An Introduction to the Wavelet Transform

IT530, Lecture Notes,
Ajit Rajwade

Sources: (1) MATLAB tutorial on wavelets,
(2) Chapter on wavelets from the book "Sparse Image and Signal Processing" by Starck, Murtagh and Fadili
(3) Two-part tutorial "Wavelets in Computer Graphics", Stollnitz, DeRose and Salesin

Wavelet transform: applications

- Compression (JPEG 2000)
- Denoising, Restoration of Images
- Texture Synthesis
- Medical imaging (ECG, MRI, CT, EEG)
- Machine learning/statistics: function regression and probability density estimation
- Economics
- Fluid Dynamics
- You name it! ☺

Fourier Transform

$$F_k = \sqrt{N}\sum_{n=0}^{N-1} f_n e^{-i2\pi kn/N}$$

k-th Fourier coefficient

$$f_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} F_k e^{i2\pi kn/N}$$

Fourier coefficient

(complex) sinusoidal signals

Value of signal ***f*** at location *n* (***f*** is a vector of size *N*)

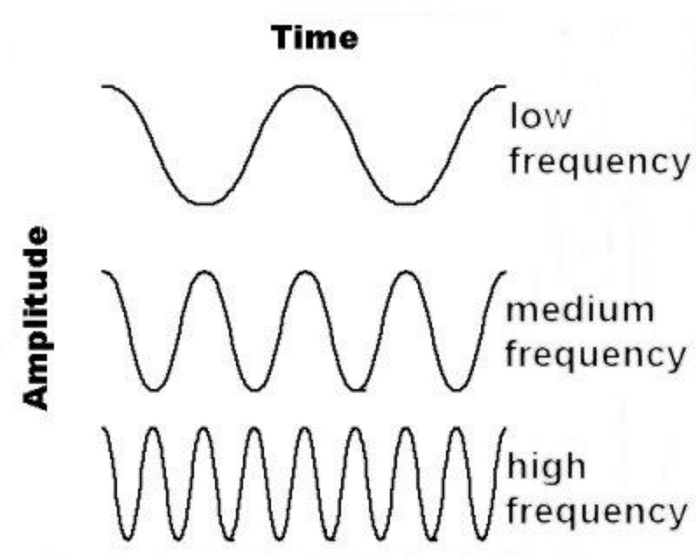
$$f = HF$$

Vector of *N* Fourier coefficients

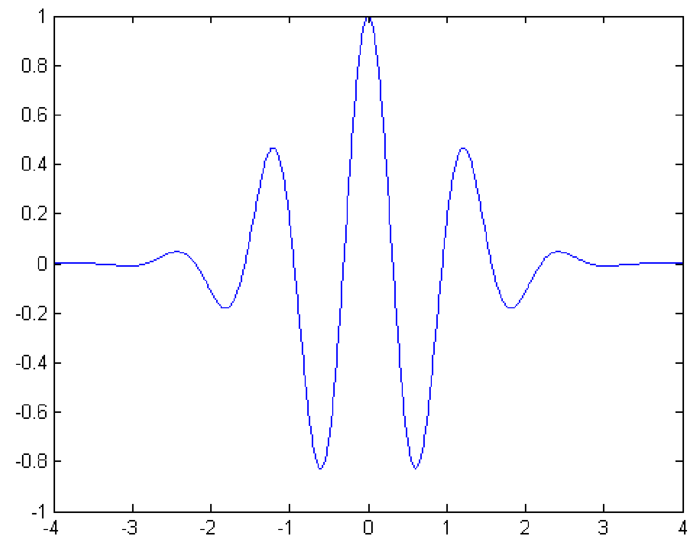
N x *N* orthonormal matrix (Fourier basis matrix)

What is a wavelet?

- A waveform of limited duration.
- Wavelet analysis: signal decomposition as a linear combination of scaled and shifted waveforms.



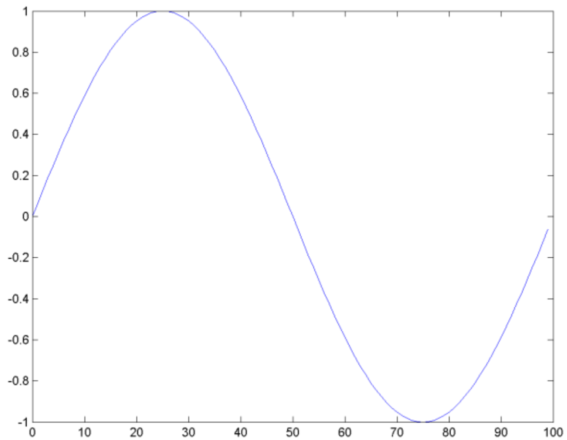
Compactly supported, asymmetric,
Better for signals with sharp changes



