

Wavelets

4C8: Digital Media Processing

Ussher Assistant Professor François Pitié
2021/2022

Department of Electronic & Electrical Engineering , Trinity College Dublin

adapted from original material written by Prof. Anil Kokaram.

This lecture develops the idea of the Haar Transform into an introduction to wavelet theory. The goals here are to

- introduce Filter Banks
- introduce the 1D and 2D discrete wavelet transform
- observe the use of wavelets for compression
- introduce the use of wavelet analysis for de-noising
- understand that the wavelet decomposition is not necessarily shift-invariant

Filter-Banks

Haar: 2-band filter bank

Recall the 1-D Haar transform from earlier lectures.

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

now, if the sequence is long enough, the results can be placed in two separate sequences y_0 and y_1 :

$$y_0(n) = \frac{1}{\sqrt{2}} (x(n-1) + x(n))$$

$$y_1(n) = \frac{1}{\sqrt{2}} (x(n-1) - x(n))$$

These can be expressed as 2 FIR filters:

$$Y_0(z) = H_0(z)X(z) \quad \text{where } H_0(z) = \frac{1}{\sqrt{2}}(z^{-1} + 1)$$

$$Y_1(z) = H_1(z)X(z) \quad \text{where } H_1(z) = \frac{1}{\sqrt{2}}(z^{-1} - 1)$$

Haar: Analysis

In Haar, every pair of input samples (x_n, x_{n-1}) is transformed into an output pair (y_n, y_{n-1}) .

All pairs (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. Thus only the output $y_0(n), y_0(n-2), y_0(n-4), \dots$ and $y_1(n), y_1(n-2), y_1(n-4), \dots$ are in fact ever calculated.

$$y_n = y_0(n)$$

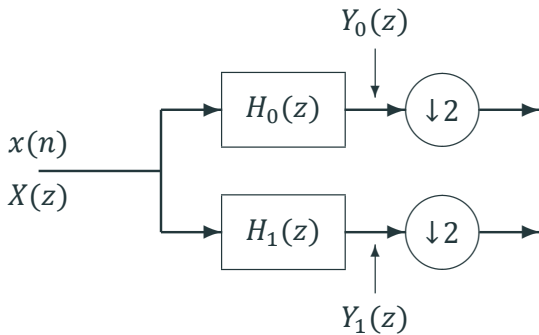
$$y_{n+1} = y_1(n)$$

$$y_{n+2} = y_0(n+2)$$

$$y_{n+3} = y_1(n+2)$$

$$\vdots$$

We may thus represent the Haar transform operation by a pair of filters followed by downsampling (see next slide).



Haar: Reconstruction

Similarly, to reconstruct $\hat{x}(n)$ from $\hat{y}(n)$, we can write for long sequences:

$$\hat{x}(2n-1) = \frac{1}{\sqrt{2}} (\hat{y}_0(2n) + \hat{y}_1(2n))$$

$$\hat{x}(2n) = \frac{1}{\sqrt{2}} (\hat{y}_0(2n) - \hat{y}_1(2n))$$

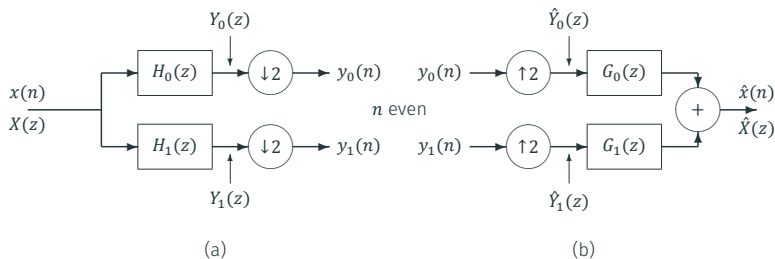
$\hat{y}_0(n)$ and $\hat{y}_1(n)$ are only calculated at even values of n , thus we may assume $\hat{y}(2n+1) = 0$ and we can now write:

$$\hat{x}(n) = \frac{1}{\sqrt{2}} [\hat{y}_0(n+1) + \hat{y}_0(n)] + \frac{1}{\sqrt{2}} [\hat{y}_1(n+1) - \hat{y}_1(n)]$$

or as z-transforms:

$$\hat{X}(z) = G_0(z) \hat{Y}_0(z) + G_1(z) \hat{Y}_1(z) \quad \text{with} \quad \begin{cases} G_0(z) = \frac{1}{\sqrt{2}}(z+1) \\ G_1(z) = \frac{1}{\sqrt{2}}(z-1) \end{cases}$$

Haar: 2-Band Filter Bank



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

Haar: upsampling/downsampling

Let's look at the upsampling/downsampling part:



We can show (next slide) that $Y_0(z)$ is simply related to $\hat{Y}_0(z)$ as follows:

$$\hat{Y}_0(z) = \frac{1}{2}[Y_0(z) + Y_0(-z)]$$

Let's convince ourselves that this is the case. Let's first expand the z -transforms $Y_0(z)$ and $Y_0(-z)$:

$$\frac{1}{2}[Y_0(z) + Y_0(-z)] = \frac{1}{2} \sum_n y_0(n) z^{-n} + y_0(n) (-z)^{-n}$$

