Image Transforms

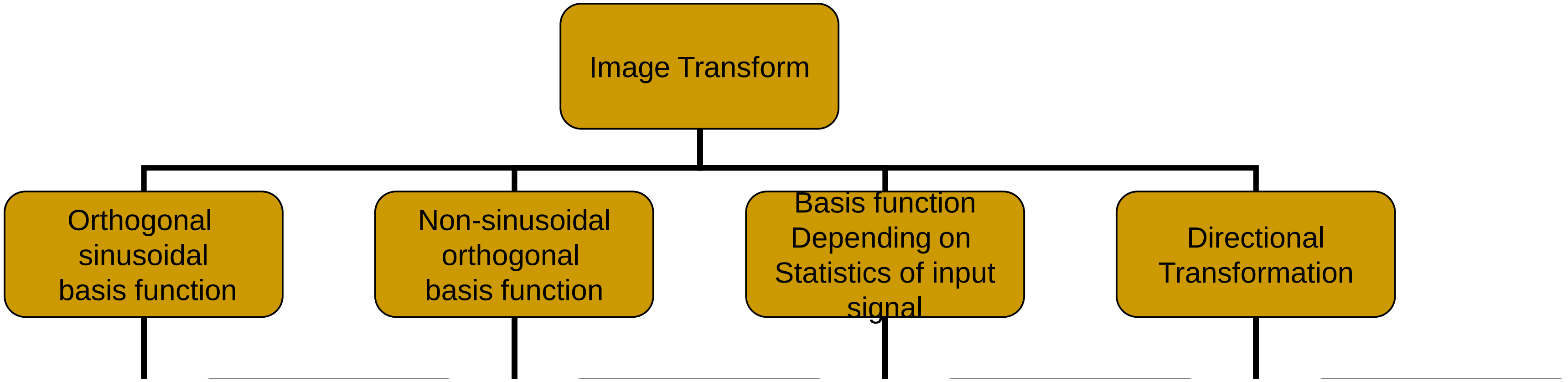
Why Image Transform?

- Allows to move from one domain to another to perform a task in an easier manner
- Useful for fast computation of convolution and correlation
- Changes the representation of signal
- Does not change the information content of the signal
- Transform is reversible, i.e., transform domain can revert to the spatial domain

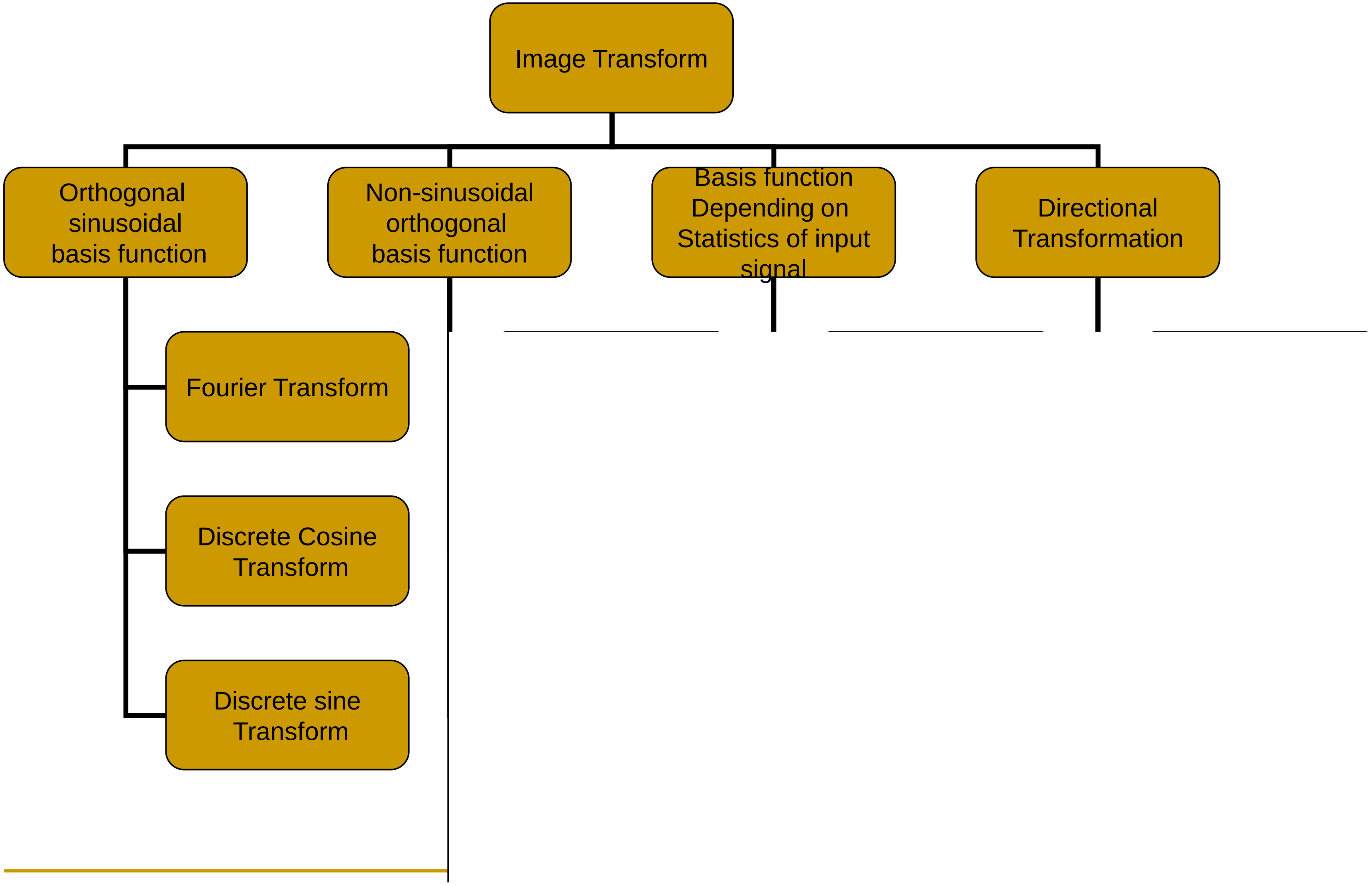
Why Image Transform? Contd..

- May isolate critical components of the image pattern so that they are directly accessible for analysis
- Information present in the image is preserved in the transformed domain
- Used for image analysis, enhancement, filtering and compression
- Ex: DCT gives information about frequency content of the image