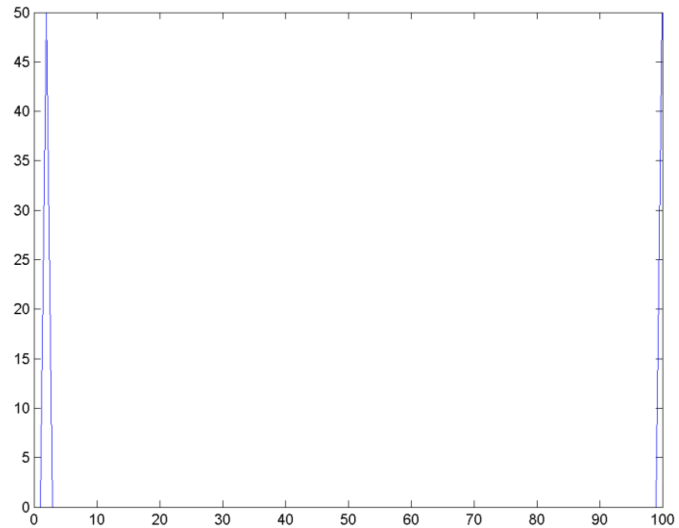
Limitations of Fourier Analysis

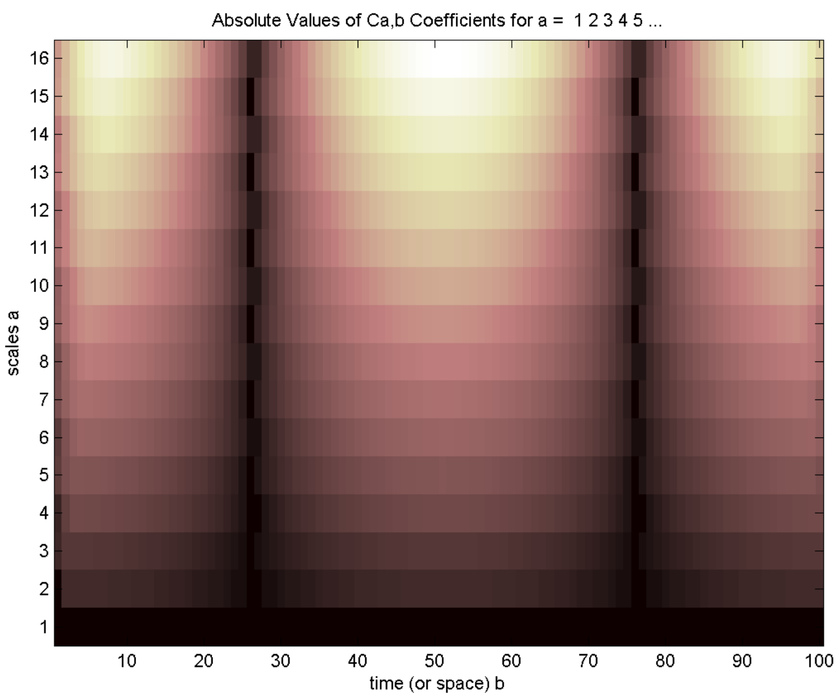
- Tells you ***whether*** a signal contains components of a certain frequency.
- Tells you the ***strength*** of that frequency component.
- But does not tell you ***where/when*** the frequency component occurred (too global!).
- Wavelets try to alleviate this limitation!

Signal



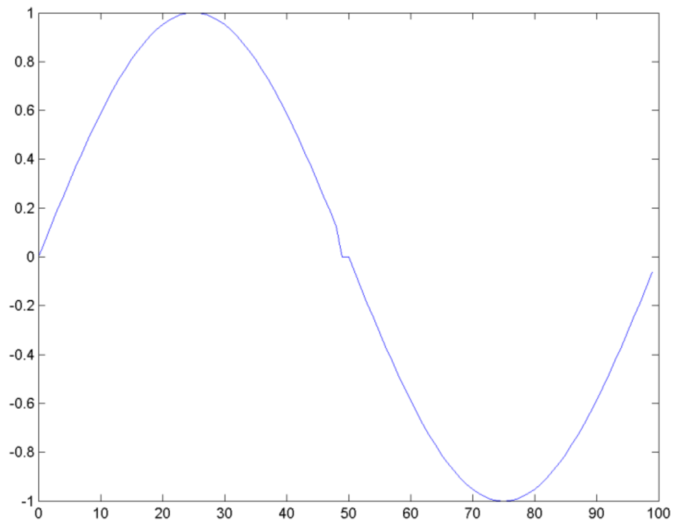
Fourier coefficients



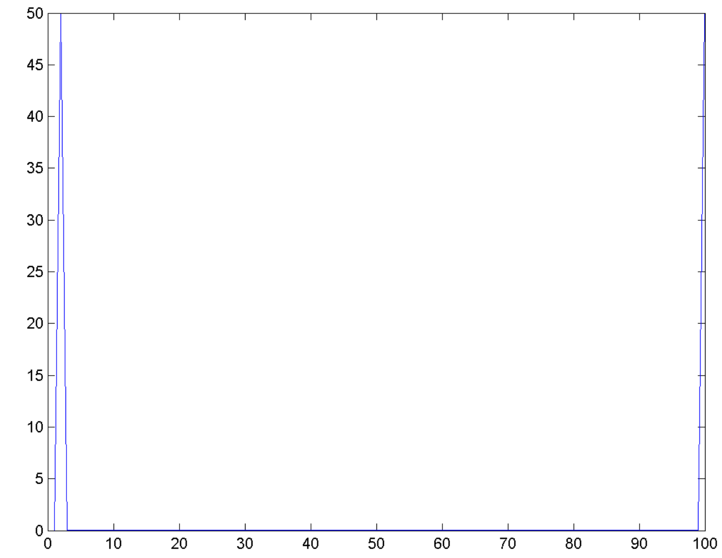


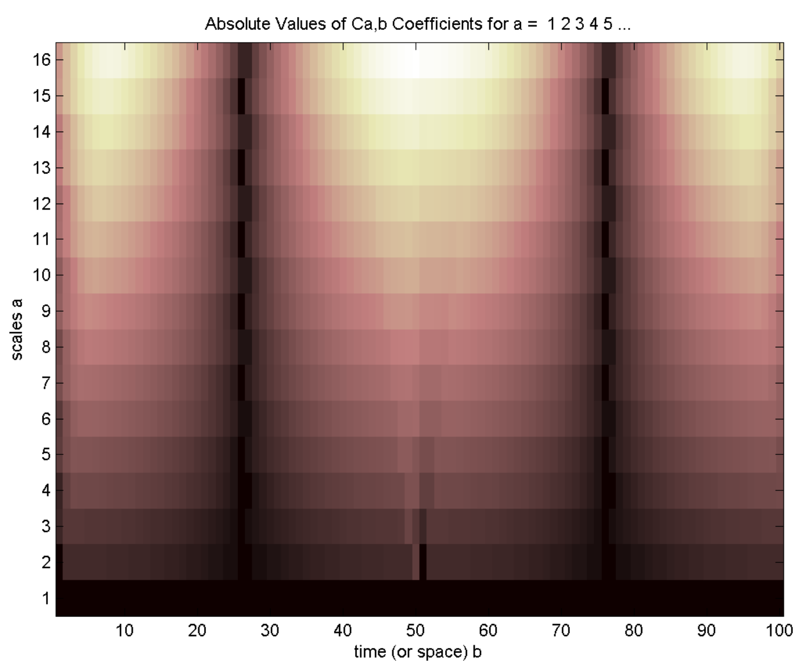
Wavelet coefficients

Signal



Fourier coefficients





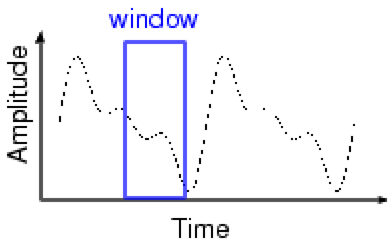
Wavelet coefficients

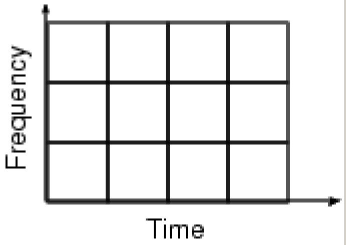
Short-time Fourier Transform (STFT)

- Perform windowing of the signal, and perform Fourier analysis only within the window (repeat over all windows).
- The window function is time-limited, determined by a scale parameter and is shifted around in time.

