Introduction

- **Short time Fourier transform**
  – Time-frequency representation
  – Two-dimensional space (matrix) from a one-dimensional function (vector)
  – Linear frequency distribution – the absolute bandwidths are the same

- **Wavelet transform**
  – Time-scale representation
  – Logarithmic frequency distribution – constant Q
  – The “butterfly” in the FFT is like the lowpass-highpass bank for wavelets
  [But the FFT admits all frequencies 0, 1, 2, 3, ... where the wavelets are dyadic and octave-based (1, 2, 4, 8, ...)!]

- **History**
  - First results by Alfred Haar in 1910
  - “French-school”, Morlet, Grossman and Meyer in the mid-eighties
  - Daubechies, Mallat

- **Application fields**
  - Compression (high-fidelity audio, ECG)
  - Identification (noise-removal, discontinuity detection, multiscale feature detection)
  - Telecommunication (transmultiplexers, discrete multitone modulation transceivers)
  - Process control (adaptive filtering/regulation)
  - Pattern-recognition of non-stationary signals

Wavelet Theory

The windowed Fourier transform:

$$F(\omega,\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(t)g(t-\tau)e^{i\omega t}dt$$

Reconstruction from STFT:

$$f(t) = \frac{1}{\|g\|}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\omega,\tau)g(t-\tau)e^{i\omega t}d\omega d\tau$$

For wavelet transform we have a *scale variable:*

$$w_{a,\tau}(t) = \frac{1}{\sqrt{|a|}}w\left(\frac{t-\tau}{a}\right)$$

Continuous Wavelet Transform **(CWT), or the Integral Wavelet Transform:**

$$F_W(a,\tau) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\, w\!\left(\frac{t-\tau}{a}\right) dt$$

Reconstruction:

$$f(t) = \frac{1}{C} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F_W(a,\tau) \frac{1}{\sqrt{|a|}}\, w\!\left(\frac{t-\tau}{a}\right) \frac{da\, d\tau}{a^2}$$

$$C = 2\pi \int |\hat{w}|^2\, d\omega \,/\, |\omega|.$$

Condition of invertability:

– *C* is finite if the transform is zero for ω=0
– i.e. the integral of the wavelet is zero (it has no DC component)
– The signal *f*(*t*) is supposed to be in $\mathbf{L}^2(\mathbf{R})$
– For finite-length sampled signals it is fulfilled.

Basis functions (mother wavelets)

– The compact support of *w*(*t*) is needed for good time localisation
– *w*(*t*) should be band limited in order to have good frequency localisation
– *w*(*t*) can be real or complex. In the latter case the complex conjugate of the basis function is used in the equations.

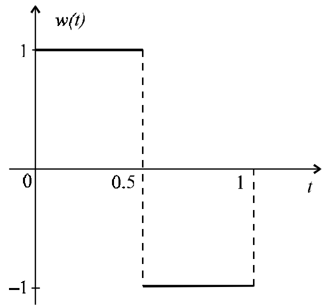
Some often used wavelet basis function with their Fourier transform

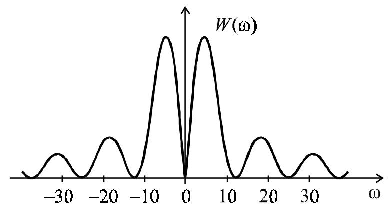
Daubechies (maxflat orthogonal with *2p* filter coefficients, used in identification)

Mallat (two-dimensional, used for image compression)

Haar

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

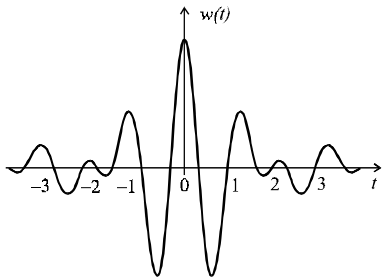


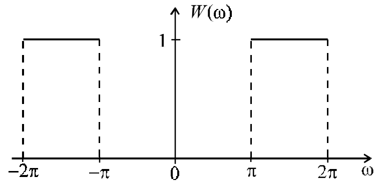


Shannon

$$w(t) = \frac{\sin(\pi t/2)}{\pi t/2}\cos\left(\frac{3\pi t}{2}\right)$$

$$W(\omega) = \begin{cases} 1 & \text{for } \pi \le |\omega| < 2\pi \\ 0 & \text{otherwise} \end{cases}$$

