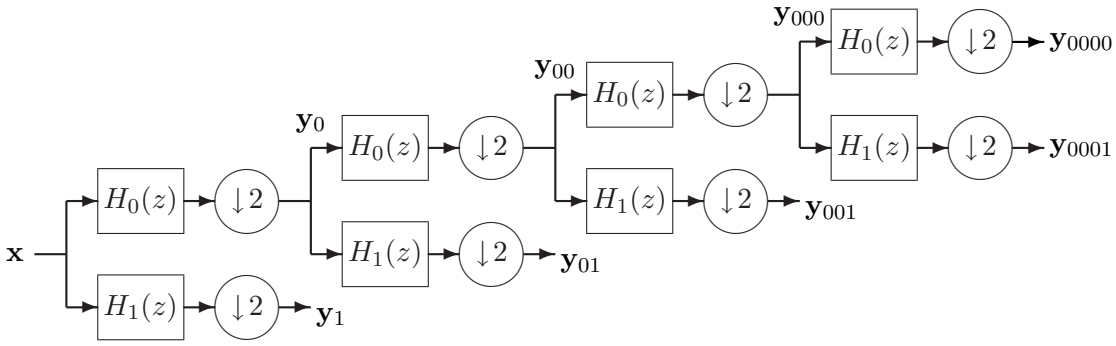


Fig 4.2: Extension of the 2-band filter bank into a binary filter tree.

1.3 The binary filter tree

Recall that for image compression, the purpose of the 2-band filter bank in the Haar transform is to compress most of the signal energy into the low-frequency band.

We may achieve greater compression if the low band is further split into two. This may be repeated a number of times to give the binary filter tree, shown with 4 levels in fig 4.2.

In 1-D, this is analogous to the way the 2-D Haar transform of section 2.1 was extended to the multi-level Haar transform in section 2.3.

For an N-sample input vector $\mathbf{x}$, the sizes and bandwidths of the signals of the 4-level filter tree are:

Signal	No. of samples	Approximate pass band
$\mathbf{x}$	N	$0 \rightarrow \frac{1}{2}\ f_s$
$\mathbf{y}_1$	$N/2$	$\frac{1}{4} \rightarrow \frac{1}{2}\ f_s$
$\mathbf{y}_{01}$	$N/4$	$\frac{1}{8} \rightarrow \frac{1}{4}\ f_s$
$\mathbf{y}_{001}$	$N/8$	$\frac{1}{16} \rightarrow \frac{1}{8}\ f_s$
$\mathbf{y}_{0001}$	$N/16$	$\frac{1}{32} \rightarrow \frac{1}{16}\ f_s$
$\mathbf{y}_{0000}$	$N/16$	$0 \rightarrow \frac{1}{32}\ f_s$

Because of the downsampling (decimation) by 2 at each level, the total number of output samples $= N$, regardless of the number of levels in the tree.

The H_0 filter is normally designed to be a lowpass filter with a passband from 0 to approximately

$\frac{1}{4}$ of the input sampling frequency for that stage; and H_1 is a highpass (bandpass) filter with a pass band approximately from $\frac{1}{4}$ to $\frac{1}{2}$ of the input sampling frequency.

When formed into a 4-level tree, the filter outputs have the approximate pass bands given in the above table. The final output $\mathbf{y}_{0000}$ is a lowpass signal, while the other outputs are all bandpass signals, each covering a band of approximately one octave.

An inverse tree, mirroring fig 4.2, may be constructed using filters G_0 and G_1 instead of H_0 and H_1, as shown for just one level in fig 4.1b. If the PR conditions of (13) and (14) are satisfied, then the output of each level will be identical to the input of the equivalent level in fig 4.2, and the final output will be a perfect reconstruction of the input signal.

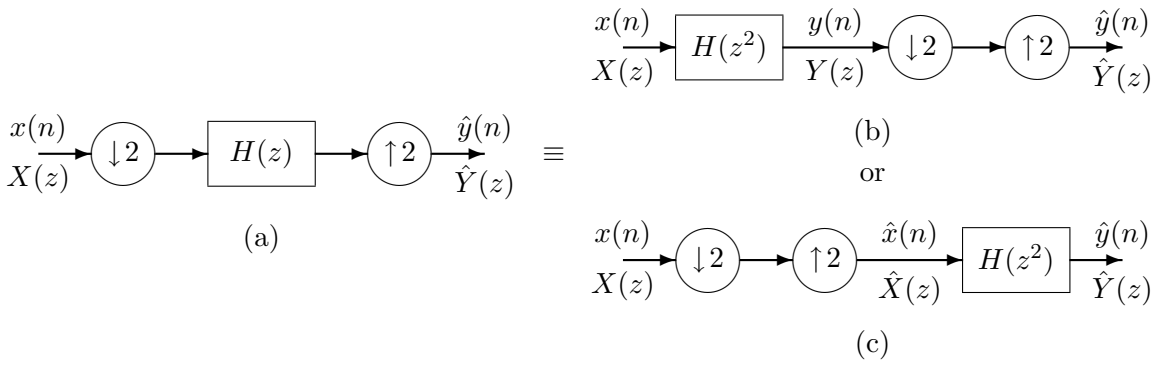


Fig 4.3: Multi-rate filtering — the result of shifting a filter ahead of a downsampling operation or after an upsampling operation.

Multi-rate filtering theorem

To calculate the impulse and frequency responses for a multistage network with down-sampling at each stage, as in fig 4.2, we must first derive an important theorem for multi-rate filters.

Theorem:

The downsample–filter–upsample operation of fig 4.3a is equivalent to either the filter–downsample–upsample operation of fig 4.3b or the downsample–upsample–filter operation of fig 4.3c, if the filter is changed from $H(z)$ to $H(z^2)$.

Proof:

From fig 4.3a:

$$\begin{aligned} \hat{y}(n) &= \sum_i x(n-2i)\ h(i) \quad \text{for } n \text{ even} \\ &= 0 \qquad\qquad\qquad\quad \text{for } n \text{ odd} \end{aligned} \tag{15}$$

Take z-transforms:

$$\hat{Y}(z) = \sum_n \hat{y}(n)\ z^{-n} = \sum_{\text{even } n} \sum_i x(n-2i)\ h(i)\ z^{-n} \tag{16}$$

Reverse the order of summation and let $m = n - 2i$:

$$\begin{aligned} \therefore\ \hat{Y}(z) &= \sum_i h(i) \sum_{\text{even } m} x(m)\ z^{-m}\ z^{-2i} \\ &= \sum_i h(i)\ z^{-2i} \sum_{\text{even } m} x(m)\ z^{-m} \\ &= H(z^2)\ \tfrac{1}{2}[X(z) + X(-z)] \\ &= \tfrac{1}{2}[H(z^2)\ X(z) + H((-z)^2)\ X(-z)] \\ &= \tfrac{1}{2}[Y(z) + Y(-z)] \qquad \text{where} \quad Y(z) = H(z^2)\ X(z) \end{aligned} \tag{17}$$

This describes the operations of fig 4.3b. Hence the first result is proved.

The result from line 3 above that

$$\hat{Y}(z) = \tfrac{1}{2}[X(z) + X(-z)]\ H(z^2) = \hat{X}(z)\ H(z^2) \tag{18}$$

shows that the filter $H(z^2)$ may be placed after the down/up-sampler as in fig 4.3c, which proves the second result.

General results for M:1 subsampling

It can be shown that:

- $H(z)$ becomes $H(z^M)$ if shifted ahead of an $M:1$ downsampler or following an $M:1$ upsampler.
- $M:1$ down/up-sampling of a signal $X(z)$ produces:

$$\hat{X}(z) = \frac{1}{M} \sum_{m=0}^{M-1} X(z\ e^{j2\pi m/M}) \tag{19}$$

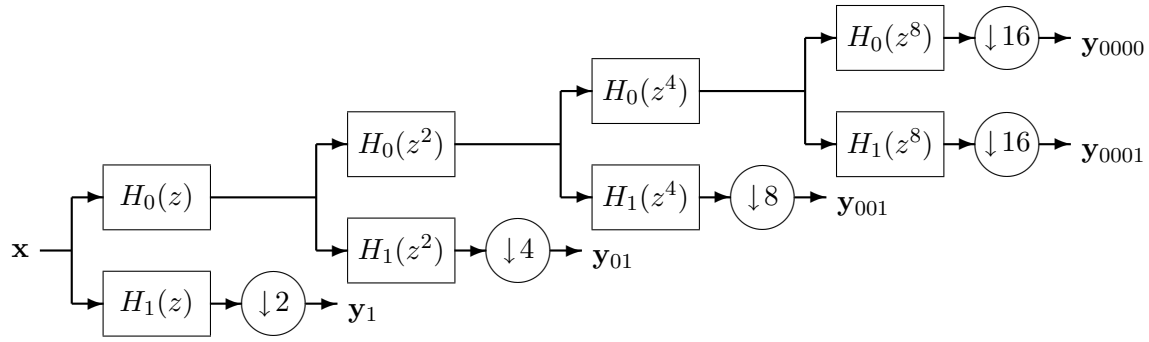


Fig 4.4: Binary filter tree, transformed so that all downsampling operations occur at the outputs.