Then split the even and odd terms:

$$\begin{aligned} &= \frac{1}{2} \sum_n y_0(2n) z^{-2n} + y_0(2n+1) z^{-2n-1} + \\ &\quad y_0(2n) (-z)^{-2n} + y_0(2n+1) (-z)^{-2n-1} \\ &= \sum_n y_0(2n) z^{-2n} \end{aligned}$$

On the other hand we have:

$$\hat{Y}_0(z) = \sum_n \hat{y}_0(n)z^{-n} = \sum_n \hat{y}_0(2n)z^{-2n} + \hat{y}_0(2n+1)z^{-2n-1}$$

since $\hat{y}_0(2n) = y_0(2n)$ and $\hat{y}_0(2n+1) = 0$.

$$\hat{Y}_0(z) = \sum_n \hat{y}_0(2n)z^{-2n}$$

Thus have indeed $\frac{1}{2}[Y_0(z) + Y_0(-z)] = \hat{Y}_0(z)$

A note here: we've already seen that with resampling.

Recall that the Fourier Transform for frequency ω radians/pixel can be computed by taking the z -Transform at $z = e^{j\omega}$.

Hence

$$X(-z) = X(-e^{j\omega}) = X(e^{j(\omega+\pi)})$$

The term $X(-z)$ then corresponds to a copy of the original spectrum shifted by π . Now π corresponds to half the sampling frequency. Thus, $X(-z)$ is simply that replication of the spectrum at half the sampling frequency that occurs when downsampling by 2.

Perfect Reconstruction

We are not restricted to using the Haar filters, and we could get better compression with more elaborate filters.

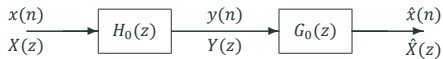
We need to choose 2 low pass filters ($H_0(z)$ and $G_0(z)$) and 2 high pass filters ($H_1(z)$ and $G_1(z)$).

But the reconstruction filters must give exactly the same signal as the input to the analysis filter bank.

This is known as **perfect reconstruction** (PR).

Perfect Reconstruction (PR)

Let's stop for a second and realise that PR is non-trivial by considering the 1-band case:



For the perfect reconstruction to hold, we need to have

$$G_0(z)H_0(z) \equiv 1 \quad \text{and thus} \quad G_0(z) = \frac{1}{H_0(z)}$$

which is simply not possible to achieve with non-trivial FIR filters: if H_0 is FIR then G_0 is IIR. Also if H_0 cuts off frequencies, then there is no way for G_0 to recover these lost frequencies.

Thus PR is non-trivial and it is only by splitting the signal into multiple bands that we can achieve PR with FIR filters.

Perfect Reconstruction

Let's go back to the two-band problem. We are now able to generalise our analysis for arbitrary filters H_0 , H_1 , G_0 and G_1 .

$$\begin{aligned}\hat{X}(z) &= \frac{1}{2}G_0(z)\hat{Y}_0(z) + \frac{1}{2}G_1(z)\hat{Y}_1(z) \\&= \frac{1}{2}G_0(z)[Y_0(z) + Y_0(-z)] + \frac{1}{2}G_1(z)[Y_1(z) + Y_1(-z)] \\&= \frac{1}{2}G_0(z)H_0(z)X(z) + \frac{1}{2}G_0(z)H_0(-z)X(-z) \\&\quad + \frac{1}{2}G_1(z)H_1(z)X(z) + \frac{1}{2}G_1(z)H_1(-z)X(-z) \\&= \frac{1}{2}X(z)[G_0(z)H_0(z) + G_1(z)H_1(z)] \\&\quad + \frac{1}{2}X(-z)[G_0(z)H_0(-z) + G_1(z)H_1(-z)]\end{aligned}$$

Perfect Reconstruction

The Perfect Reconstruction (PR) condition requires $\hat{X}(z) \equiv X(z)$, which is given by:

$$G_0(z)H_0(z) + G_1(z)H_1(z) \equiv 2 \quad (1)$$

and

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) \equiv 0 \quad (2)$$

The Identity (1) is known as the **no-distortion** condition because the term in $X(z)$ corresponds to distortion caused by the filters.

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

PR for the Haar Wavelet

Let's see how the Perfect Reconstruction works for Haar.

$$H_0(z) = \frac{1}{\sqrt{2}}(z^{-1} + 1) \quad G_0(z) = \frac{1}{\sqrt{2}}(z + 1)$$

$$H_1(z) = \frac{1}{\sqrt{2}}(z^{-1} - 1) \quad G_1(z) = \frac{1}{\sqrt{2}}(z - 1)$$

no-distortion:

$$\begin{aligned} G_0(z)H_0(z) + G_1(z)H_1(z) &= \frac{1}{2}(z + 1)(z^{-1} + 1) + \frac{1}{2}(z - 1)(z^{-1} - 1) \\ &= \frac{1}{2}(z^{-1} + z + 2) + \frac{1}{2}(-z^{-1} - z + 2) = 2 \end{aligned}$$