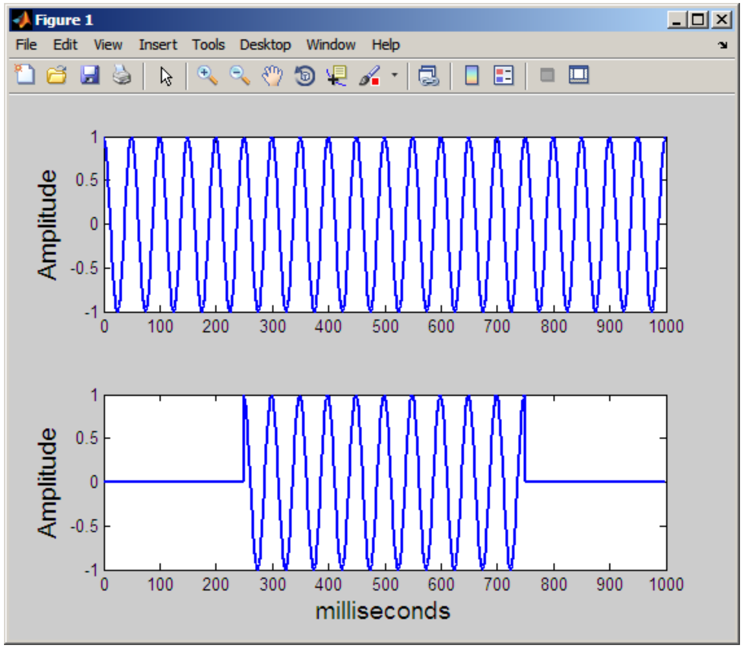
$$w(t) = \sqrt{\frac{\alpha}{\pi}} e^{-\alpha t^2} \xrightarrow{\text{Shifting}} w(t-\tau) = \sqrt{\frac{\alpha}{\pi}} e^{-\alpha(t-\tau)^2}$$

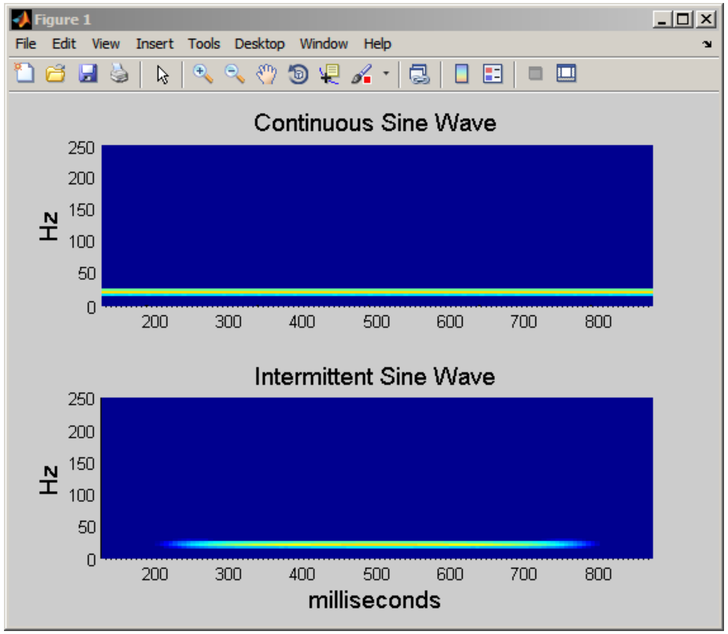
$$STF(f;\omega,t)=\int f(t)w(t-\tau)e^{-j\omega t}dt$$





STFT is a function of two variables, the frequency (omega) and the shift (tau). Precision of the representation is determined by length of time-window (i.e. the scale). The same time-window is applicable for all frequencies.





$$STF(f;\omega,\tau) = \int f(t)w(t-\tau)e^{-j\omega t}dt$$

$$= F(f;\omega) * F_\tau(w;\omega)$$

$$w(t) = \sqrt{\frac{\alpha}{\pi}}e^{-\alpha t^2} \quad \Longleftrightarrow \quad F(w;\omega) = e^{-\omega^2/4\alpha}$$

High precision in time (high alpha) = **Low precision in frequency.**

Low precision in time (low alpha) = **High precision in frequency.**

The (continuous) wavelet transform is an extension of the STFT where **multiple scales (i.e. window sizes)** are used to analyze the signal!

Continuous Wavelet Transform

- Compares a signal to shifted and scaled (stretched or compressed) versions of a wavelet function.
- Redundant transform

$$W(f;a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi^*(\frac{t-b}{a})dt$$

Wavelet function (e.g. Daubechies-n, Morel, Mexican Hat, etc.)

$$f(t) \in L_2(R)$$

Positive scale parameter

Shift parameter (real-valued)

Properties

$$W(v_1 f + v_2 f_2; a,b) = v_1 W(f_1; a,b) + v_2 W(f_2; a,b)$$

Linearity

$$g(t) = f(t-u) \Leftrightarrow W(g; a,b) = W(f; a, b-u)$$

Shifting

$$f_s(t) = f(st) \Leftrightarrow W(f_s; a,b) = \frac{1}{\sqrt{s}} W_f(f; sa, sb)$$

Scaling

Inverse CWT

$$f(t) = \frac{1}{C_\chi} \int\limits_0^{+\infty} \int\limits_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} W(a,b) \chi\left(\frac{t-b}{a}\right) \frac{da.db}{a^2}$$

$$C_\chi = \int\limits_0^{+\infty} \frac{\hat{\psi}^*(\nu)\hat{\chi}(\nu)}{\nu} d\nu = \int\limits_{-\infty}^{0} \frac{\hat{\psi}^*(\nu)\hat{\chi}(\nu)}{\nu} d\nu.$$