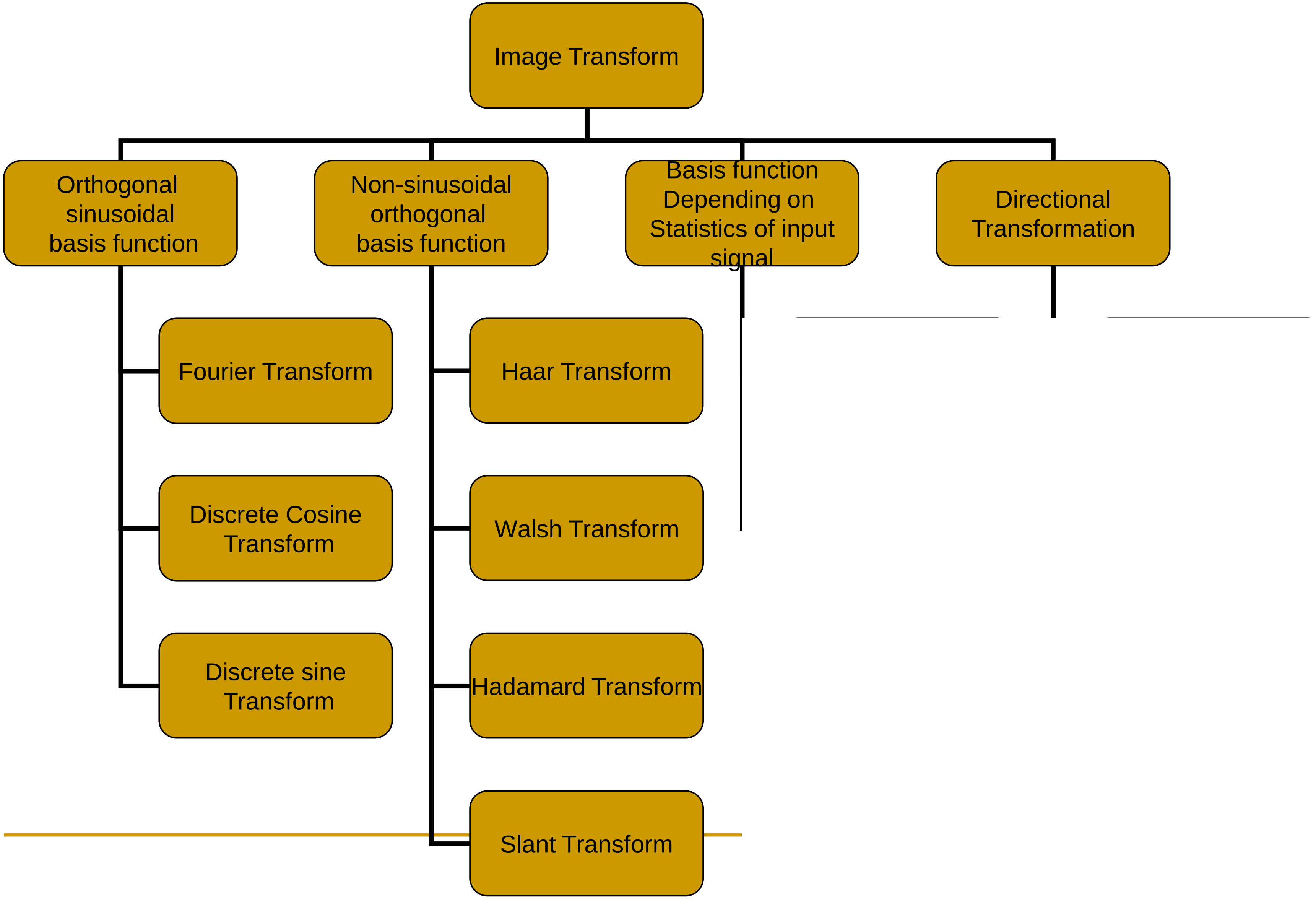
Classification of Image Transforms



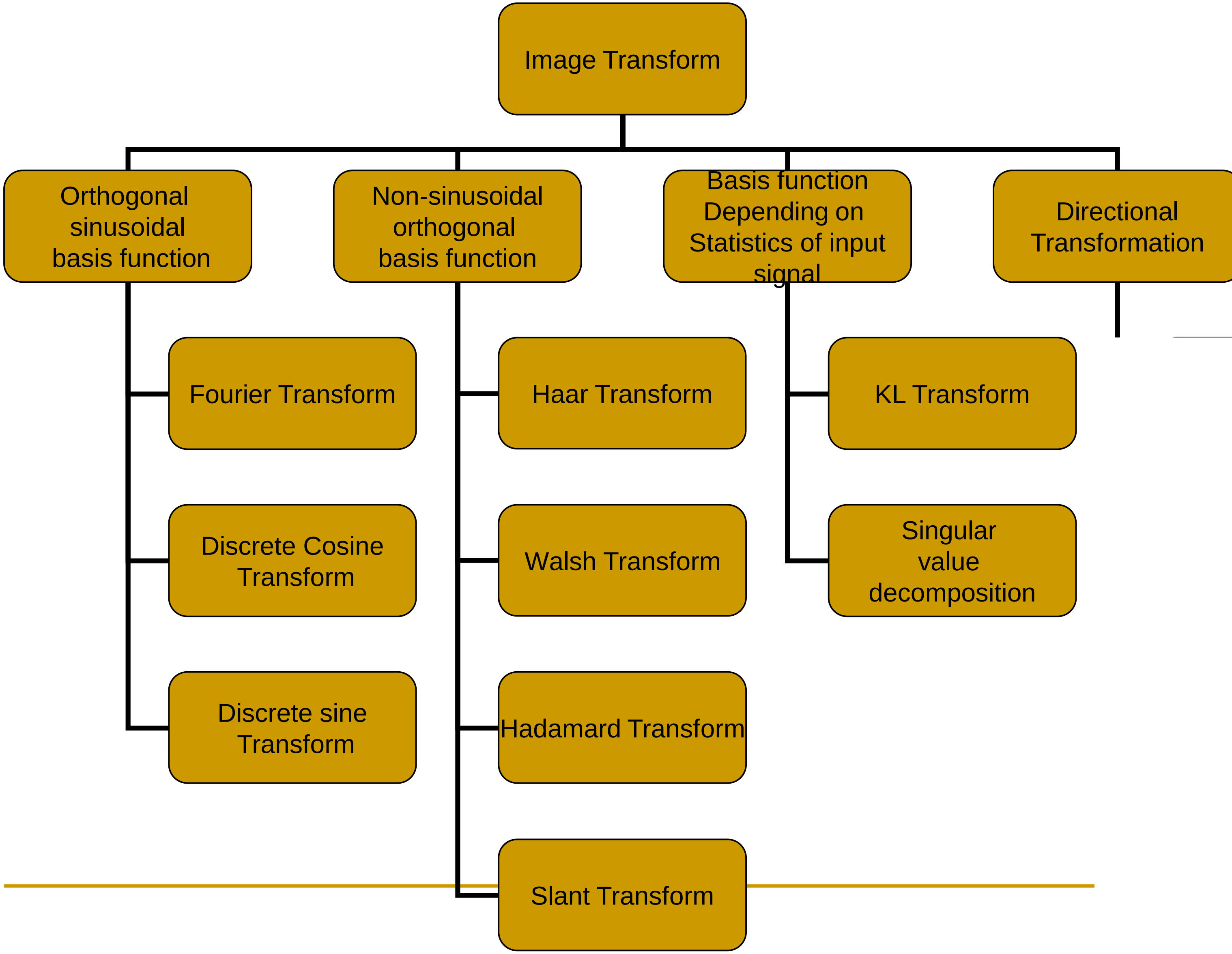
Classification of Image Transforms



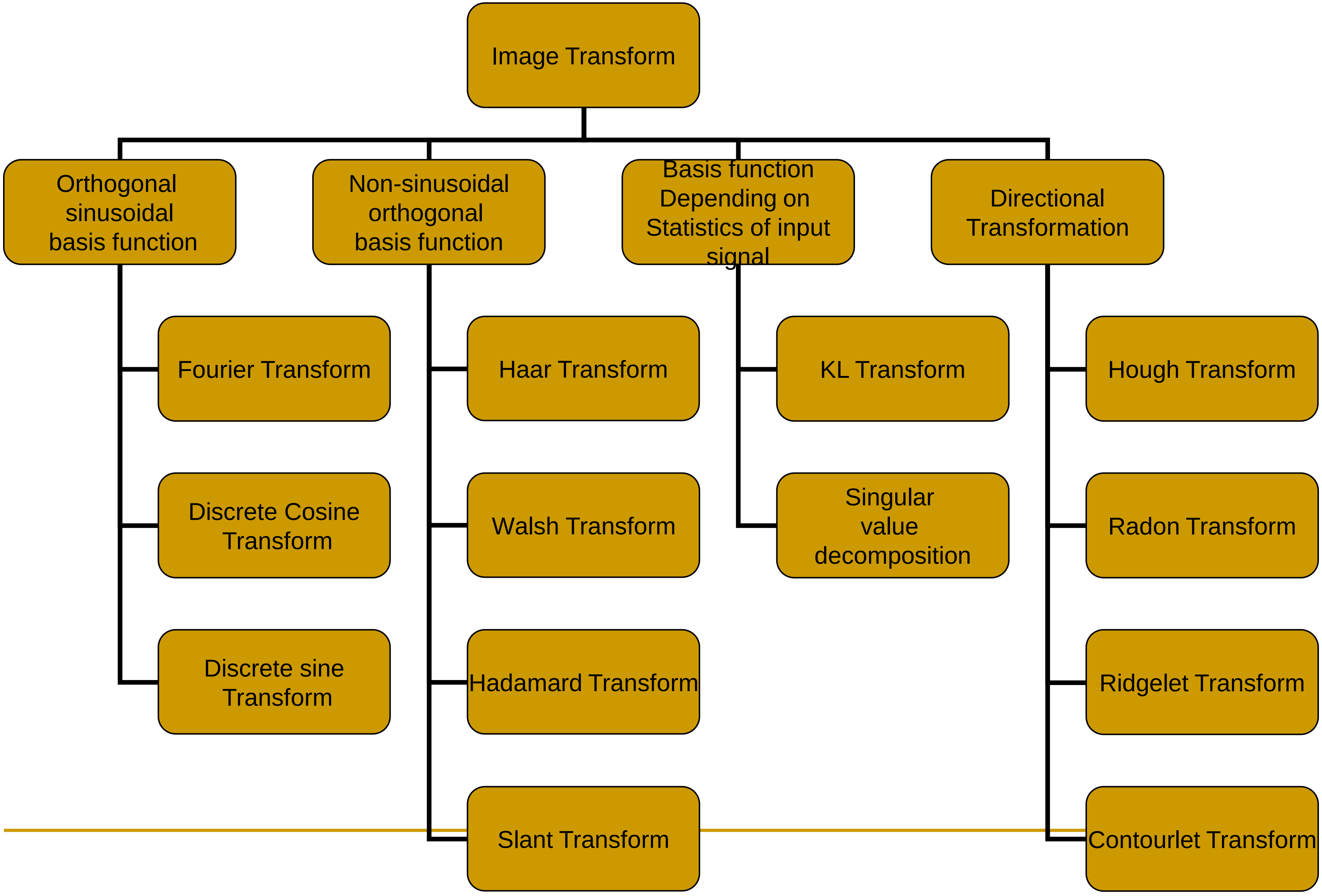
Classification of Image Transforms



Classification of Image Transforms



Classification of Image Transforms



Unitary Transform

- A discrete linear transform is unitary if its transform matrix follows unitary condition

  $A \times A^H = I$

- A is transform matrix, A^H represents Hermitian matrix and I is identity matrix
- $A^H = A^{*T} = A^{-1}$
- A^* and A^T are complex conjugate and transpose of A respectively
- For an image, f, transform is F

  $F = A f A^T$, $f = A^T F A$

Walsh Transform

- Generalized class of Fourier Transform
- Uses 2-point DFT
- 2-dft matrix is $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
- Based on orthogonal square wave functions
- Coefficients are real (+1 or -1)
- Therefore computationally simple

Walsh Transform

$$g(n,k)=\frac{1}{N}\prod_{i=0}^{m-1}(-1)^{b_i(n)b_{m-1-i}(k)}$$

N is order of matrix
m is number of bits to represent N
$b_i(n)$ is i^{th} bit from LSB

Walsh Transform Matrix

$$g(n,k)=\frac{1}{N}\prod_{i=0}^{m-1}(-1)^{b_i(n)b_{m-1-i}(k)}$$

- N is size of matrix, m = $\log_2 N$, n,k=0,1,..,N-1, b_i is i^{th} bit of n
- N = 4, m = $\log_2 4$ = 2 bits

Decimal	binary	
n	$b_1(n)$	$b_0(n)$
0	0	0
1	0	1
2	1	0
3	1	1

$$g(2,3)=\frac{1}{4}\left[(-1)^{b_0(2)b_{2-1-0}(3)}(-1)^{b_1(2)b_{2-1-1}(3)}\right]$$

$$=\frac{1}{4}\left[(-1)^{b_0(2)b_1(3)}(-1)^{b_1(2)b_0(3)}\right]$$

$$=\frac{1}{4}\left[(-1)^{0\times 1}(-1)^{1\times 1}\right]$$

$$=-\frac{1}{4}$$

Walsh Transform Matrix (N=4)

$$g(n,k) = \frac{1}{N} \prod_{i=0}^{m-1} (-1)^{b_i(n) b_{m-1-i}(k)}$$

- N = 4, m = $\log_2 4$ = 2

Dec	binary	
n	$b_1(n)$	$b_0(n)$
0	0	0
1	0	1
2	1	0
3	1	1

g(n,k)

n ↓ k→	0	1	2	3
0				