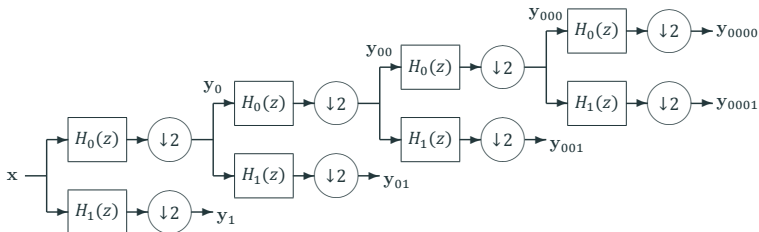
no-aliasing:

$$\begin{aligned} G_0(z)H_0(-z) + G_1(z)H_1(-z) &= \frac{1}{2}(z + 1)(-z^{-1} + 1) + \frac{1}{2}(z - 1)(-z^{-1} - 1) \\ &= \frac{1}{2}(-z^{-1} + z) + \frac{1}{2}(+z^{-1} - z) = 0 \end{aligned}$$

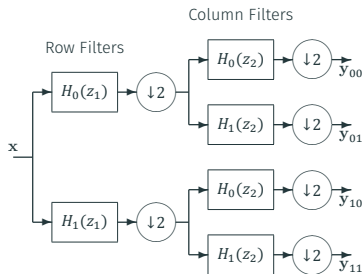
Discrete Wavelet Transform

Binary Filter Tree

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.

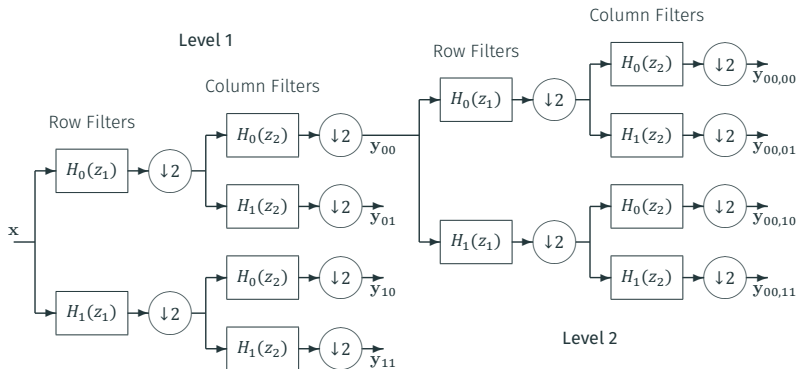


2D Wavelet Transform



y_{00} is mapped to the LoLo, y_{01} to the HiLo, y_{10} to the LoHi and y_{11} to the HiHi.

The Multilevel 2D Discrete Wavelet Xform



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

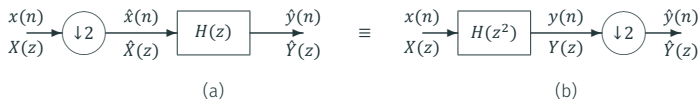
What does it do to a signal?

We need to work out the impulse response of each filter output \mathbf{y}_1 , \mathbf{y}_{01} , \mathbf{y}_{001} , \mathbf{y}_{000} , etc.

But for this, we need to understand the combined effect of the down-sample operations.

Multi-Rate Filtering Theorem

The downsample-filter operation of (a) is equivalent to the filter-downsample of (b) if the filter is changed from $H(z)$ to $H(z^2)$.



Multi-Rate Filtering Theorem: Proof

From (a), we derive

$$\hat{y}_{(a)}[n] = \sum_i \hat{x}[n-i] h[i] = \sum_i x[2n-2i] h[i]$$

From (b), we derive

$$\hat{y}_{(b)}[n] = y[2n] = \sum_i x[2n-i] h'[i]$$

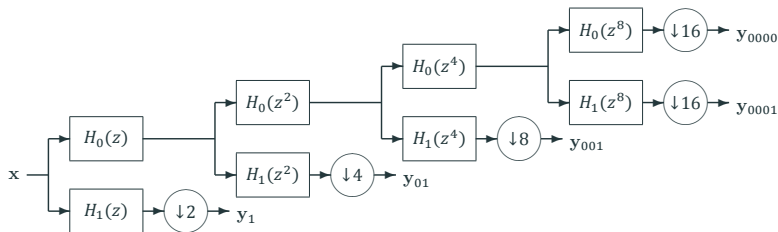
where h' is the impulse response of $H(z^2)$.

$$H(z^2) = \sum_n h'[n] z^{-n} = \sum_n h[n] z^{-2n}$$

note that $h'[2n] = h[n]$ and $h'[2n+1] = 0$, thus

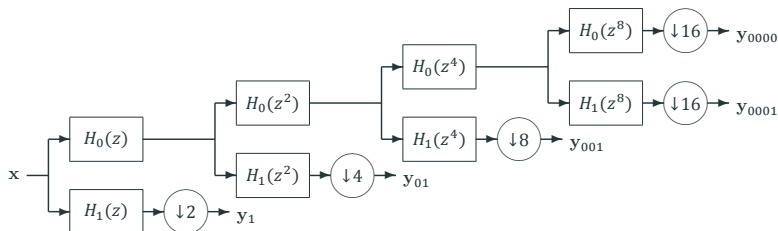
$$\begin{aligned} \hat{y}_{(b)}[n] &= \sum_i x[2n-i] h'[i] = \sum_i x[2n-2i] h'[2i] + x[2n-2i] h'[2i+1] \\ &= \sum_i x[2n-2i] h[i] = \hat{y}_{(a)}[n] \end{aligned}$$

What does it do to a signal?