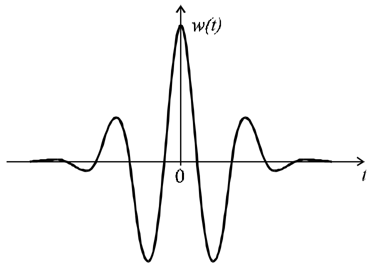


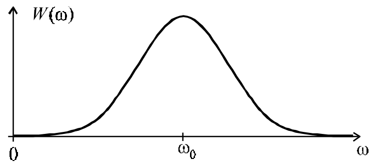


Modulated Gauss (Morlet)

$$w(t) = e^{i\omega_0 t} e^{-t^2/2}$$

$$W(\omega) = \sqrt{2\pi}\, e^{-\frac{(\omega-\omega_0)^2}{2}}$$

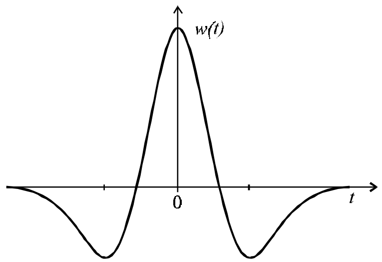


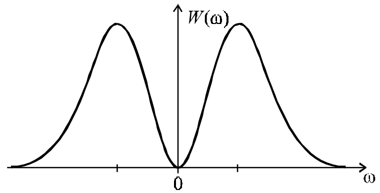


Gauss second derivative

$$w(t) = \left(1 - t^2\right) e^{-t^2/2}$$

$$W(\omega) = \sqrt{2\pi}\, \omega^2 e^{-\omega^2/2}$$



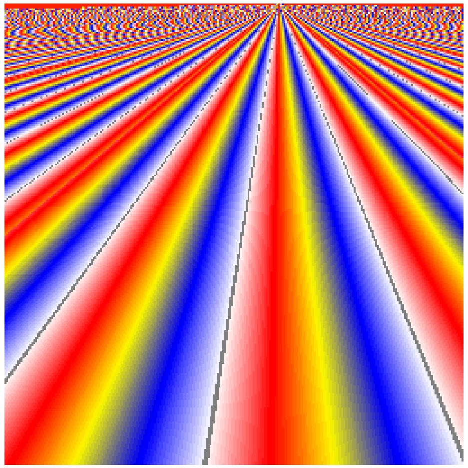


Example

Having a Dirac-delta at $t = t_0$, with modulated Gauss wavelet:

$$CWT(a,\tau) = \frac{1}{\sqrt{a}} w\left(\frac{t_0 - \tau}{a}\right) = \frac{1}{\sqrt{a}} e^{-j\omega_0\left(\frac{t_0-\tau}{a}\right)} e^{-\left(\frac{t_0-\tau}{a}\right)^2}$$

Its phase is $\theta = -\omega_0\left(\frac{t_0 - \tau}{a}\right)$:



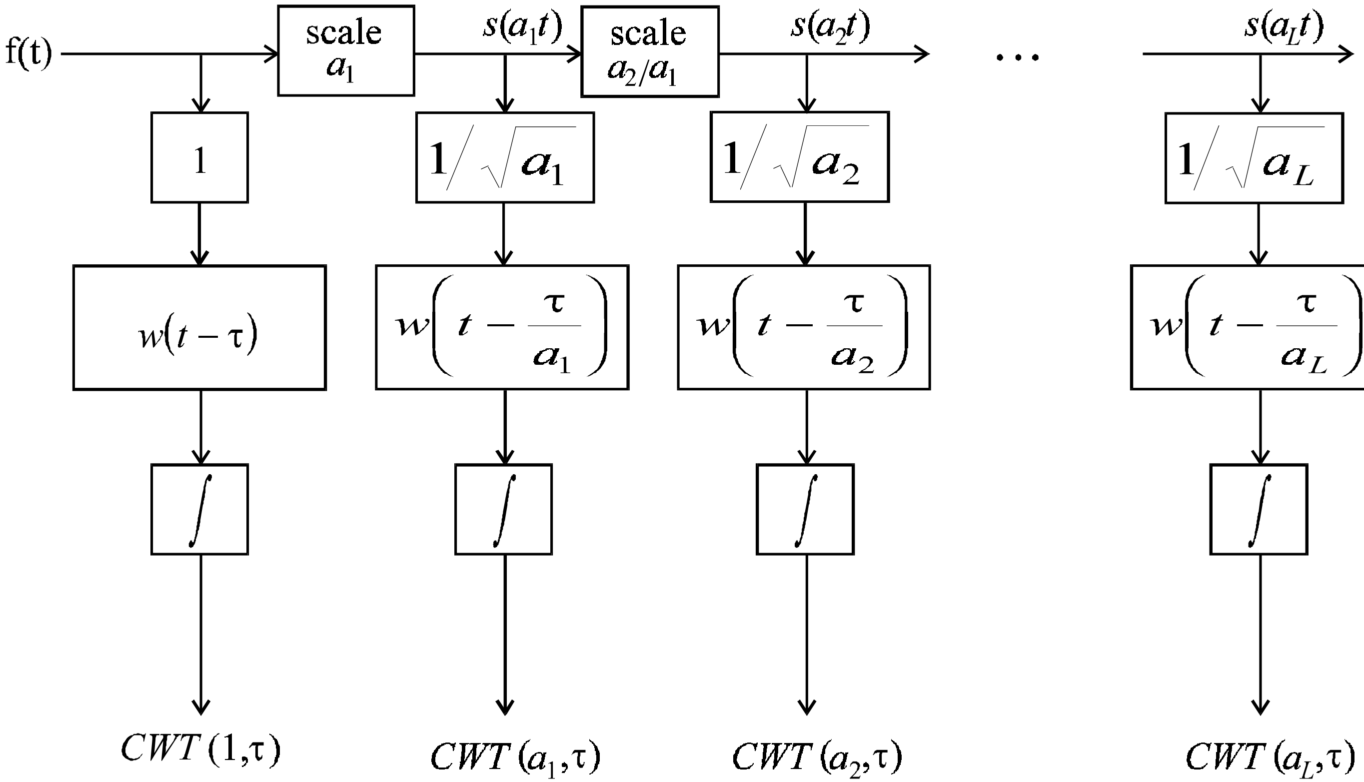
[Horizontal axis: time 0 to 1. Vertical axis: scale]

Implementations of the Wavelet Transform

There are four ways to view the integral operation (or the inner product) in the wavelet transform:

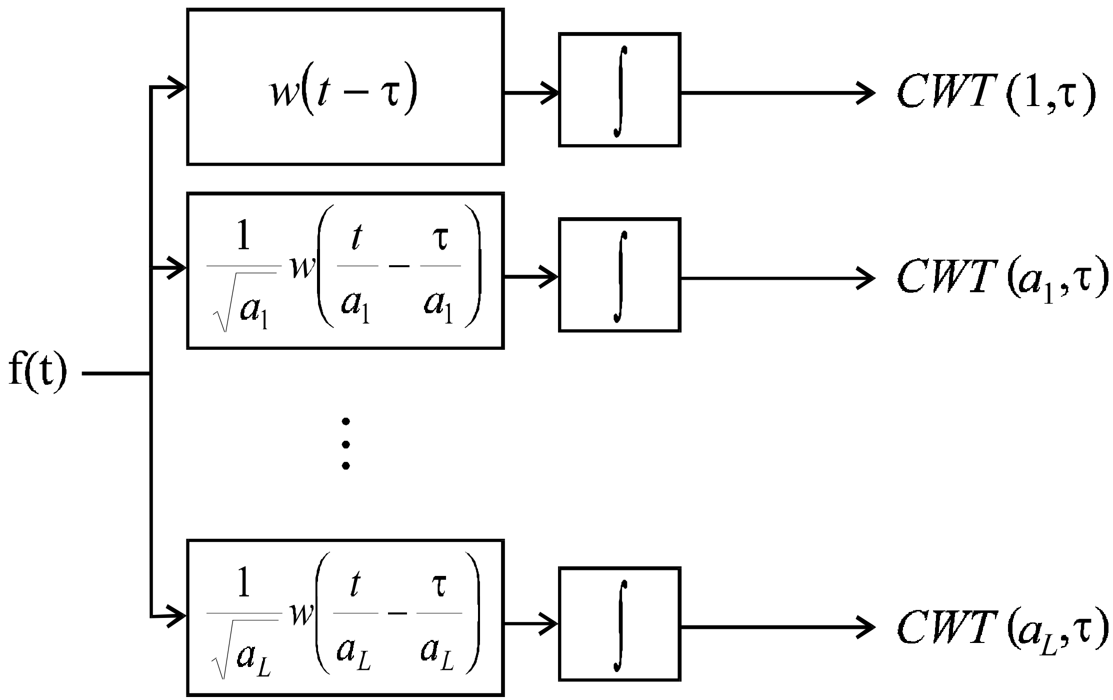
1. Cross-correlation of input signal *f*(*t*) with scaled wavelet $w(t/a)/\sqrt{a}$, at time shift τ/a. It therefore computes the "similarity" between them, or the component of the signal that is "common" to the wavelet.
2. Output of a bandpass filter of impulse response $w(-t/a)/\sqrt{a}$, of input *f*(*t*), at the instant τ/a.
3. Cross-correlation of a scaled signal *f*(*at*) with the wavelet $\sqrt{a}\, w(t)$, at time shift τ/a.
4. Output of a bandpass filter of impulse response $\sqrt{a}\, w(-t)$, of input *f*(*at*), at the instant τ/a.

The first realisation is simpler, assumed we can scale the signal *f*(*t*). In discrete time and decimation by 2 it is easily done — this is the base of discrete FWT.



Computing FWT by scaling the signal

The second one computes the same transform in a different way:



Filter bank implementation of the wavelet transform

The Discrete Wavelet Transform

Similarly to Fourier transform, wavelet transform also has different discrete versions:

Discrete parameter wavelet transform

$$DPWT(j,k) = a_0^{-j/2} \int_{-\infty}^{\infty} f(t) w\left(a_0^{-j} t - k\tau_0\right) dt$$

Discrete time wavelet transform

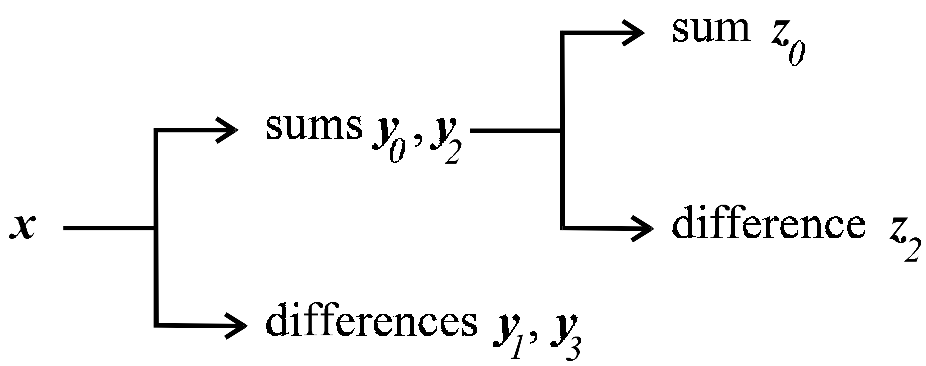
$$DTWT(j,k) = a_0^{-j/2} \sum_r f(r) w\left(a_0^{-j} r - k\tau_0\right)$$

