Supplementary

The Wavelet Transform

Jean Baptiste Joseph Fourier (1768 – 1830)

MGP (Mathematics Genealogy Project): Leibniz - Bernoulli - Bernoulli - Euler - Lagrange - Fourier – Dirichlet - ….

1787: Train for priest (Left but Never married!!!).

1793: Involved in the local Revolutionary Committee.

1974: Jailed for the first time.

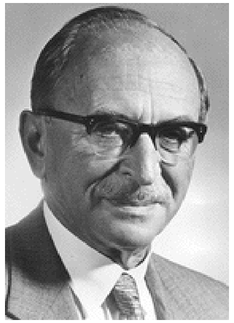
1797: Succeeded Lagrange as chair of analysis and mechanics at École Polytechnique.

1798: Joined Napoleon's army in its invasion of Egypt.

1804-1807: Political Appointment. Work on Heat. **Expansion of functions as trigonometrical series.** Objections made by Lagrange and Laplace.

1817: Elected to the Académie des Sciences in and served as secretary to the mathematical section. Published his prize winning essay *Théorie analytique de la chaleur*.

1824: Credited with the discovery that gases in the atmosphere might increase the surface temperature of the Earth (sur les températures du globe terrestre et des espaces planétaires). He established the concept of planetary energy balance. Fourier called infrared radiation "chaleur obscure" or "dark heat“.



Dennis Gabor

Windowed (Short-Time) Fourier Transform (1946)

Winner of the 1971 Nobel Prize for contributions to the principles underlying the science of holography, published his now-famous paper “Theory of Communication.”2

James W. Cooley and John W. Tukey

Fast Fourier Transform

James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.* **19**, 297–301 (1965).

Independently re-invented an algorithm known to Carl Friedrich Gauss around 1805

C. F. Gauss

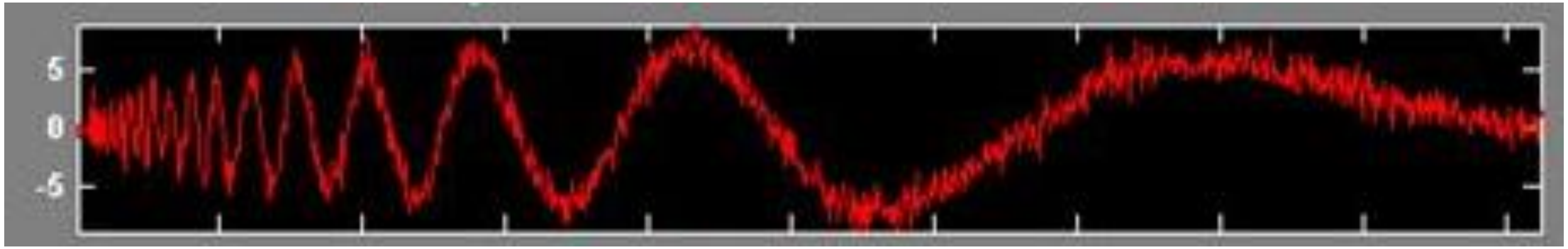
Jean Morlet

Presented the concept of wavelets (ondelettes) in its present theoretical form when he was working at the Marseille Theoretical Physics Center (France). (Continuous Wavelet Transform)

Stephane Mallat, Yves Meyer

(Discrete Wavelet Transform) The main algorithm dates back to the work of Stephane Mallat in 1988. Then joined Y. Meyer.

Motivation

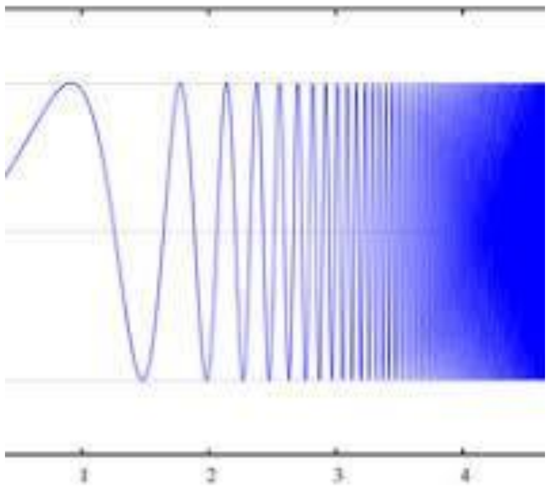


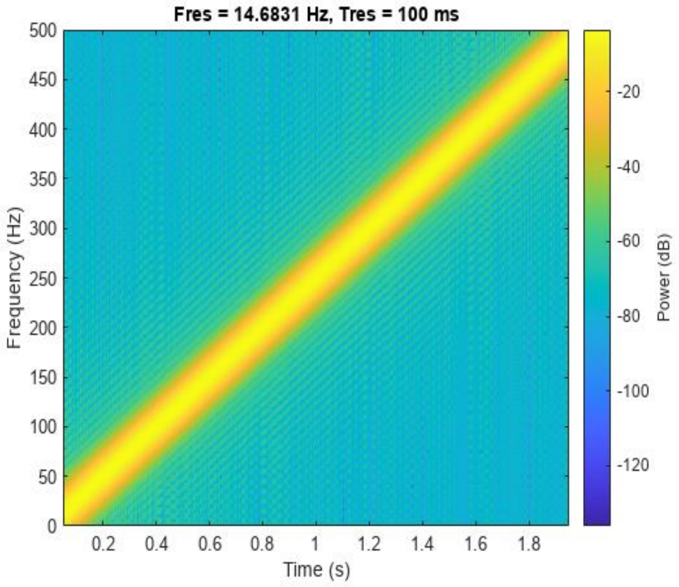
Some signals obviously have spectral characteristics that vary with time

STATIONARITY OF SIGNAL

- Stationary Signal
  - Signals with frequency content unchanged in time
  - All frequency components exist at all times

- Non-stationary Signal
  - Frequency changes in time
  - One example: the “Chirp Signal”

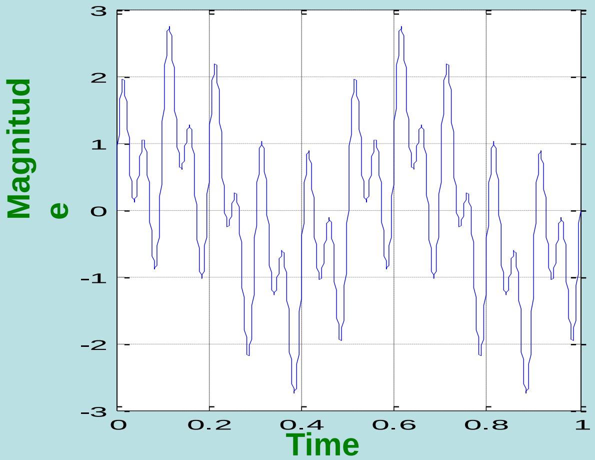


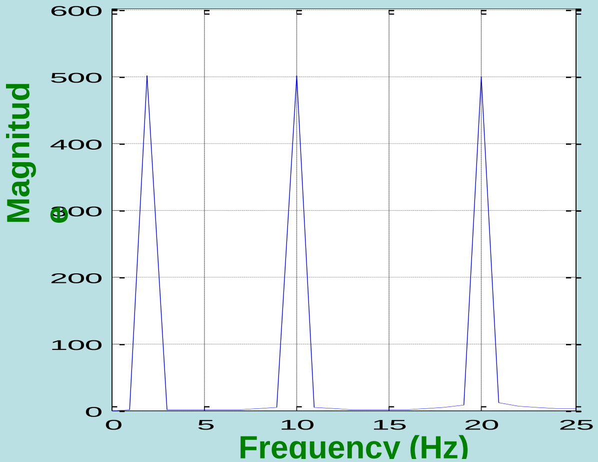


STATIONARITY OF SIGNAL

2 Hz + 10 Hz + 20Hz

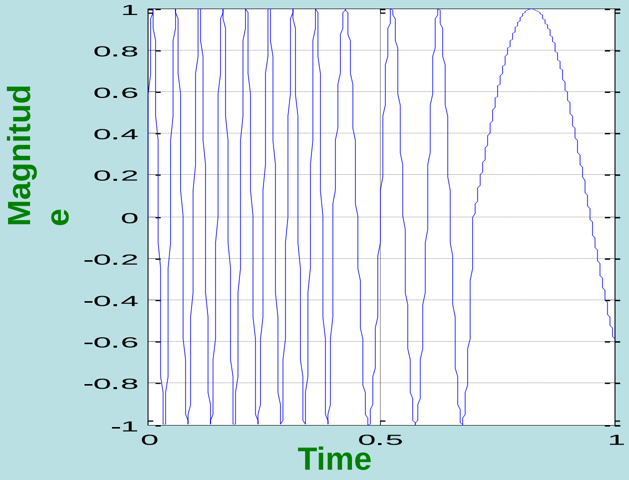
Stationary

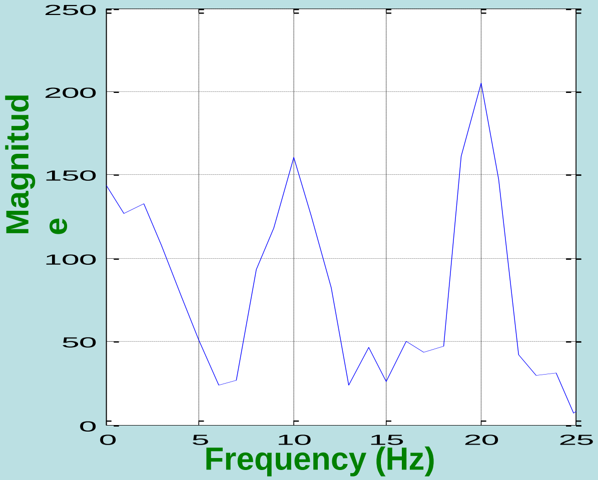




0.0-0.4: 20 Hz +
0.4-0.7: 10 Hz +
0.7-1.0: 2 Hz

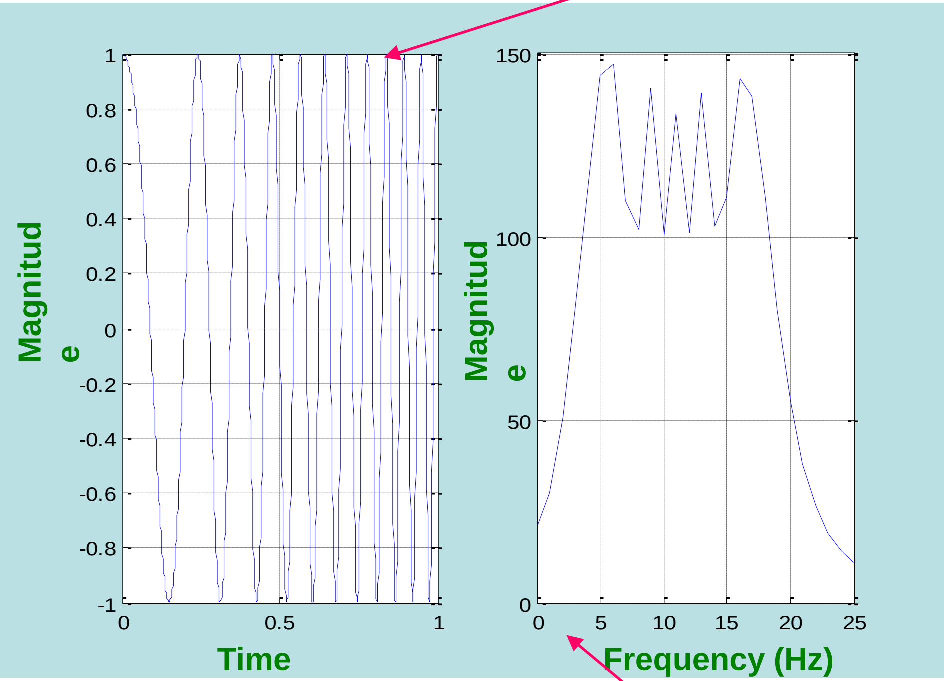
Non-Stationary





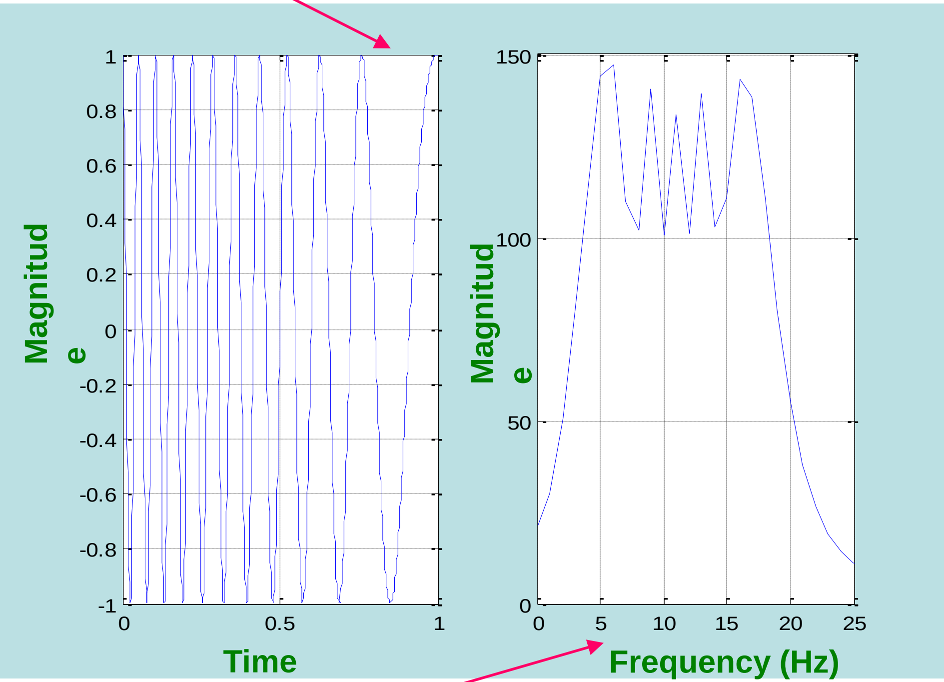
CHIRP SIGNALS

Frequency: 2 Hz to 20 Hz



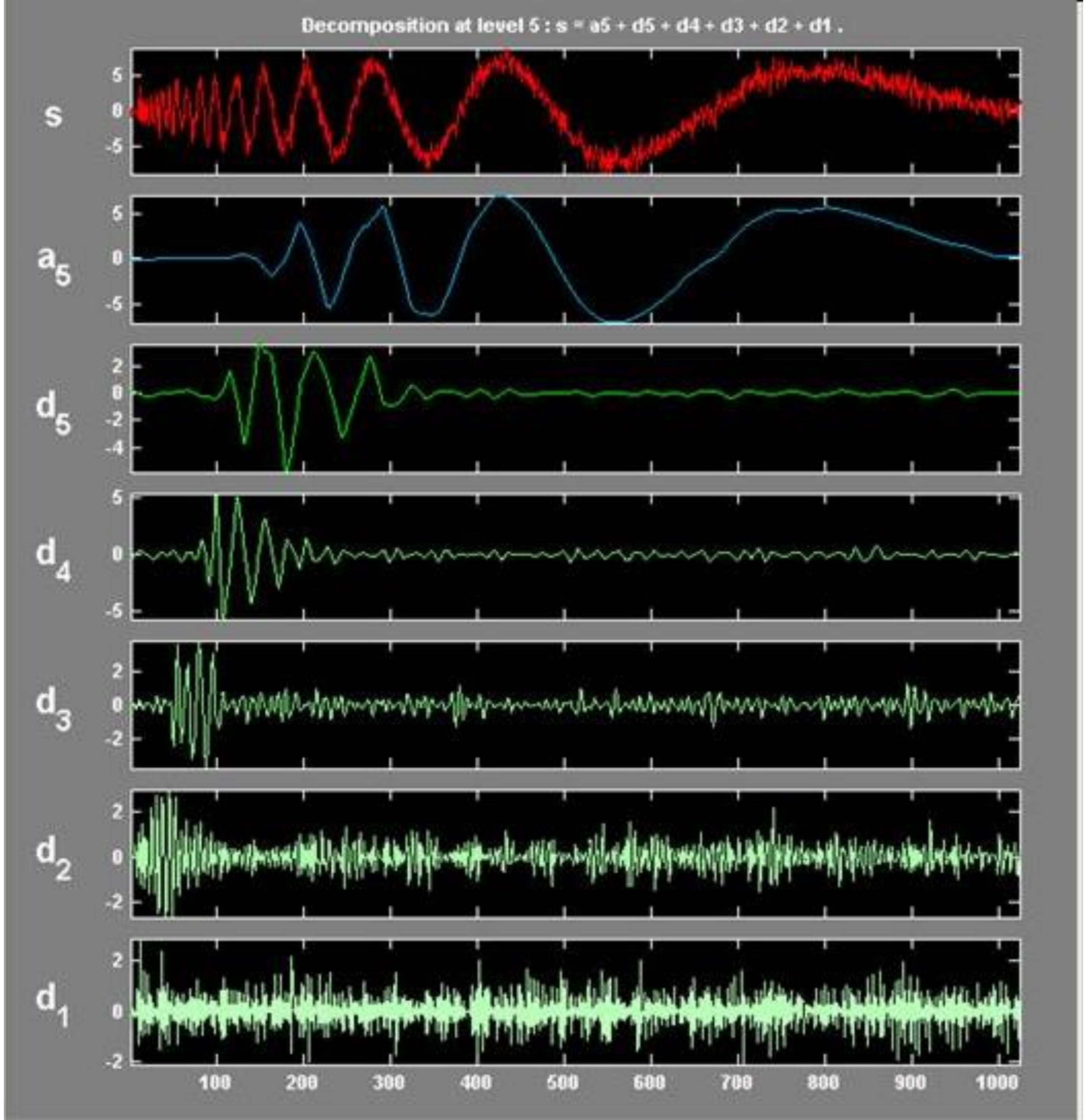
Different in Time Domain

Frequency: 20 Hz to 2 Hz



Same in Frequency Domain

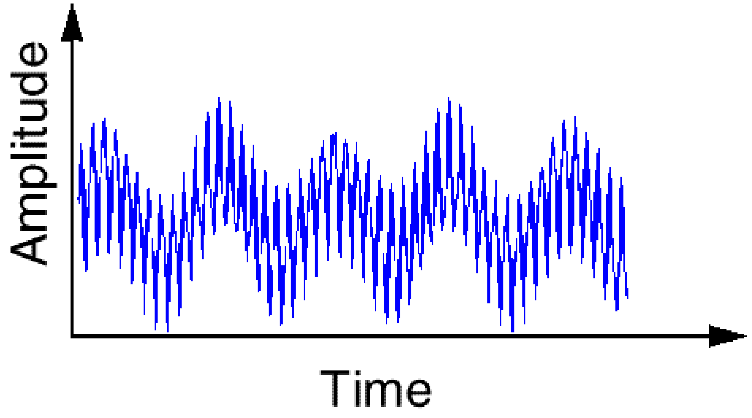
At what time the frequency components occur? FT can not tell!