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

What does it do to a signal? (Haar)



$$H_{01}(z) = H_0(z) H_1(z^2)$$

$$H_{001}(z) = H_0(z) H_0(z^2) H_1(z^4)$$

$$H_{0001}(z) = H_0(z) H_0(z^2) H_0(z^4) H_1(z^8)$$

$$H_{0000}(z) = H_0(z) H_0(z^2) H_0(z^4) H_0(z^8)$$

What does it do to a signal? (Haar)

Transfer functions to the outputs of the 4-level Haar:

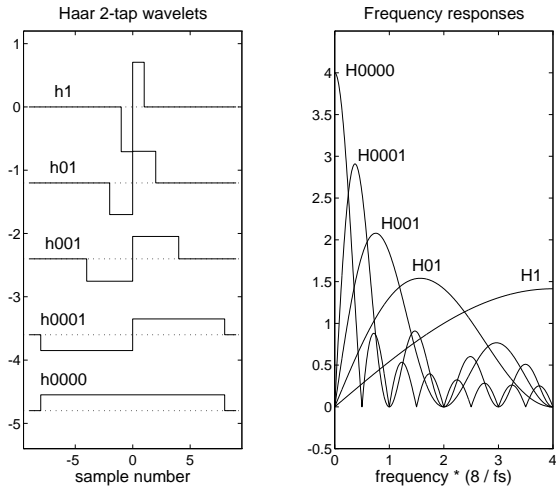
$$H_{01}(z) = \frac{1}{2} [(z^{-3} + z^{-2}) - (z^{-1} + 1)]$$

$$H_{001}(z) = \frac{1}{2\sqrt{2}} [(z^{-7} + z^{-6} + z^{-5} + z^{-4}) - (z^{-3} + z^{-2} + z^{-1} + 1)]$$

$$H_{0001}(z) = \frac{1}{4} [(z^{-15} + z^{-14} + z^{-13} + z^{-12} + z^{-11} + z^{-10} + z^{-9} + z^{-8}) \\ - (z^{-7} + z^{-6} + z^{-5} + z^{-4} + z^{-3} + z^{-2} + z^{-1} + 1)]$$

$$H_{0000}(z) = \frac{1}{4} [z^{-15} + z^{-14} + z^{-13} + z^{-12} + z^{-11} + z^{-10} + z^{-9} + z^{-8} \\ + z^{-7} + z^{-6} + z^{-5} + z^{-4} + z^{-3} + z^{-2} + z^{-1} + 1]$$

Impulse Responses (Haar)



Impulse responses and frequency responses of the 4-level Haar filters.

The process of creating y_1 , y_{01} , etc. is the Wavelet Transform.

The “Wavelet” refers to the impulse response of the cascade of filters.

The shape of impulse response is similar at each level. It is derived from something called a “Mother wavelet”.

The low pass Impulse response to level k is called the “scaling function at level k ”.

Good Wavelets for Compression

There are better filters than the Haar filters.

Wavelet filter design is art and science, and we won't go into this at all in this course. You will just be exposed to a couple of wavelets that are used in the literature. But there are many wavelets! Only some are good for compression and others for analysis.

Wavelet Design

Recall the PR conditions:

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

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

The usual way of satisfying the anti-aliasing condition is to set:

$$H_1(z) = z^{-k}G_0(-z) \quad \text{and} \quad G_1(z) = z^kH_0(-z)$$

where k must be odd. Now we define the lowpass product

$$P(z) = H_0(z)G_0(z)$$

and the no-distortion condition becomes:

$$P(z) + P(-z) = 2$$

which means that $P(z)$ is of the form $P(z) = 1 + \sum_n p_{2n+1}z^{-2n-1}$. And since $P(z)$ should also be zero phase, $P(z)$ should be symmetric:

$$P(z) = \dots + p_3z^{-3} + p_1z^{-1} + 1 + p_1z^{-1} + p_3z^{-3} + \dots$$

The design of a set of $H_0(z)$, $H_1(z)$, $G_0(z)$, $G_1(z)$ filters can now be summarized as follows:

1. choose p_1, p_3 , etc. to design a zero-phase lowpass filter $P(z)$ with some desired qualities.
2. factorize $P(z)$ into $H_0(z)$ and $G_0(z)$ to have similar frequency response
3. derive $H_1(z)$ and $G_1(z)$ from previous slide equation

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 $z = -1$.

The simplest case is

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

which gives the familiar Haar filters.

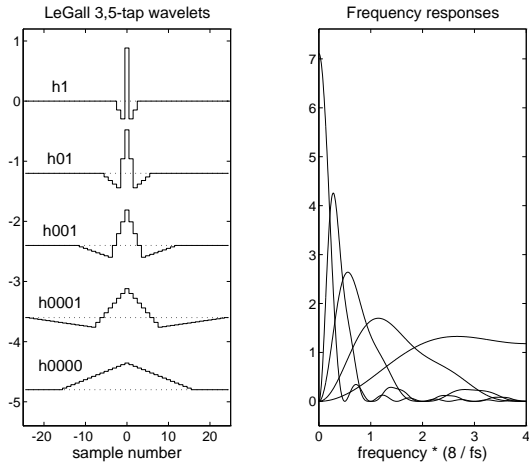
The LeGall 3,5-tap filter set. Given that name because it was first published in the context of 2-band filter banks by Didier LeGall in 1988.