1				
2				-1/4
3				

g(2,3)

Walsh Transform Matrix (N=4)

$$g(n,k)=\frac{1}{N}\prod_{i=0}^{m-1}(-1)^{b_i(n)b_{m-1-i}(k)}$$

- N = 4, m = $\log_2 4$ = 2

g(n,k)

Dec	binary	
n	$b_1(n)$	$b_0(n)$
0	0	0
1	0	1
2	1	0
3	1	1

n ↓ k →	0	1	2	3
0	1/4	1/4	1/4	1/4
1	1/4	1/4	-1/4	-1/4
2	1/4	-1/4	1/4	-1/4
3	1/4	-1/4	-1/4	1/4

Walsh Matrix

A =

n k→	0	1	2	3
0	1/4	1/4	1/4	1/4
1	1/4	1/4	-1/4	-1/4
2	1/4	-1/4	1/4	-1/4
3	1/4	-1/4	-1/4	1/4

$$A = g(n,k) = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

- f(x,y) is image, A is transform matrix
- $F(u,v)=A\, f(x,y)\, A^T$
- $f(x,y)=N^2[A^T\, F(u,v)\, A)]$

Hadamard Transform

- Elements of basis function are +1 or -1 which result in very low computational complexity.
- Generally $N = 2^n$
- Also called Walsh Hadamard because elements are +1 or -1
- Used for data encryption, data compression (JPEG-XR, MPEG 4) and video compression

Hadamard matrix

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

H is real, symmetric and orthogonal

$H = H^* = H^T = H^{-1}$

Computation of Hadamard transform

F(x,y) is image, A is transform matrix

$F(u,v)=A\,f(x,y)\,A^T$

$f(x,y)=[A^T\,F(u,v)\,A]/N^2$

$$f=\begin{bmatrix} 2 & 4 & 1 & 6 \\ 3 & 2 & 1 & 4 \\ 6 & 5 & 3 & 2 \\ 1 & 2 & 4 & 3 \end{bmatrix} \qquad A=H_4=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$F=\begin{bmatrix} 49 & -7 & 1 & 5 \\ 9 & -3 & 9 & 1 \\ -3 & -11 & -3 & 9 \\ -3 & -7 & -11 & -3 \end{bmatrix}$$

Hadamard Transform for N = 8

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Properties of Hadamard Transform

- H is real, symmetric and orthogonal

  $H = H^* = H^T = H^{-1}$
- Fast transform. No multiplication are required.
- Good energy compaction for highly correlated images

Slant Transform

- Order $N=2^n$
- Used for watermarking
- Used for detecting diagonal edges
- Fast computational algorithm is possible