Reconstruction is possible only if it is finite, which means that

Can be different from the original wavelet function

$$\hat{\psi}(0) = 0$$

$$\hat{\psi} = F(\psi)$$

Discretized CWT

- Select minimum and maximum scales.
- For each scale defined in this range (discretized), convolve the signal with the following function:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) dt = f * \bar{\psi}_a(b)$$

$$\bar{\psi}_a(t) = (1/\sqrt{a})\psi^*(-t/a)$$

Morlet Function

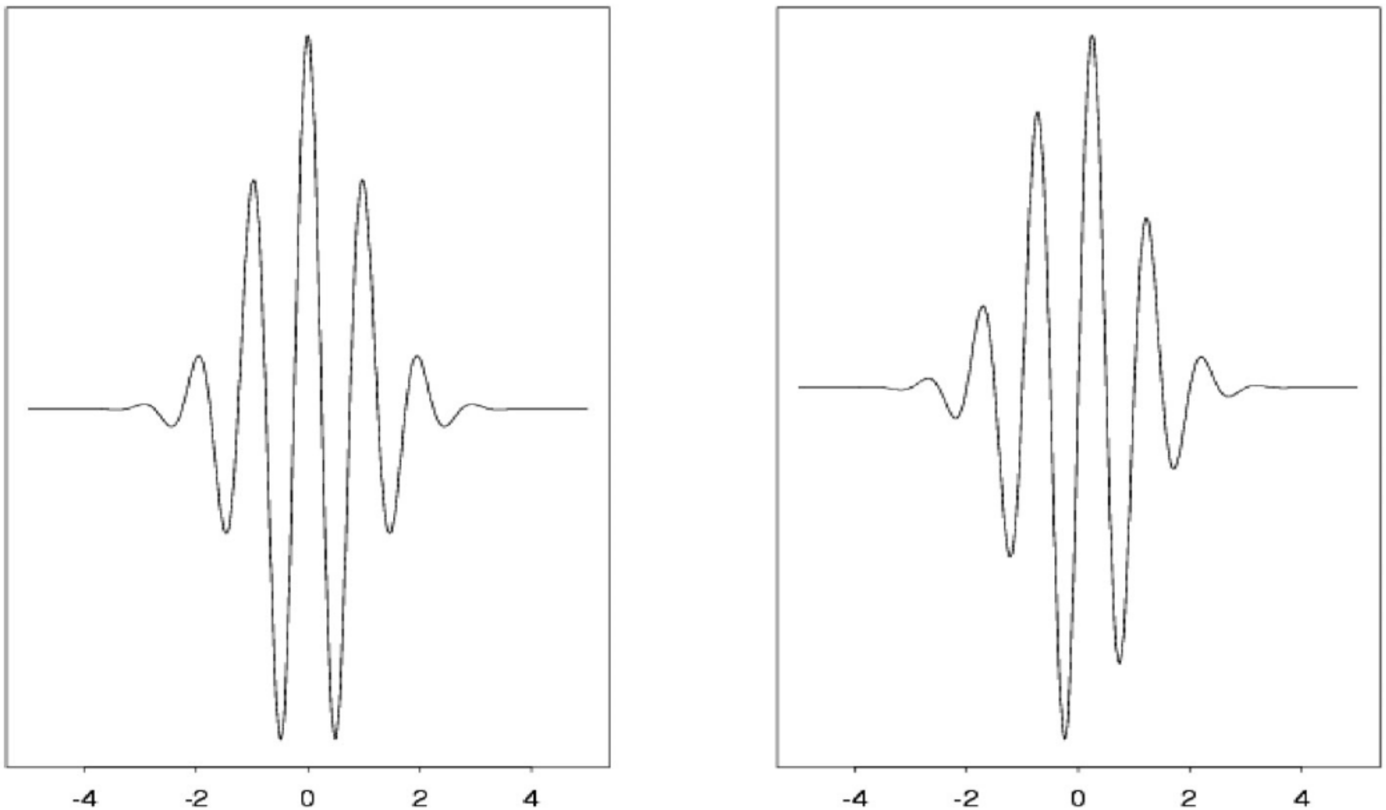


Figure 2.1. Morlet's wavelet: (left) real part and (right) imaginary part.

$$\hat{\psi}(\nu) = e^{-2\pi^2(\nu-\nu_0)^2}.$$

$$\mathrm{Re}(\psi(t)) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \cos(2\pi\nu_0 t)$$

$$\mathrm{Im}(\psi(t)) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \sin(2\pi\nu_0 t),$$

Mexican Hat Wavelet

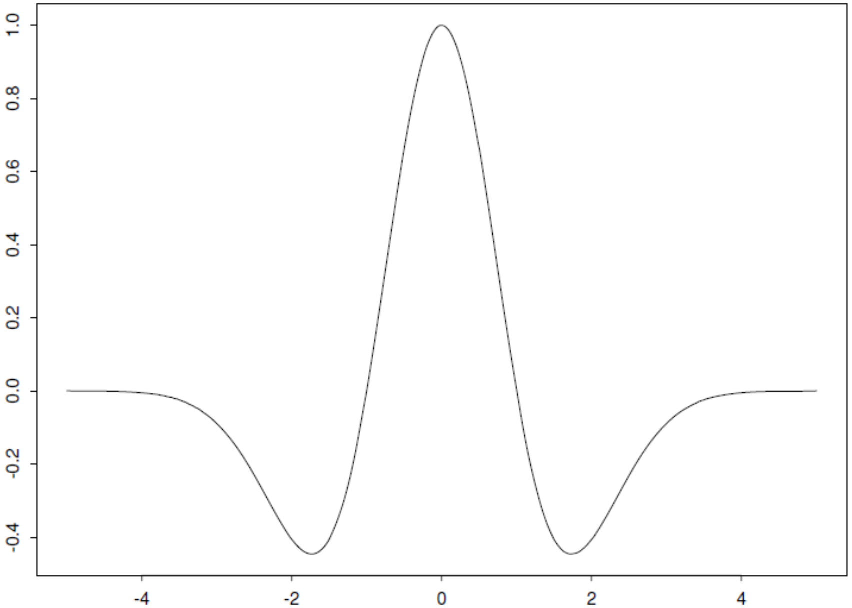
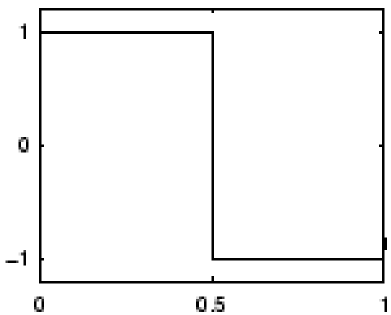