$$H_0(z) = \frac{1}{2}(z + 2 + z^{-1})$$

$$\begin{aligned} G_0(z) &= \frac{1}{8}(z + 2 + z^{-1})(-z + 4 - z^{-1}) \\ &= \frac{1}{8}(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}) \end{aligned}$$

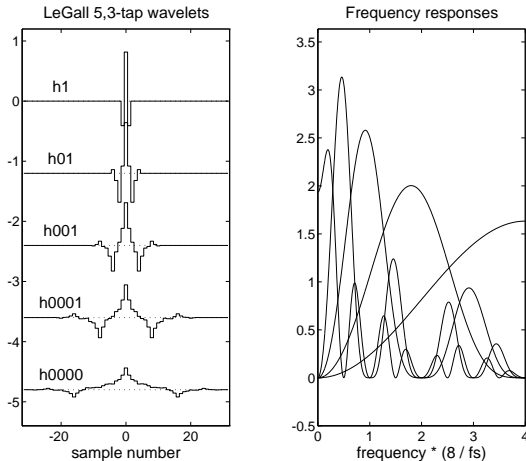
$$G_1(z) = z H_0(-z) = \frac{1}{2}z(-z + 2 - z^{-1})$$

$$H_1(z) = z^{-1} G_0(-z) = \frac{1}{8}z^{-1}(-z^2 - 2z + 6 - 2z^{-1} - z^{-2})$$



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

LeGall 5,3 (Inverse LeGall)



Impulse and freq responses of the LeGall 5,3-tap filters. These are the reconstruction filters of the LeGall 3,5, formed from the G_0 , G_1 filters.

The LeGall 3,5 filters are pretty good for image processing because of the smooth nature of the analysis filters and they are symmetric.

But the reconstruction filters are not smooth.

It turns out that you can swap the analysis and reconstruction filters around and use G_0, G_1 as analysis filters and H_0, H_1 as reconstruction filters. This is called the LeGall 5,3.

Unbalanced System

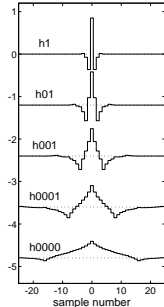
Unbalance between analysis and reconstruction filters is also termed a bi-orthogonal transformation.

It is often regarded as being undesirable, particularly as it prevents the filtering process from being represented as an orthogonal transformation of the input signal.

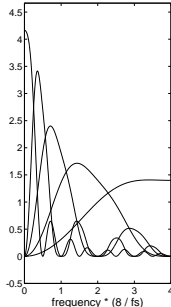
We can obtain more balanced system by using longer filters.

5,7 filters

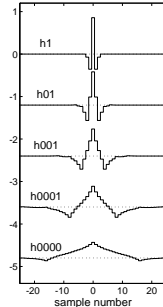
Near-balanced 5,7-tap wavelets



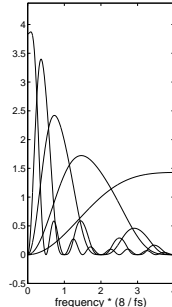
Frequency responses



Near-balanced 7,5-tap wavelets



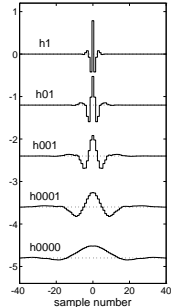
Frequency responses



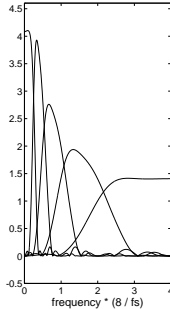
near-balanced 5,7 filters. Left: analysis, Right: reconstruction.

13,19 filters

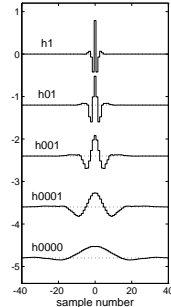
Near-balanced 13,19-tap wavelets



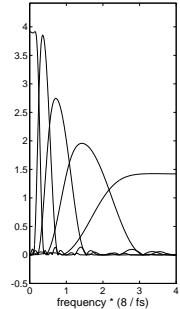
Frequency responses



Near-balanced 19,13-tap wavelets



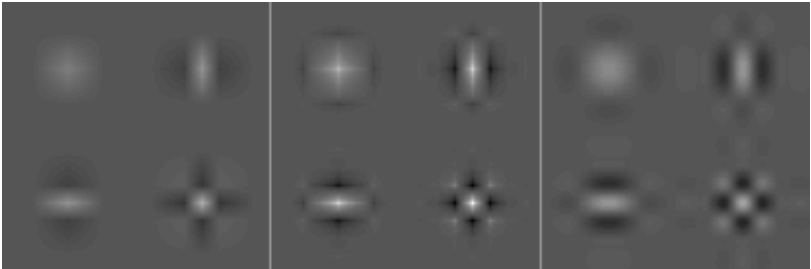
Frequency responses



near-balanced 13,19 filters. Left: analysis, Right: reconstruction.

2D Impulse responses of the separable filters

Legall 3,5-tap, and near balanced 5,7-tap and 13,19-tap 2-D wavelets at level 4



Coding with Wavelets

So we have free choice over what wavelet filters to use, but how do they compare with the DCT?

To compare we must look at which filters give the best compromise between quality and compression efficiency. The tool for this are the rate distortion curves.