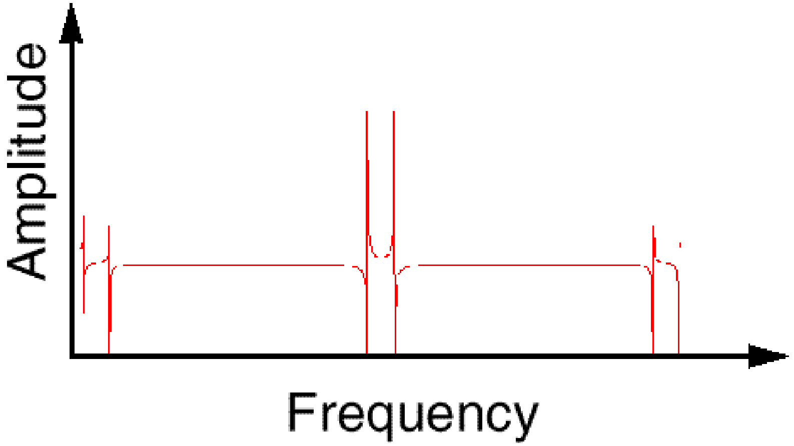


Decomposition at level 5 : s = a5 + d5 + d4 + d3 + d2 + d1 .
s
a5
d5
d4
d3
d2
d1
100 200 300 400 500 600 700 800 900 1000

Fourier Analysis

- Breaks down a signal into constituent sinusoids of different frequencies





In other words: Transform the view of the signal from time-base to frequency-base.

The Fourier Transform (FT)

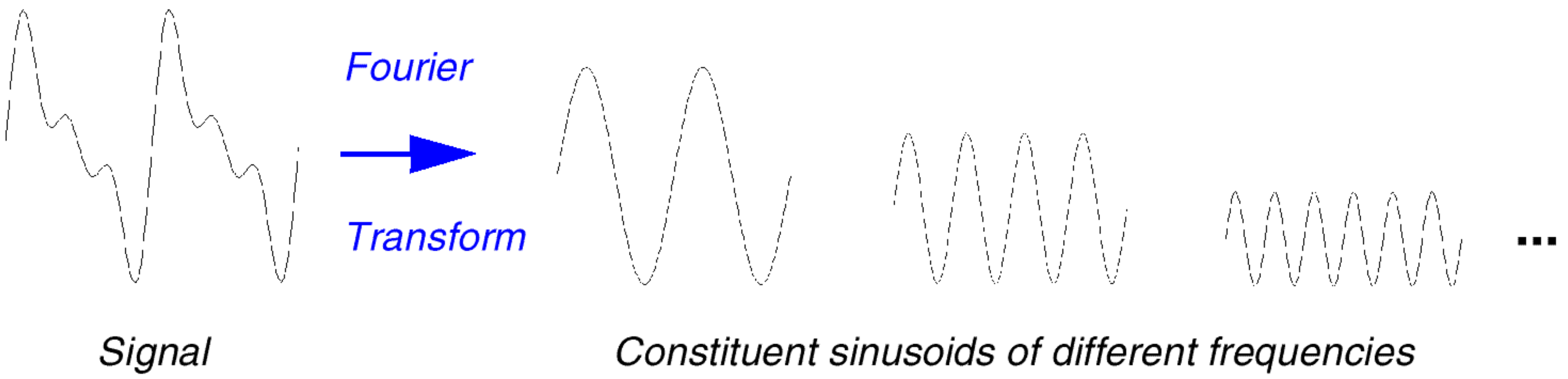
- A mathematical representation of the Fourier transform:

$$F(w) = \int_{-\infty}^{\infty} f(t) e^{-iwt} dt$$

- Meaning: the sum over all time of the signal *f(t)* multiplied by a complex exponential, and the result is the Fourier coefficients $F(\omega)$.

Fourier Transform

- Those coefficients, when multiplied by a sinusoid of appropriate frequency ω, yield the constituent sinusoidal component of the original signal:

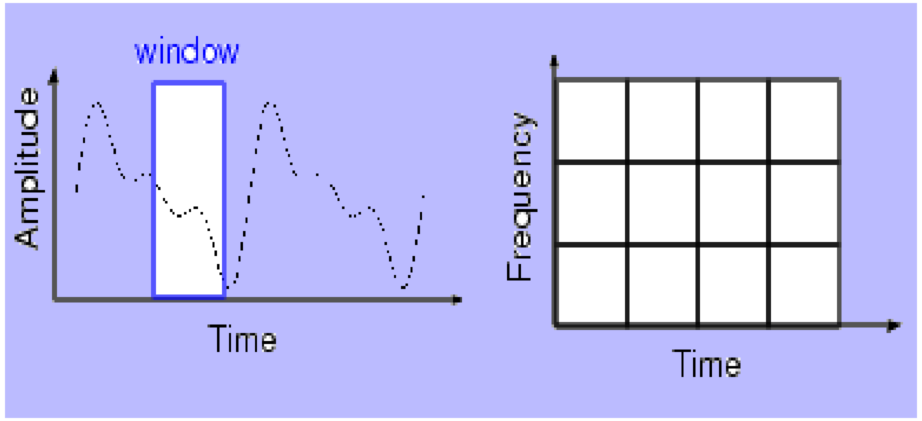


What's wrong with Fourier?

- By using Fourier Transform , we lose the time information : WHEN did a particular event take place ?
- FT can not locate drift, trends, abrupt changes, beginning and ends of events, etc.
- Calculating use complex numbers.

Short Time Fourier Transform

- Dennis Gabor (1946) Used STFT
  - To analyze only a small section of the signal at a time -- a technique called *Windowing the Signal*.
- The Segment of Signal is Assumed *Stationary*
- A 3D transform



$$\mathrm{STFT}_X^{(\omega)}(t', f) = \int_t \left[x(t) \bullet \omega^*(t - t')\right] \bullet e^{-j2\pi ft} dt$$

$\omega(t)$: the window function

A function of time and frequency

STFT (or: Gabor Transform)

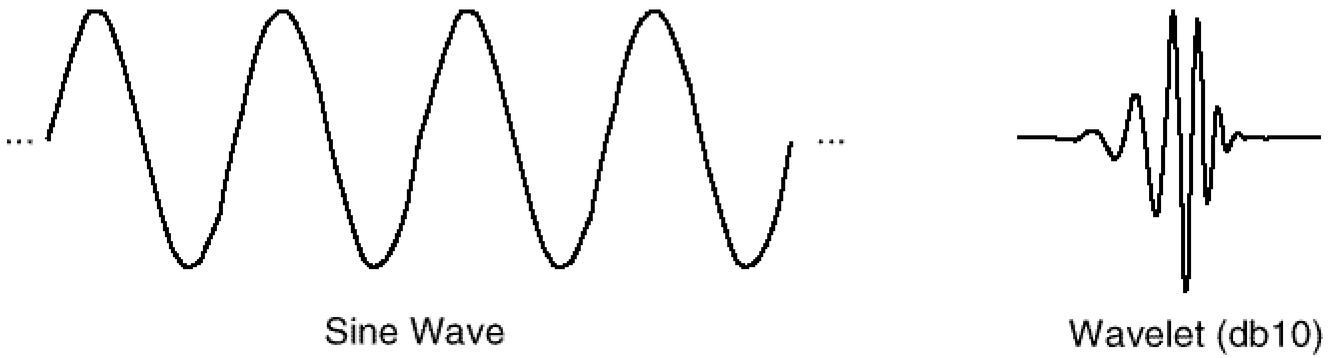
- A compromise between time-based and frequency-based views of a signal.
- both time and frequency are represented in limited precision.
- The precision is determined by the size of the window.
- Once you choose a particular size for the time window - it will be the same for all frequencies.

What's wrong with Gabor?

- Many signals require a more flexible approach - so we can vary the window size to determine more accurately either time or frequency.

What is Wavelet Analysis ?

- And…what is a wavelet…?



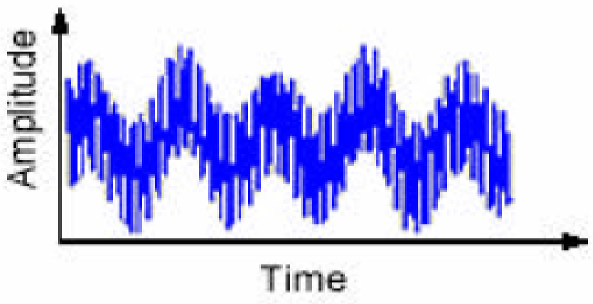
- A wavelet is a waveform of effectively limited duration that has an average value of zero.

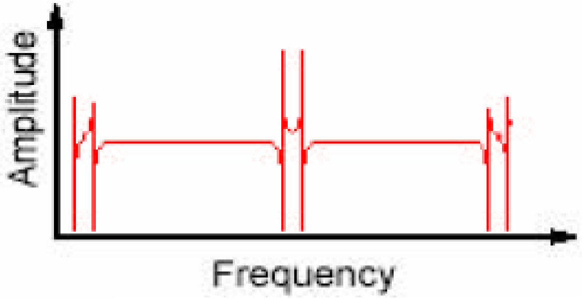
Wavelet's properties

- Short time localized waves with zero integral value.
- Possibility of time shifting.
- Flexibility.

Fourier vs. Wavelet

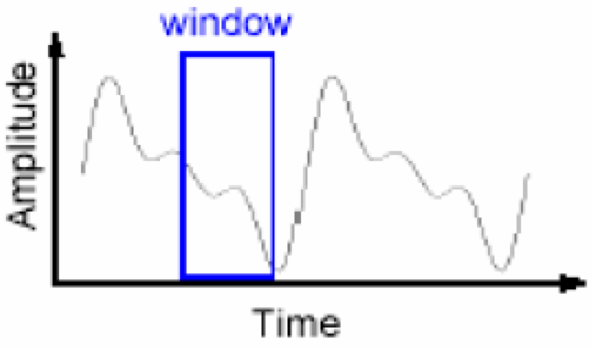
- FFT, basis functions: sinusoids
- Wavelet transforms: small waves, called wavelet
- FFT can only offer frequency information
- Wavelet: frequency + temporal information
- Fourier analysis doesn’t work well on discontinuous, “bursty” data
  - music, video, power, earthquakes,…

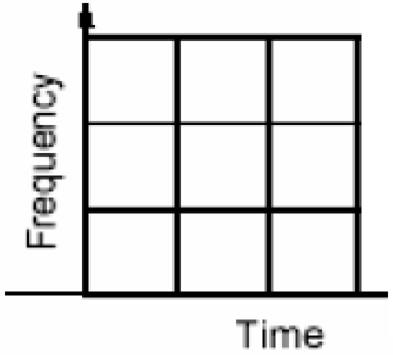




frequency

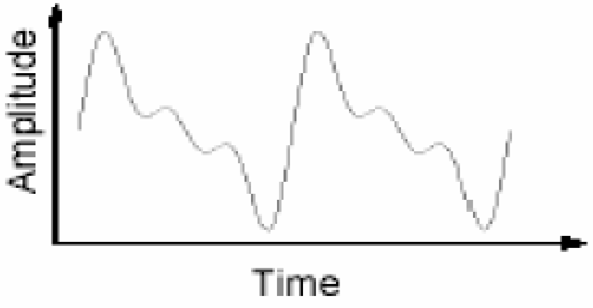
$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} df$$

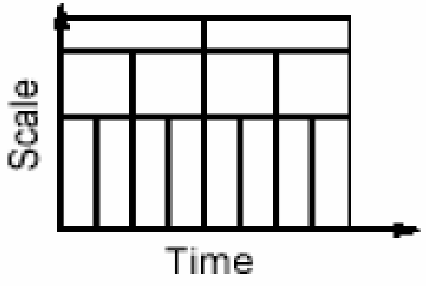




frequency + time (equal time intervals)

$$STFT(\tau, f) = \int_{-\infty}^{+\infty} x(t) \overline{w(t-\tau)} e^{-j2\pi ft} dt$$





Wavelet Analysis

frequency + time

Scale

- Scale
  - S>1: dilate the signal
  - S<1: compress the signal
- Low Frequency -> High Scale -> Non-detailed Global View of Signal -> Span Entire Signal
- High Frequency -> Low Scale -> Detailed View Last in Short Time
- Only Limited Interval of Scales is Necessary