Transformation of the filter tree

Using the result of equation (17), fig 4.2 can be redrawn as in fig 4.4 with all downsamplers moved to the outputs. (Note fig 4.4 requires much more computation than fig 4.2.) We can now calculate the transfer function to each output (before the downsamplers) as:

$$\begin{aligned} H_{01}(z) &= H_0(z)\ H_1(z^2) \\ H_{001}(z) &= H_0(z)\ H_0(z^2)\ H_1(z^4) \\ H_{0001}(z) &= H_0(z)\ H_0(z^2)\ H_0(z^4)\ H_1(z^8) \\ H_{0000}(z) &= H_0(z)\ H_0(z^2)\ H_0(z^4)\ H_0(z^8) \end{aligned} \tag{20}$$

In general the transfer functions to the two outputs at level k of the tree are given by:

$$\begin{aligned} H_{k,1} &= \prod_{i=0}^{k-2} H_0(z^{2^i})\ \ H_1(z^{2^{k-1}}) \\ H_{k,0} &= \prod_{i=0}^{k-1} H_0(z^{2^i}) \end{aligned} \tag{21}$$

For the Haar filters of equation (3), the transfer functions to the outputs of the 4-level tree become:

$$\begin{aligned} H_{01}(z) &= \tfrac{1}{2}\ [\ (z^{-3}+z^{-2}) - (z^{-1}+1)\ ] \\ H_{001}(z) &= \tfrac{1}{2\sqrt{2}}\ [\ (z^{-7}+z^{-6}+z^{-5}+z^{-4}) - (z^{-3}+z^{-2}+z^{-1}+1)\ ] \\ H_{0001}(z) &= \tfrac{1}{4}\ [\ (z^{-15}+z^{-14}+z^{-13}+z^{-12}+z^{-11}+z^{-10}+z^{-9}+z^{-8}) \\ &\quad - (z^{-7}+z^{-6}+z^{-5}+z^{-4}+z^{-3}+z^{-2}+z^{-1}+1)\ ] \\ H_{0000}(z) &= \tfrac{1}{4}\ [\ z^{-15}+z^{-14}+z^{-13}+z^{-12}+z^{-11}+z^{-10}+z^{-9}+z^{-8} \\ &\quad + z^{-7}+z^{-6}+z^{-5}+z^{-4}+z^{-3}+z^{-2}+z^{-1}+1\ ] \end{aligned} \tag{22}$$

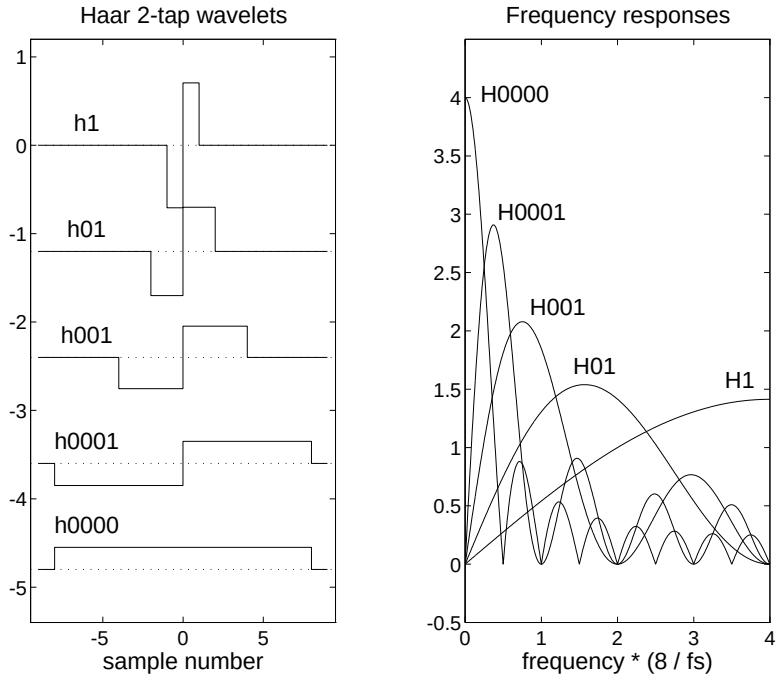


Fig 4.5: Impulse responses and frequency responses of the 4-level tree of Haar filters.

1.4 Wavelets

The process of creating the outputs $\mathbf{y}_1$ to $\mathbf{y}_{0000}$ from $\mathbf{x}$ is known as the discrete wavelet transform (DWT); and the reconstruction process is the inverse DWT.

The word *wavelet* refers to the impulse response of the cascade of filters which leads to a given bandpass output. The frequency response of the wavelet at level k is obtained by substituting $z = e^{j\omega T_s}$ in the z-transfer function $H_{k,1}$ from equation (21). T_s is the sampling period at the input to the filter tree.

Since the frequency responses of the bandpass bands are scaled down by 2:1 at each level, their impulse responses become longer by the same factor at each level, BUT their shapes remain very similar. The basic impulse response wave shape is almost independent of scale and is known as the *mother wavelet*.

The impulse response to a lowpass output $H_{k,0}$ is called the *scaling function* at level k.

Fig 4.5 shows these effects using the impulse responses and frequency responses for the five outputs of the 4-level tree of Haar filters, based on the z-transforms given in equations (22). Notice the abrupt transitions in the middle and at the ends of the Haar wavelets. These result in noticeable *blocking* artefacts in decompressed images (as in fig 2.13b).

1.5 Good filters / wavelets for compression

Our main aim now is to search for better filters / wavelets which result in compression performance that rivals or beats the DCT.

We assume that perfect reconstruction is a prime requirement, so that the only image degradations are caused by coefficient quantisation, and may be made as small as we wish by increasing bit rate.

Wavelet filter design is both an art and a science. There are numerous filters which could satisfy the PR condition yet they may not all be suitable for image analysis or compression. To give you a taste of how a wavelet filter can be designed, we start our search with the two PR identities from section 4.2, equations (13) and (14), which we repeat here:

$$G_0(z)H_0(z) + G_1(z)H_1(z) \equiv 2 \tag{23}$$

and

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) \equiv 0 \tag{24}$$

The usual way of satisfying the anti-aliasing condition (24), while permitting H_0 and G_0 to have lowpass responses (passband where $Re[z] > 0$) and H_1 and G_1 to have highpass responses (passband where $Re[z] < 0$), is with the following relations:

$$H_1(z) = z^{-k}G_0(-z) \qquad \text{and} \qquad G_1(z) = z^k H_0(-z) \tag{25}$$

where k must be odd so that:

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = G_0(z)H_0(-z) + z^k H_0(-z)\ (-z)^{-k}G_0(z) = 0$$

Now define the lowpass product filter:

$$P(z) = H_0(z)G_0(z) \tag{26}$$

and substitute relations (25) into identity (23) to get:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = G_0(z)H_0(z) + H_0(-z)G_0(-z) = P(z) + P(-z) = 2 \tag{27}$$

This requires all $P(z)$ terms in even powers of z to be zero, except the z^0 term which should be 1. The $P(z)$ terms in odd powers of z may take any desired values since they cancel out in (27).

A further constraint on $P(z)$ is that it should be zero phase, in order to **minimise the visibility of any distortions** due to the high-band being quantised to zero. Hence $P(z)$ should be of the form:

$$P(z) = \ldots + p_5 z^5 + p_3 z^3 + p_1 z + 1 + p_1 z^{-1} + p_3 z^{-3} + p_5 z^{-5} + \ldots \tag{28}$$

The design of a set of PR filters H_0, H_1 and G_0, G_1 can now be summarised as:

1. Choose a set of coefficients $p_1, p_3, p_5 \ldots$ to give a zero-phase lowpass product filter $P(z)$ with desirable characteristics. (This is non-trivial and is discussed below.)
2. Factorize $P(z)$ into $H_0(z)$ and $G_0(z)$, preferably so that the two filters have similar lowpass frequency responses.
3. Calculate $H_1(z)$ and $G_1(z)$ from equations (25).

Wavelet filter design is clearly non-trivial for filters of any use. A Belgian mathematician, Ingrid Daubechies, did much pioneering work on wavelets in the 1980s. She discovered that to achieve smooth wavelets after many levels of the binary tree, the lowpass filters $H_0(z)$ and $G_0(z)$ must both have a number of zeros at half the sampling frequency (at $z = -1$). These will also be zeros of $P(z)$, and so $P_t(Z)$ will have zeros at $Z = -1$.

The simplest case is a single zero at $Z = -1$, so that $P_t(Z) = 1 + Z$.

$$\therefore \; P(z) = \tfrac{1}{2}(z + 2 + z^{-1}) = \tfrac{1}{2}(z+1)(1+z^{-1}) = G_0(z)\; H_0(z)$$

which gives the familiar Haar filters.

We will not push on with the derivation of the other popular wavelet filters which have been discovered. Here we simply state some of the more well known wavelet filters. The design of wavelets for different purposes remains an important research topic.

1.5.1 LeGall Filters

The LeGall 3,5-tap filter set. Given that name because it was first published in the context of 2-band filter banks by Didier LeGall in 1988.

$$\begin{aligned} H_0(z) &= \tfrac{1}{2}(z + 2 + z^{-1}) \\ G_0(z) &= \tfrac{1}{8}(z + 2 + z^{-1})(-z + 4 - z^{-1}) \;=\; \tfrac{1}{8}(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}) \end{aligned} \tag{29}$$

$$\begin{aligned} G_1(z) &= z\; H_0(-z) \;=\; \tfrac{1}{2}z(-z + 2 - z^{-1}) \\ H_1(z) &= z^{-1}\; G_0(-z) \;=\; \tfrac{1}{8}z^{-1}(-z^2 - 2z + 6 - 2z^{-1} - z^{-2}) \end{aligned} \tag{30}$$

Note that H_1 and G_1 are shifted by one sample, z, z^{-1} respectively. This implies that in an implementation of these filters, we can use the un-shifted versions of these filters but select the *ODD* samples when downsampling by a factor of 2 at each level. Similarly, the implementation of H_0, G_0 requires the selection of the *EVEN* samples. In reconstruction the same rlues apply. Reconstruction with G_1 requires the upsampling step to place the non-zero coefficients in the *ODD* locations and reconstruction with G_0 requires the upsampling step to place the non-zero coefficients in the *EVEN* locations.

The wavelets of the LeGall 3,5-tap filters, H_0 and H_1 above, and their frequency responses are shown in fig 4.6. The scaling function (bottom left) converges to a pure triangular pulse and the wavelets are the superposition of two triangular pulses.

The triangular scaling function produces linear interpolation between consecutive lowband coefficients and also causes the wavelets to be linear interpolations of the coefficients of the H_1 filter, $-1, -2, 6, -2, -1$ (scaled appropriately).

These wavelets have quite desirable properties for image compression (note the absence of waveform discontinuities and the much lower sidelobes of the frequency responses), and they represent probably the simplest useful wavelet design. Unfortunately there is one drawback — the inverse wavelets are not very good. These are formed from the LeGall 5,3-tap filter pair, G_0 and G_1 above, whose wavelets and frequency responses are shown in fig 4.7.