Coding with Wavelets

Similarly to the DCT, we should also use some kind of subjectively weighted quantisation and set a lower Q_{step} for low frequency bands (higher level number) and higher Q_{step} for high frequency bands.

In the following, we have used these quantiser steps to the 4-level DWT bands:

| | |
|-----------------------------------|------------------|
| All bands at levels 3 and 4: | $Q_{step} = 50$ |
| Hi-Lo and Lo-Hi bands at level 2: | $Q_{step} = 50$ |
| Hi-Hi band at level 2: | $Q_{step} = 100$ |
| Hi-Lo and Lo-Hi bands at level 1: | $Q_{step} = 100$ |
| Hi-Hi band at level 1: | $Q_{step} = 200$ |

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

Coding with Wavelets



8 × 8 DCT with $Q_{step} = 5Q_{lum}$

RMS: 10.49

Entropy: 0.2910 bits/pixel



Haar with Q_{step} from prev slide

RMS: 10.61

Entropy: 0.3820 bits/pixel

Coding with Wavelets



LeGall 3,5



Inverse LeGall 5,3

Having smooth reconstruction filters is key to reducing artefacts. Hence it is no surprise that LeGall 5,3 performs better for compression.

Coding with Wavelets

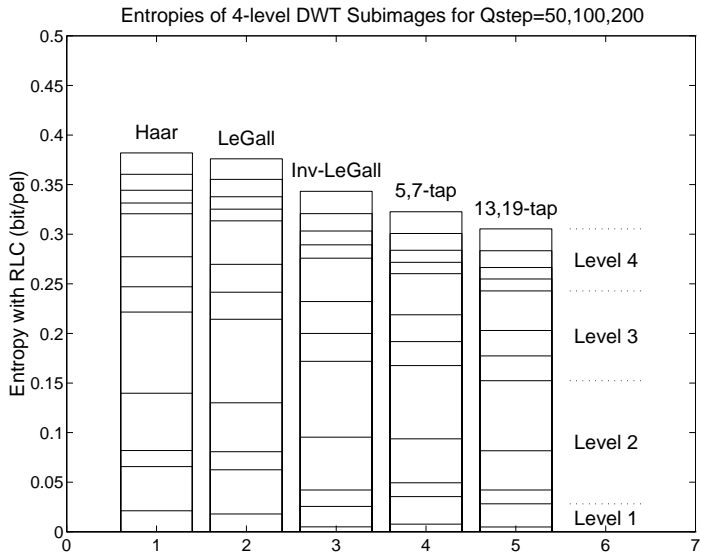


5,7 filters



13,19 filters

Coding with Wavelets



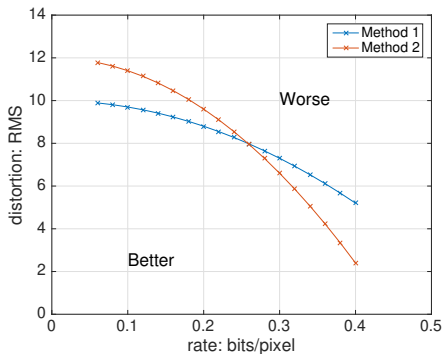
Rate Distortion Curve

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.

What we need to do is graph the distortion (eg. RMS) vs. the entropy. This is the rate distortion curve.

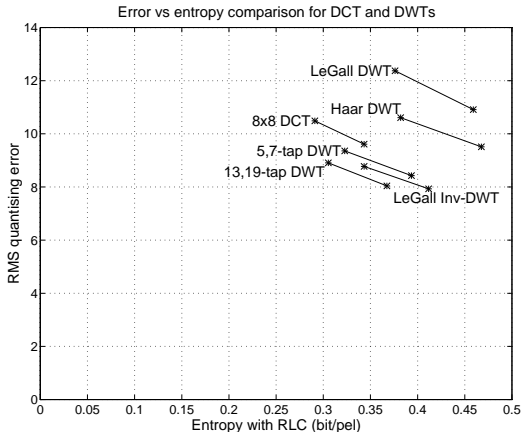
Rate Distortion Curve



With rate-distortion curves, we can compare the performance of different compression methods (eg. different wavelet filters).

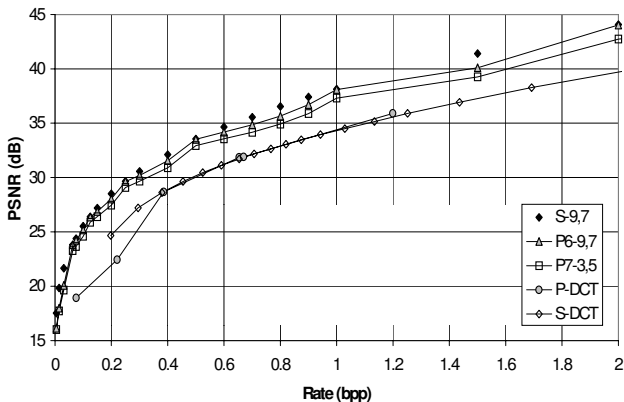
Each point of a curve is obtained by varying one parameter of method (eg. the scale of the quantisation). With JPEG, we can set the quantisation matrix to be λQ_{lum} and vary the scaling factor λ .

Rate Distortion Curve



RMS error vs. entropy for the 8×8 DCT and five wavelet filter pairs. For the DCT, $Q_{step} = 5Q_{lum}$ and $4Q_{lum}$; for the DWT, Q_{step} is 1.0 and 0.8 of the values in prev slide).