Comparison of resolution

- Windowed Fourier Transform

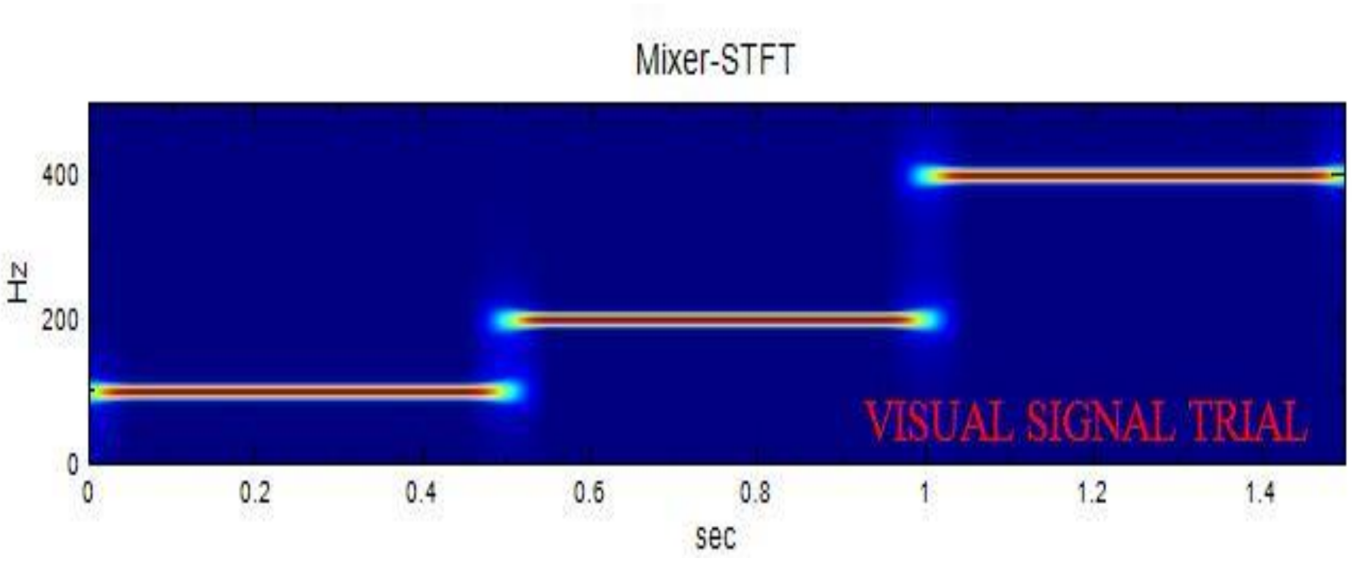


Fig. 18 the result using Windowed Fourier Transform

Comparison of resolution

- Discrete Wavelet Transform

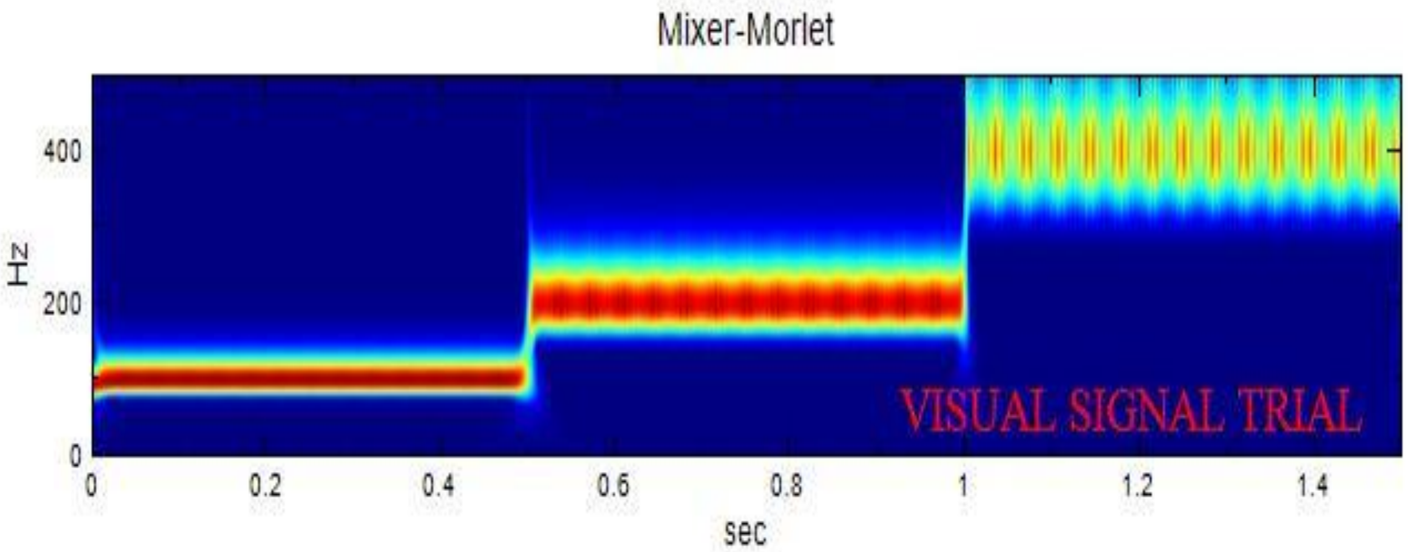
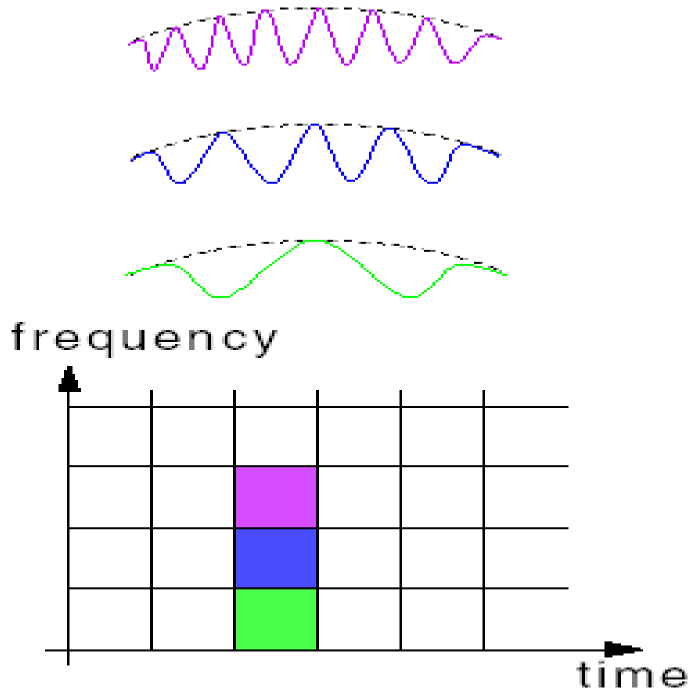
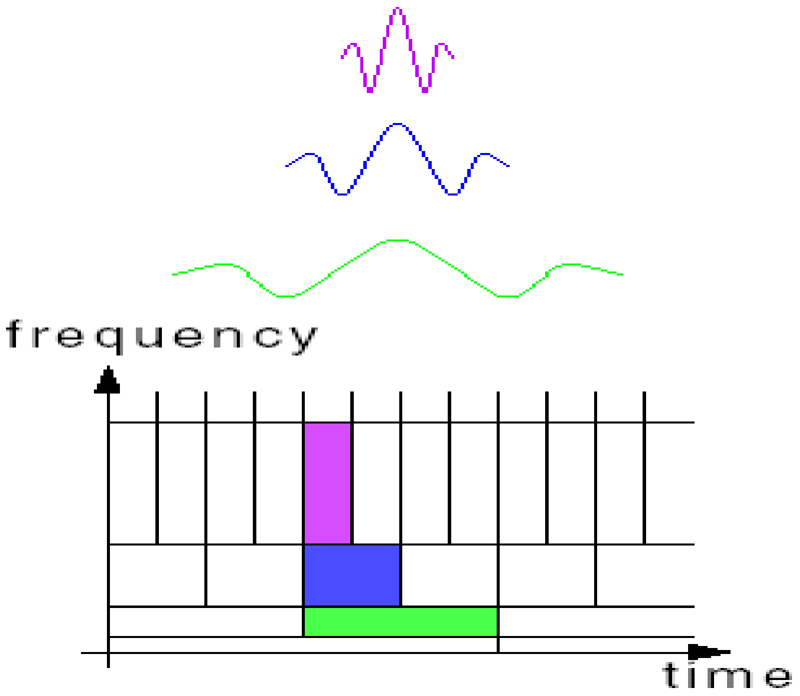


Fig. 19 the result using Discrete Wavelet Transform

STFT and Wavelets



short-time Fourier transform



wavelet transform

What is wavelet transform?

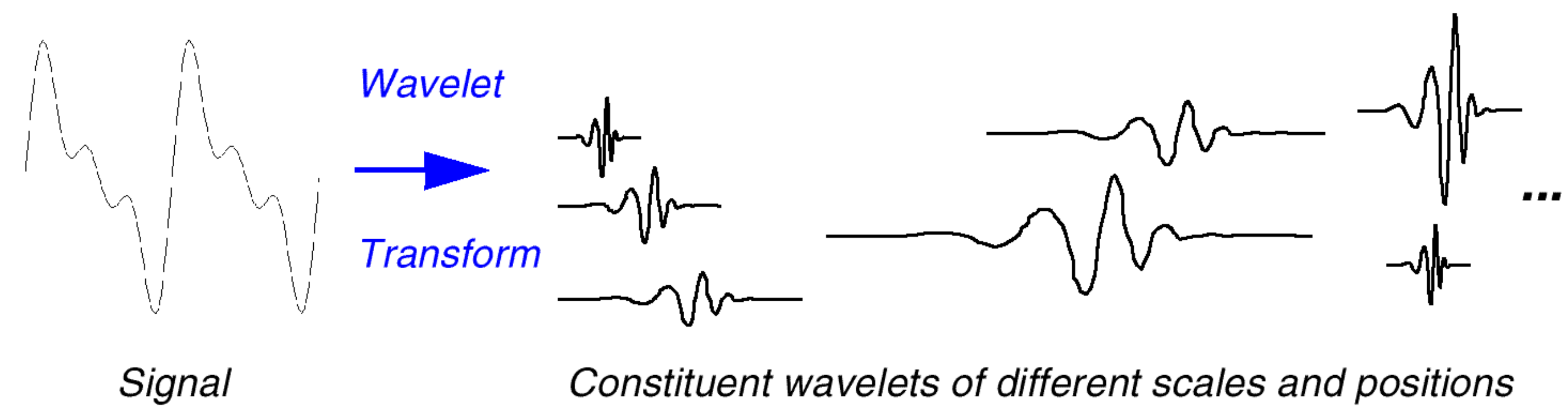
- Provides time-frequency representation
- Wavelet transform decomposes a signal into a set of basis functions (wavelets)
- Wavelets are obtained from a single prototype wavelet Ψ(t) called mother wavelet by dilations and shifting:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$$

- where a is the scaling parameter and b is the shifting parameter

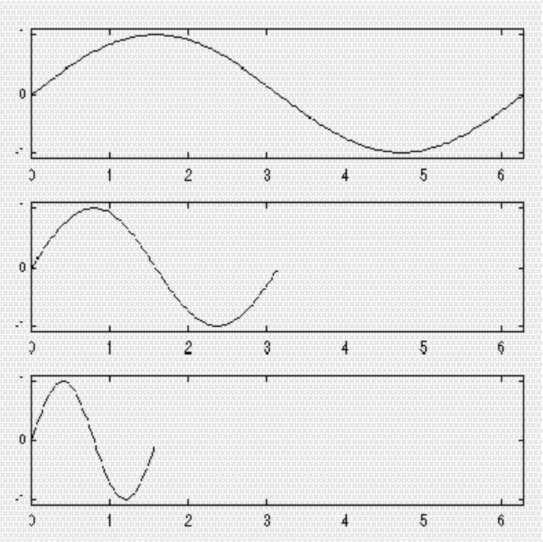
Wavelet Transform

- The result of the Continuous WT are Wavelet coefficients .
- Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelet of the original signal:



Scaling

- Wavelet analysis produces a time-scale view of the signal.
- *Scaling* means stretching or compressing of the signal.
- scale factor (a) for sine waves:



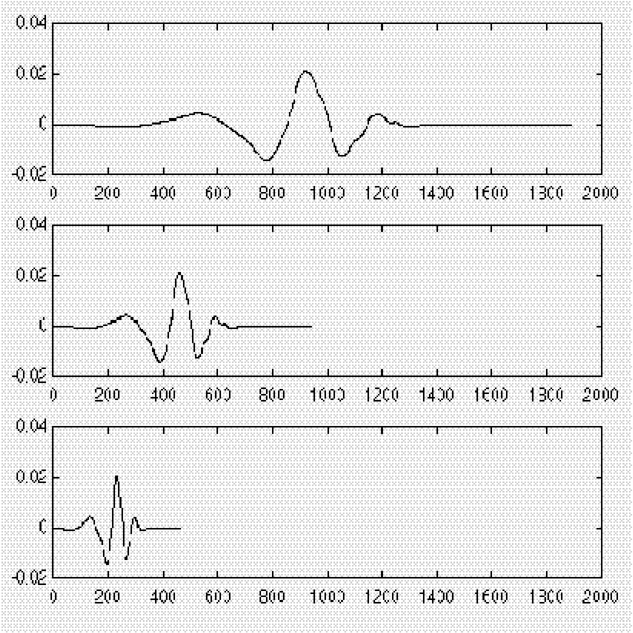
$$f(t) = \sin(t) \quad ; \quad a = 1$$

$$f(t) = \sin(2t) \quad ; \quad a = \tfrac{1}{2}$$

$$f(t) = \sin(4t) \quad ; \quad a = \tfrac{1}{4}$$

Scaling (Cont'd)

- Scale factor works exactly the same with wavelets:



$$f(t) = \Psi(t) \;\; ; \;\; a = 1$$

$$f(t) = \Psi(2t) \;\; ; \;\; a = \frac{1}{2}$$

$$f(t) = \Psi(4t) \;\; ; \;\; a = \frac{1}{4}$$

Wavelet function

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{a}}\Psi\left(\frac{x-b}{a}\right)$$

- b – shift coefficient
- a – scale coefficient

$$\Psi_{a,b_x,b_y}(x,y) = \frac{1}{|a|}\Psi\left(\frac{x-b_x}{a}, \frac{y-b_y}{a}\right)$$

- 2D function

Wavelet Transform

- Continuous Wavelet Transform (CWT)
- Discrete Wavelet Transform (DWT)

Basis Functions Using Wavelets

- Like sin( ) and cos( ) functions in the Fourier Transform, wavelets can define a set of basis functions $\psi_k(t)$:

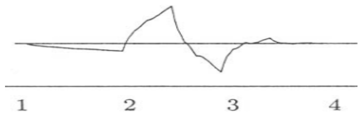
$$f(t) = \sum_k a_k \psi_k(t)$$

- Span of $\psi_k(t)$: vector space S containing all functions f(t) that can be represented by $\psi_k(t)$.

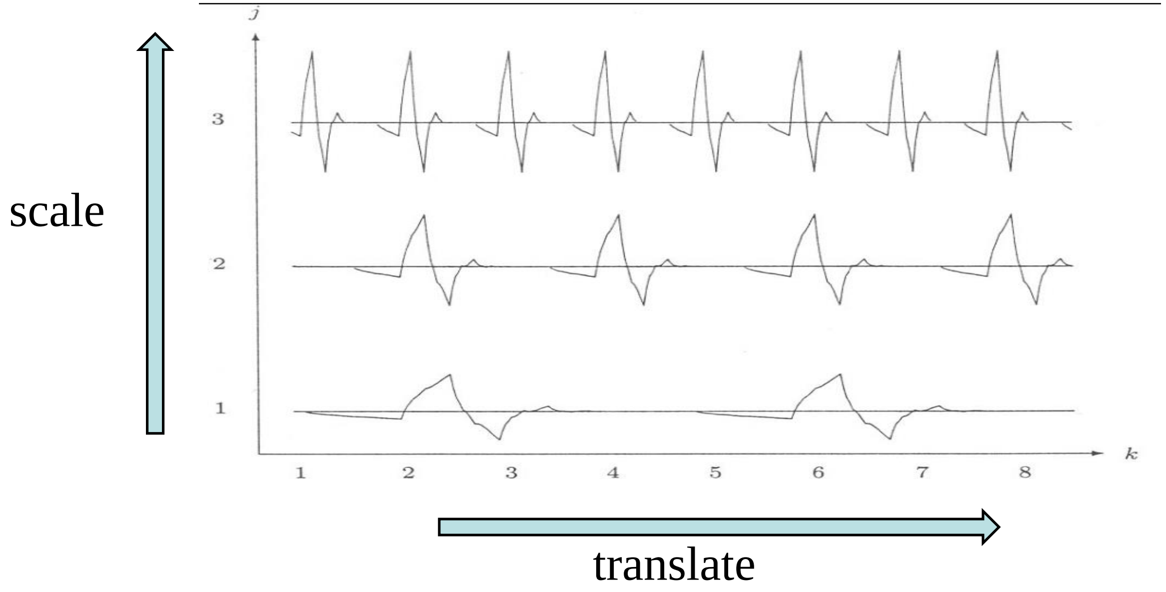
Basis Construction – “Mother” Wavelet

The basis can be constructed by applying translation and scaling (stretch/compress) on the “mother” wavelet ψ(t):

Example:



ψ(t)



Continuous Wavelet Transform (CWT)

translation parameter (measure of time)

scale parameter (measure of frequency)

scale $=1/2^j$ (1/frequency)

Forward CWT:

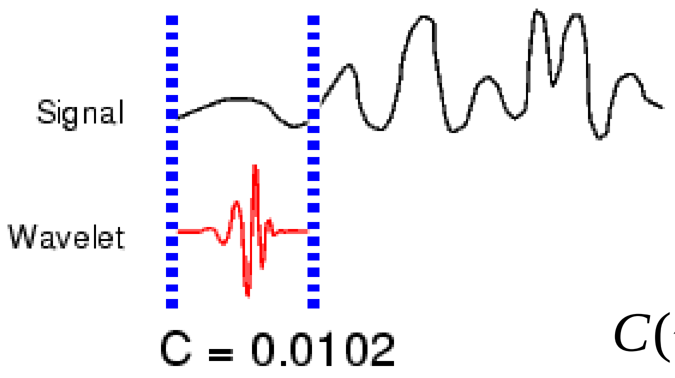
$$C(\tau, s) = \frac{1}{\sqrt{s}} \int_t f(t)\, \psi^*\left(\frac{t-\tau}{s}\right) dt$$

normalization constant

mother wavelet (i.e., window function)

Five Easy Steps to a Continuous Wavelet Transform

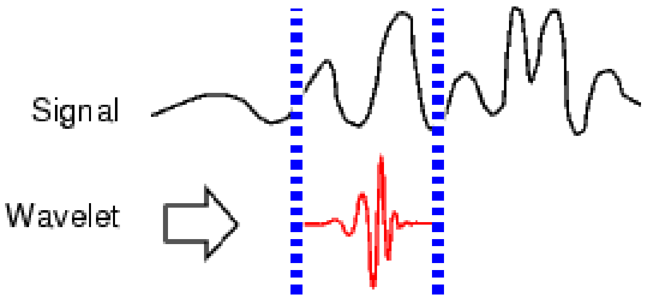
1. Take a wavelet and compare it to a section at the start of the original signal.
2. Calculate a correlation coefficient c



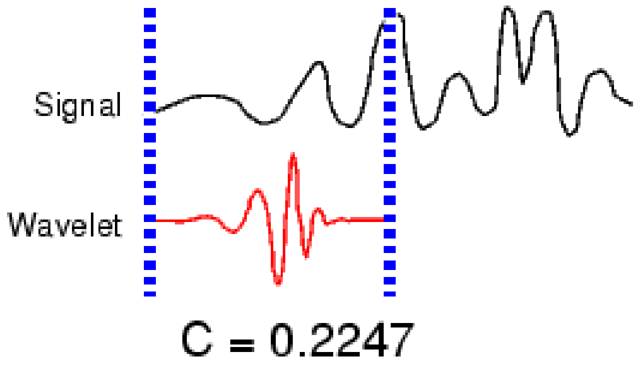
$$C(\tau, s) = \frac{1}{\sqrt{s}} \int_t f(t) \psi^* \left(\frac{t - \tau}{s} \right) dt$$