The main problem is that the wavelets do not converge after many levels to a smooth function and hence the frequency responses have large unwanted sidelobes. The jaggedness of the scaling function and wavelets causes highly visible coding artefacts if these filters are used for reconstruction of a compressed image.

However the allocation of the factors of $P_t(Z)$ to H_0 and G_0 is a free design choice, so we may swap the factors (and hence swap G and H[1])

However the assignment of factors in the PR condition is a free design choice so G and H can be swapped in order that the smoother 3,5-tap filters become G_0, G_1 and are used for reconstruction. We shall show later that this leads to a good low-complexity solution for image compression and that the jaggedness of the analysis filters is not critical.

Unbalance between analysis and reconstruction filters / wavelets is nevertheless often regarded as being undesirable, particularly as it prevents the filtering process from being represented as an orthonormal transformation of the input signal (since an orthonormally transformed signal may be reconstructed simply by transposing the transform matrix). An unbalanced PR filter system is often termed a *bi-orthogonal* transformation.

[1] And then recalculate H_1, G_1

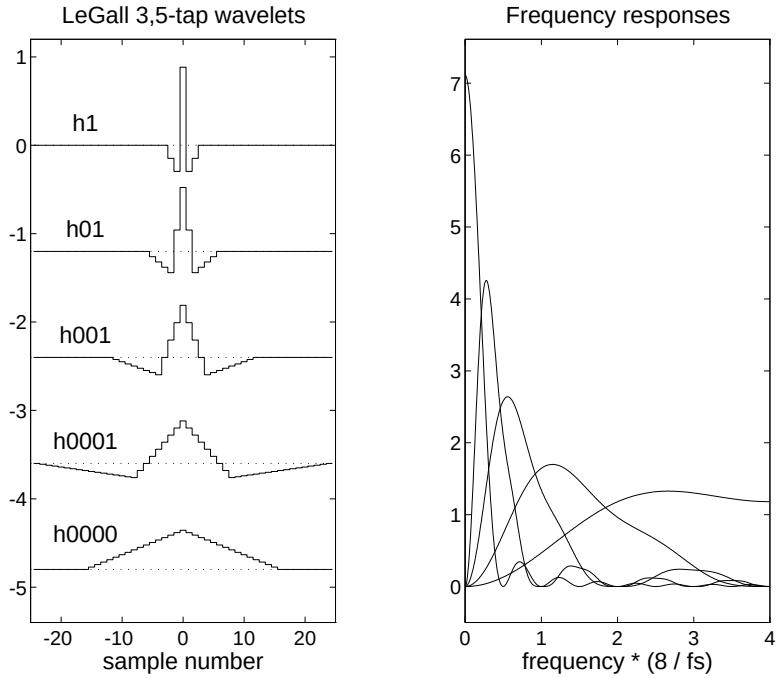


Fig 4.6: Impulse responses and frequency responses of the 4-level tree of LeGall 3,5-tap filters.

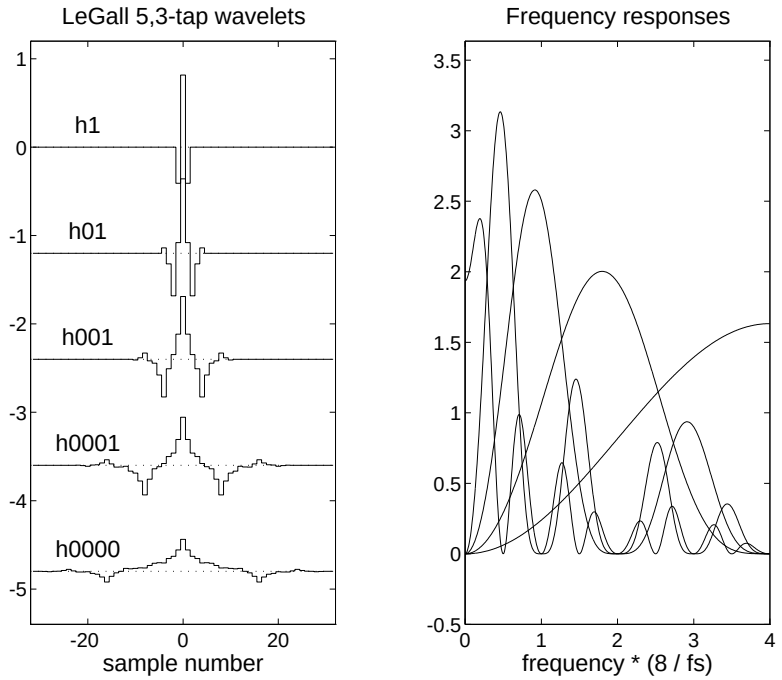


Fig 4.7: Impulse responses and frequency responses of the 4-level tree of LeGall 5,3-tap filters.

1.5.2 Daubechies wavelets

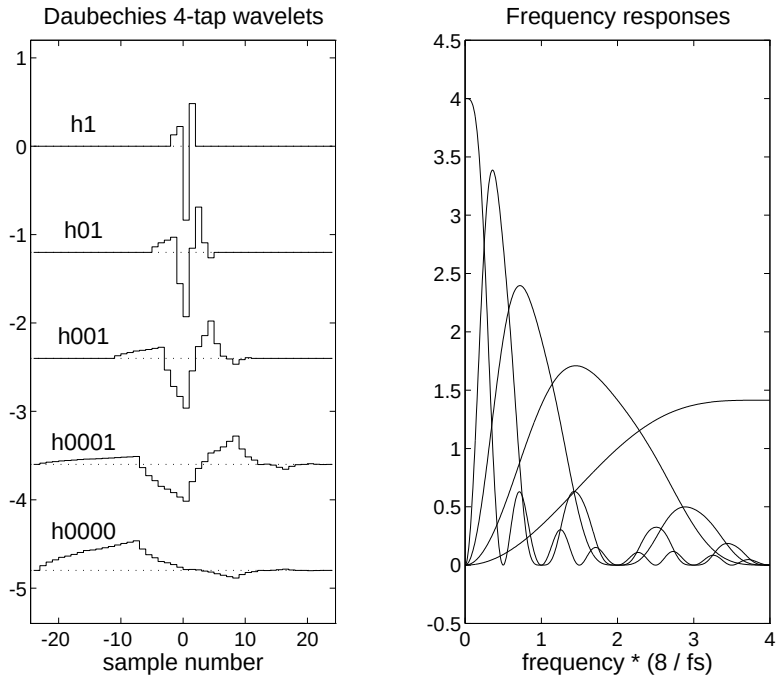


Fig 4.8: Impulse responses and frequency responses of the 4-level tree of Daubechies 4-tap filters.

These filters have balanced H and G frequency responses but non-linear phase responses. The 4 tap Daubechies filters can be derived as

$$H_0(z) = \frac{1}{2\sqrt{1+\alpha^2}}(1+z^{-1})^2\ (1+\alpha z^{-1}) = 0.4830 + 0.8365z^{-1} + 0.2241z^{-2} - 0.1294z^{-3}$$

and

$$H_1(z) = z^{-3}G_0(-z) = z^{-3}H_0(-z^{-1}) = 0.1294 + 0.2241z^{-1} - 0.8365z^{-2} + 0.4830z^{-3}$$

The wavelets and frequency responses for these filters are shown in fig 4.8. It is clear that the wavelets and scaling function are no longer linear phase and are less smooth than those for the LeGall 3,5-tap filters. The frequency responses also show worse sidelobes. The G_0, G_1 filters give the time reverse of these wavelets and identical frequency responses.

Higher order Daubechies filters achieve smooth wavelets but they still suffer from non-linear phase. This tends to result in more visible coding artefacts than linear phase filters, which distribute any artefacts equally on either side of sharp edges in the image.

Linear phase filters also allow an elegant technique, known as symmetric extension, to be used at the outer edges of images, where wavelet filters would otherwise require the size of the transformed image to be increased to allow for convolution with the filters. Symmetric extension assumes that

the image is reflected by mirrors at each edge, so that an infinitely tessellated plane of reflected images is generated. Reflections avoid unwanted edge discontinuities. If the filters are linear phase, then the DWT coefficients also form reflections and no increase in size of the transformed image is necessary to accommodate convolution effects.

1.5.3 Filters with linear phase and nearly balanced frequency responses:

A set of filters (5,7-tap) satisfying these requirements are as follows. The wavelets and frequency responses are shown in fig 4.9.

$$H_0(z) = \frac{1}{14}[-z^{-2} + 5z^{-1} + 12 + 5z - z^2]$$
$$H_1(z) = \frac{1}{69.4896}[\frac{1}{8}z^{-3} - \frac{5}{8}z^{-2} - \frac{73}{24}z^{-1} + \frac{85}{12} - \frac{73}{12}z - \frac{5}{8}z^2 + \frac{1}{8}z^3]$$

The near balance of the responses may be seen from fig 4.10 which shows the alternative 7,5-tap versions (i.e. with H and G swapped). It is quite difficult to spot the minor differences between these figures.

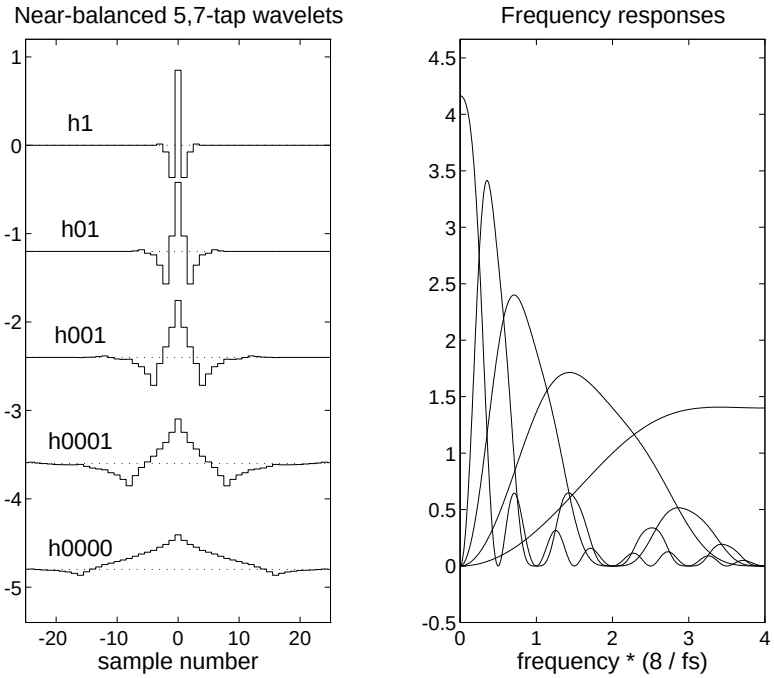


Fig 4.9: Impulse responses and frequency responses of the 4-level tree of near-balanced 5,7-tap filters.