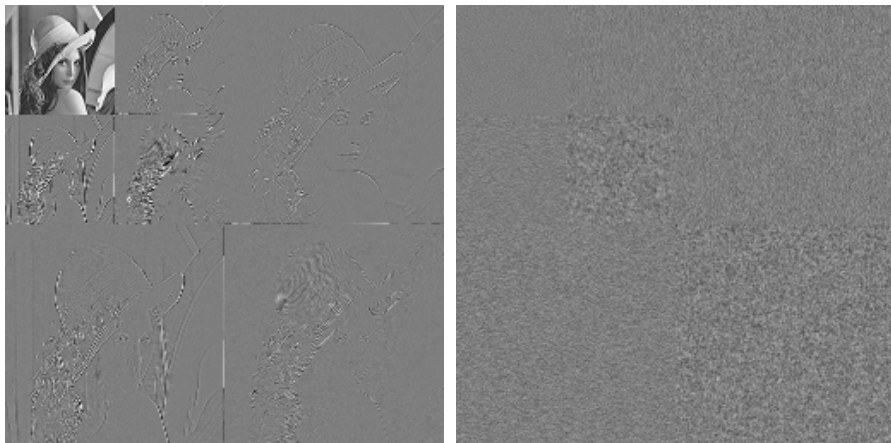
Rate Distortion Curve: baseline JPEG vs JPEG 2000



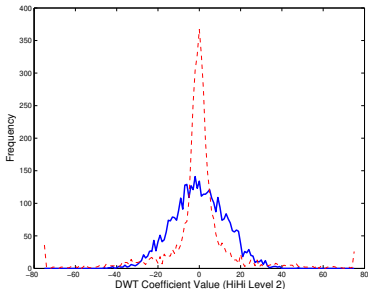
JPEG (for two schemes of DCT) vs JPEG2000 (using 9/7 and 3/5 filters). From *Marcellin et al., "An overview of JPEG2000"*. Recall that high PSNR means better visual quality.

Noise Reduction

As with the Fourier Transform, noise is poorly compacted by the DWT.



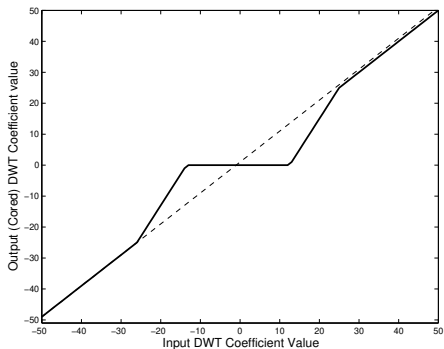
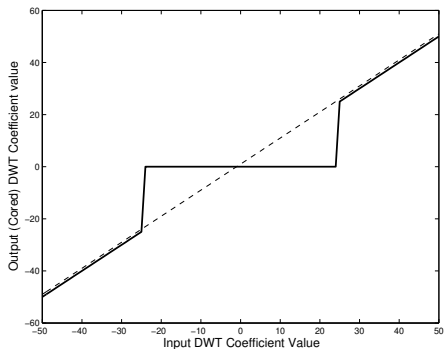
LeGall DWT for Lenna (Left) and Gaussian Noise $\sigma_v^2 = 100$ (Right).



Consider the distributions of the coefficient values for both the clean image Lenna (red) and the Noise (blue). The values for Lenna are mostly small and very large in a few localised areas.

The idea for denoising is thus to set mid-range values to zero and leave the others untouched.

This is called Wavelet Shrinkage or Wavelet Coring



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

Denoising Results



Denoising is important in video for compression efficiency and for image quality as well as image processing/editing.

SONY, Philips, Snell and Wilcox, Foundry, Digital Vision all use wavelet noise reduction of some kind.

Wavelet denoising tends to create less unnatural artefacts than Wiener denoising.

Shift-invariance

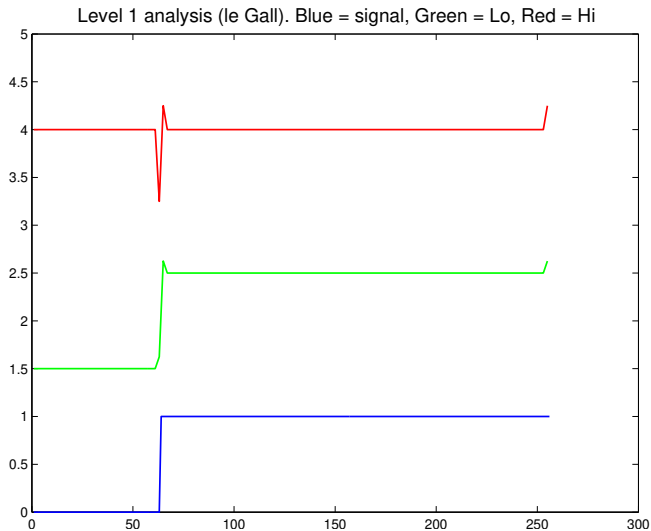
The price for decimation is aliasing.

Remember that the high-pass filters are downsampled. It works because the Perfect reconstruction condition makes sure that aliasing is cancelled by the end of the reconstruction.