Slant Transform matrix(N=4, n=2)

$$S_{2^1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$a_1 = 1$$
$$a_n = 2b_n a_{n-1}$$
$$b_n = \left(1 + 4a_{n-1}^2\right)^{-1/2}$$

$$S_{2^n} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc:c:cc:c} 1 & 0 & \multirow{2}{*}{0} & 1 & 0 & \multirow{2}{*}{0} \\ a_n & b_n & & -a_n & b_n & \\ \hdashline \multicolumn{2}{c:}{0} & I_{(N/2)-2} & \multicolumn{2}{c:}{0} & I_{(N/2)-2} \\ \hdashline 0 & 1 & \multirow{2}{*}{0} & 0 & -1 & \multirow{2}{*}{0} \\ -b_n & a_n & & b_n & a_n & \\ \hdashline \multicolumn{2}{c:}{0} & I_{(N/2)-2} & \multicolumn{2}{c:}{0} & -I_{(N/2)-2} \end{array}\right]\left[\begin{array}{c:c} S_{2^{n-1}} & 0 \\ \hdashline 0 & S_{2^{n-1}} \end{array}\right]$$

I_N is identity matrix of size N×N

O is all zero matrix

Slant Transform matrix(N=4, n=2)

$$S_{2^1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$a_1 = 1$$
$$a_n = 2b_n a_{n-1}$$
$$b_n = \left(1 + 4a_{n-1}^2\right)^{-1/2}$$

$$S_{2^n} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc:c:cc:c} 1 & 0 & \multirow{2}{*}{0} & 1 & 0 & \multirow{2}{*}{0} \\ a_n & b_n & & -a_n & b_n & \\ \hdashline \multicolumn{2}{c:}{0} & I_{(N/2)-2} & \multicolumn{2}{c:}{0} & I_{(N/2)-2} \\ \hdashline 0 & 1 & \multirow{2}{*}{0} & 0 & -1 & \multirow{2}{*}{0} \\ -b_n & a_n & & b_n & a_n & \\ \hdashline \multicolumn{2}{c:}{0} & I_{(N/2)-2} & \multicolumn{2}{c:}{0} & -I_{(N/2)-2} \end{array}\right]\left[\begin{array}{c:c} S_{2^{n-1}} & 0 \\ \hdashline 0 & S_{2^{n-1}} \end{array}\right]$$

$$a_1 = 1$$
$$b_2 = \left(1 + 4a_1^2\right)^{-1/2} = 1/\sqrt{5}$$
$$a_2 = 2b_2 a_1 = 2/\sqrt{5}$$

$$S_{2^2} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 & 0 \\ a_2 & b_2 & -a_2 & b_2 \\ 0 & 1 & 0 & -1 \\ -b_2 & a_2 & b_2 & a_2 \end{bmatrix}\begin{bmatrix} S_1 & 0 \\ 0 & S_1 \end{bmatrix}$$

Slant Transform matrix (N=4)

$$b_2 = 1/\sqrt{5}, a_2 = 2/\sqrt{5}$$

$$S_{2^2} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 & 0 \\ a_2 & b_2 & -a_2 & b_2 \\ 0 & 1 & 0 & -1 \\ -b_2 & a_2 & b_2 & a_2 \end{bmatrix}\begin{bmatrix} S_1 & 0 \\ 0 & S_1 \end{bmatrix}$$

$$S_{2^2} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 & 0 \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ 0 & 1 & 0 & -1 \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$S_{2^2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

Slant Transform matrix(N=8, n=3)

$$a_1 = 1$$
$$a_n = 2b_n a_{n-1}$$
$$b_n = \left(1 + 4a_{n-1}^2\right)^{-1/2}$$

$$a_3 = \sqrt{\frac{3 \times 8 \times 8}{8 \times 8 \times 8 - 1}} = \frac{4}{\sqrt{21}} \quad b_3 = \sqrt{\frac{8 \times 8 - 1}{8 \times 8 \times 8 - 1}} = \sqrt{\frac{5}{21}}$$

$$b_2 = 1/\sqrt{5}, a_2 = 2/\sqrt{5}$$

Slant Transform matrix(N=8, n=3)

$$a = a_3 = \frac{4}{\sqrt{21}} \qquad b = b_3 = \sqrt{\frac{5}{21}}$$

$$S_n = S_{2^n} = S_{2^3} = \left[\begin{array}{cc:cc:cc:cc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a & b & 0 & 0 & -a & b & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -b & a & 0 & 0 & b & a & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} S_{2^2} & 0_{2^2} \\ 0_{2^2} & S_{2^2} \end{bmatrix}$$

Properties of Slant Transform

- Real
- $S=S^*$ and $S^{-1}=S^T$
- Good energy compaction

Discrete Cosine Transform

- Real valued and unitary
- Basis vectors are sampled form of cosine signal
- Some of the DCT coefficients are ignored (converted to zeros)
- Therefore, DCT transform is widely used for image compression

2-D DCT matrix

$$A = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix}$$

- $f(x,y) \rightarrow F(u,v)$ and $F(u,v) \rightarrow f(x,y)$
- $F = AfA^T$
- $f = A^T F A$

Example DCT Transform

- Determine DCT of the following image matrix and prove that DCT has good energy compaction property
- Repeat the above for slant transform of the image

$$f = \begin{bmatrix} 2 & 3 & 3 & 3 \\ 2 & 3 & 3 & 2 \\ 3 & 2 & 2 & 2 \\ 2 & 3 & 2 & 3 \end{bmatrix}$$

Computation of 2-D DCT matrix

$$f = \begin{bmatrix} 2 & 3 & 3 & 3 \\ 2 & 3 & 3 & 2 \\ 3 & 2 & 2 & 2 \\ 2 & 3 & 2 & 3 \end{bmatrix} \qquad A = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix}$$

$$Af = \begin{bmatrix} 4.5 & 5.5 & 5 & 5.0 \\ -0.27 & 0.27 & 0.92 & 0 \\ -0.5 & 0.5 & 0.0 & 1.0 \\ 0.65 & -0.65 & -0.38 & 0.0 \end{bmatrix}$$

$$AfA^T = F = \begin{bmatrix} 10 & -0.19 & -0.5 & -0.46 \\ 0.46 & -0.35 & -0.73 & 0.35 \\ 0.5 & -0.84 & 0.0 & -0.73 \\ -0.19 & 0.35 & 0.84 & 0.35 \end{bmatrix}$$

Energy of DCT Transform

- Energy of each DCT coefficient
- $E(u,v) = \{F(u,v)\}^2$

$$E = \begin{bmatrix} 10^2 & (-0.19)^2 & (-0.5)^2 & (-0.46)^2 \\ (0.46)^2 & (-0.35)^2 & (-0.73)^2 & (0.35)^2 \\ (0.5)^2 & (-0.84)^2 & 0.0 & (-0.73)^2 \\ (-0.19)^2 & (0.35)^2 & (0.84)^2 & (0.35)^2 \end{bmatrix}$$

$$E = \begin{bmatrix} 100 & 0.04 & 0.25 & 0.21 \\ 0.21 & 0.12 & 0.54 & 0.12 \\ 0.25 & 0.71 & 0.0 & 0.54 \\ 0.04 & 0.12 & 0.71 & 0.12 \end{bmatrix}$$

Energy of DCT Transform

$$E = \begin{bmatrix} 100 & 0.04 & 0.25 & 0.21 \\ 0.21 & 0.12 & 0.54 & 0.12 \\ 0.25 & 0.71 & 0.0 & 0.54 \\ 0.04 & 0.12 & 0.71 & 0.12 \end{bmatrix}$$

- Total energy= sum(E)=104
- Percentage of total energy contained in E(0,0)
- E(0,0)=100(100/104) = 96%
- Percentage of total energy contained in first four coefficients