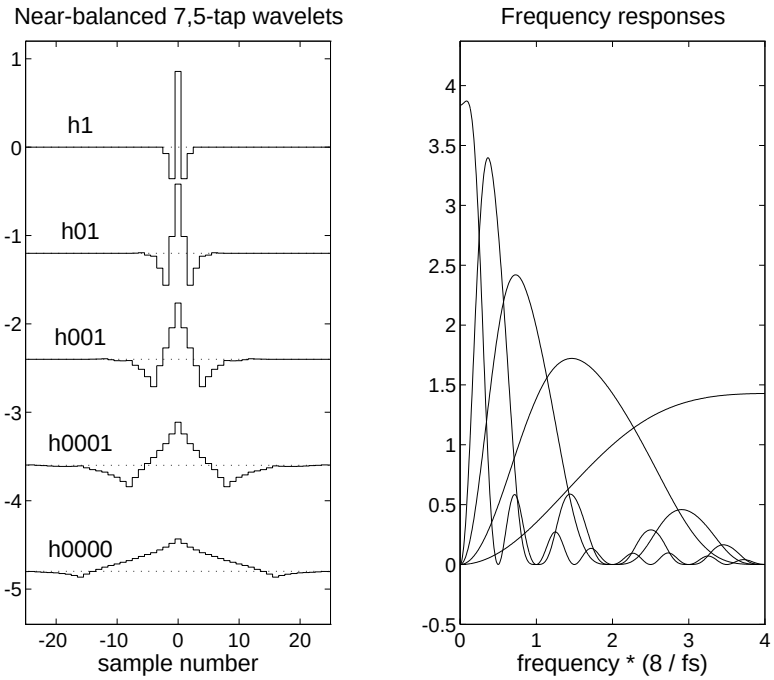


Fig 4.10: Impulse responses and frequency responses of the 4-level tree of near-balanced 7,5-tap filters.

1.5.4 Smoother Wavelets

Fig 4.11 shows the wavelets and frequency responses of a 13,19-tap filter set. Note the smoother wavelets and scaling function and the much lower sidelobes in the frequency responses from these higher order filters.

Fig 4.12 demonstrates that the near balanced properties are preserved in the high order filters.

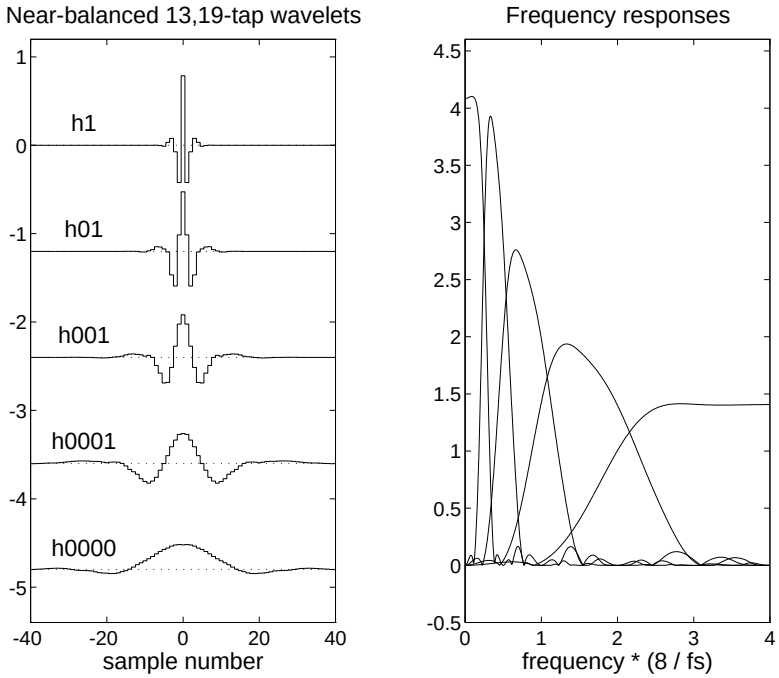


Fig 4.11: Impulse responses and frequency responses of the 4-level tree of near-balanced 13,19-tap filters.

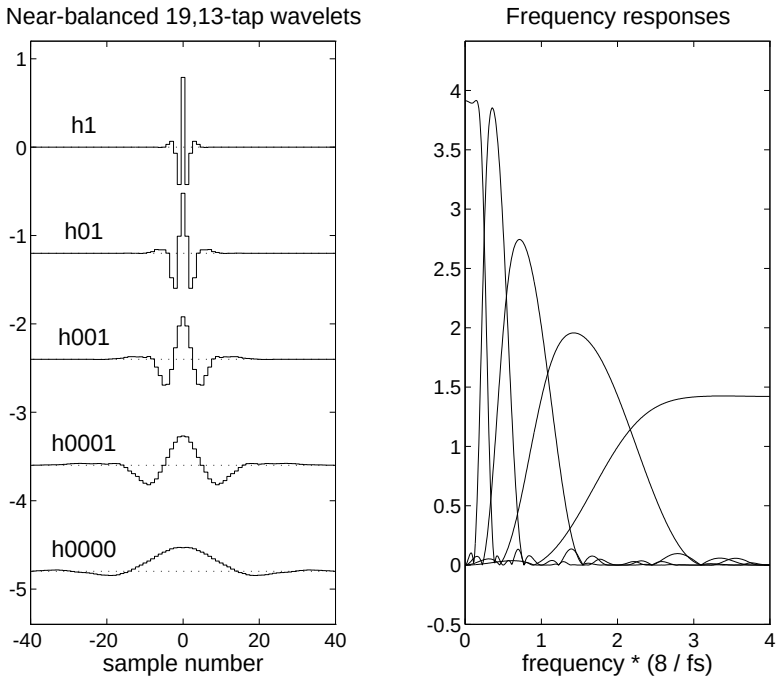


Fig 4.12: Impulse responses and frequency responses of the 4-level tree of near-balanced 19,13-tap filters.

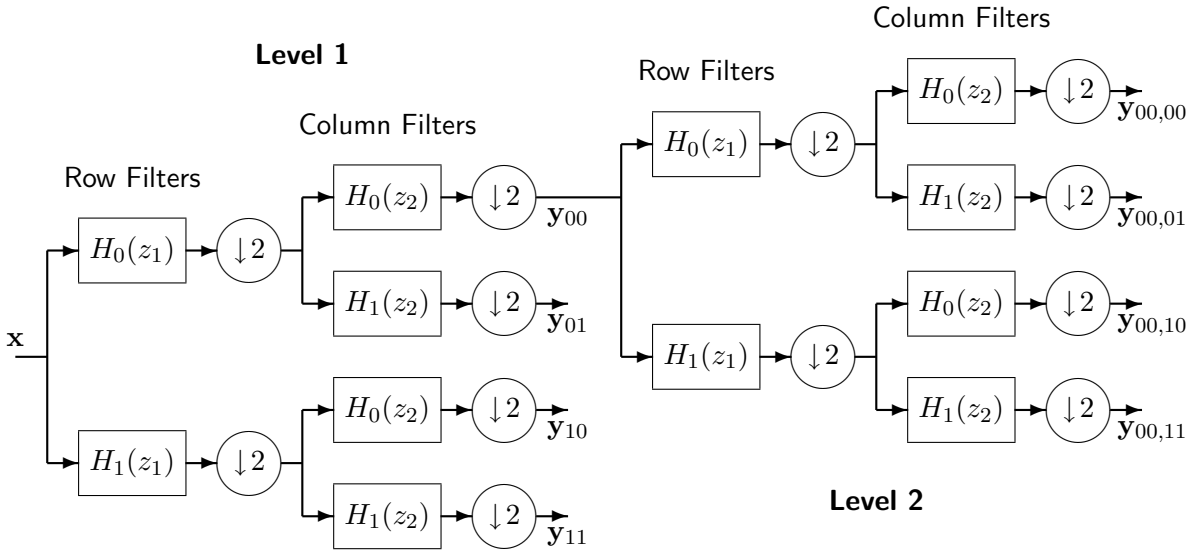


Fig 4.13: Two levels of a 2-D filter tree, formed from 1-D lowpass (H_0) and highpass (H_1) filters.

1.6 The 2-D DWT

We have already seen in section 2.1 how the 1-D Haar transform (or wavelet) could be extended to 2-D by filtering the rows and columns of an image separably.

All 1-D 2-band wavelet filter banks can be extended in a similar way. Fig 4.13 shows two levels of a 2-D filter tree. The input image at each level is split into 4 bands (Lo-Lo = $\mathbf{y}_{00}$, Lo-Hi = $\mathbf{y}_{01}$, Hi-Lo = $\mathbf{y}_{10}$, and Hi-Hi = $\mathbf{y}_{11}$) using the lowpass and highpass wavelet filters on the rows and columns in turn. The Lo-Lo band subimage $\mathbf{y}_{00}$ is then used as the input image to the next level. Typically 4 levels are used, as for the Haar transform.