Five Easy Steps to a Continuous Wavelet Transform

3. Shift the wavelet to the right and repeat steps 1 and 2 until you've covered the whole signal.

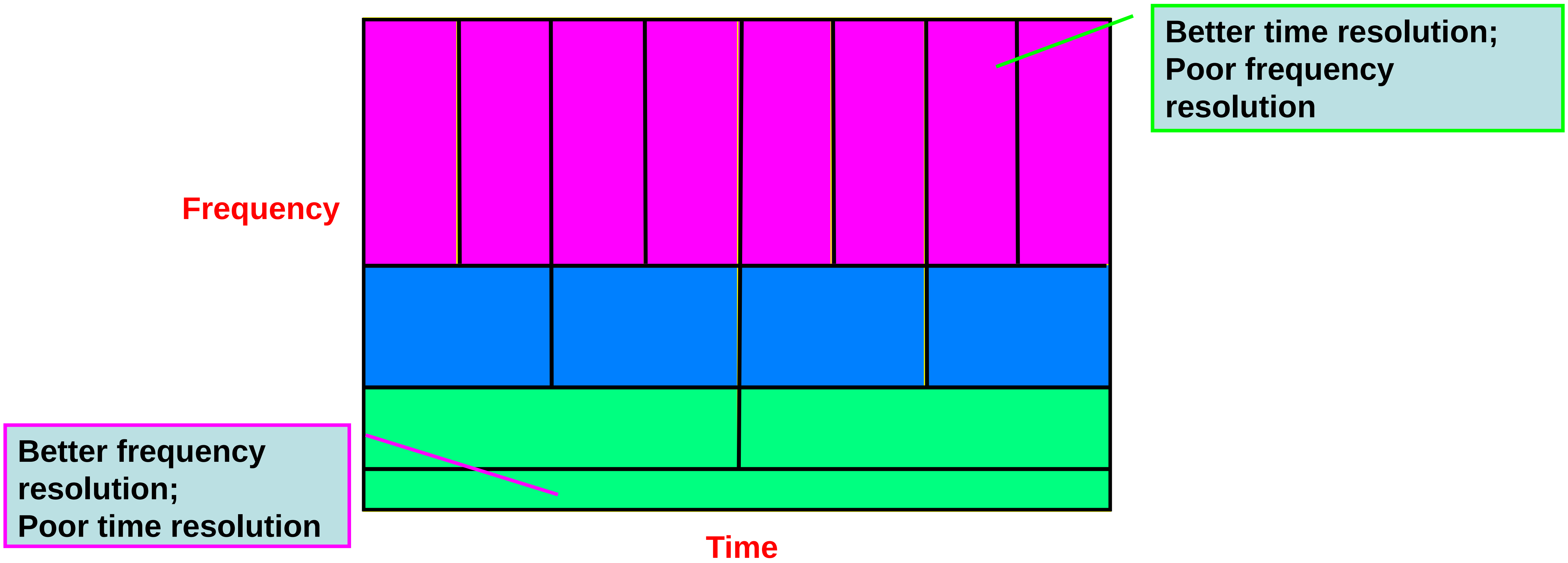


4. Scale (stretch) the wavelet and repeat steps 1 through 3.



5. Repeat steps 1 through 4 for all scales.

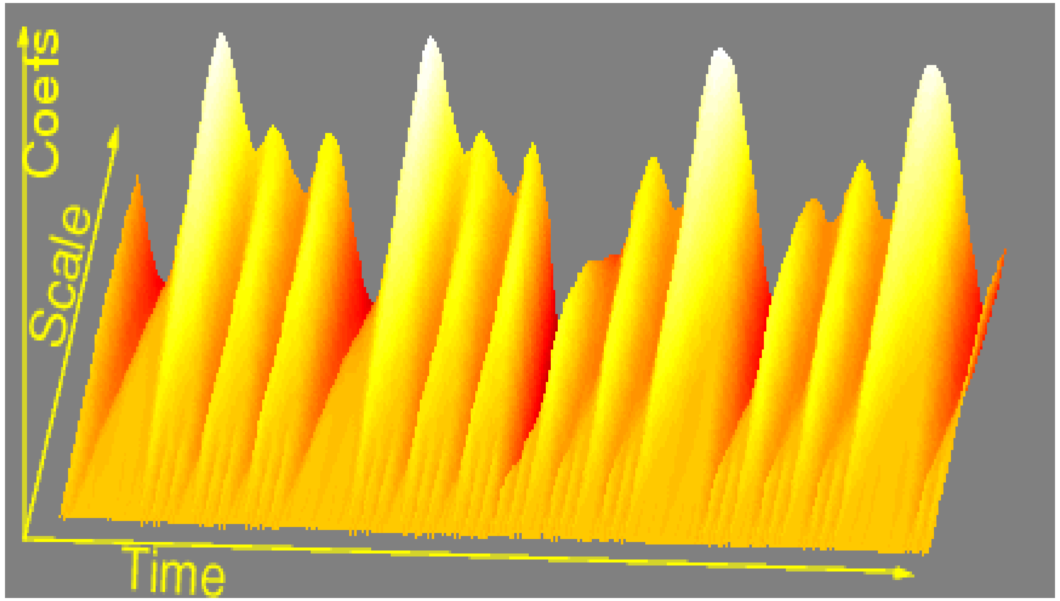
Resolution of Time and Frequency



- Each box represents a equal portion
- Resolution in STFT is selected once for entire analysis

Visualize CTW Transform

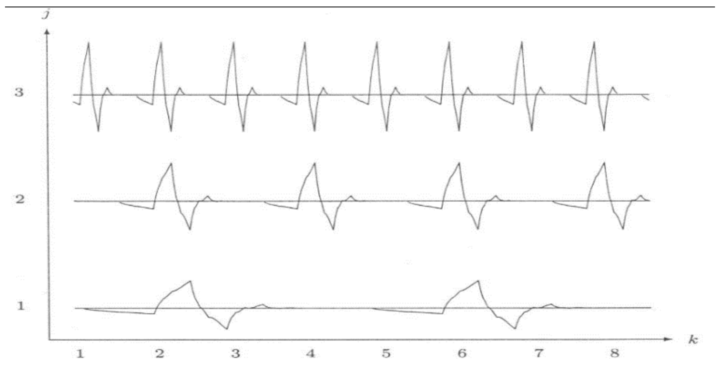
- Wavelet analysis produces a time-scale view of the input signal or image.

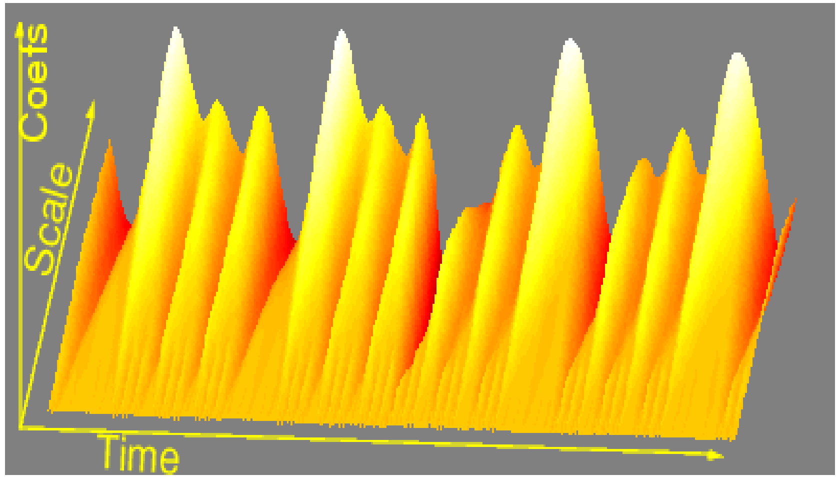


$$C(\tau, s) = \frac{1}{\sqrt{s}} \int_t f(t)\psi^*\left(\frac{t-\tau}{s}\right)dt$$

Properties of Wavelets

- Simultaneous localization in time and scale
  - The location of the wavelet allows to explicitly represent the location of events in time.
  - The shape of the wavelet allows to represent different detail or resolution.





Properties of Wavelets (cont'd)

- Sparsity: for functions typically found in practice, many of the coefficients in a wavelet representation are either zero or very small.

$$f(t) = \frac{1}{\sqrt{s}} \int_{\tau} \int_{s} C(\tau, s) \psi\left(\frac{t-\tau}{s}\right) d\tau ds$$

Properties of Wavelets (cont’d)

- Adaptability: Can represent functions with discontinuities or corners more efficiently.
- Linear-time complexity: many wavelet transformations can be accomplished in O(N) time.

Discretization of CWT

- It is Necessary to Sample the Time-Frequency (scale) Plane.
- At High Scale *s* (Lower Frequency *f*), the Sampling Rate *N* can be Decreased.
- The Scale Parameter *s* is Normally Discretized on a Logarithmic Grid.
- The most Common Value is 2.
- The Discretized CWT is not a True Discrete Transform

- Discrete Wavelet Transform (DWT)
  - Provides sufficient information both for analysis and synthesis
  - Reduce the computation time sufficiently
  - Easier to implement
  - Analyze the signal at different frequency bands with different resolutions
  - Decompose the signal into a coarse approximation and detail information

Discrete Wavelet Transform (DWT)

$$a_{jk} = \sum_t f(t)\psi^*_{jk}(t) \qquad \text{(forward DWT)}$$

$$f(t) = \sum_k \sum_j a_{jk}\psi_{jk}(t) \qquad \text{(inverse DWT)}$$

where $$\psi_{jk}(t) = 2^{j/2}\psi\left(2^j t - k\right)$$

DFT vs DWT

- DFT expansion:

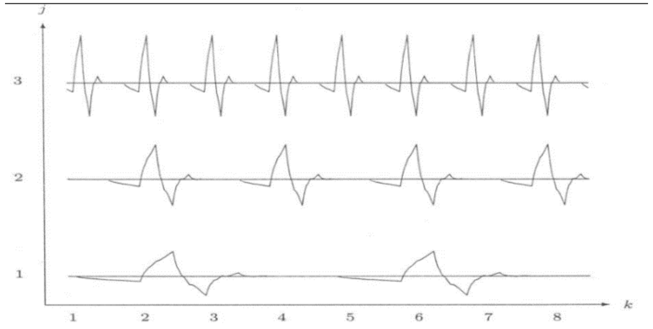
one parameter basis

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi ux}{N}}, \quad \text{or} \quad f(t) = \sum_{l} a_l \psi_l(t)$$

- DWT expansion

two parameter basis

$$f(t) = \sum_{k} \sum_{j} a_{jk} \psi_{jk}(t)$$



Haar Wavelets

- Scaling functions

Haar scaling function is defined by

$$\phi(x) = \begin{cases} 1 & for \quad 0 \le x < 1 \\ 0 & otherwise \end{cases}$$

and is shown in Figure 1.

Some examples of its translated and scaled versions are shown in Figures 2-4.

1D Haar

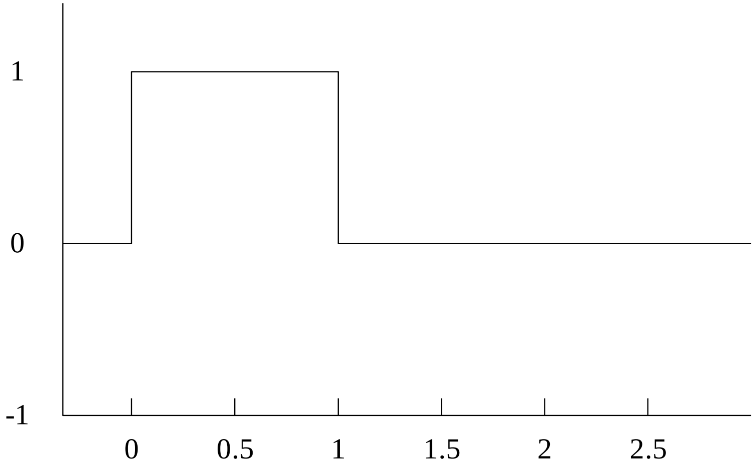


Fig.1: Haar scaling function ϕ(x).

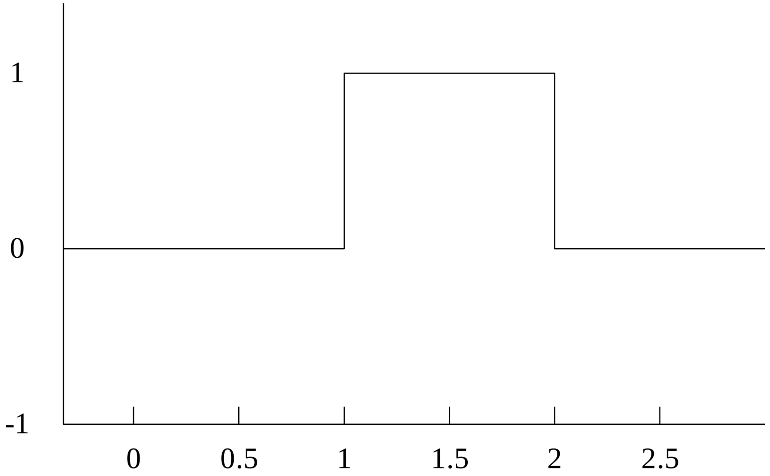


Fig.2: Haar scaling function ϕ(x-1).

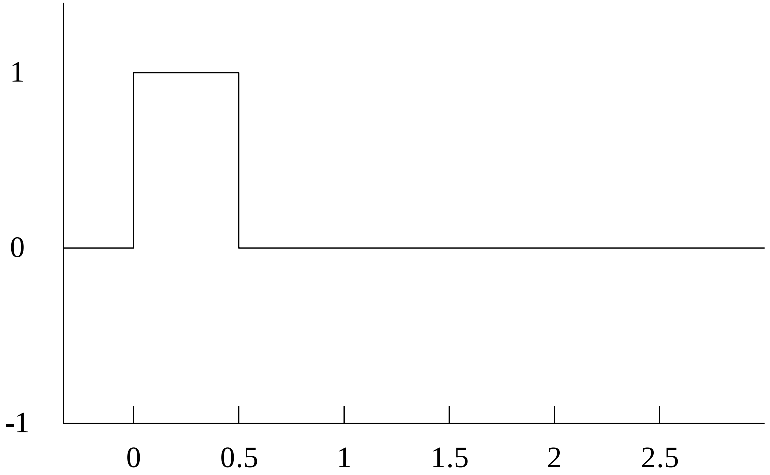


Fig.3: Haar scaling function ϕ(2x).

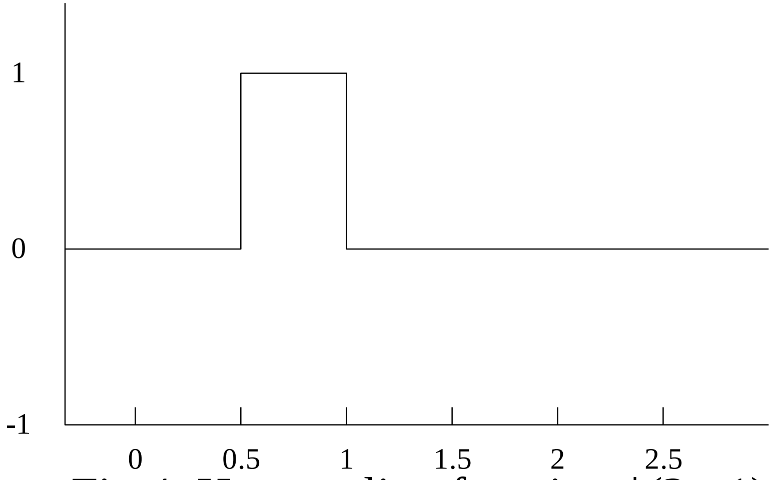
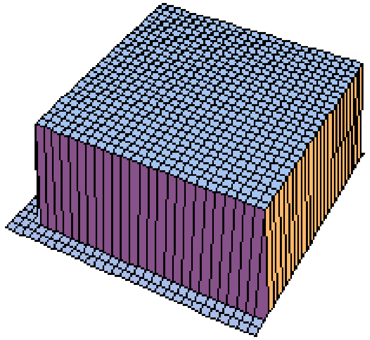


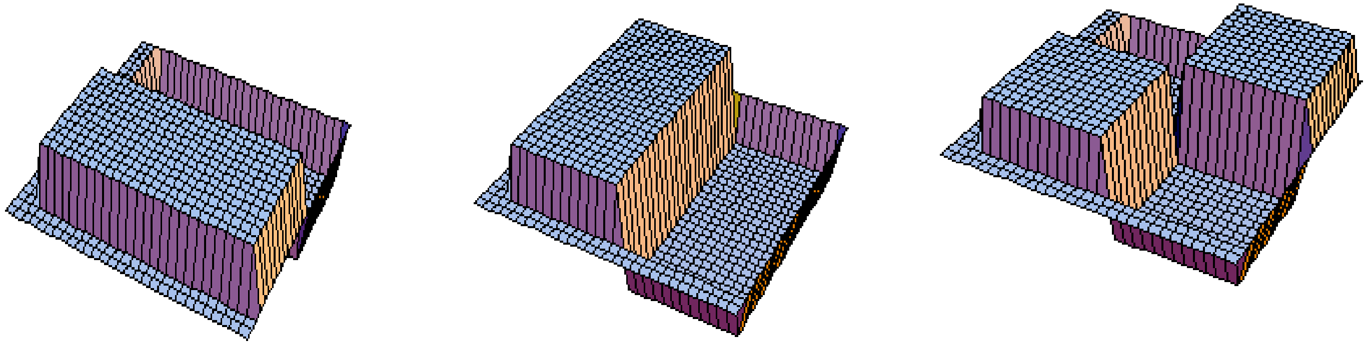
Fig.4: Haar scaling function ϕ(2x-1).

2D Haar

- 2D Haar scaling:



- 2D Haar wavelets:



■ Wavelets

1. The Haar wavelet ψ (x) is given by

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

   and is shown in Figure 5.