  = (100+0.04+0.21+0.12)100/104 = 96.5%

Example DCT

- 256x256 image
- Image as 65596 pixels

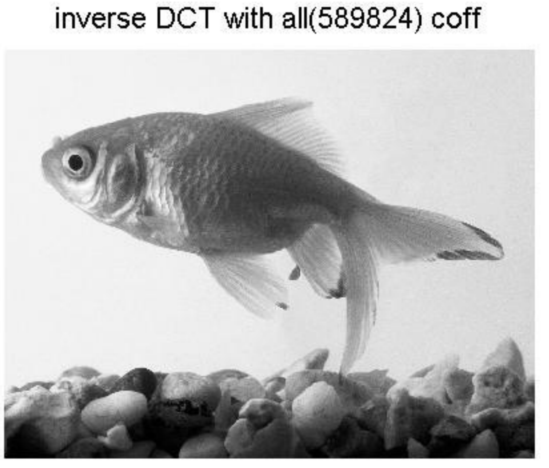
Image

DCT Coefficients

- Low frequency coefficients have higher values than higher frequencies

DCT compression for 768x768 image

- Original image has 589824 pixels
- DCT of image as 589824 pixels



589824 coefficients are stored and used for inverse DCT

DCT compression for 768x768 image

- Original image has 589824 pixels
- DCT of image as 589824 pixels

inverse DCT with all(589824) coff

589824 coefficients are stored and used for inverse DCT

inverse DCT with 943/589824 coeff

- Highest 943 coefficients are stored and used for inverse DCT
- Other coefficients are ignored (converted to zeros)
- Thus image is compressed

Example DCT

- Original image has 589824 pixels
- DCT of image as 589824 pixels

inverse DCT with all(589824) coff

589824 coefficients are stored and used for inverse DCT

inverse DCT with 59/589824 coeff

Highest 59 coefficients are stored and used for inverse DCT

Fourier transform and Wavelet transform

Fourier	Wavelet
• Basis function is a sinusoid • Provides only frequency information, temporal information is lost	• Basis function is a small wavelet of varying frequency and limited time duration • Provides time as well as frequency information

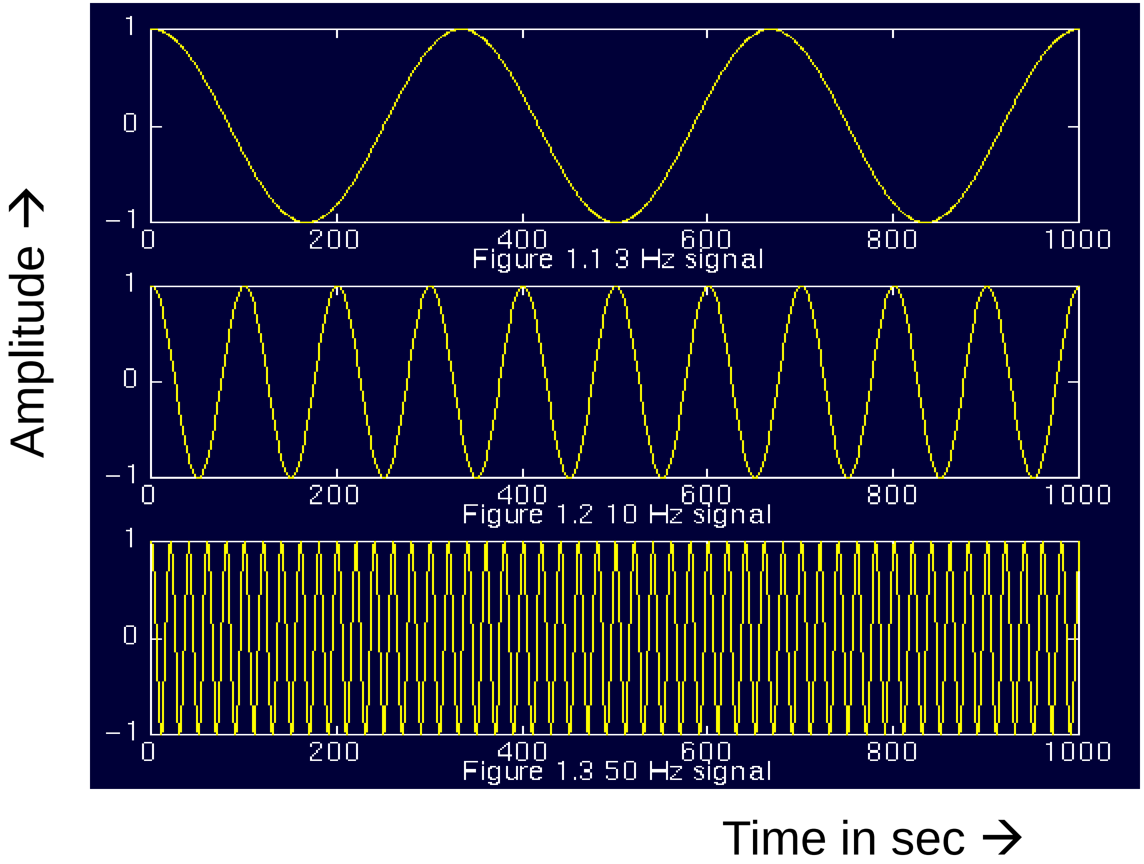
Background for Wavelet Transforms

Signals

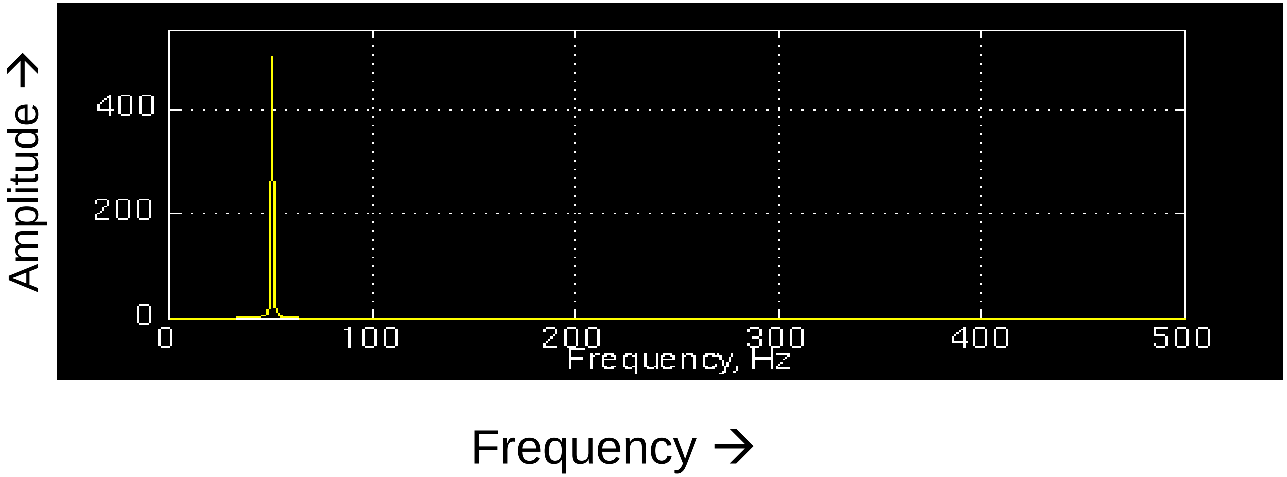
- Stationary or Time signals:
  - aveforms of constant frequency throughout the time
  - Can be processed by Fast Fourier Transform (FFT)

- Non Stationary Signals
  - Frequency content change over time
  - Example: Real world signals like health of human heart, blip on Radar
  - Can be processed by Wavelets