Filtering of the rows of an image by $H_a(z_1)$ and of the columns by $H_b(z_2)$, where $a, b = 0$ or 1, is equivalent to filtering by the 2-D filter:

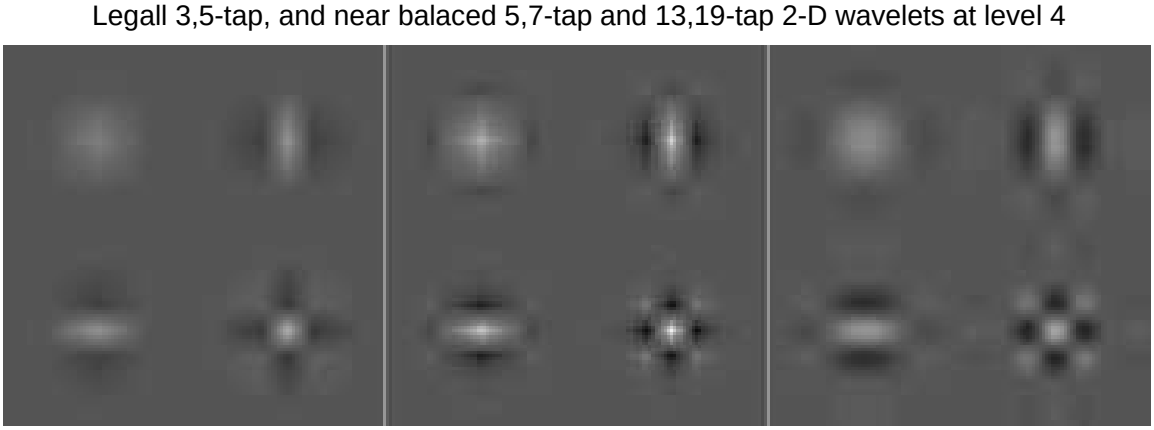
$$H_{ab}(z_1, z_2) = H_a(z_1)\ H_b(z_2) \tag{31}$$

In the spatial domain, this is equivalent to convolving the image matrix with the 2-D impulse response matrix

$$\mathbf{h}_{ab} = \mathbf{h}_a\ \mathbf{h}_b^T \tag{32}$$

where $\mathbf{h}_a$ and $\mathbf{h}_b$ are column vectors of the 1-D filter impulse responses. However note that performing the filtering separably (i.e. as separate 1-D filterings of the rows and columns) is much more computationally efficient.

To obtain the impulse responses of the four 2-D filters at each level of the 2-D DWT we form $\mathbf{h}_{ab}$ from $\mathbf{h}_0$ and $\mathbf{h}_1$ using equation (32) with $ab = 00, 01, 10$ and 11.



(a) (b) (c)

Fig 4.14: 2-D impulse responses of the level-4 wavelets and scaling functions derived from the LeGall 3,5-tap filters (a), and the near-balanced 5,7-tap (b) and 13,19-tap (c) filters.

Fig 4.14 shows the impulse responses at level 4 as images for three 2-D wavelet filter sets, formed from the following 1-D wavelet filter sets:

1. The LeGall 3,5-tap filters: H_0 and H_1 from equations (29) and (30).
2. The near-balanced 5,7-tap filters.
3. The near-balanced 13,19-tap filters.

Note the sharp points in fig 4.14b, produced by the sharp peaks in the 1-D wavelets of fig 4.9. These result in noticeable artefacts in reconstructed images when these wavelets are used. The smoother wavelets of fig 4.14c are much better in this respect.

The 2-D frequency responses of the level 1 filters, derived from the LeGall 3,5-tap filters, are shown in figs 4.15 (in mesh form) and 4.16 (in contour form). These are obtained by substituting $z_1 = e^{j\omega_1}$ and $z_2 = e^{j\omega_2}$ into equation (31). This equation demonstrates that the 2-D frequency response is just the product of the responses of the relevant 1-D filters.

Figs 4.17 and 4.18 are the equivalent plots for the 2-D filters derived from the near-balanced 13,19-tap filters. We see the much sharper cut-offs and better defined pass and stop bands of these filters. The high-band filters no longer exhibit gain peaks, which are rather undesirable features of the LeGall 5-tap filters.

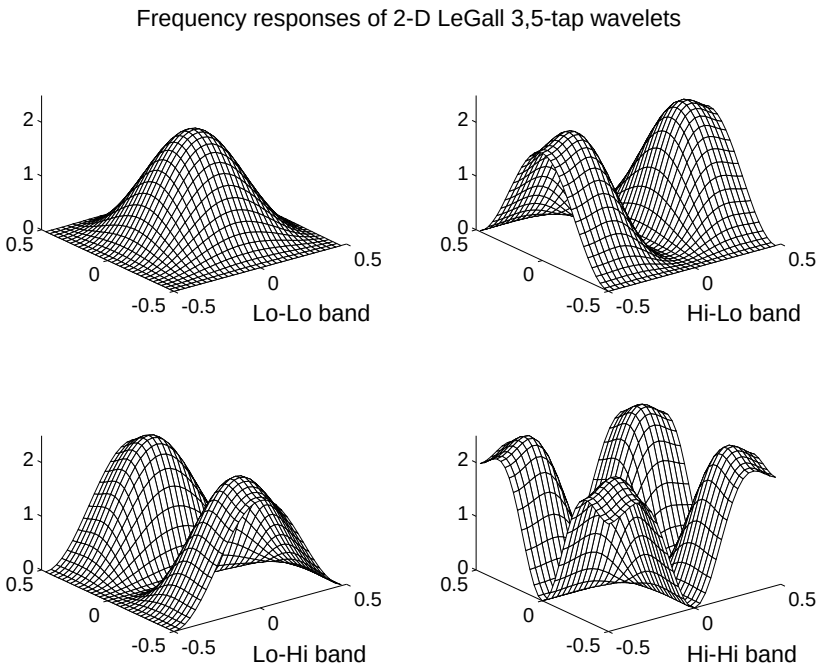


Fig 4.15: Mesh frequency response plots of the 2-D level 1 filters, derived from the LeGall 3,5-tap filters.

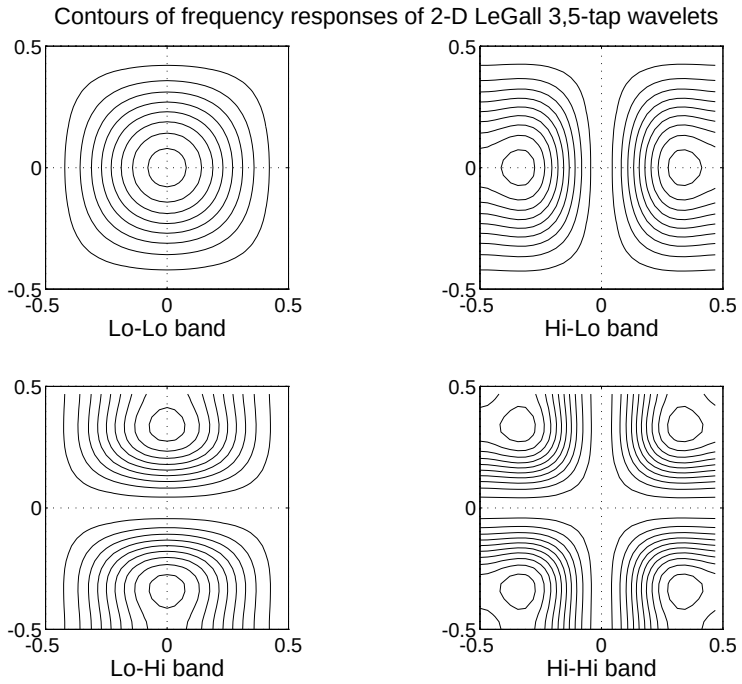


Fig 4.16: Contour plots of the frequency responses of fig 4.15.

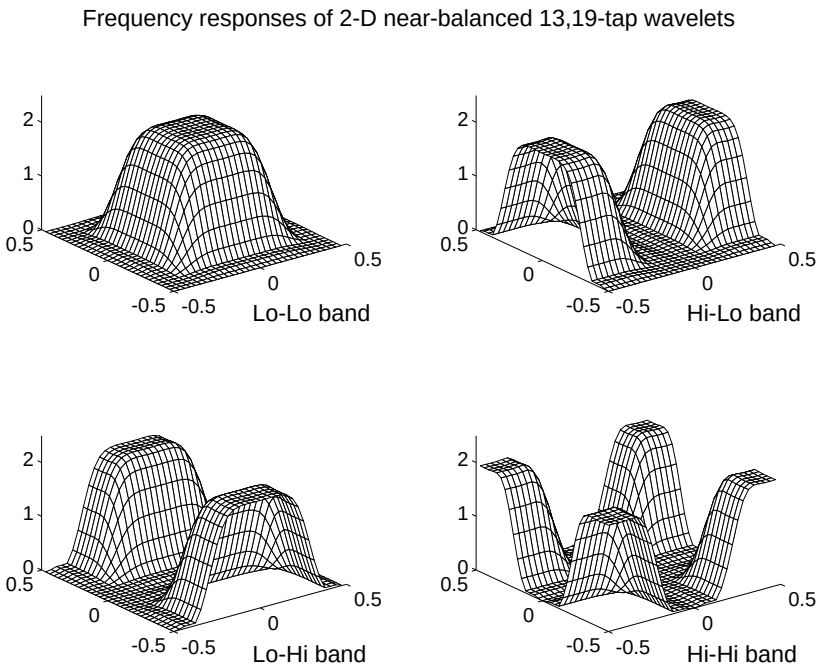


Fig 4.17: Mesh frequency response plots of the 2-D level 1 filters, derived from the near-balanced 13,19-tap filters.

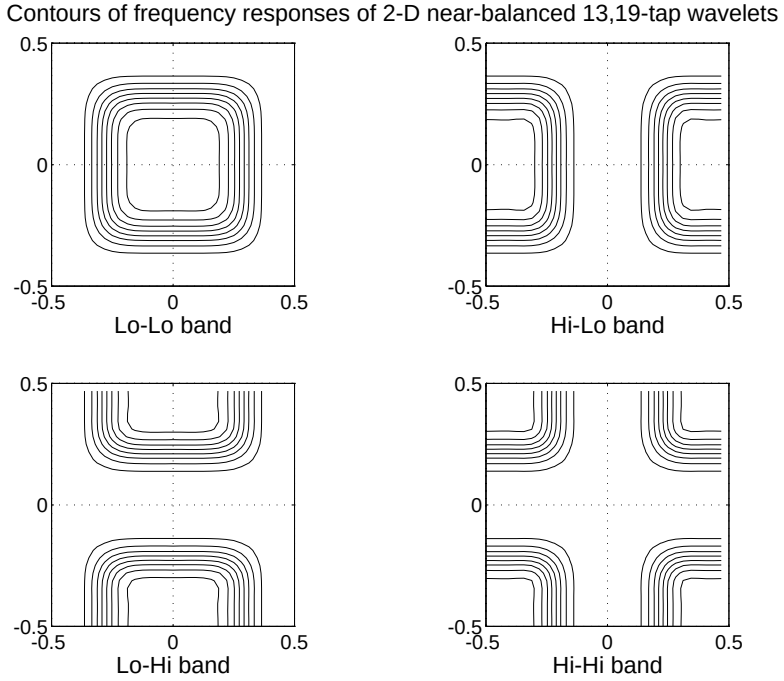


Fig 4.18: Contour plots of the frequency responses of fig 4.17.