Figure 2.2. Mexican hat wavelet.

$$\psi(t) = (1 - t^2)e^{-\frac{t^2}{2}}.$$

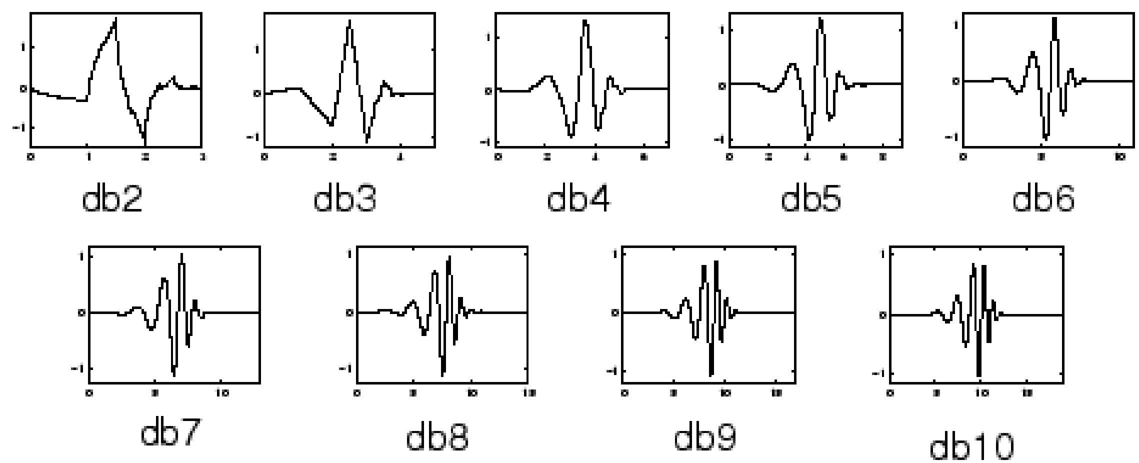
Daubechies Wavelets

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \le t < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \le t < 1 \\ 0 & \text{otherwise.} \end{cases}$$

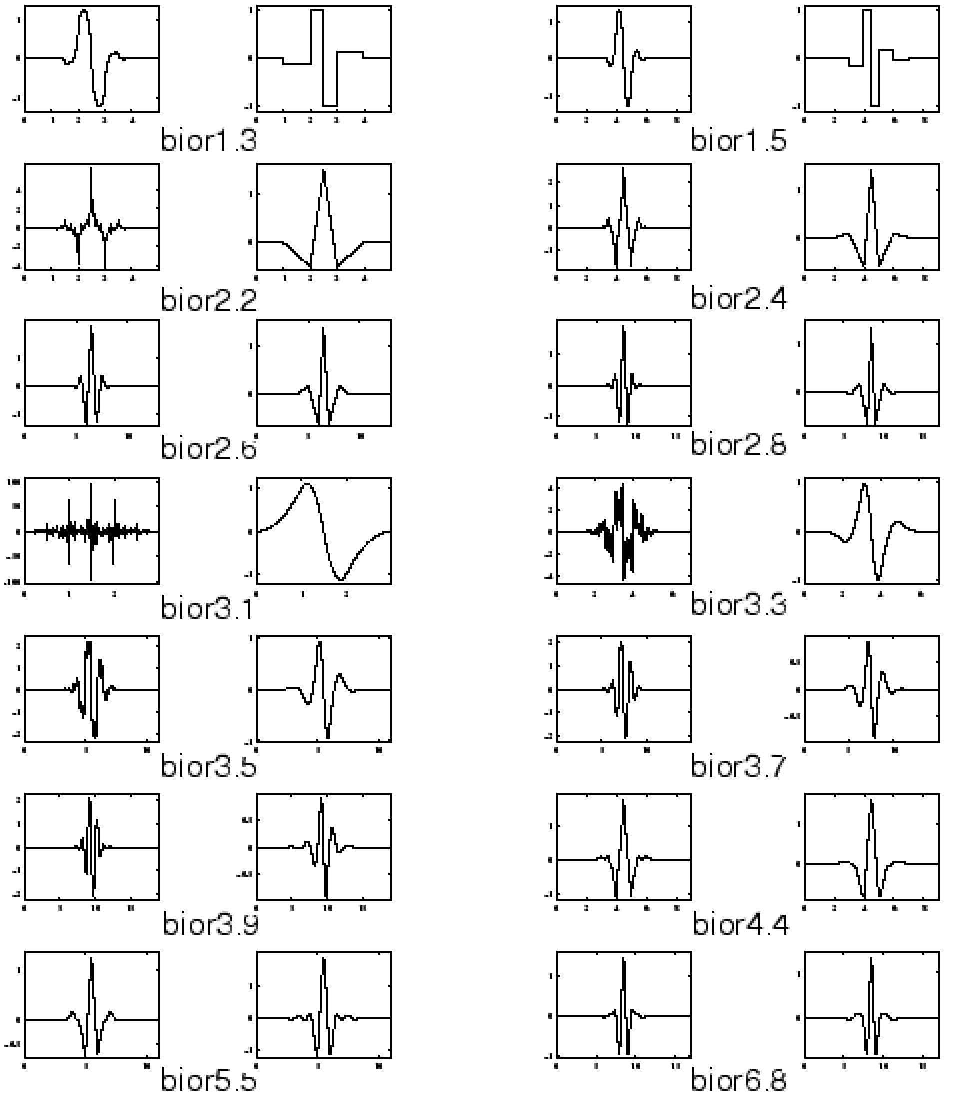


Wavelet function psi

Haar wavelet (db1)