Time domain signal



Fourier Transform for 50 Hz signal

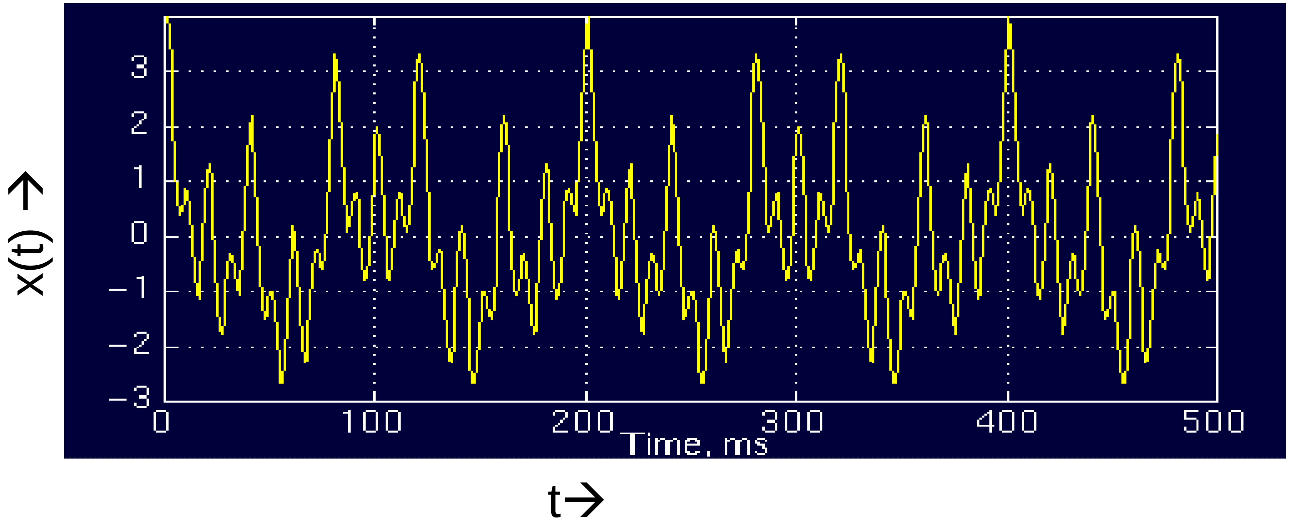


Fourier Transform (FT)

- FT gives the frequency information of the signal
- Gives information about the amplitude of each frequency which exists in the signal
- Does not tell when in time the frequency components exist
- Time information is not required when the signal is stationary
- For non-stationary signal it is required to know about the time frequency components exist
- FT assumes that all frequency components exist at all times

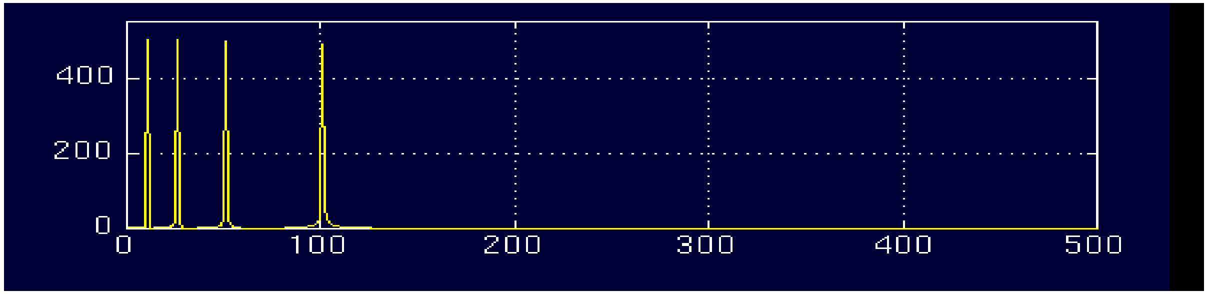
Stationary signal

- $x(t)=\cos(2\pi 10t)+\cos(2\pi 25t)+\cos(2\pi 50t)+\cos(2\pi 100t)$
- Signal has components at 10, 25, 50 and 100 Hz



Fourier Transform of stationary signal

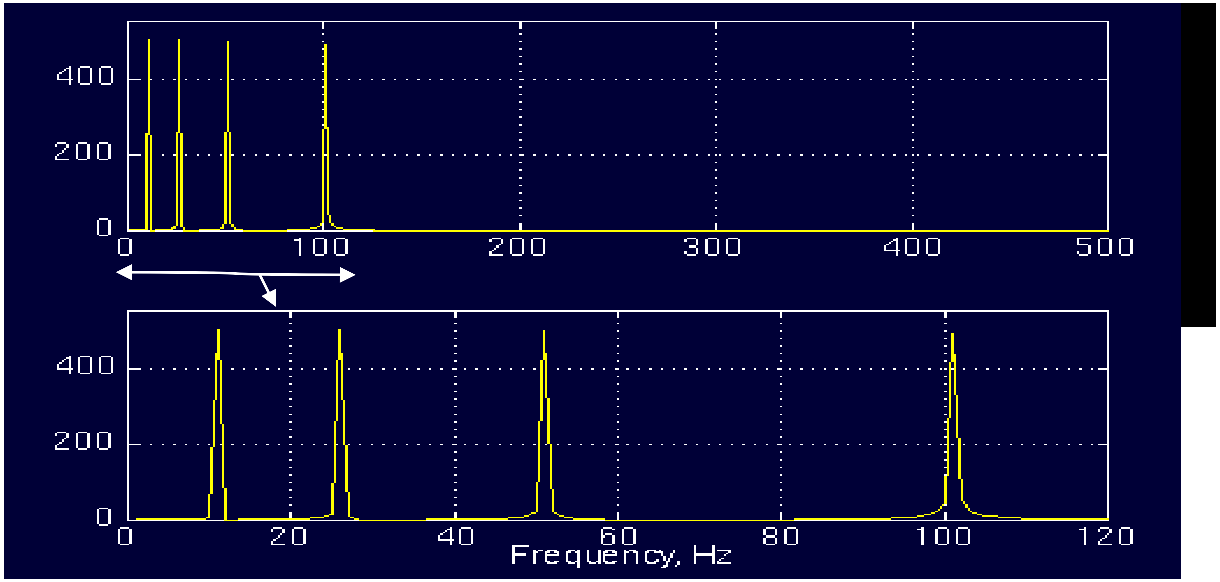
- Frequency components are 10, 25, 50 and 100 Hz
- Frequency components exist for the entire duration of the signal



Fourier Transform of stationary signal

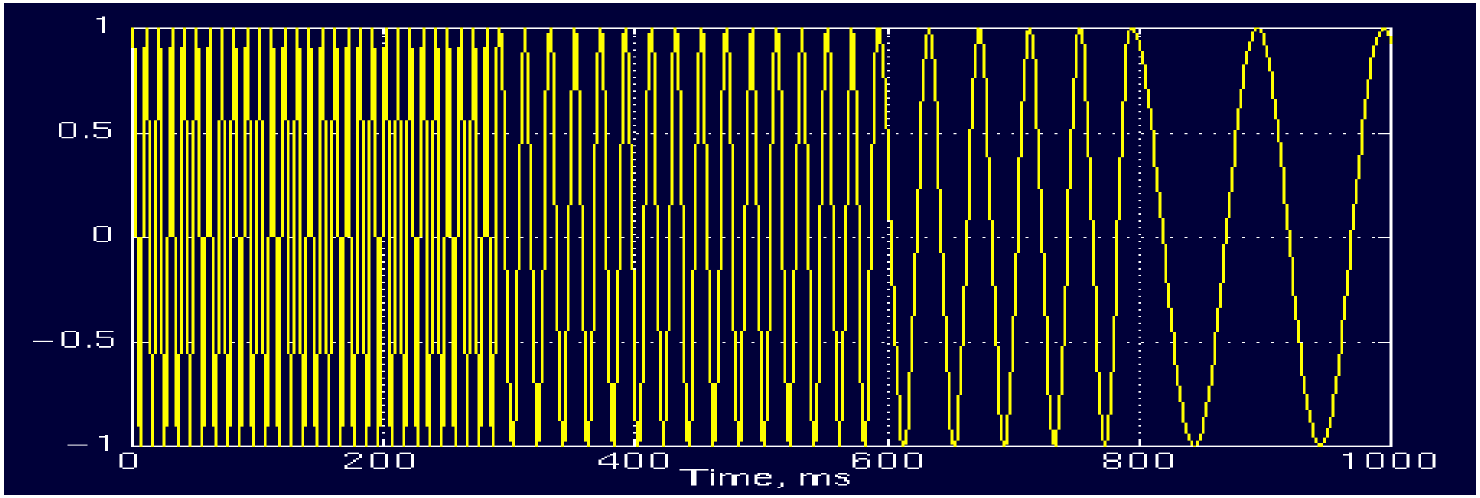
- Frequency components are 10, 25, 50 and 100 Hz
- Frequency components exist for the entire duration of the signal