Discrete wavelet transform

$$DWT(j,k) = 2^{-j/2} \sum_r f(r) w\left(2^{-j} r - k\right)$$

Multilevel transforms by recursion

– Sums and differences of neighbours are at the finest scale
– Larger scale: taking sums and differences again – recursion
– It leads to a **multiresolution** of the original signal



Multilevel by recursion

– This is the “logarithmic” or “dyadic” tree
– Walsh-Hadamard transform: all possible independent sums and differences, complete binary tree

Computational complexity

– Signal length $N=2^L$
– The Hadamard matrix has N^2 nonzeros
– The wavelet matrix has only $N + NL = N + N\log_2 N$

In case of recursive computations, we have a product of very sparse matrices

– The flow-graph gives $2NL = 2N\log_2 N$ for the complete Walsh-Hadamard transform (*exactly matching the FFT*)
– For wavelets the transform computes the sum and difference of $N-1$ pairs. This needs only $2N-2$ calls to memory.

Wavelets in Discrete Time

– Wavelets come from the **iteration** of filters (with rescaling)
– If we use the same filter coefficients, **relatively** it will have the same effect as if we halved filter cut-off frequencies at each level
– In each level, we separate the signal with a low- and a highpass filter after downsampling, thus we get the *logarithmic frequency distribution*

Connecting filters and wavelet functions

– The "*highpass filter*" leads to $w(t)$
– The "*lowpass filter*" leads to a scaling function $\phi(t)$
– In most cases the lowpass filter comes first — **the scaling function is obtained before the wavelet**.
– The scaling function (in continuous time) comes from infinite repetition **L L ... L** of the lowpass filter, with rescaling each iteration.
– The wavelet follows from $\phi(t)$ by just *one* application of the highpass filter.
– All functions $w_{jk}(t)$ (the so-called "baby-wavelets") are constructed from a single mother wavelet $w(t)$. This wavelet is a small wave (a pulse). Normally it starts at time $t=0$ and ends at time $t=N$.
– Scaling functions and wavelets inherit orthogonality, or biorthogonality, from the filter bank.
– There may or may not be a solution $\phi(t)$.
– The solution is zero outside the interval $0 \le t < N$.
– The solution seldom has an elementary formula.
– The solution is not likely to be a smooth function.

Continuous time	**Discrete time**
multiresolution	filter bank tree
rescaling $t \to 2t$	downsampling $\omega \to \frac{\omega}{2}$
averaging with $f(t)$	lowpass filter
averaging with $w(t)$	highpass filter
orthogonal bases	orthogonal matrices
wavelet coefficients	analysis bank output
sum of wavelet series	synthesis bank output
fast wavelet transform	product of filter matrices
compact support	FIR filters

The dilation equation:

$$\phi(t) = 2\sum_{k=0}^{N} \underline{\mathbf{h}}(k)\,\phi(2t-k)$$

The Wavelet Equation

$$w(t) = 2\sum \underline{\mathbf{h}}(k)\,\phi(2t-k)$$

Theorem: If the coefficients c(k) and d(k) come from an orthogonal filter bank, so they have double-shift orthogonality, then

a. The scaling functions $\phi(t-n)$ are orthonormal to each other:

$$\int_{-\infty}^{\infty} \phi(t-n)\phi(t-m)\,dt = \delta(m-n)$$

b. The scaling functions are orthogonal to the wavelets:

$$\int_{-\infty}^{\infty} \phi(t-m)w(t-n)\,dt = 0$$

c. The wavelets $w_{jk}(t) = 2^{j/2} w\left(2^j t - k\right)$ at all scales are orthonormal:

$$\int_{-\infty}^{\infty} w_{jk}(t) w_{JK}(t)\,dt = \delta(j-J)\delta(k-K)$$

Synthesis

$$a_{0k} = \sum \underline{\mathbf{c}}(l - 2k) a_{1l} \quad \textit{and} \quad b_{0k} = \sum \underline{\mathbf{d}}(l - 2k) a_{1l}$$

$$\begin{tikzcd}
a_{j+1} \arrow[r, "C^T"] \arrow[dr, "D^T"'] & a_j \arrow[r, "C^T"] \arrow[dr, "D^T"'] & a_{j-1} & \cdots & a_1 \arrow[r, "C^T"] \arrow[dr, "D^T"'] & a_0 \\
& b_j & b_{j-1} & & & b_0
\end{tikzcd}$$