1.7 Wavelets used in MPEG4

The MPEG4 standard uses Wavelet compression in that part of the standard called *Texture Coding*. The default analysis filters are a 9,3-tap Biorthogonal filter set (i.e. unbalanced filter set). The filter coefficients for the analysis filters are as follows

$$h_0 = \sqrt{2}[3,\ -6,\ -16,\ 38,\ 90,\ 38,\ -16,\ -6,3]/128$$
$$h_1 = \sqrt{2}[-32,\ 64,\ -32]/128$$

The synthesis filter set is

$$g_0 = \sqrt{2}[3,\ 6,\ -16,\ -38,\ 90,\ -38,\ -16,\ 6,3]/128$$
$$g_1 = \sqrt{2}[32,\ 64,\ 32]/128$$

The filters have been chosen not only for compression performance, but also because they allow low complexity implementation. The coefficients are mostly factor of 2 and the division necessary is just by 128. The required arithmetic operations can therefore be implemented mostly via shift registers if a hardware implementation was attempted. Furthermore, the synthesis and analysis filters use similar coefficients, excepting the sign change.

For more information see *Multimedia Systems Standards and Networks*, Atul Puri and Tsuhan Chen, Marcel Dekker, 2000, ISBN 0-8247-9303-X.

1.8 The 2D DWT applied to Lenna

Figure 1 shows the 2D DWT applied to Lenna using the Haar and LeGall filterbanks. The energy compaction in the subbands is clear. The orientation of the image features in the subbands is similar. Two levels of decomposition are shown. The energy distribution of the two decompositions is different, with the LeGall DWT perhaps giving better energy compaction.

1.9 Compression properties of wavelets

We now look at how well some of the various wavelet filters perform in practice. We have used them in place of the Haar transform of section 2, and have measured the entropies and reconstructed the images from quantised coefficients.

In order to allow a fair comparison with the JPEG DCT results, we have modified the DWT quantising strategy to take advantage of the reduced visibility of the higher frequency wavelets. This approximately matches the effects achieved by the JPEG $\mathbf{Q}_{lum}$ matrix introduced in the handout on

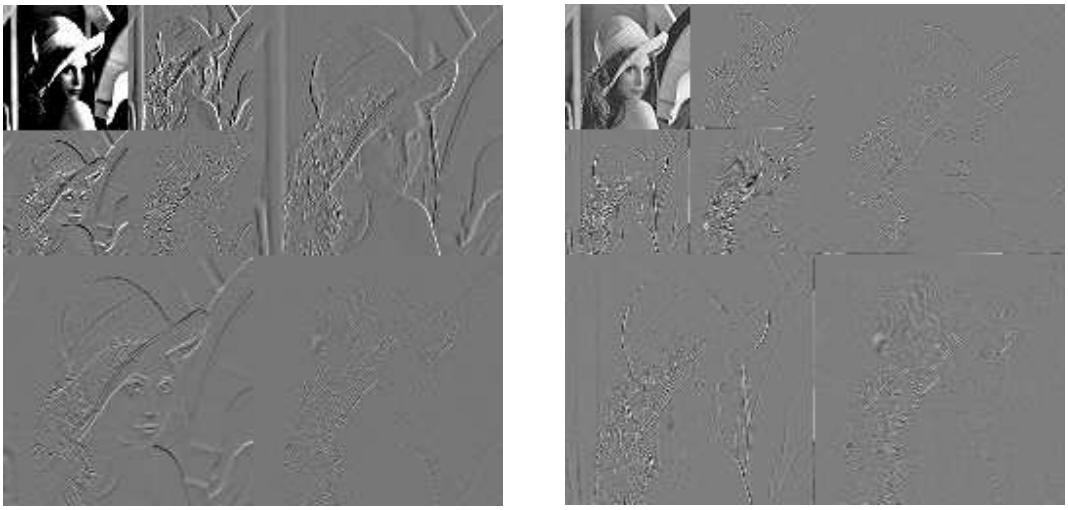


Figure 1: 2D DWT using two different Wavelets Left: Haar wavelet, Right: LeGall Wavelet

the DCT and JPEG. To achieve a high degree of compression we have used the following allocation of quantiser step sizes to the 4-level DWT bands:

$$\begin{aligned} \text{All bands at levels 3 and 4:} \quad & Q_{step} = 50 \\ \text{Hi-Lo and Lo-Hi bands at level 2:} \quad & Q_{step} = 50 \\ \text{Hi-Hi band at level 2:} \quad & Q_{step} = 100 \\ \text{Hi-Lo and Lo-Hi bands at level 1:} \quad & Q_{step} = 100 \\ \text{Hi-Hi band at level 1:} \quad & Q_{step} = 200 \end{aligned} \tag{33}$$

A similar compressed bit rate is produced by the 8×8 DCT when $\mathbf{Q}_{step} = 5\mathbf{Q}_{lum}$.

For reference, fig 4.19 compares the DCT and Haar transforms using these two quantisers. The rms errors between the reconstructed images and the original are virtually the same at 10.49 and 10.61 respectively, but the DCT entropy of 0.2910 bit/pel is significantly lower than the Haar entropy of 0.3820 bit/pel. Both images display significant *blocking artefacts* at this compression level.

Fig 4.20 shows the reconstructed images for the following four DWTs using the quantiser of equations (33):

1. The LeGall 3,5-tap filters: H_0, H_1 and G_0, G_1 from equations (29) and (30).
2. The inverse-LeGall 5,3-tap filters: equations (29) and (30) with H_0, H_1 and G_0, G_1 swapped.
3. The near-balanced 5,7-tap filters.
4. The near-balanced 13,19-tap filters.

(a) (b)

Fig 4.19: Reconstructions after coding using the 8×8 DCT (a) with $\mathbf{Q}_{step} = 5\mathbf{Q}_{lum}$, and (b) using the Haar transform with $\mathbf{Q}_{step}$ from equations (33).

We see that the LeGall 3,5-tap filters (fig 4.20a) produce a poor image, whereas the other three images are all significantly better. The poor image is caused by the roughness of the LeGall 5,3-tap filters (shown in fig 4.7) which are used for reconstructing the image when the 3,5-tap filters are used for analysing the image. When these filters are swapped, so that the reconstruction filters are the 3,5-tap ones of fig 4.6, the quality is greatly improved (fig 4.20b).

The near-balanced 5,7-tap filters (fig 4.20c) produce a relatively good image but there are still a few bright or dark point-artefacts produced by the sharp peaks in the wavelets (shown in fig 4.9). The smoother 13,19-tap wavelets (fig 4.10) eliminate these, but their longer impulse responses tend to cause the image to have a slightly *blotchy* or *mottled* appearance.

(a) (b)

(c) (d)

Fig 4.20: Reconstructions after coding using $\mathbf{Q}_{step}$ from equations (33) with (a) the LeGall 3,5-tap filters, (b) the inverse-Legall 5,3-tap filters, (c) the near-balanced 5,7-tap filters, and (d) the near-balanced 13,19-tap filters.

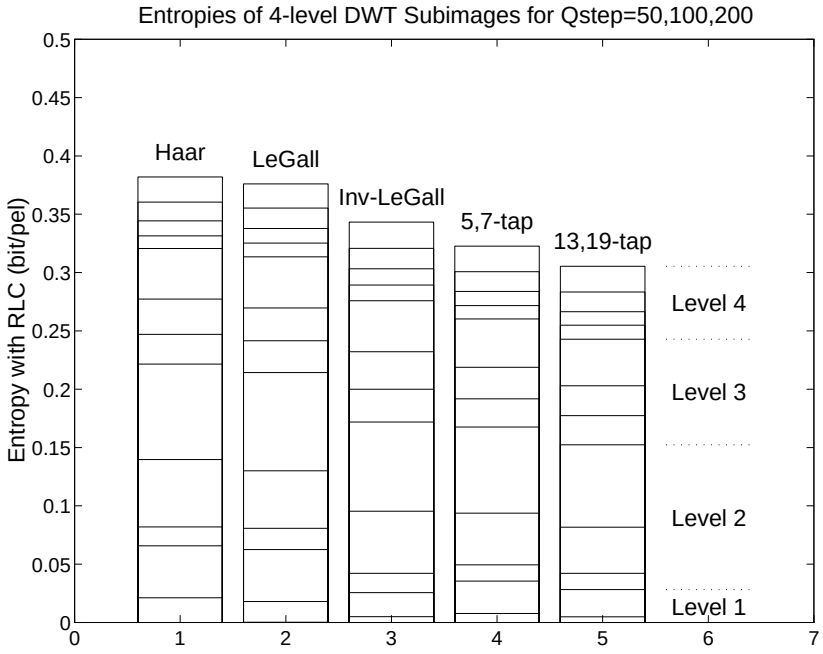


Fig 4.21: Entropies of 4-level DWT subimages using $\mathbf{Q}_{step}$ defined by equations (33), for five different wavelet filter pairs.

Fig 4.21 shows the entropies (with RLC) of the separate subimages of the 4-level DWT for the Haar filter set and the four filter sets of fig 4.20. $\mathbf{Q}_{step}$ is defined by equations (33) and it is particularly noticeable how the higher step sizes at levels 1 and 2 substantially reduce the entropy required to code these levels (compare with fig 2.12). In fact the Hi-Hi band at level 1 is not coded at all! The reduction of entropy with increasing filter smoothness is also apparent.

We see that we have now been able to reduce the bit rate to around 0.3 bit/pel.