2. The two-scale relation for Haar wavelet is

$$\psi(x) = \phi(2x) - \phi(2x-1).$$

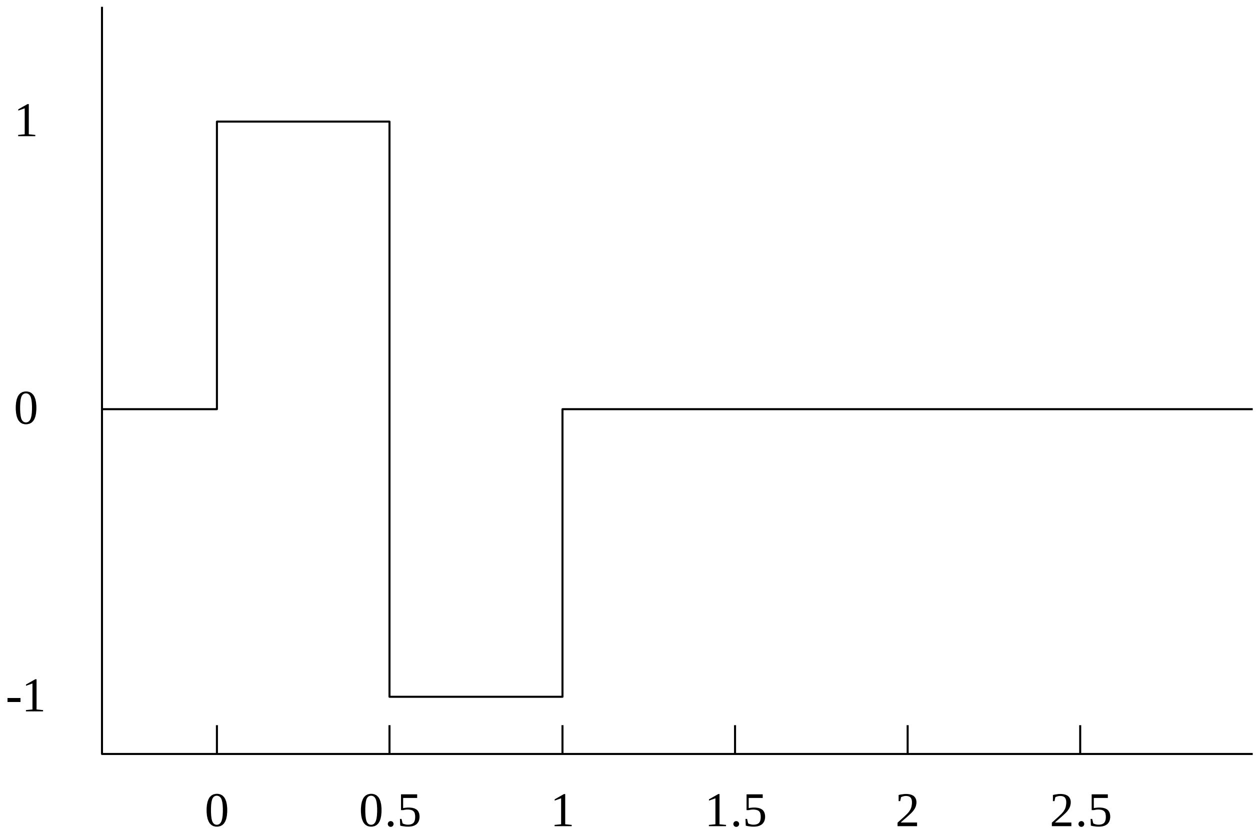


Figure 5: Haar Wavelet $\psi(x)$.

Wavelet expansion

- Wavelet decompositions involve a pair of waveforms:

encodes low resolution info $\varphi(t)$

$\psi(t)$ encodes details or high resolution info

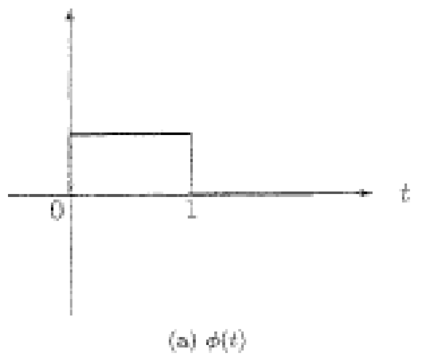
$$f(t)=\sum_k c_k \varphi(t-k)+\sum_k \sum_j d_{jk} \psi(2^j t-k)$$

Terminology: scaling function (φ)   wavelet function (ψ)

1D Haar Wavelets

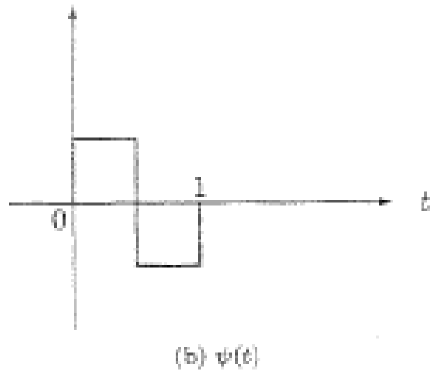
- Haar scaling and wavelet functions:

φ(t)



computes average
(low pass)

ψ(t)



computes details
(high pass)

Haar Filter Bank

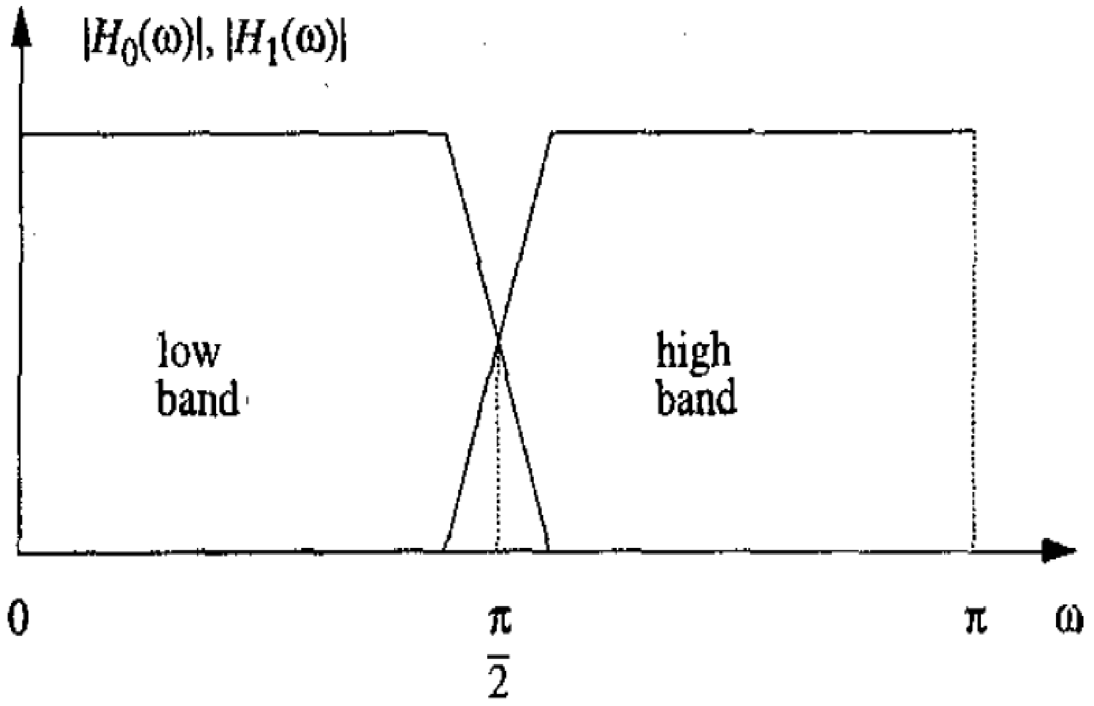
- The simplest orthogonal filter bank is Haar
- The lowpass filter is

$$h_0[n] = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0, -1 \\ 0, & \text{otherwise} \end{cases}$$

- And the highpass filter

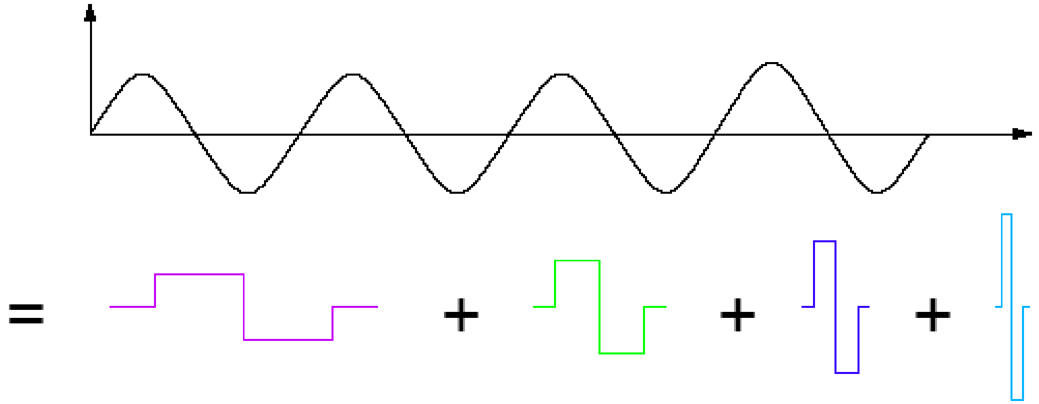
$$h_1[n] = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0 \\ -\frac{1}{\sqrt{2}}, & n = -1 \\ 0, & \text{otherwise} \end{cases}$$

Two-Channel Filter Banks



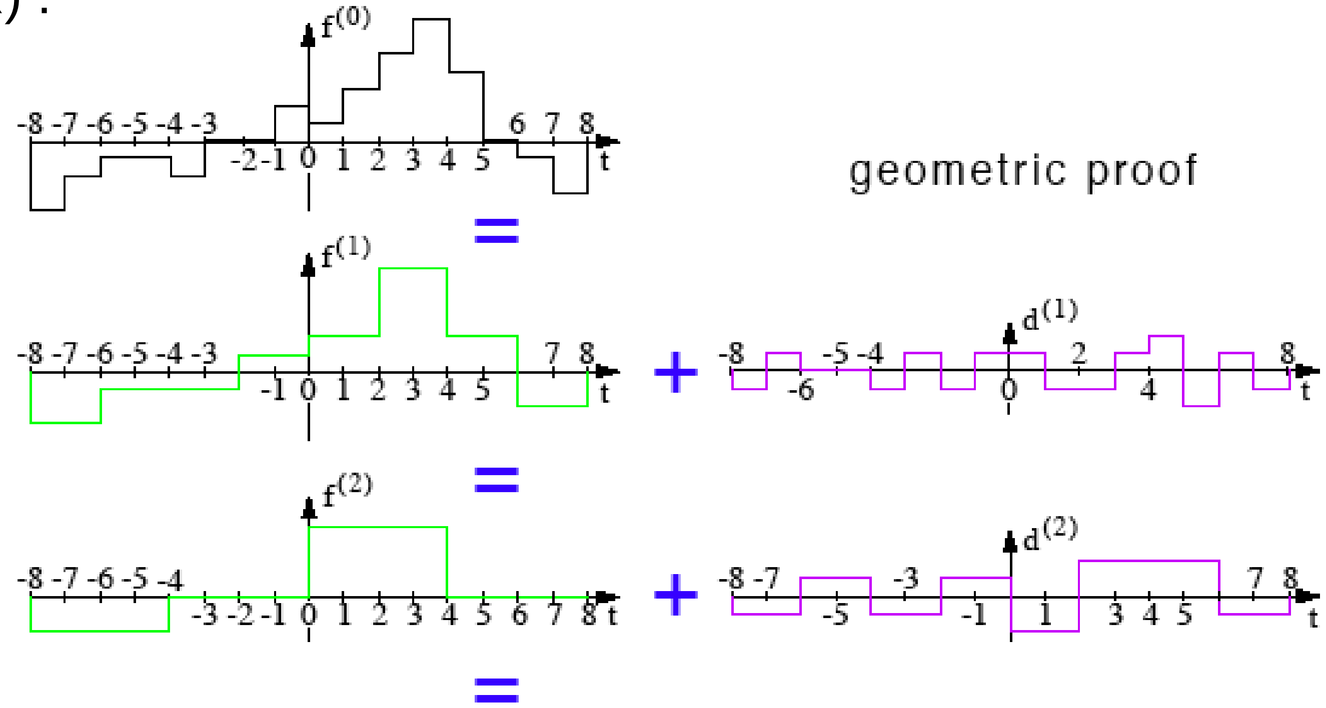
The Haar wavelet

1910: Alfred Haar discovers the Haar wavelet
dual to the Fourier construction



•A basis for $L_2(R)$:

Averaging
and
differencing

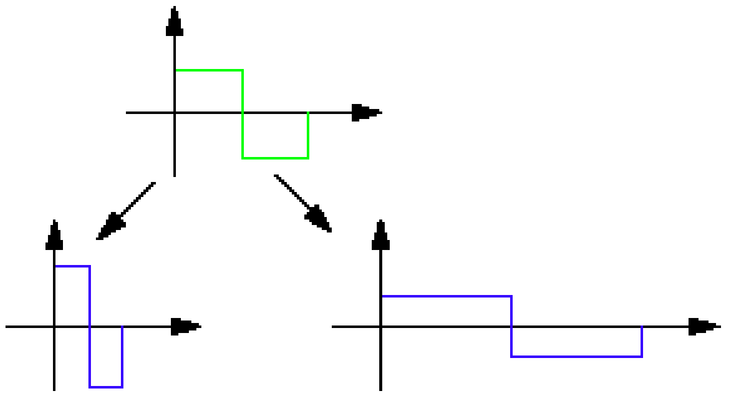


The Haar wavelet

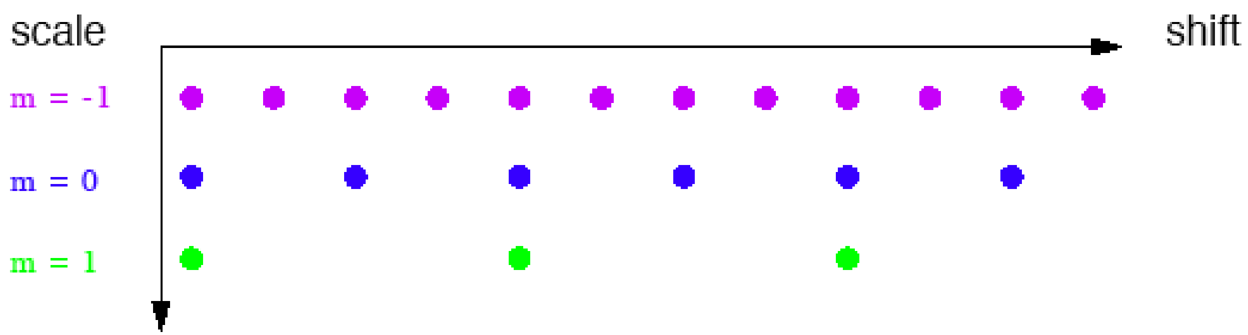
Basis functions

$$\psi(t) = \begin{cases} 1 & 0 \le t < 0.5 \\ -1 & 0.5 \le t < 1 \\ 0 & \text{else} \end{cases}$$

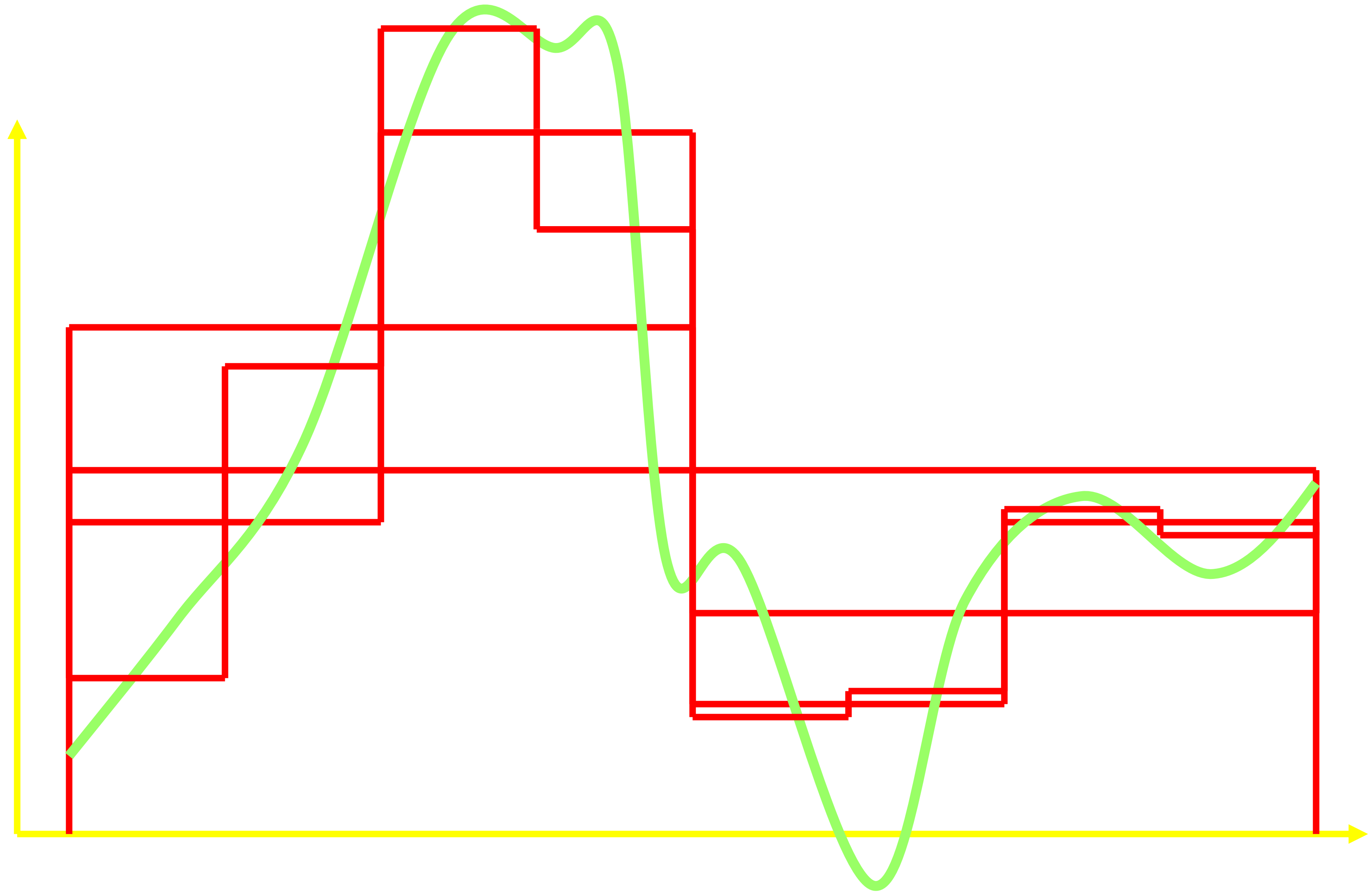
$$\psi_{m,n}(t) = 2^{-m/2}\psi(2^{-m}t - n)$$



Compute WT on a discrete grid



Haar transform

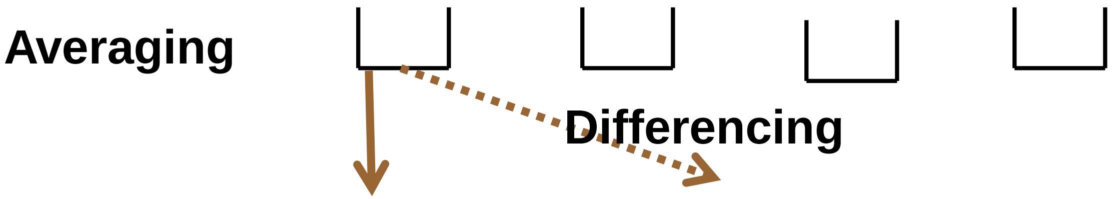


Haar Wavelet Transform

- Find the average of each pair of samples
- Find the difference between the average and sample
- Fill the first half with averages
- Fill the second half with differences
- Repeat the process on the first half

- Step 1:

[3 5 4 8 13 7 5 3]



[4 6 10 4 -1 -2 3 1]

Haar Wavelet Transform

- Step 2

[4 6 10 4 -1 -2 3 1]

Averaging

Differencing

[5 7 -1 3 -1 -2 3 1]

ex. (4 + 6)/2 = 5

4 - 5 = -1

Haar Wavelet Transform

- Step 3

[5 7 -1 3 -1 -2 3 1]

Averaging

Differencing

[6 -1 -1 3 -1 -2 3 1]

row average

ex. (5 + 7)/2 = 6

5 - 6 = -1

Wavelet Transform Example

- Suppose we are given the following input sequence.