Synthesis filter bank:

$$a_{j+1,l} = \sum \mathbf{c}(2k - l) a_{jk} + \sum \mathbf{d}(2k - l) b_{jk}$$

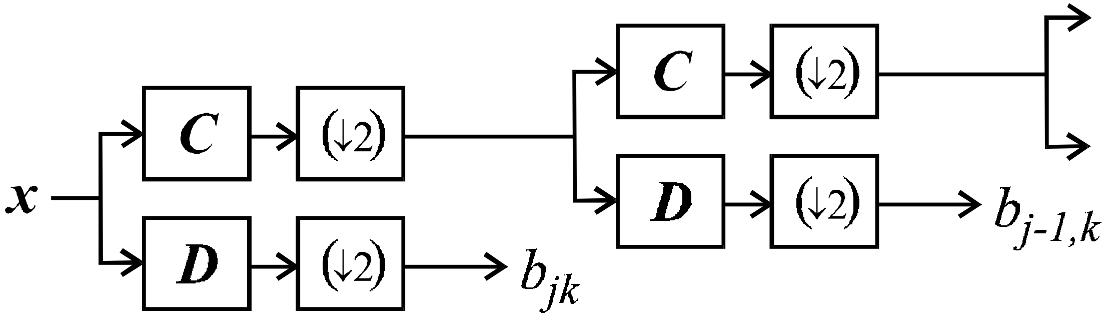
The inverse pyramid is the **fast inverse wavelet transform**:

$$\begin{tikzcd}
a_0 \arrow[r, "C"] & a_1 \arrow[r, "C"] & a_2 & \cdots & a_j \arrow[r, "C"] & a_{j+1} \\
b_0 \arrow[ur, "D"'] & b_1 \arrow[ur, "D"'] & & & b_j \arrow[ur, "D"'] &
\end{tikzcd}$$

Convergence of $\phi^{(i)}(t)$ to $\phi(t)$ needs $H = 0$ at $z = -1$

Orthogonality of $\phi(t-k)$ needs $H(z)H(z^{-1})$ to be halfband.

Tree-Structured Filter Bank – Fast Wavelet Transform (FWT)



Analysis tree – in matrix form of the pyramid algorithm:

$$\underline{\underline{\mathbf{A}}} = \left[\begin{array}{c|c} \underline{\underline{\mathbf{L}}} & \\ \underline{\underline{\mathbf{B}}} & \\ \hline & \underline{\underline{\mathbf{I}}} \end{array}\right] \begin{bmatrix} \underline{\underline{\mathbf{L}}} \\ \underline{\underline{\mathbf{B}}} \end{bmatrix}$$

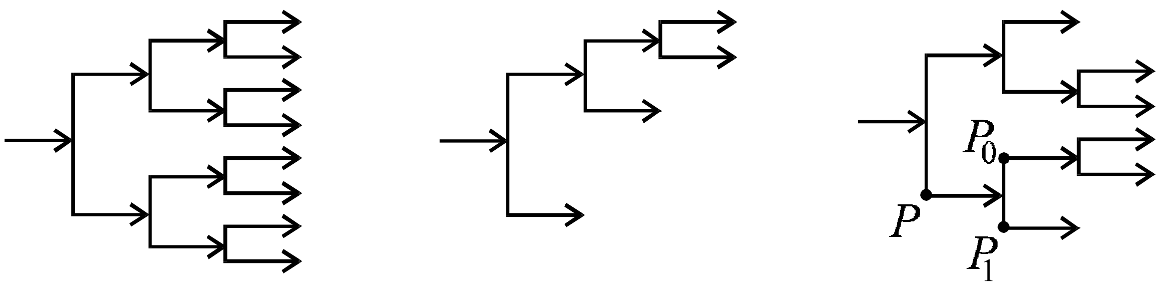
The total computational need for the FWT:

$$TL\left(1+\frac{1}{2}+\frac{1}{4}+\cdots+\frac{1}{2^{J-1}}\right)=TL\sum_{i=0}^{J-1}\frac{1}{2^i}<2TL$$

Wavelet Packets and the Best Basis

– A wavelet packet basis allows any dyadic tree structure
– At each point in the tree the signal through the lowpass-highpass filter bank, **or not**.