zoomed



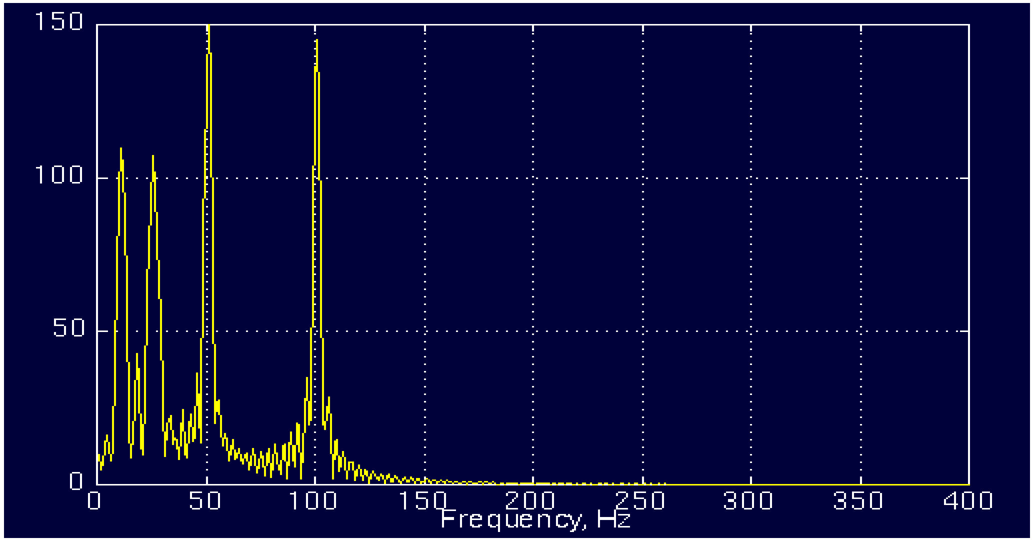
Non-stationary signal

- The interval
  - 0 to 300 ms has a 100 Hz sinusoid,
  - 300 to 600 ms has a 50 Hz sinusoid
  - 600 to 800 ms has a 25 Hz sinusoid
  - 800 to 1000 ms has a 10 Hz sinusoid
- Frequency components change continuously



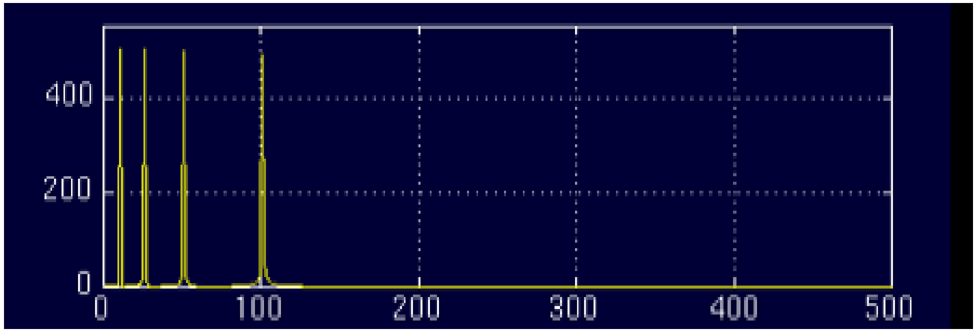
Fourier Transform of non- stationary signal

Frequencies present in the signal are 10 Hz, 25 Hz, 50 Hz and 100 Hz

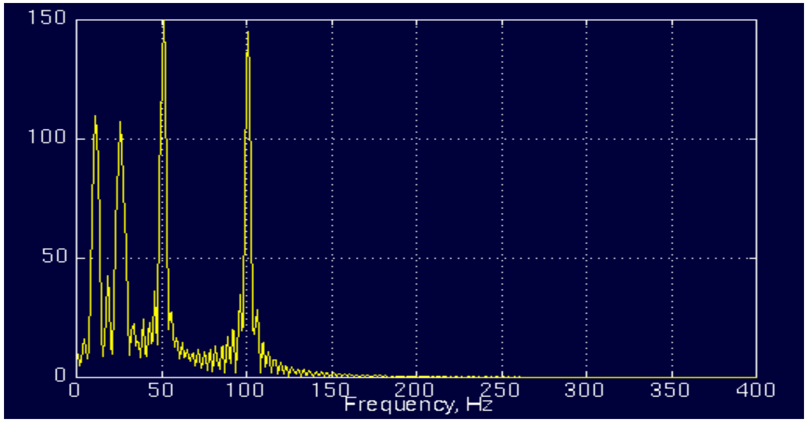


Fourier Transform of non- stationary signal

- FT of stationary and non-stationary signals show four spectral components at exactly the same frequencies, i.e., at 10, 25, 50, and 100 Hz.
- Two spectrums are almost identical
- Corresponding time-domain signals are not similar



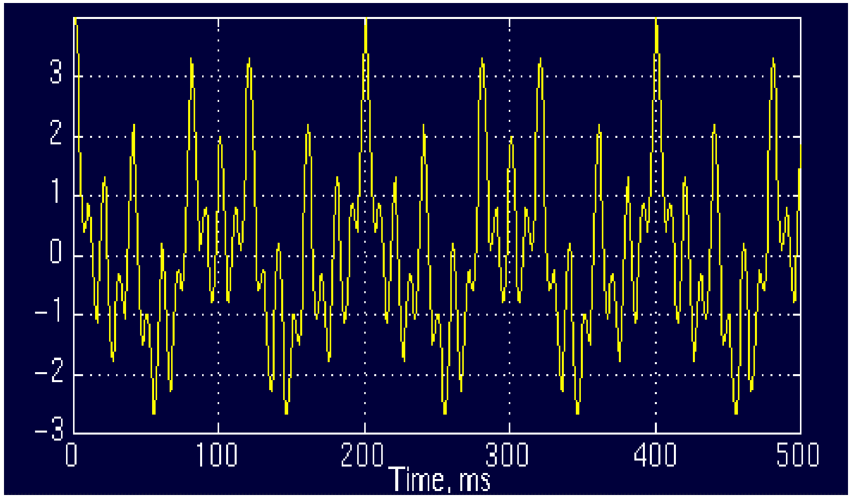
FT of stationary signal

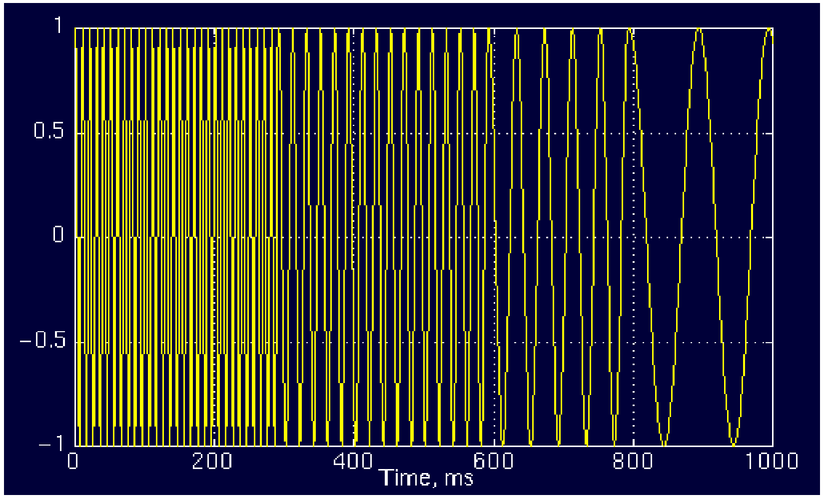


FT of non stationary signal

Fourier Transform of non- stationary signal

- Stationary signal has these frequencies at all times
- Non-stationary signals have these frequencies at different intervals
- FT gives the spectral content of the signal
- FT does not show information regarding where in time those spectral components appear
- FT is not a suitable technique for non-stationary signal