Biorthogonal Wavelets

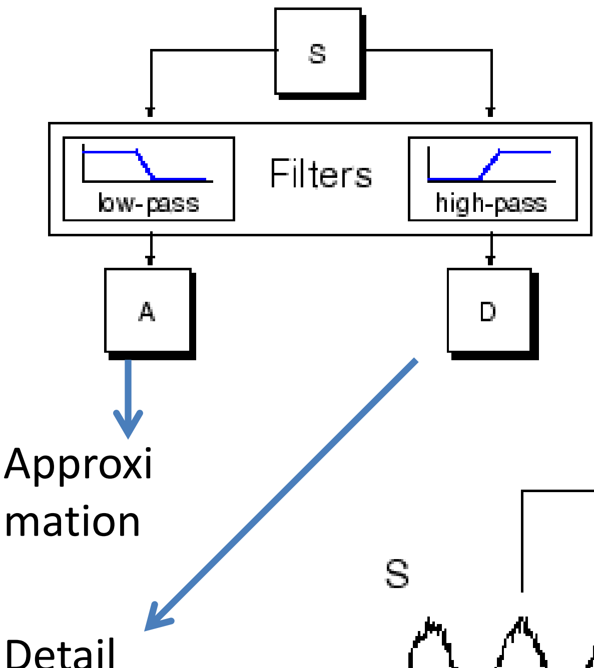


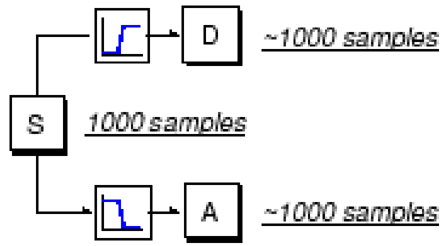
Functions for reconstruction and decomposition are different

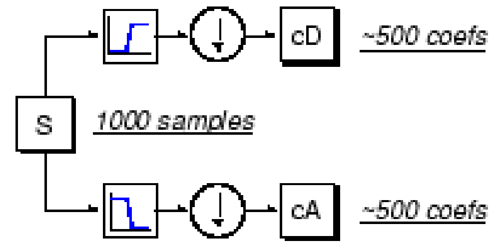
Choice of Wavelet Function

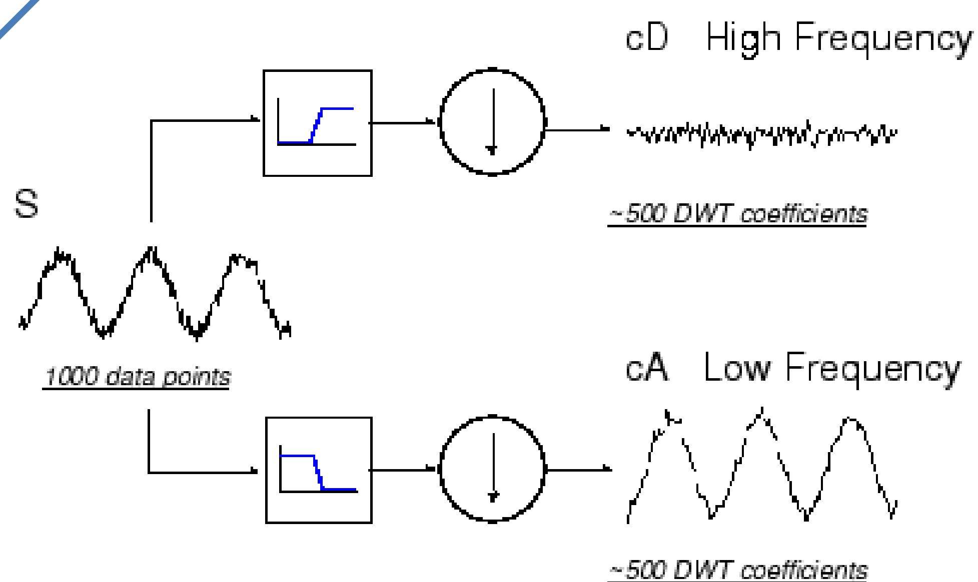
- Depends on user/application (a variety of choices, unlike the case with the Fourier Transform)
- For signals with oscillation, you can use Morlets or db8, db9 or db10.
- For smooth signals, you can use Mexican hat wavelets.
- For piecewise flat signals, you can use db1.