However measurement of entropy is not the whole story, as it is the tradeoff of entropy vs quantising error which is important. Fig 4.22 attempts to show this trade-off by plotting rms quantising error (obtained by subtracting the reconstructed image from the original) versus the entropy for the 8×8 DCT and the five DWTs. To show the slope of the curves, the measurements are repeated with an 80% lower quantiser step-size, giving lower rms errors and higher entropies. The pair of points for each configuration are joined by lines which indicate the slope of the rate-distortion curve.

Measurements at many more step sizes can be taken in order to give more complete rate-distortion curves if required.

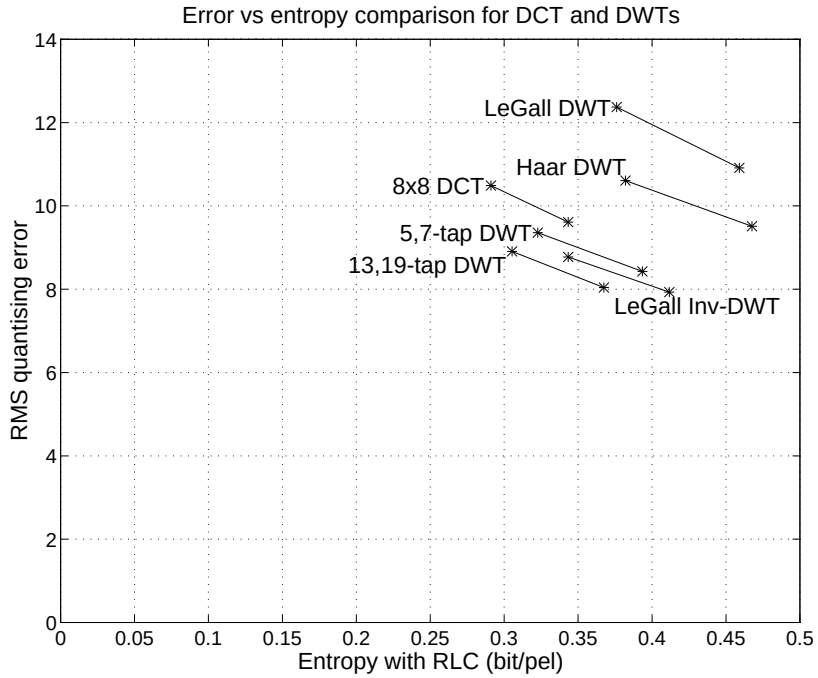


Fig 4.22: RMS error vs. entropy for the 8×8 DCT and five wavelet filter pairs. For the DCT, $\mathbf{Q}_{step} = 5\mathbf{Q}_{lum}$ and $4\mathbf{Q}_{lum}$; for the DWT, $\mathbf{Q}_{step}$ is 1.0 and 0.8 of the values in (33).

The good performance of the 13,19-tap filters is clear, but the inverse-LeGall filters do surprisingly well — showing that the poor smoothness of the analysis filters does not seem to matter. Correct ways to characterise unbalanced filter sets to account properly for this phenomenon are still the subject of current research.

What is clear is that when filters are unbalanced between analysis and reconstruction, the ones which give smoother wavelets *must* be used for reconstruction.

Finally, in these tests, the assessments of subjective image quality approximately match the assessments based on rms errors. However this is not always true and one must be careful to backup any conclusions from rms error measurements with at least some subjective tests.

1.10 Wavelets for Image Analysis

The wavelet decomposition of images is useful for many other tasks aside from compression. The low-pass part of the wavelet tree can be used for *coarse to fine* analysis of image features for example in motion estimation or edge detection. Wavelet decompositions can also be used for de-noising images. This latter application is discussed next.

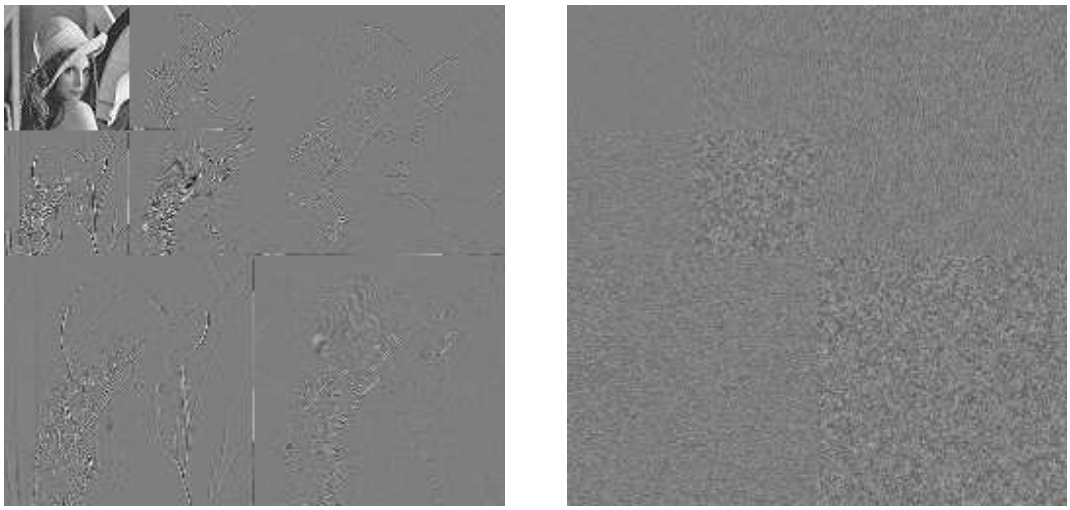


Figure 2: 2 Levels of Wavelet Decomposition using Le Gall 3,5 Tap Filters. Left: Lenna, Right: White, Gaussian random noise of variance $\sigma_v^2 = 100$. There are only a few non-zero coefficients in the DWT coefficients of the Lenna image in the bandpass subimages. The noise field shows many more non-zero coefficients in all bands, especially the Bandpass subimages.

1.10.1 Wavelet De-noising

Throughout this last set of handouts the notion of energy compaction has been used to explain why wavelets are good for compression. In effect the Band Pass wavelet subimages contain only a few HIGH valued coefficients, because most of the DWT (Discrete Wavelet Transform) values are zero. It was noticed some time ago (Donoho et al [2]) that while wavelets compact image information very well, they do not compact noisy data as well. This is because of the low correlation inherent in noise when it occurs as an image artefact.

Figure 2 shows the energy compaction of the DWT (using the LeGall filter bank). The left hand display shows the 7 subbands of the 2 level decomposition of the clean image. Again, we can see that the bandpass images contain only a FEW high valued coefficients. The right hand display shows the same decomposition performed on an image containing noise alone. The noise was Gaussian with variance $\sigma_v^2 = 100$. The images show that the wavelet decomposition of the noise results in wavelet coefficients that are not as compacted as Lenna. This is emphasised in figure 3 which shows the histogram of the HiHi subband of level 2 for the noise and original Lenna data superimposed on each other. The left hand display in figure 3 also shows the 2 Level decomposition of the Lenna image corrupted with the noise used for the right hand display in figure 2. Given the original (clean) image is defined as $I(i,j)$ and the observed (dirty) image is defined as $Y(i,j)$, the corrupted Lenna was created using $Y(i,j) = I(i,j) + v(i,j)$ where $v(i,j) \sim \mathcal{N}(0, \sigma_v^2)$.

A simple recipe, therefore, for de-noising an image is to truncate, or core, the size of the wavelet

[2] Donoho and Johnstone, *Ideal spatial adaptation via wavelet shrinkage.* Biometrika, Vol 81, pages 425-455, December1994.

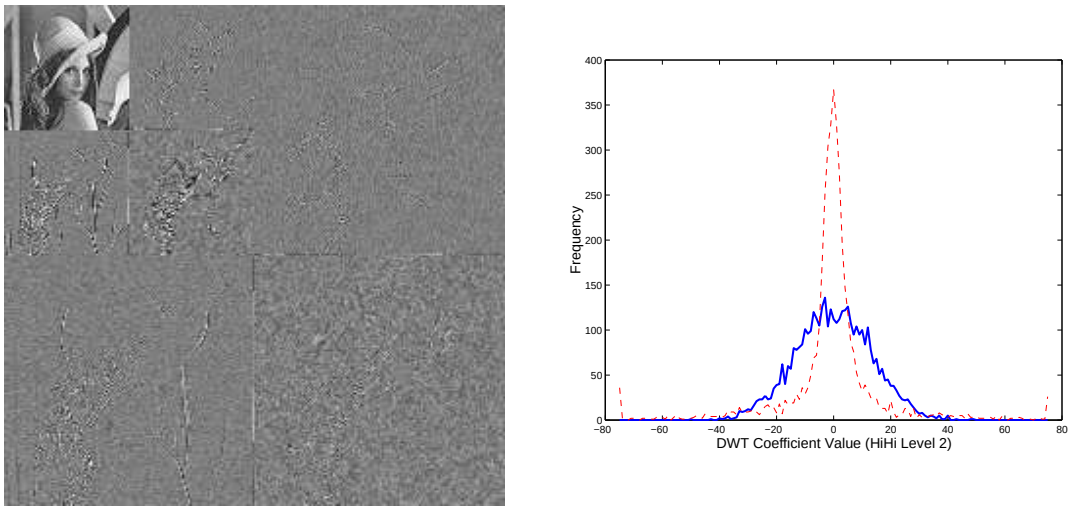


Figure 3: 2 Levels of Wavelet Decomposition using Le Gall 3,5 Tap Filters. Left: Noisy Lenna, Right: Histograms of the HiHi band in the 2nd level decomposition of the noise (solid) and image (dashed).

coefficients in the noisy image. This is called *Wavelet Shrinkage* or *Wavelet Coring*. The idea is as follows. Note that the DWT coefficients of a typical image tend to be small over most of the subimage, and very large in a few localised areas. Therefore if we find DWT values in the corrupted within a small range of zero, those DWT values *should probably have been* 0 if the purely clean image were being analysed. Therefore we can set those DWT coeffs to 0 and leave the others untouched.

Given an image $I(i,j)$ with wavelet coefficients at level l given by $W_l(i,j)$, the truncated coefficients in the output image are given by $\hat{W}_l(i,j)$.

$$\hat{W}_l(i,j) = \begin{cases} W_l(i,j) \text{ If } |W_l(i,j)| > t_l \\ 0 \text{ Otherwise} \end{cases}$$

This is known as *hard* thresholding of the wavelet coefficients (see figure 4), using a threshold of t_l at all levels. Ideally one would use different thresholds at each level since the distribution of wavelet coefficients at each level is different.