$\{x_{n,i}\}$ = {10, 13, 25, 26, 29, 21, 7, 15}

- Consider the transform that replaces the original sequence with its pairwise *average* x_{n-1},i and *difference* $d_{n-1,i}$ defined as follows:

$$x_{n-1,i} = \frac{x_{n,2i} + x_{n,2i+1}}{2}$$

$$d_{n-1,i} = \frac{x_{n,2i} - x_{n,2i+1}}{2}$$

- The averages and differences are applied only on consecutive *pairs* of input sequences whose first element has an even index. Therefore, the number of elements in each set $\{x_{n-1,i}\}$ and $\{d_{n-1,i}\}$ is exactly half of the number of elements in the original sequence.

- Form a new sequence having length equal to that of the original sequence by concatenating the two sequences $\{x_{n-1,i}\}$ and $\{d_{n-1,i}\}$. The resulting sequence is

    - $\{x_{n-1,i}, d_{n-1,i}\} = \{11.5, 25.5, 25, 11, -1.5, -0.5, 4, -4\}$

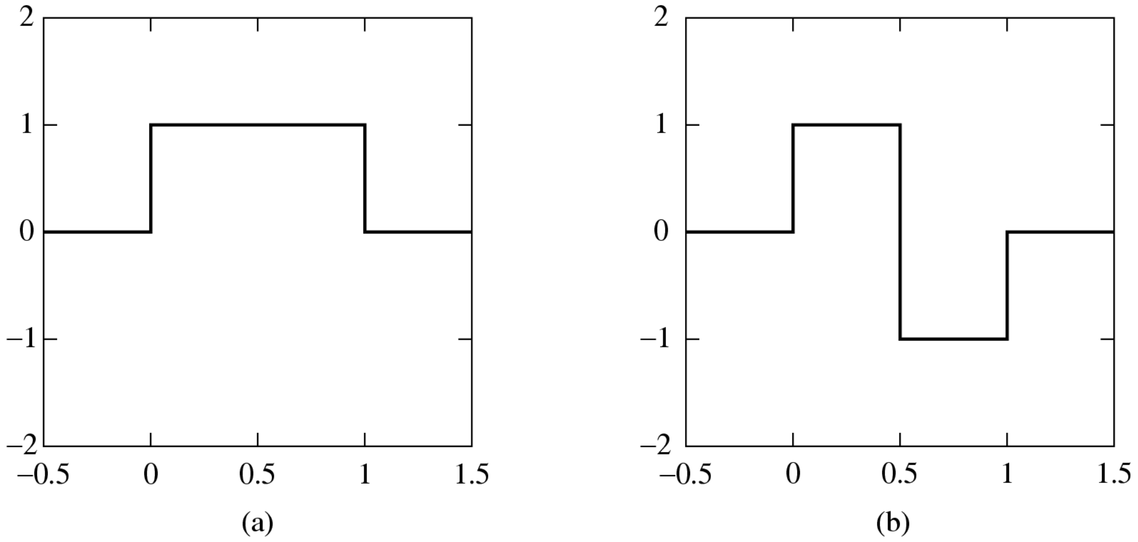
- This sequence has exactly the same number of elements as the input sequence — the transform did not increase the amount of data.

- Since the first half of the above sequence contain averages from the original sequence, we can view it as a coarser approximation to the original signal. The second half of this sequence can be viewed as the details or approximation errors of the first half.

- It is easily verified that the original sequence can be reconstructed from the transformed sequence using the relations

$$x_{n,\,2i} = x_{n-1,\,i} + d_{n-1,\,i}$$
$$x_{n,\,2i+1} = x_{n-1,\,i} - d_{n-1,\,i} \qquad (8.49)$$

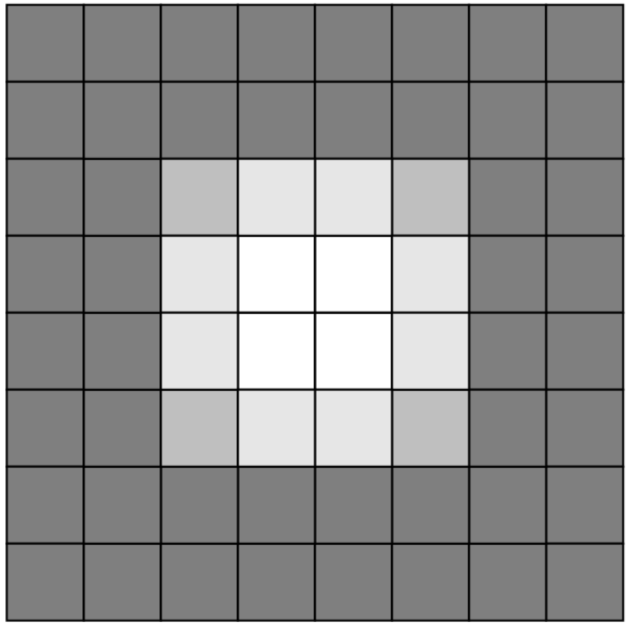
- This transform is the discrete Haar wavelet transform.



Haar Transform: (a) scaling function, (b) wavelet function.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	63	127	127	63	0	0
0	0	127	255	255	127	0	0
0	0	127	255	255	127	0	0
0	0	63	127	127	63	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)



(b)

Input image for the 2D Haar Wavelet Transform.

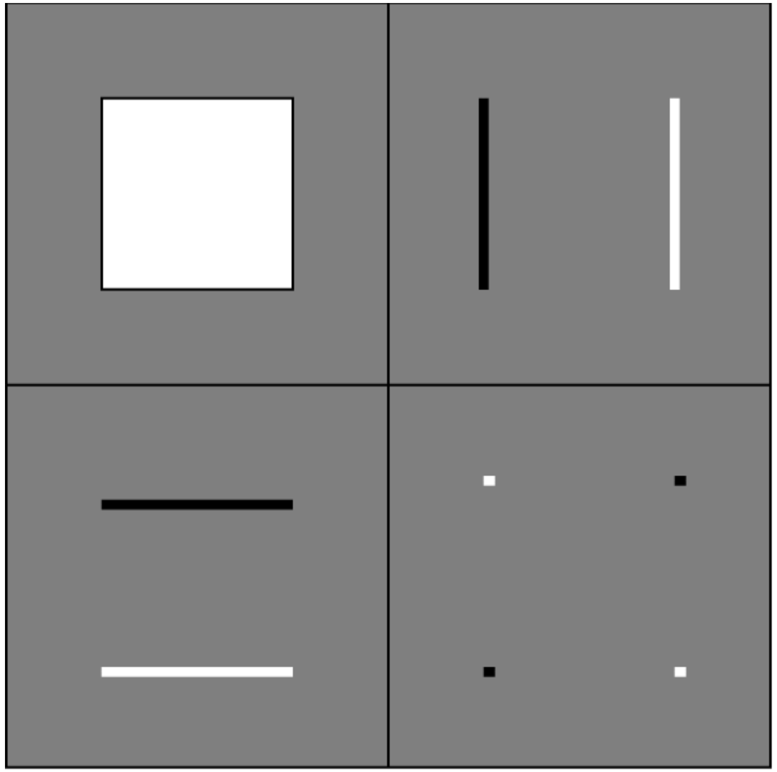
(a) The pixel values. (b) Shown as an 8 × 8 image.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	95	95	0	0	–32	32	0
0	191	191	0	0	–64	64	0
0	191	191	0	0	–64	64	0
0	95	95	0	0	–32	32	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Intermediate output of the 2D Haar Wavelet Transform.

0	0	0	0	0	0	0	0
0	143	143	0	0	–48	48	0
0	143	143	0	0	–48	48	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	–48	–48	0	0	16	–16	0
0	48	48	0	0	–16	16	0
0	0	0	0	0	0	0	0

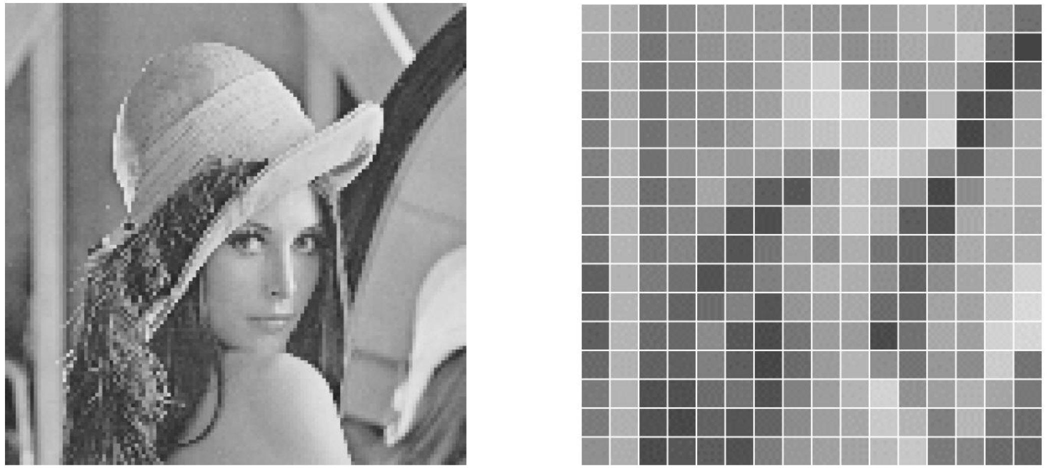
Output of the first level of the 2D Haar Wavelet Transform.



A simple graphical illustration of Wavelet Transform.

2D Wavelet Transform Example

The input image is a sub-sampled version of the image **Lena**. The size of the input is 16×16. The filter used in the example is the Antonini 9/7 filter set



(a) (b)

The Lena image: (a) Original 128 × 128 image. (b) 16 × 16 sub-sampled image.

- The input image is shown in numerical form below.

$$I_{00}(x,y) = \begin{bmatrix}
158 & 170 & 97 & 104 & 123 & 130 & 133 & 125 & 132 & 127 & 112 & 158 & 159 & 144 & 116 & 91 \\
164 & 153 & 91 & 99 & 124 & 152 & 131 & 160 & 189 & 116 & 106 & 145 & 140 & 143 & 227 & 53 \\
116 & 149 & 90 & 101 & 118 & 118 & 131 & 152 & 202 & 211 & 84 & 154 & 127 & 146 & 58 & 58 \\
95 & 145 & 88 & 105 & 188 & 123 & 117 & 182 & 185 & 204 & 203 & 154 & 153 & 229 & 46 & 147 \\
101 & 156 & 89 & 100 & 165 & 113 & 148 & 170 & 163 & 186 & 144 & 194 & 208 & 39 & 113 & 159 \\
103 & 153 & 94 & 103 & 203 & 136 & 146 & 92 & 66 & 192 & 188 & 103 & 178 & 47 & 167 & 159 \\
102 & 146 & 106 & 99 & 99 & 121 & 39 & 60 & 164 & 175 & 198 & 46 & 56 & 56 & 156 & 156 \\
99 & 146 & 95 & 97 & 144 & 61 & 103 & 107 & 108 & 111 & 192 & 62 & 65 & 128 & 153 & 154 \\
99 & 140 & 103 & 109 & 103 & 124 & 54 & 81 & 172 & 137 & 178 & 54 & 43 & 159 & 149 & 174 \\
84 & 133 & 107 & 84 & 149 & 43 & 158 & 95 & 151 & 120 & 183 & 46 & 30 & 147 & 142 & 201 \\
58 & 153 & 110 & 41 & 94 & 213 & 71 & 73 & 140 & 103 & 138 & 83 & 152 & 143 & 128 & 207 \\
56 & 141 & 108 & 58 & 92 & 51 & 55 & 61 & 88 & 166 & 58 & 103 & 146 & 150 & 116 & 211 \\
89 & 115 & 188 & 47 & 113 & 104 & 56 & 67 & 128 & 155 & 187 & 71 & 153 & 134 & 203 & 95 \\
35 & 99 & 151 & 67 & 35 & 88 & 88 & 128 & 140 & 142 & 176 & 213 & 144 & 128 & 214 & 100 \\
89 & 98 & 97 & 51 & 49 & 101 & 47 & 90 & 136 & 136 & 157 & 205 & 106 & 43 & 54 & 76 \\
44 & 105 & 69 & 69 & 68 & 53 & 110 & 127 & 134 & 146 & 159 & 184 & 109 & 121 & 72 & 113
\end{bmatrix}$$

- First, we need to compute the analysis and synthesis high-pass filters.

$$h_1[n] = [-0.065, 0.041, 0.418, -0.788, 0.418, 0.041, -0.065]$$

$$\tilde{h}_1[n] = [-0.038, -0.024, 0.111, 0.377, -0.853, 0.377, 0.111, -0.024, -0.038] \qquad (8.70)$$

- Convolve the first row with both $h_0[n]$ and $h_1[n]$ and discarding the values with odd-numbered index. The results of these two operations are:

$$(I_{00}(:,0) * h_0[n]) \downarrow 2 = [245, 156, 171, 183, 184, 173, 228, 160]$$

$$(I_{00}(:,0) * h_1[n]) \downarrow 2 = [-30, 3, 0, 7, -5, -16, -3, 16]$$

- Form the transformed output row by concatenating the resulting coefficients. The first row of the transformed image is then:

  - [245, 156, 171, 183, 184, 173, 228, 160, −30, 3, 0, 7, −5, −16, −3, 16]

- Continue the same process for the remaining rows.

- The result after all rows have been processed:

$$I_{00}(x, y) =$$

$$\begin{bmatrix}
245 & 156 & 171 & 183 & 184 & 173 & 228 & 160 & -30 & 3 & 0 & 7 & -5 & -16 & -3 & 16 \\
239 & 141 & 181 & 197 & 242 & 158 & 202 & 229 & -17 & 5 & -20 & 3 & 26 & -27 & 27 & 141 \\
195 & 147 & 163 & 177 & 288 & 173 & 209 & 106 & -34 & 2 & 2 & 19 & -50 & -35 & -38 & -1 \\
180 & 139 & 226 & 177 & 274 & 267 & 247 & 163 & -45 & 29 & 24 & -29 & -2 & 30 & -101 & -78 \\
191 & 145 & 197 & 198 & 247 & 230 & 239 & 143 & -49 & 22 & 36 & -11 & -26 & -14 & 101 & -54 \\
192 & 145 & 237 & 184 & 135 & 253 & 169 & 192 & -47 & 38 & 36 & 4 & -58 & 66 & 94 & -4 \\
176 & 159 & 156 & 77 & 204 & 232 & 51 & 196 & -31 & 9 & -48 & 30 & 11 & 58 & 29 & 4 \\
179 & 148 & 162 & 129 & 146 & 213 & 92 & 217 & -39 & 18 & 50 & -10 & 33 & 51 & -23 & 8 \\
169 & 159 & 163 & 97 & 204 & 202 & 85 & 234 & -29 & 1 & -42 & 23 & 37 & 41 & -56 & -5 \\
155 & 153 & 149 & 159 & 176 & 204 & 65 & 236 & -32 & 32 & 85 & 39 & 38 & 44 & -54 & -31 \\
145 & 148 & 158 & 148 & 164 & 157 & 188 & 215 & -55 & 59 & -110 & 28 & 26 & 48 & -1 & -64 \\
134 & 152 & 102 & 70 & 153 & 126 & 199 & 207 & -47 & 38 & 13 & 10 & -76 & 3 & -7 & -76 \\
127 & 203 & 130 & 94 & 171 & 218 & 171 & 228 & 12 & 88 & -27 & 15 & 1 & 76 & 24 & 85 \\
70 & 188 & 63 & 144 & 191 & 257 & 215 & 232 & -5 & 24 & -28 & -9 & 19 & -46 & 36 & 91 \\
129 & 124 & 87 & 96 & 177 & 236 & 162 & 77 & -2 & 20 & -48 & 1 & 17 & -56 & 30 & -24 \\
103 & 115 & 85 & 142 & 188 & 234 & 184 & 132 & -37 & 0 & 27 & -4 & 5 & -35 & -22 & -33
\end{bmatrix}$$

- Apply the filters to the columns of the resulting image. Apply both $h_0[n]$ and $h_1[n]$ to each column and discard the odd indexed results:

$$(I_{11}(0,:) * h_0[n]) \downarrow 2 = [353, 280, 269, 256, 240, 206, 160, 153]^T$$

$$(I_{11}(0,:) * h_1[n]) \downarrow 2 = [-12, 10, -7, -4, 2, -1, 43, 16]^T$$

- Concatenate the above results into a single column and apply the same procedure to each of the remaining columns.