Discrete Wavelet Transform

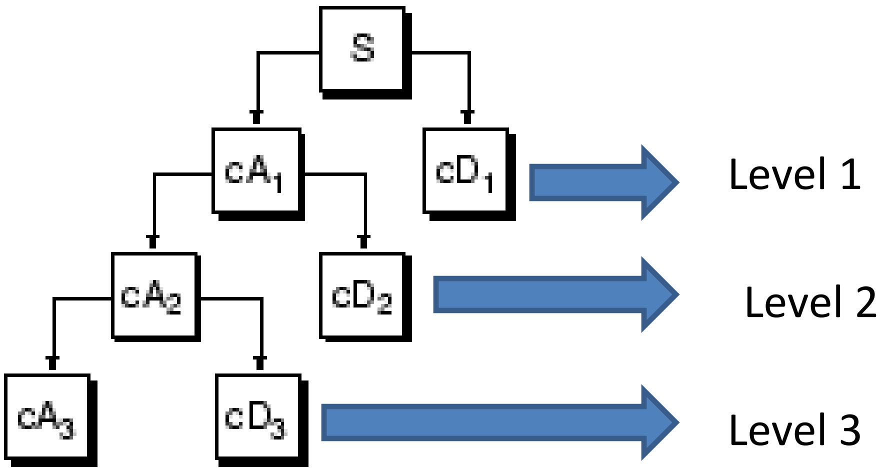




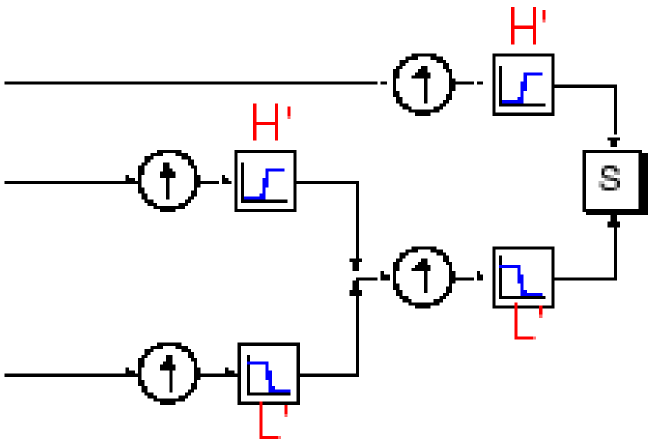


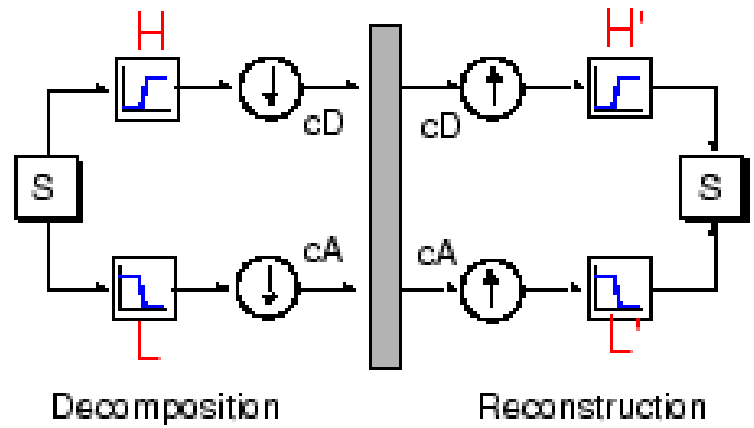


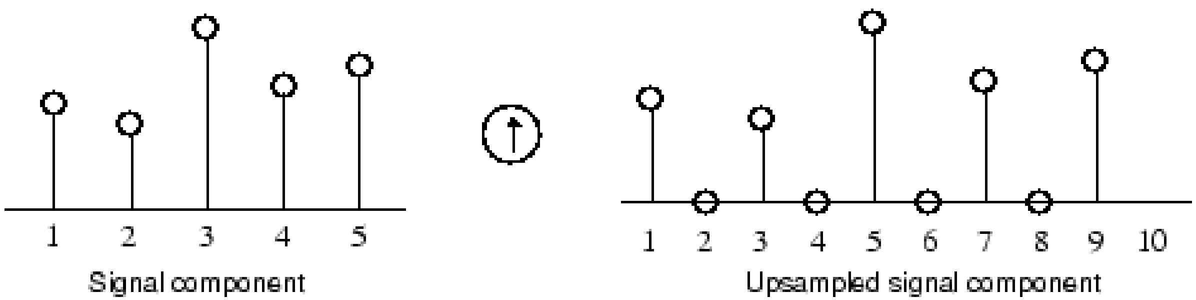
DWT: Decomposition tree

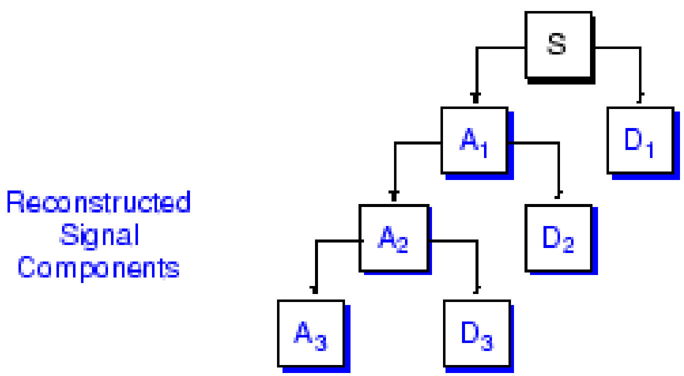


DWT: Synthesis/Reconstruction









$$S = A_1 + D_1$$

$$= A_2 + D_2 + D_1$$

$$= A_3 + D_3 + D_2 + D_1$$

(H,L) and **(H',L')** form what are called as **quadrature mirror filters**

Discrete Wavelet Transform: Multiresolution Analysis

$$\cdots \subset V_3 \subset V_2 \subset V_1 \subset V_0 \cdots$$
$$\text{if } f(t) \in V_j, \text{ then } f(2t) \in V_{j+1}.$$

Nested subspaces generated from different dyadic scales (successive smoothing)

$$c_j[l] = \langle f, \phi_{j,l} \rangle = \langle f, 2^{-j}\phi(2^{-j}. - l) \rangle.$$

$$\frac{1}{2}\phi\left(\frac{t}{2}\right) = \sum_k h[k]\phi(t-k),$$

$$\hat{h}(\nu) = \sum_k h[k] e^{-2\pi i k \nu}.$$

Low pass filter

Projection of signal onto the space $\mathbf{V_j}$ – defined by inner product of the signal and the dilated and shifted version of the scaling function (father wavelet)

$$c_{j+1}[l] = \sum_k h[k-2l] c_j[k].$$

Direct computation of coefficients from one level to another

At each level of smoothing, the size of the signal is halved. The higher frequency details are lost during this process. But they can be recovered from $\mathbf{W}_{j+1}$, the orthogonal complement of $\mathbf{V}_{j+1}$ in $\mathbf{V}_j$.

$$\frac{1}{2}\psi\left(\frac{t}{2}\right) = \sum_k g[k]\phi(t-k),$$

High pass filter

$$w_{j+1}[l] = \langle f, \psi_{j+1,l} \rangle = \langle f, 2^{-(j+1)}\psi(2^{-(j+1)}. - l) \rangle$$
$$= \sum_k g[k-2l]c_j[k].$$

Mother wavelet

Algorithm 2 1-D DWT Algorithm

Task: Compute DWT of discrete finite-length signal X.
Parameters: Filters $h, \tilde{h}$.
Initialization: $c_0 = X$, $J = \log_2 N$.
for $\underline{j = 0 \text{ to } J - 1}$ **do**

- Compute $c_{j+1} = \bar{h} \star c_j$, down-sample by a factor 2.
- Compute $w_{j+1} = \bar{g} \star c_j$, down-sample by a factor 2.

Output: $\mathcal{W} = \{w_1, \ldots, w_J, c_J\}$, the DWT of X.