It is possible to model the distribution of the wavelet coefficients of the image and noise using a Laplacian distribution. This can then be used to select optimal thresholds for the coring operation at each level. However this is not discussed here. For more information see *A Wavelet Tour of Signal Processing*, Stefane Mallat, Academic Press, ISBN 0-12-466605-1. The design of optimal schemes for wavelet denoising is a subject of much recent research, although the technique itself is now well established. Many industrial broadcast television companies now use some form of wavelet denoising in professional Digital Television and film equipment (SONY Telecine Products, Snell and Wilcox, Philips Telecine Products).

Hard thresholding of the wavelet coefficients sometimes causes artefacts in the output image

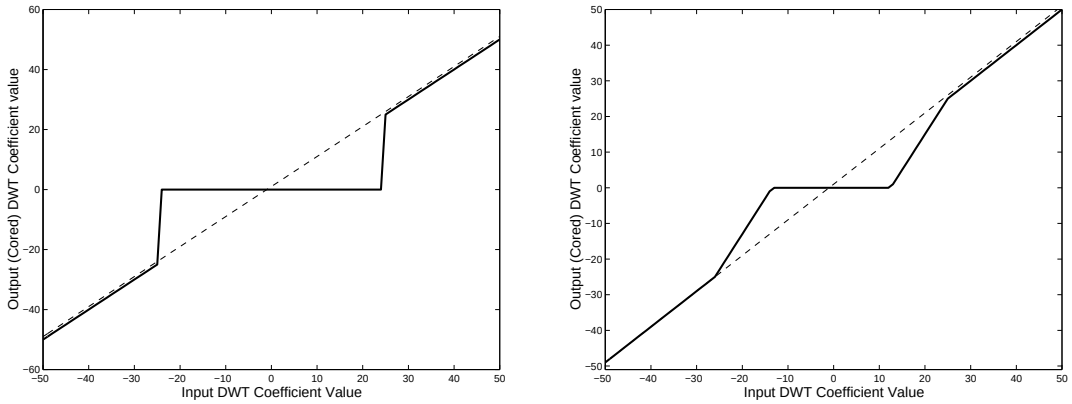


Figure 4: Two simple thresholding or coring schemes for wavelet subbands. Left: Hard, Right: Soft.

Figure 5: Left: Noisy lenna image, Right: Denoised using Wavelet Coring of the LeGall DWT, $t_1 = 20,\ t_2 = 45$.

when it is synthesised. A number of soft thresholding schemes can be defined instead to avoid this. One such form is shown in figure 4.

Figure 5 shows the result of wavelet de-noising applied to the Lena image. The main artefact in the de-noised image is a kind of 'blotchiness'. This is due to the LeGall filters themselves, and the lack of adaptation in this de-noising implementation. Furthermore, this kind of DWT does not allow the analysis of many different orientations. There are just 3 wavelet subbands at each scale which implies the analysis is performed on only 3 principal directions. The image features are orientated in many more directions than these and therefore this DWT implementation is at best achieving some kind of compromise amongst these directions.

Wavelet de-noising has been successful because it does not suffer from the blocking artefacts that many of the standard Wiener-based noise reducers suffer from. Furthermore, it allows the better

preservation of detail since it inherently allows the notion of image 'scale' to be taken into account by adjusting the thresholds at each level of decomposition. However adaptive selection of the thresholds is crucial to the correct application of this method, and this is difficult to engineer.

1.11 Shift-invariance

One of the major drawbacks of the use of wavelet analysis is that the usual wavelet structures for decomposition result in *SHIFT VARIANT* decompositions. That means, when we shift the input signal, the resulting wavelet decomposition can be completely different from the unshifted signal. This is due to the aliasing caused by downsampling the wavelets after each level of decomposition. For compression applications this is not an important drawback. This is because a compression system is always transmitting coded coefficients anyway. The fact that they change because of a trivial change in the image space would only affect the bit rate of the compressed stream of data. If the system is designed correctly, this effect should be small.

However, for pattern recognition tasks and video processing, this feature can be a serious problem. The shift variance of the wavelet decomposition would cause the decomposition to vary as an object is moved across a scene for instance. This could cause a pattern recognition or motion estimation task to fail. A noise reducer designed for video based on a wavelet decomposition could show drastically different amounts of noise reduction as objects move across the frame. This would be unacceptable in most broadcast applications.

This unfortunate phenomenon is demonstrated during the lecture with a simple m-file that is also available for download. Figure 6 shows the 1D DWT of a step function as it translates in the horizontal direction. The lo and hi pass bands of level 1 are shown vertically displaced from the step function itself to allow easier viewing. As can be seen the DWT coefficients are not the same when the step function is shifted to different locations.

It is generally the orthogonal wavelet decompositions (having the same number of wavelet coefficients as input image samples) which suffer from this problem. For this reason it is now widely accepted that while orthogonal wavelet decompositions are useful for compression, it is necessary to use redundant wavelet decompositions (non-orthogonal bases) for signal analysis. A very simple and widely used redundant decomposition is discussed next.

1.11.1 Non-decimated Wavelet Transform

The simplest mechanism to overcome the shift-variance of the orthogonal wavelet decomposition is NOT TO USE THE DOWNSAMPLING operation. This of course results in a huge increase in the data to be processed, but the result is a *SHIFT INVARIANT* wavelet decomposition. This decomposition is known as the *À Trous* (with holes) algorithm, proposed as long ago as 1989. The idea is trivially simple and this decomposition can be used for many kinds of pattern recognition and signal processing tasks.

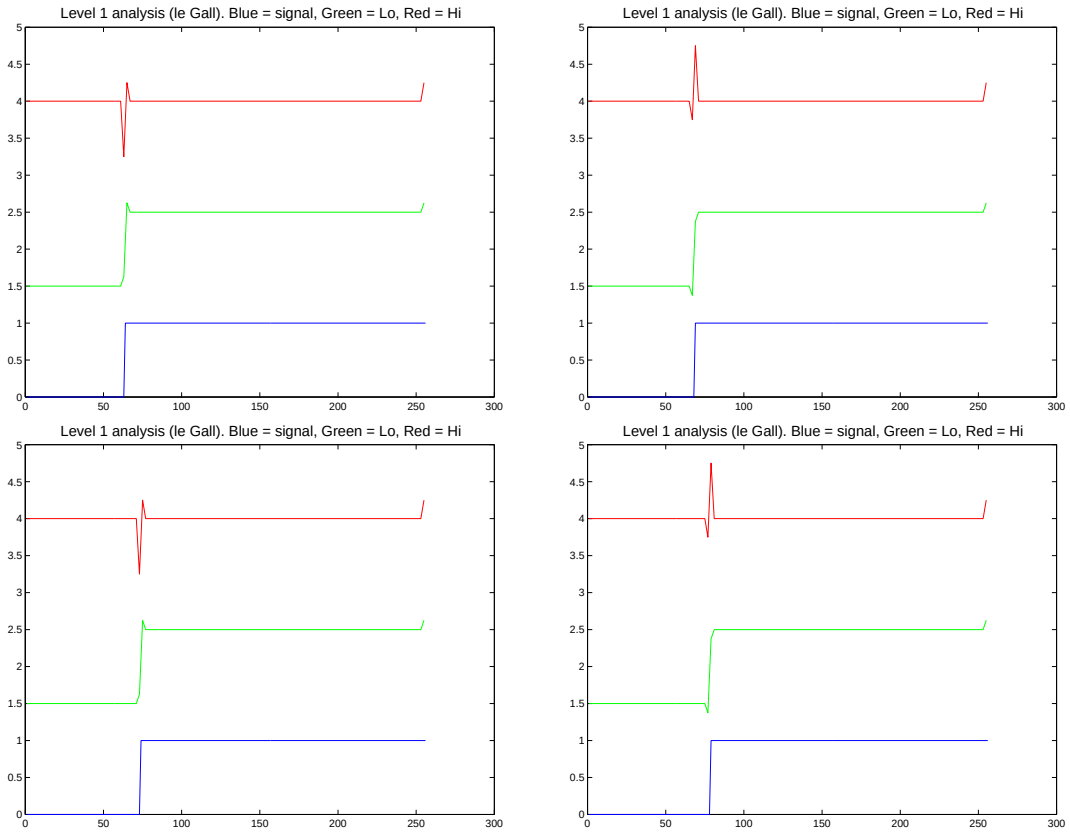


Figure 6: The 1D DWT of a step function at various positions. In each plot only the Hi (Top line) and Lo (Middle line) decompositions at level 1 are shown. As the step function is displaced, the wavelet coefficients change. The DWT is NOT shift invariant.

There are other types of decompositions which are also shift-invariant and more efficient. The Complex Discrete Wavelet Transform (Kingsbury) is a recent development that results in a shift-invariant wavelet decomposition but is much more data efficient than the À Trous algorithm. The Simoncelli Oriented Pyramid decomposition is another such (redundant) shift-invariant wavelet decomposition useful for image analysis. This latter decomposition allows the analysis of more orientations in the image, but this comes with the expense of a large explosion in data to be processed.

1.12 Wavelets and Matlab

The Matlab Wavelet toolbox has a wide range of tools to support wavelet analysis in 2D. Many filters have been implemented including Daubechies and a series of Biorthogonal filters of various lengths. Try the following

```
help wavelet\wavelet
help waveinfo
wavedemo
wavemenu
wdcbm2
```

The last function is an implementation of a wavelet de-noising scheme.

2 Summary

This handout has covered an introduction to wavelets and their use in compression and de-noising. The goal if this handout is to encourage a functional knowledge of the use of wavelets in image processing.

Lim does not cover wavelets, but there is a good coverage in the following texts.

- *Wavelets and Subband Coding*, Martin Vetterli and Jelena Kovacević, Prentice Hall, ISBN 0-13-097080-8
- *A Wavelet Tour of Signal Processing*, Stefane Mallat, Academic Press, ISBN 0-12-466605-1

Wavelet image processing and analysis remains an active research topic although there is less activity now in wavelet filter design.