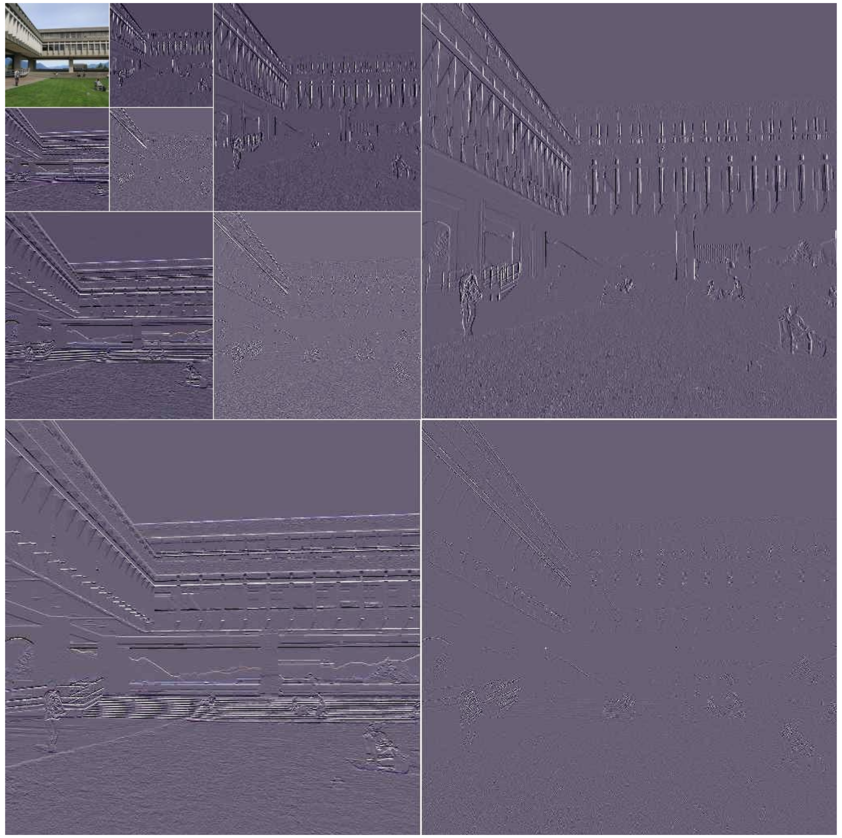
$$I_{11}(x,y) = $$

$$\begin{bmatrix}
353 & 212 & 251 & 272 & 281 & 234 & 308 & 289 & -33 & 6 & -15 & 5 & 24 & -29 & 38 & 120 \\
280 & 203 & 254 & 250 & 402 & 269 & 297 & 207 & -45 & 11 & -2 & 9 & -31 & -26 & -74 & 23 \\
269 & 202 & 312 & 280 & 316 & 353 & 337 & 227 & -70 & 43 & 56 & -23 & -41 & 21 & 82 & -81 \\
256 & 217 & 247 & 155 & 236 & 328 & 114 & 283 & -52 & 27 & -14 & 23 & -2 & 90 & 49 & 12 \\
240 & 221 & 226 & 172 & 264 & 294 & 113 & 330 & -41 & 14 & 31 & 23 & 57 & 60 & -78 & -3 \\
206 & 204 & 201 & 192 & 230 & 219 & 232 & 300 & -76 & 67 & -53 & 40 & 4 & 46 & -18 & -107 \\
160 & 275 & 150 & 135 & 244 & 294 & 267 & 331 & -2 & 90 & -17 & 10 & -24 & 49 & 29 & 89 \\
153 & 189 & 113 & 173 & 260 & 342 & 256 & 176 & -20 & 18 & -38 & -4 & 24 & -75 & 25 & -5 \\
-12 & 7 & -9 & -13 & -6 & 11 & 12 & -69 & -10 & -1 & 14 & 6 & -38 & 3 & -45 & -99 \\
10 & 3 & -31 & 16 & -1 & -51 & -10 & -30 & 2 & -12 & 0 & 24 & -32 & -45 & 109 & 42 \\
-7 & 5 & -44 & -35 & 67 & -10 & -17 & -15 & 3 & -15 & -28 & 0 & 41 & -30 & -18 & -19 \\
-4 & 9 & -1 & -37 & 41 & 6 & -33 & 2 & 9 & -12 & -67 & 31 & -7 & 3 & 2 & 0 \\
2 & -3 & 9 & -25 & 2 & -25 & 60 & -8 & -11 & -4 & -123 & -12 & -6 & -4 & 14 & -12 \\
-1 & 22 & 32 & 46 & 10 & 48 & -11 & 20 & 19 & 32 & -59 & 9 & 70 & 50 & 16 & 73 \\
43 & -18 & 32 & -40 & -13 & -23 & -37 & -61 & 8 & 22 & 2 & 13 & -12 & 43 & -8 & -45 \\
16 & 2 & -6 & -32 & -7 & 5 & -13 & -50 & 24 & 7 & -61 & 2 & 11 & -33 & 43 & 1
\end{bmatrix}$$

- This completes one stage of the discrete wavelet transform. We can perform another stage of the DWT by applying the same transform procedure illustrated above to the upper left 8 × 8 DC image of $I_{12}(x, y)$. The resulting two-stage transformed image is

$$I_{22}(x,y) =$$

$$\begin{bmatrix}
558 & 451 & 608 & 532 & 75 & 26 & 94 & 25 & -33 & 6 & -15 & 5 & 24 & -29 & 38 & 120 \\
463 & 511 & 627 & 566 & 66 & 68 & -43 & 68 & -45 & 11 & -2 & 9 & -31 & -26 & -74 & 23 \\
464 & 401 & 478 & 416 & 14 & 84 & -97 & -229 & -70 & 43 & 56 & -23 & -41 & 21 & 82 & -81 \\
422 & 335 & 477 & 553 & -88 & 46 & -31 & -6 & -52 & 27 & -14 & 23 & -2 & 90 & 49 & 12 \\
14 & 33 & -56 & 42 & 22 & -43 & -36 & 1 & -41 & 14 & 31 & 23 & 57 & 60 & -78 & -3 \\
-13 & 36 & 54 & 52 & 12 & -21 & 51 & 70 & -76 & 67 & -53 & 40 & 4 & 46 & -18 & -107 \\
25 & -20 & 25 & -7 & -35 & 35 & -56 & -55 & -2 & 90 & -17 & 10 & -24 & 49 & 29 & 89 \\
46 & 37 & -51 & 51 & -44 & 26 & 39 & -74 & -20 & 18 & -38 & -4 & 24 & -75 & 25 & -5 \\
-12 & 7 & -9 & -13 & -6 & 11 & 12 & -69 & -10 & -1 & 14 & 6 & -38 & 3 & -45 & -99 \\
10 & 3 & -31 & 16 & -1 & -51 & -10 & -30 & 2 & -12 & 0 & 24 & -32 & -45 & 109 & 42 \\
-7 & 5 & -44 & -35 & 67 & -10 & -17 & -15 & 3 & -15 & -28 & 0 & 41 & -30 & -18 & -19 \\
-4 & 9 & -1 & -37 & 41 & 6 & -33 & 2 & 9 & -12 & -67 & 31 & -7 & 3 & 2 & 0 \\
2 & -3 & 9 & -25 & 2 & -25 & 60 & -8 & -11 & -4 & -123 & -12 & -6 & -4 & 14 & -12 \\
-1 & 22 & 32 & 46 & 10 & 48 & -11 & 20 & 19 & 32 & -59 & 9 & 70 & 50 & 16 & 73 \\
43 & -18 & 32 & -40 & -13 & -23 & -37 & -61 & 8 & 22 & 2 & 13 & -12 & 43 & -8 & -45 \\
16 & 2 & -6 & -32 & -7 & 5 & -13 & -50 & 24 & 7 & -61 & 2 & 11 & -33 & 43 & 1
\end{bmatrix}$$

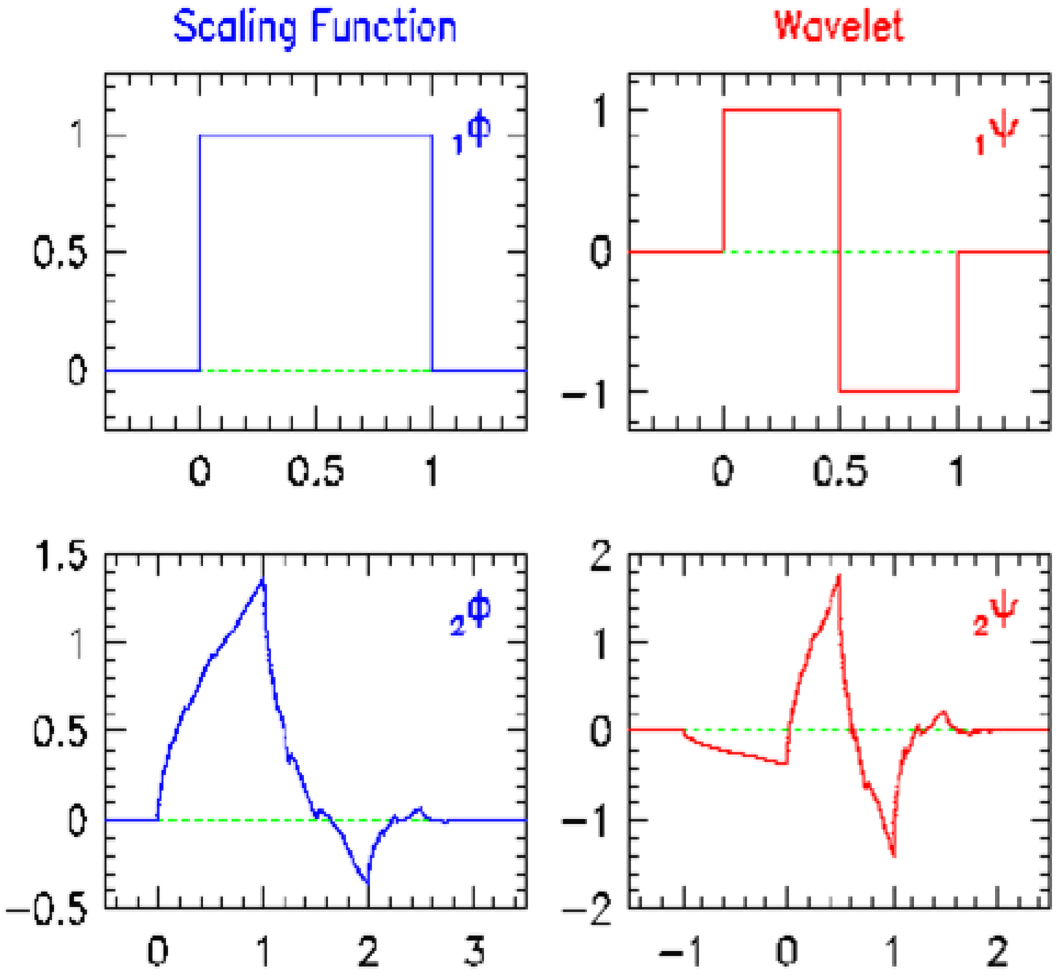


Haar wavelet decomposition.

Wavelet functions examples

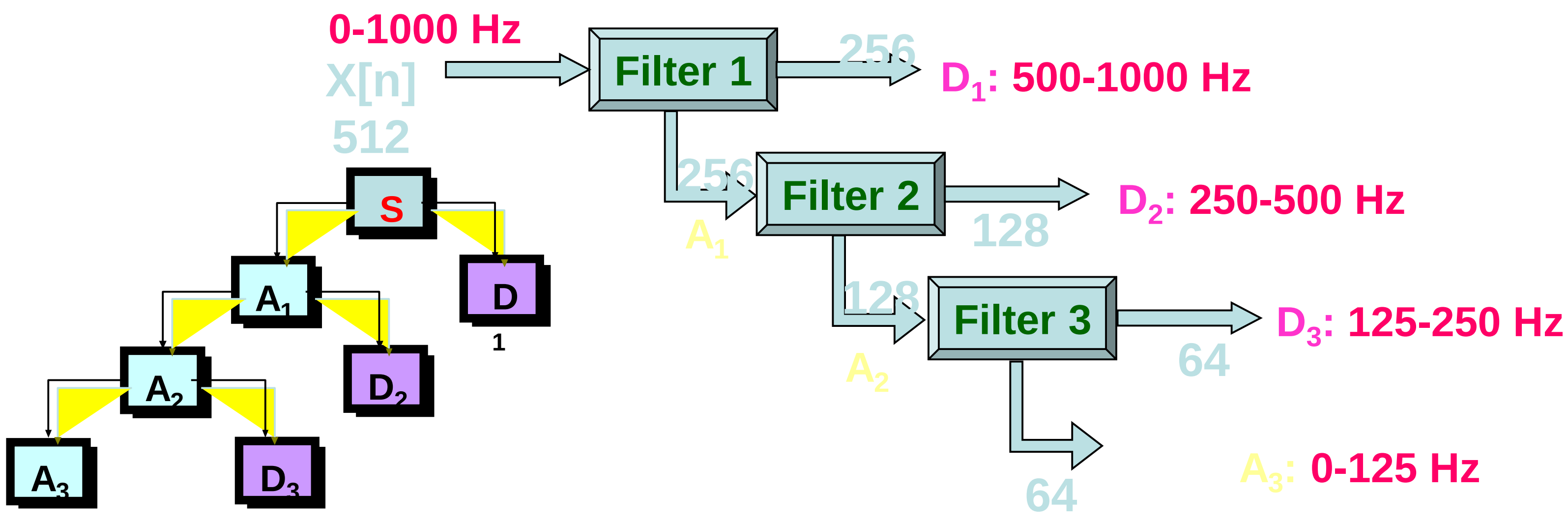
- Haar function

- Daubechies function

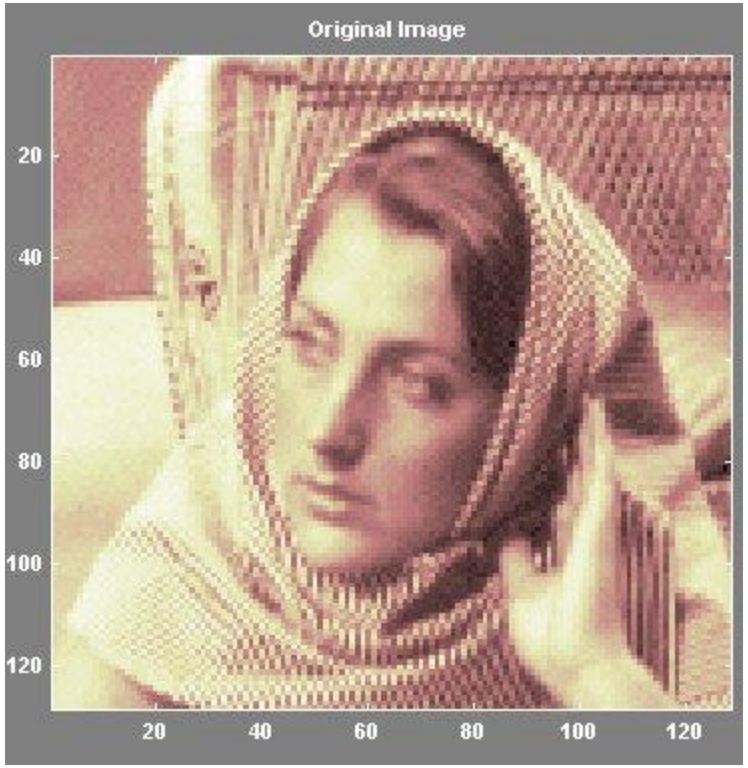


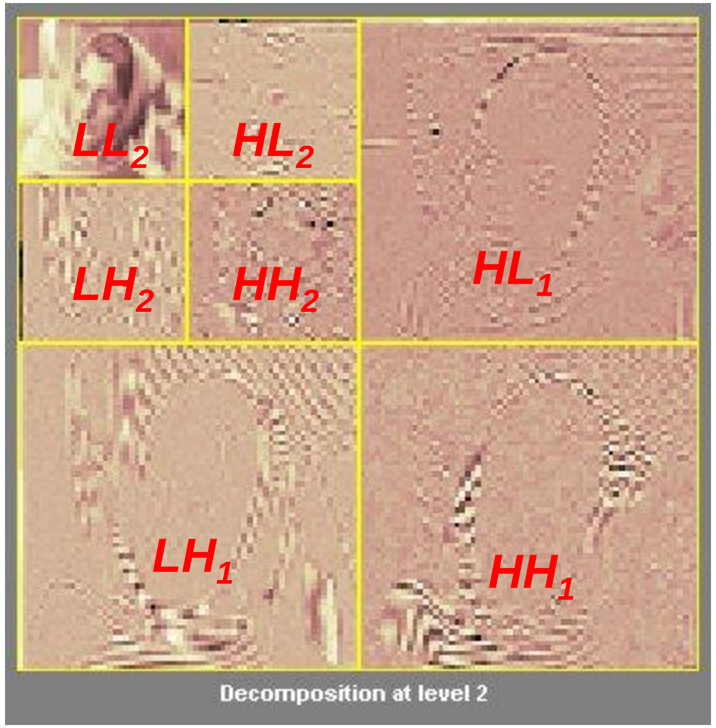
Multi-level Decomposition

- Iterating the decomposition process, breaks the input signal into many lower-resolution components: *Wavelet decomposition tree*:



Discrete Wavelet Transform





Mexican hat wavelet

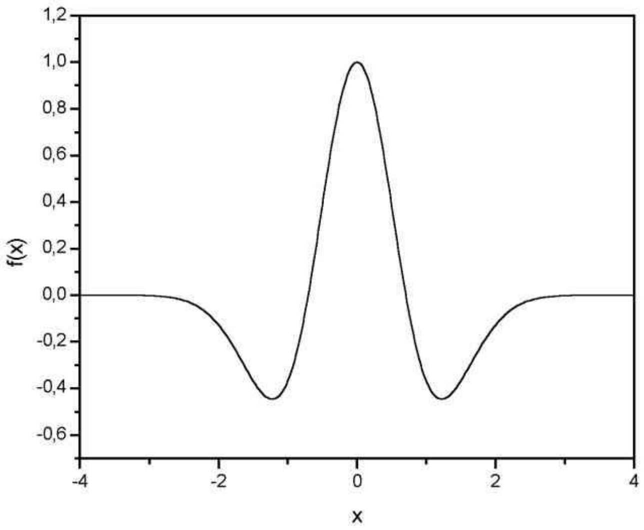


Fig. 7 The Mexican hat wavelet[5]

Also called the second derivative of the Gaussian function

$$\varphi(t) = \frac{1}{\sqrt{2\pi\sigma^3}}\left[e^{\frac{-t^2}{2\sigma^2}}\left(\frac{t^2}{\sigma^2}-1\right)\right]$$

Morlet wavelet

$$\varphi(t) = \pi^{-1/4} e^{imt} e^{-t^2/2}$$

$$\hat{\varphi}(\omega) = \pi^{-1/4} U(\omega) e^{-(\omega - m)^2/2}$$

U(ω): step function

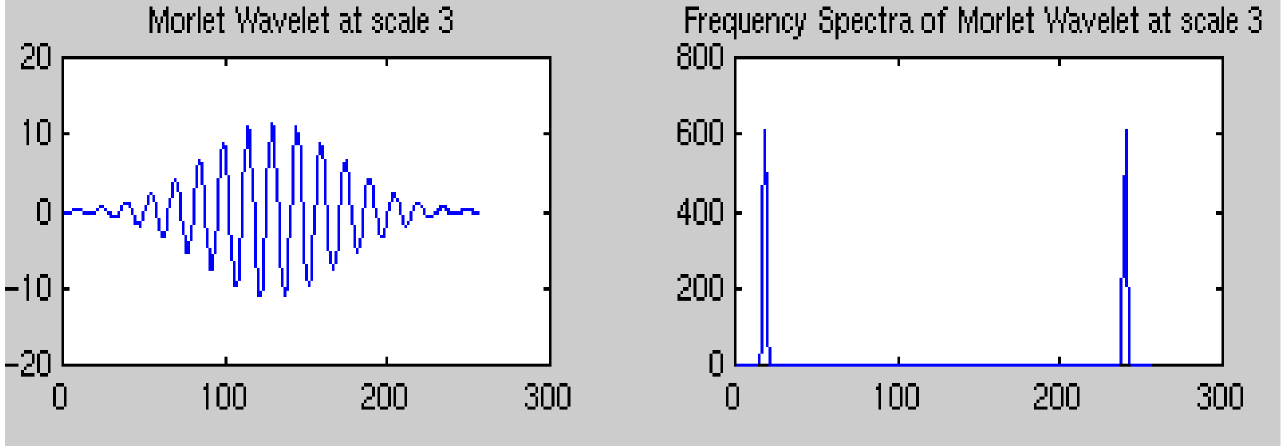
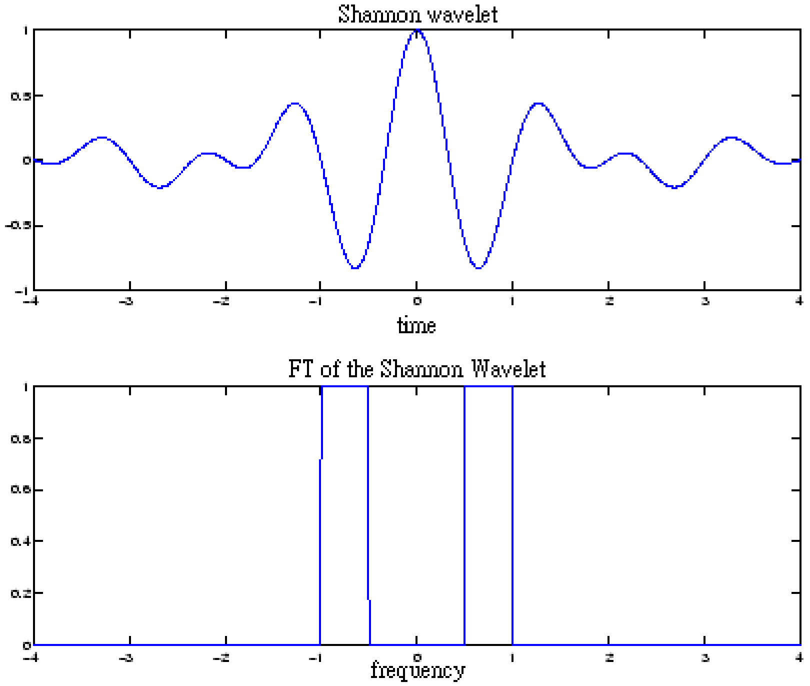


Fig. 8 Morlet wavelet with m equals to 3[4]

Shannon wavelet



$$\varphi(t) = \text{sinc}(t/2)\cos(3\pi t/2)$$

$$\hat{\varphi}(f) = \begin{cases} 1 & 0.5 \le |f| \le 1 \\ 0 & otherwise \end{cases}$$

Fig. 9 The Shannon wavelet in time and frequency domains[5]

Wavelets Applications

- Noise filtering
- Image compression
  - – Special case: fingerprint compression
- Image fusion
- Recognition

G. Bebis, A. Gyaourova, S. Singh, and I. Pavlidis, "Face Recognition by Fusing Thermal Infrared and Visible Imagery", **Image and Vision Computing**, vol. 24, no. 7, pp. 727-742, 2006.

- Image matching and retrieval

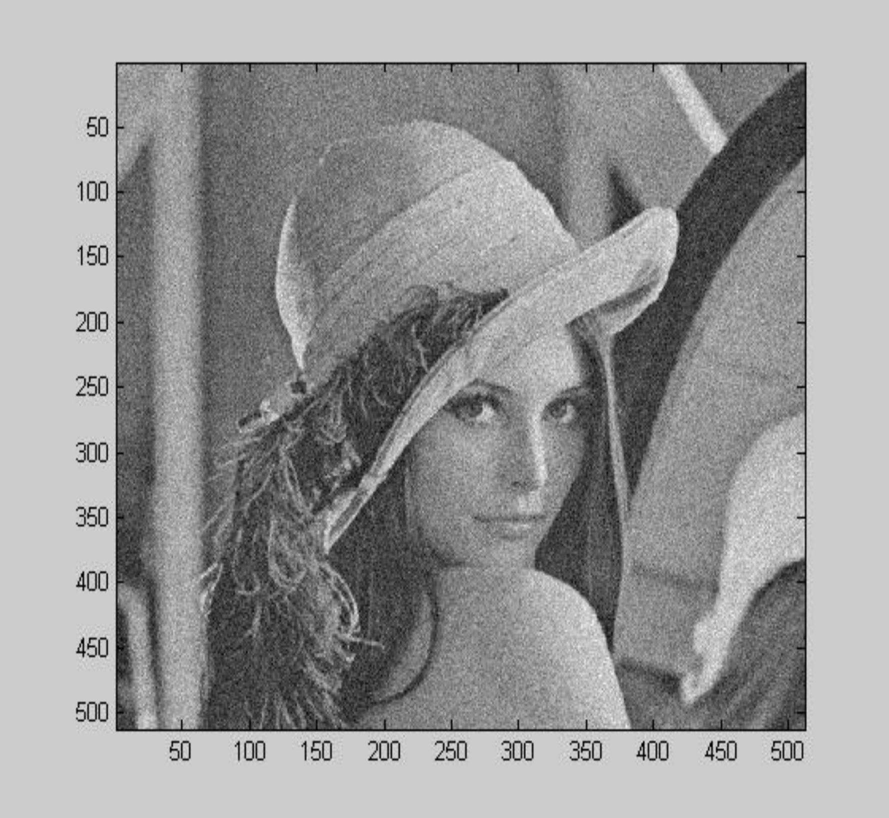
  Charles E. Jacobs Adam Finkelstein David H. Salesin, "Fast Multiresolution Image Querying", **SIGRAPH**, 1995.

Image Denoising Using Wavelets

- Calculate the DWT of the image.
- Threshold the wavelet coefficients. The threshold may be universal or subband adaptive.
- Compute the IDWT to get the denoised estimate.
- Soft thresholding is used in the different thresholding methods. Visually more pleasing images.

Application: Image Denoising Using Wavelets

- Noisy Image:



- Denoised Image:

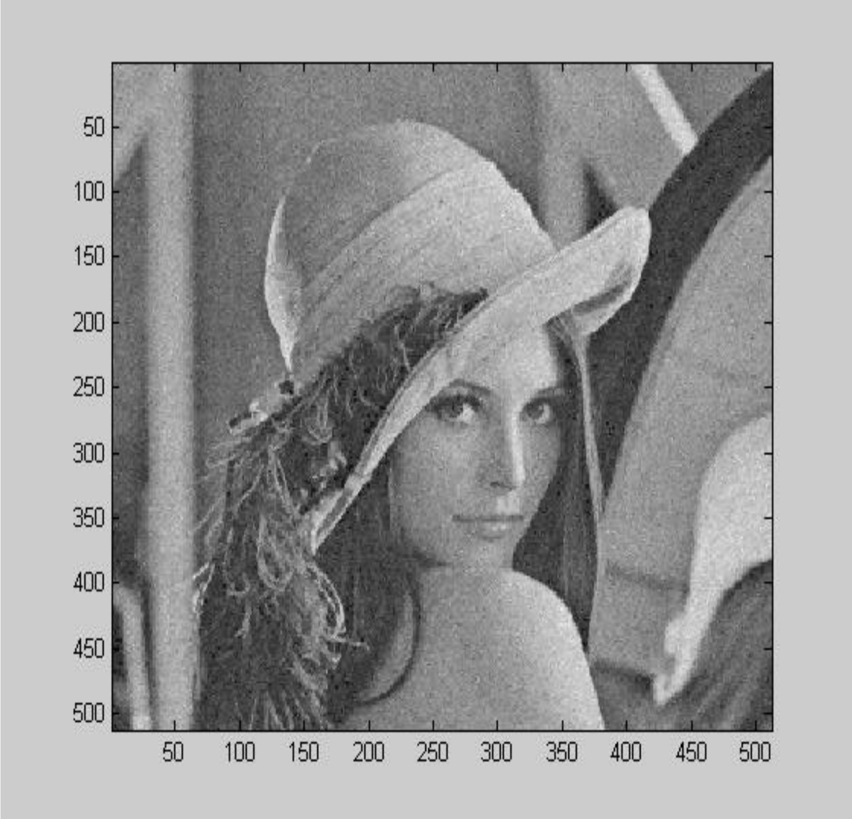


Image Enhancement

- Image contrast enhancement with wavelets, especially important in medical imaging
- Make the small coefficients very small and the large coefficients very large.
- Apply a nonlinear mapping function to the coefficients.

Experiments


(a) Original Image

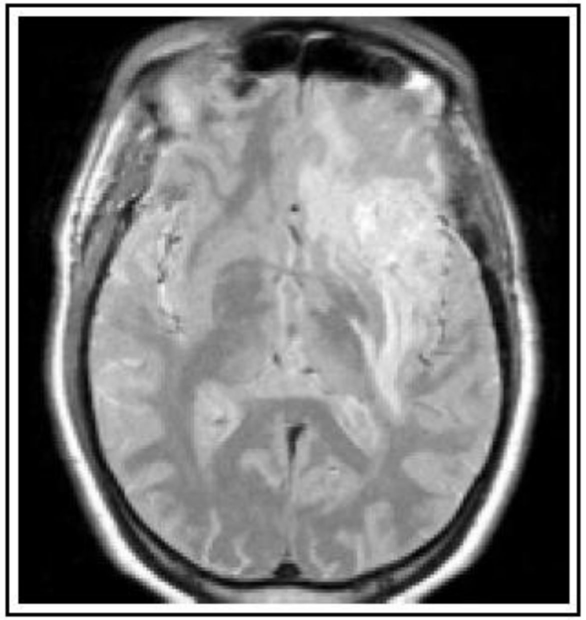

(c) Proposed Method

Denoising and Enhancement

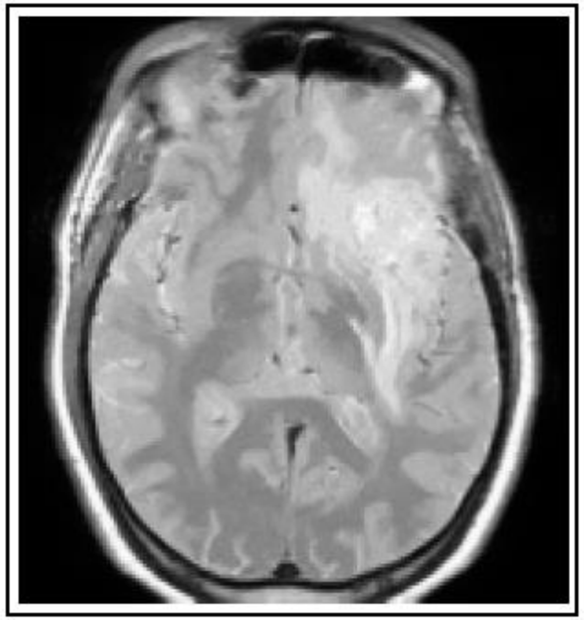
- Apply DWT
- Shrink transform coefficients in finer scales to reduce the effect of noise
- Emphasize features within a certain range using a nonlinear mapping function
- Perform IDWT to reconstruct the image.

Examples

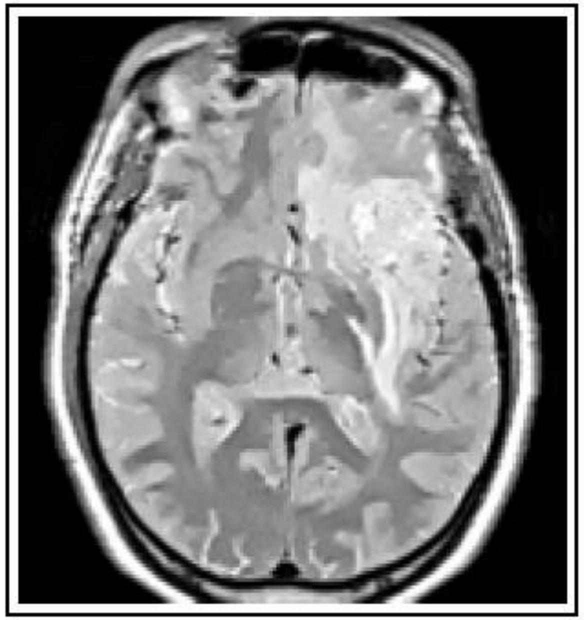
Original | Denoised | Denoising with Enhancement



(a)



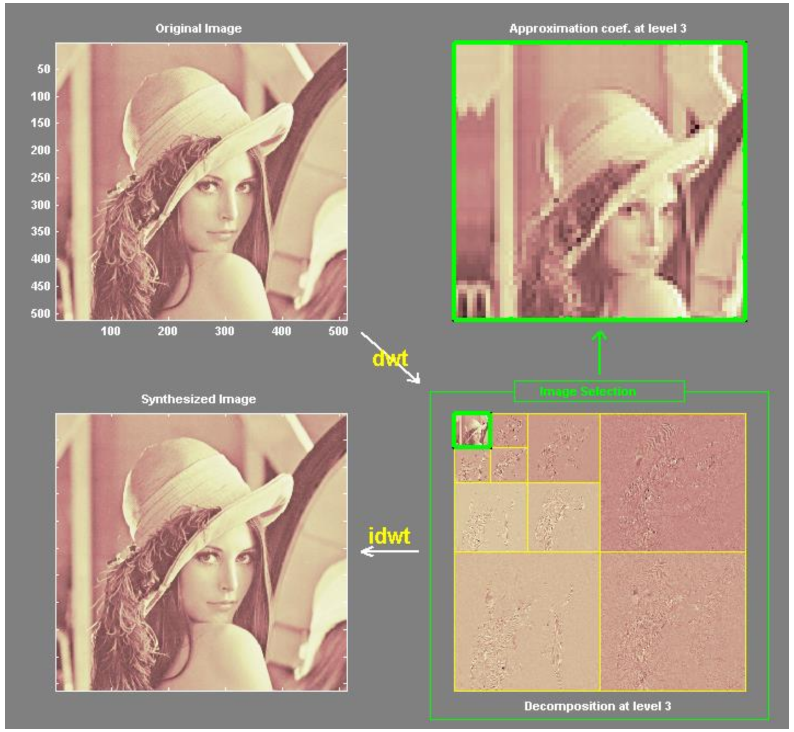
(b)

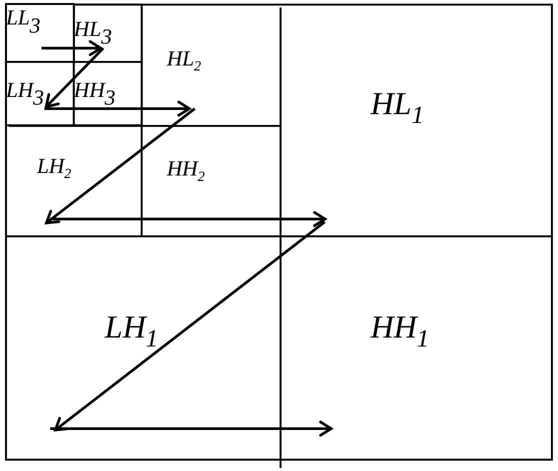


(c)

DWT for Image Compression

- **Image Decomposition**
  - Feature 1:
    - Energy distribution concentrated in low frequencies
  - Feature 2:
    - Spatial self-similarity across subbands





The scanning order of the subbands for encoding the significance map.