2D Wavelet Transform

Obtained by separable (tensor) products of
scaling function and wavelet functions

ϕ ψ

$$c_{j+1}[k,l] = \sum_{m,n} h[m-2k]h[n-2l]c_j[m,n]$$
$$= [\bar{h}\bar{h} \star c_j]_{\downarrow 2,2}[k,l],$$

The detail coefficient images are obtained from three wavelets:

- vertical wavelet: $\psi^1(t_1,t_2) = \phi(t_1)\psi(t_2)$
- horizontal wavelet: $\psi^2(t_1,t_2) = \psi(t_1)\phi(t_2)$
- diagonal wavelet: $\psi^3(t_1,t_2) = \psi(t_1)\psi(t_2)$

The detail coefficient images are obtained from three wavelets:

- vertical wavelet: $\psi^1(t_1, t_2) = \phi(t_1)\psi(t_2)$
- horizontal wavelet: $\psi^2(t_1, t_2) = \psi(t_1)\phi(t_2)$
- diagonal wavelet: $\psi^3(t_1, t_2) = \psi(t_1)\psi(t_2)$

$$w^1_{j+1}[k, l] = \sum_{m,n} g[m - 2k]h[n - 2l]c_j[m, n] = [\bar{g}\bar{h} \star c_j]_{\downarrow 2,2}[k, l],$$

$$w^2_{j+1}[k, l] = \sum_{m,n} h[m - 2k]g[n - 2l]c_j[m, n] = [\bar{h}\bar{g} \star c_j]_{\downarrow 2,2}[k, l],$$

$$w^3_{j+1}[k, l] = \sum_{m,n} g[m - 2k]g[n - 2l]c_j[m, n] = [\bar{g}\bar{g} \star c_j]_{\downarrow 2,2}[k, l].$$

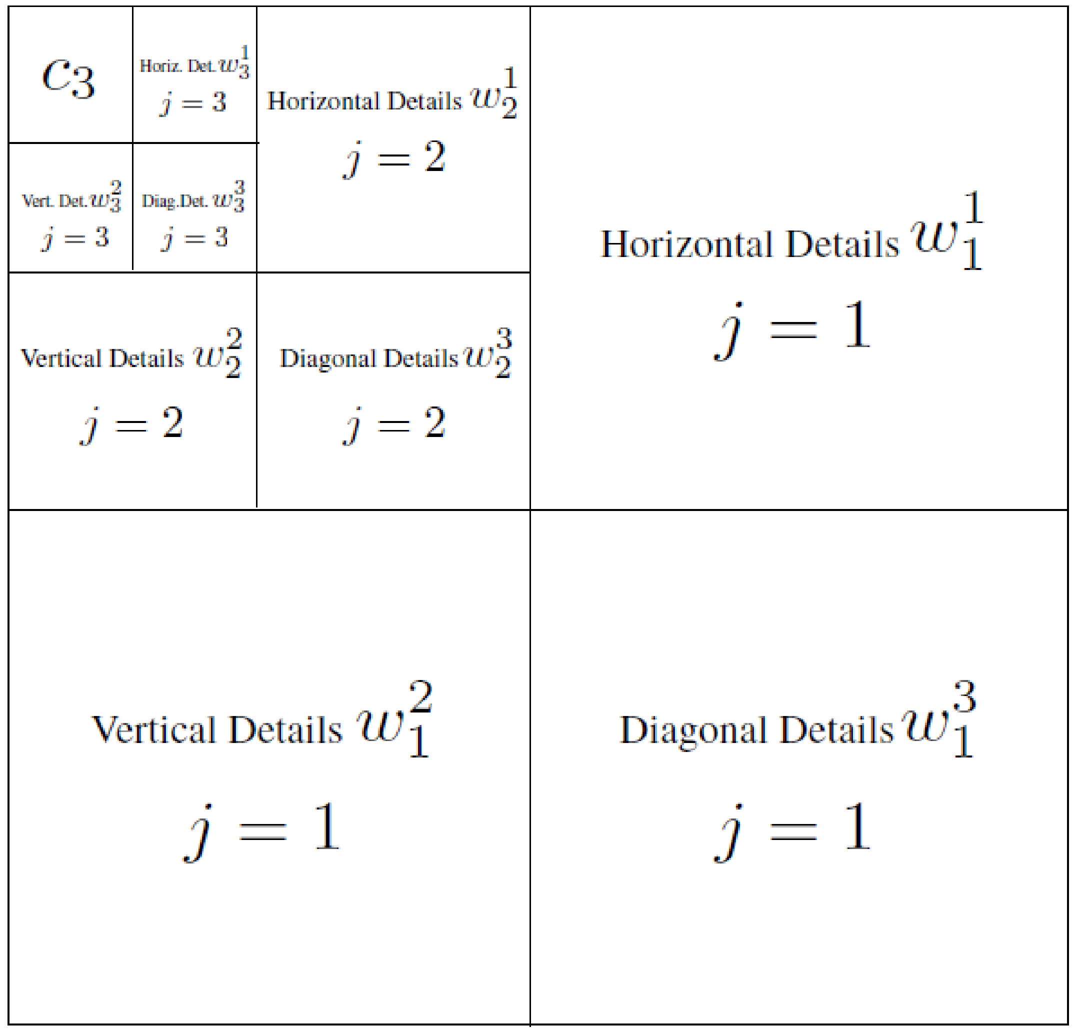
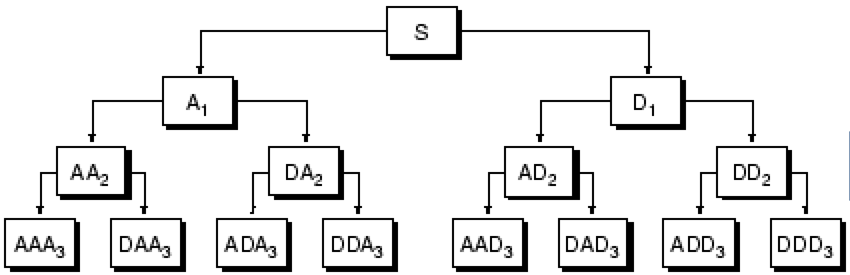
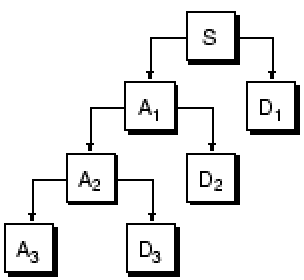


Figure 2.6. Discrete wavelet transform representation of an image with corresponding horizontal, vertical, diagonal, and approximation subbands.

Wavelet packet decomposition: divide not only the approximation, but also the detail coefficients. This leads to larger wavelet decomposition trees.



Wavelet packet decomposition tree



$$
\begin{aligned}
S &= A_1 + D_1 \\
&= A_2 + D_2 + D_1 \\
&= A_3 + D_3 + D_2 + D_1
\end{aligned}
$$

Wavelet decomposition tree

Wavelet Analysis: Issues

- Principled design of wavelet filters for orthogonal and bi-orthogonal wavelets, tuned for special signals
- Fast algorithms for wavelet and wavelet packet analysis
- Non-dyadic wavelet decomposition
- Wavelets defined on non-Euclidean spaces (example: sphere)