Complete tree (Walsh) | Wavelet tree (Haar) | Wavelet Packet tree



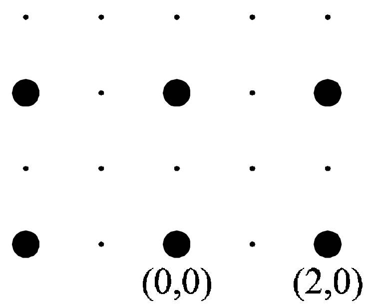
Each wavelet packet tree yields a basis, including Walsh and Haar

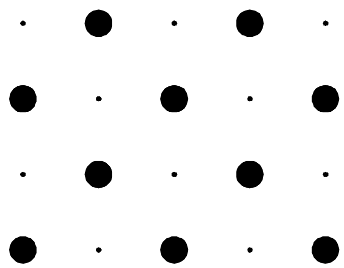
Multidimensional Filters and Wavelets

– *Separable*: Products of one-dimensional filters (easy-to-design).
– *Non-separable*: Genuinely two-dimensional filters (hard to design, especially if we want orthogonality).

(separable) $\underline{\underline{\mathbf{M}}}_s = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ (non-separable quincunx) $\underline{\underline{\mathbf{M}}}_q = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$,

n_1 even and n_2 even



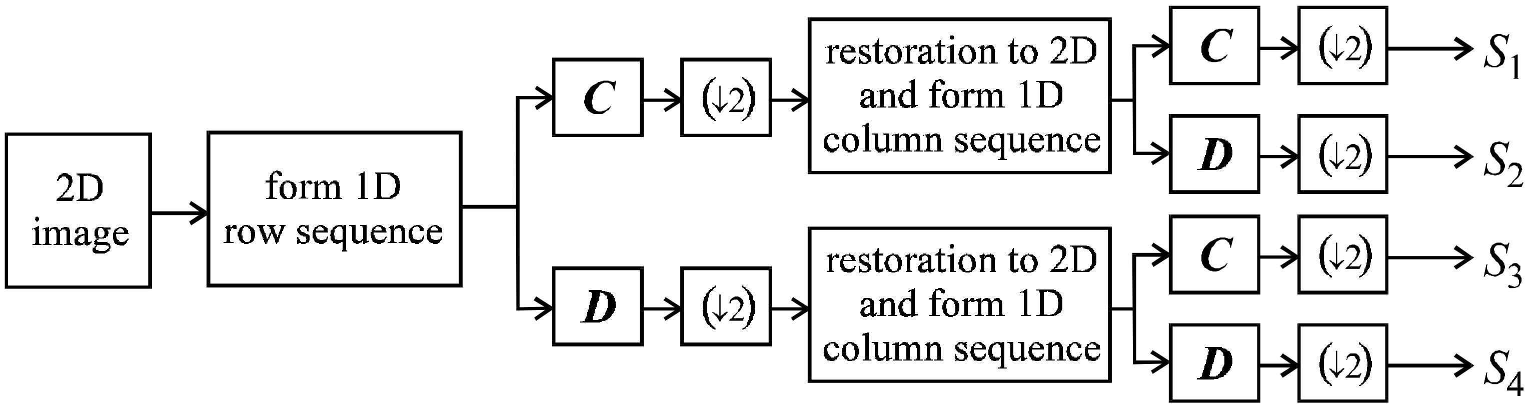


n_1+n_2 even

The sampling lattices for separable and quincunx

Separable: strongly oriented in the horizontal and vertical directions

Quincunx: extra symmetry at 45º and –45º. It is closer to isotrophic. A diagonal edge in an image will be captured far better by the quincunx.



A possible realisation of two-dimension MRA