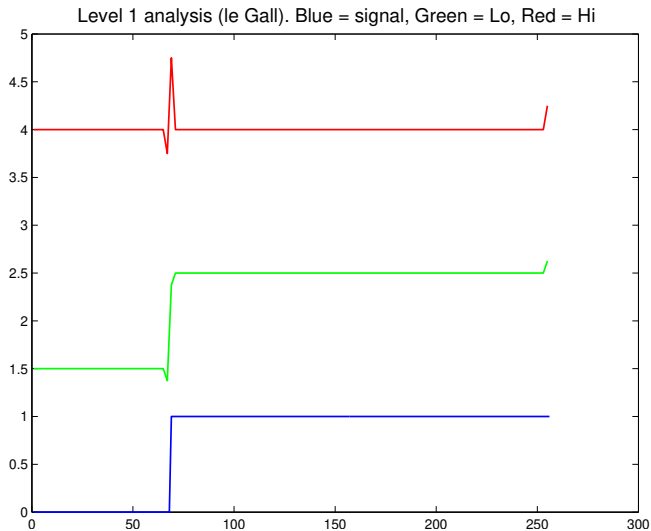
However the coefficients are aliased.

This means that DWT is shift variant! A small shift of the signal greatly perturbs the coefficient pattern at discontinuities. This is not good for analysis tasks such as pattern recognition for instance. In practice it doesn't matter too much for compression.

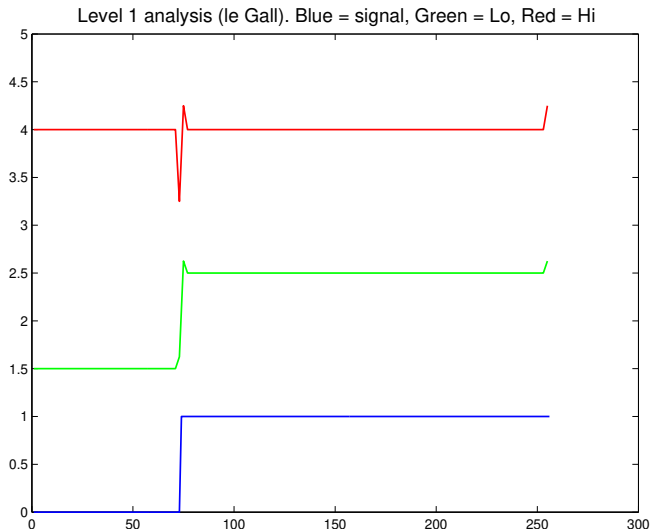
A Shift-Variance example



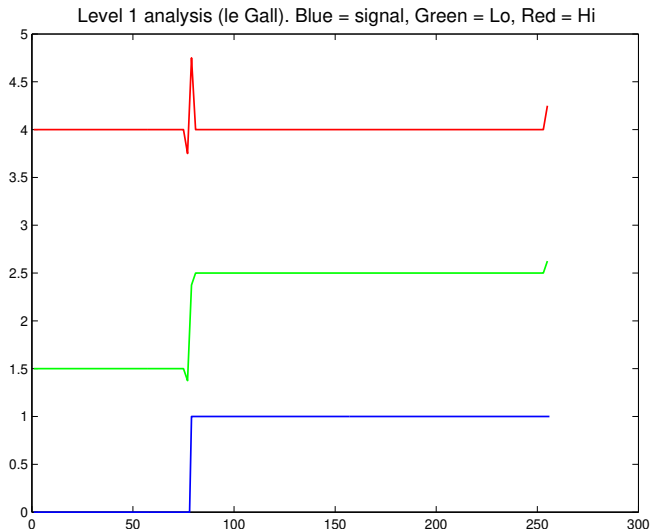
A Shift-Variance example



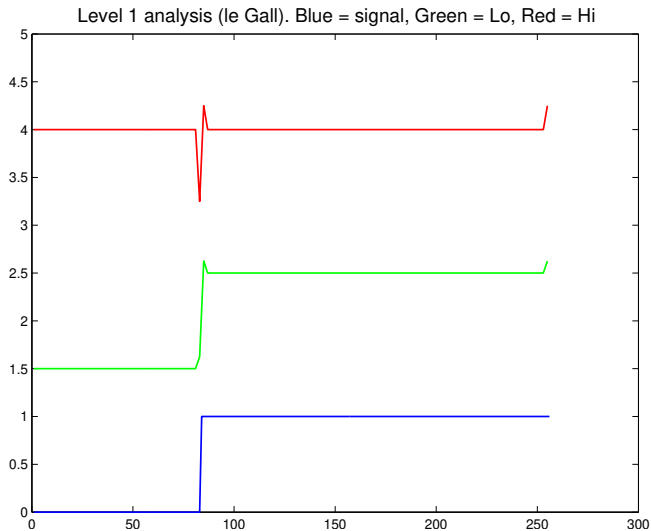
A Shift-Variance example



A Shift-Variance example



A Shift-Variance example



The solution

The solution to preserve shift-invariance is not to downsample and keep all the coefficient undecimated.

This creates a huge redundancy. As you have 4 bands per level,

$$\text{\#coeffs} = 4 \times \text{\#levels} \times \text{\#pixels}$$

OR we can use Nick Kingsbury's Dual-Tree Complex Wavelet Transform, which approximate the shift invariance at a computational cost close to normal DWT.