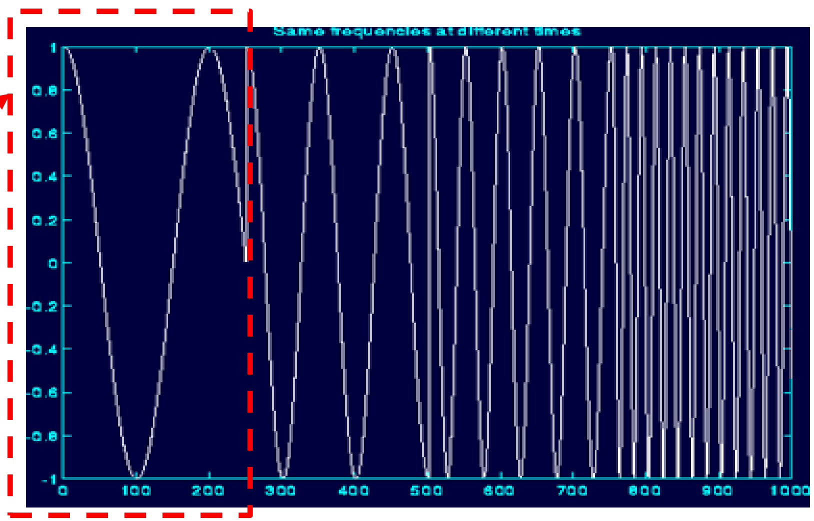




Short Time Fourier Transform

- Assumes that, some portion of a non-stationary signal is stationary
- If this region is too small, signal can be assumed to be stationary in this region
- Window is used to divide signals in small parts
- Window must be narrow enough to assume that the signal is stationary

window



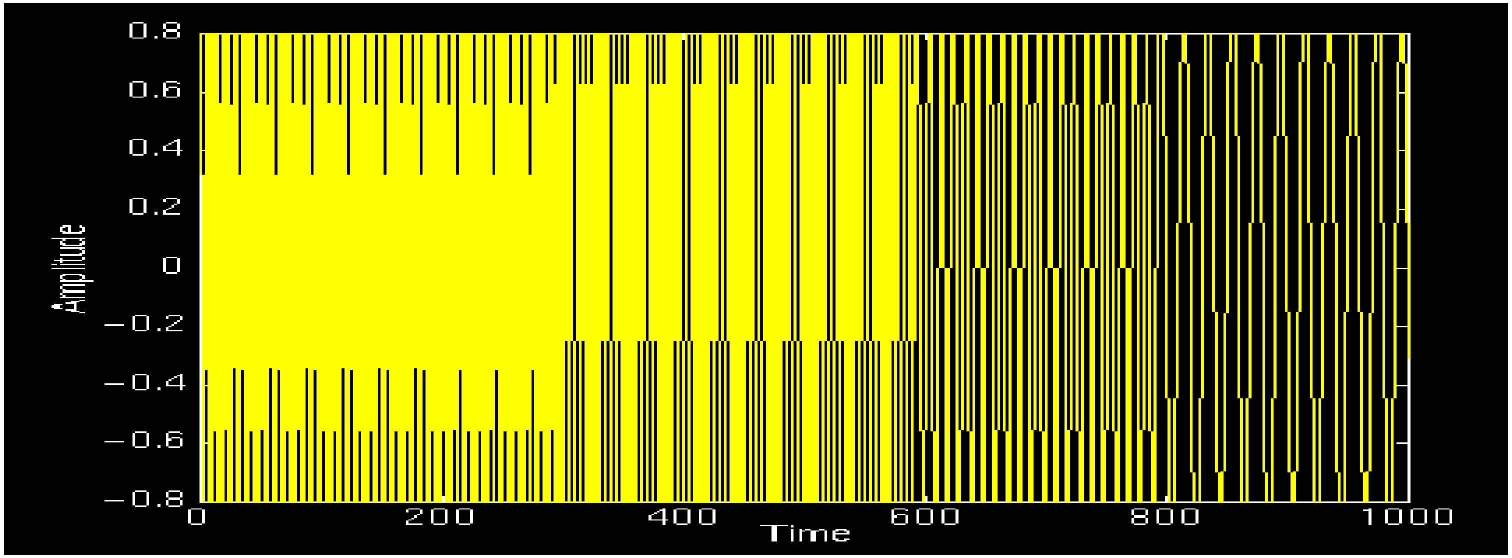
Stationary during 250 time unit intervals.

Short Time Fourier Transform (STFT)

- Revised version of the Fourier transform is The Short Time Fourier Transform
- Signal is divided into small enough segments
- Segments of the signal can be assumed to be stationary
- A window function "w" is chosen
- The width of this window must be equal to the segment of the signal where it is stationary

non-stationary signal

- Four frequency components at different times
- The interval 0 to 250 ms is a sinusoid of 300 Hz
- The other 250 ms intervals are sinusoids of 200 Hz, 100 Hz, and 50 Hz, respectively



Short Time Fourier Transform (STFT)

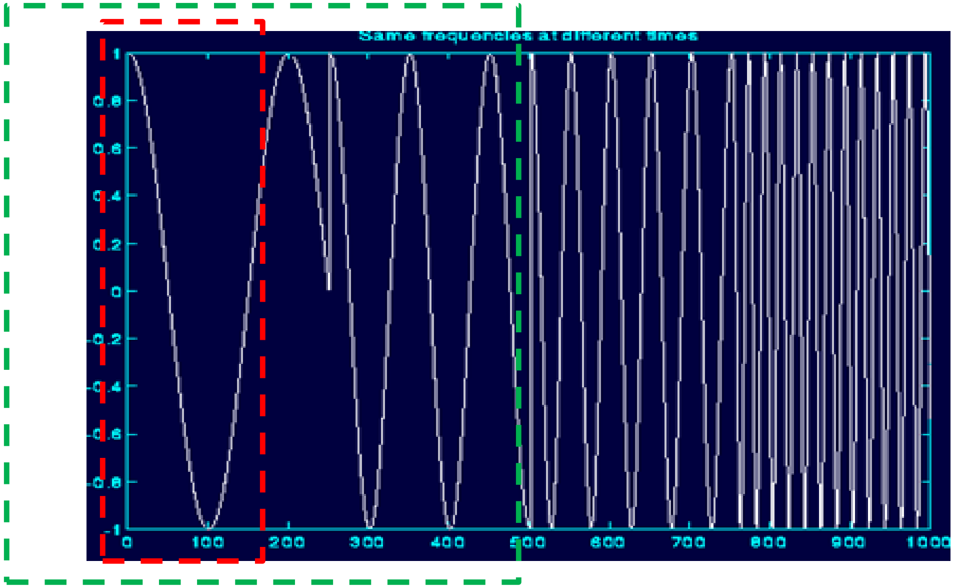
- Divide the total time interval to shorter time intervals and then apply FT for each interval
- The windowing method would narrow down the time to that of interval where the event was found
- STFT gives us information about time and frequency of the signal

Limitation of STFT

- Accuracy is limited by size and shape of the window
- Small window
  - provides good time resolution
  - but very short time of each window would not given good frequency resolution for low frequency signals
  - Because window may not cover the complete portion of the signal
- Long window
  - Provides better frequency resolution
  - but does not provide good time resolution
  - Because it may not cover only stationary signal
  - Also longer time intervals are not needed for high frequency signals

Limitation of STFT

- Accuracy is limited by size and shape of the window



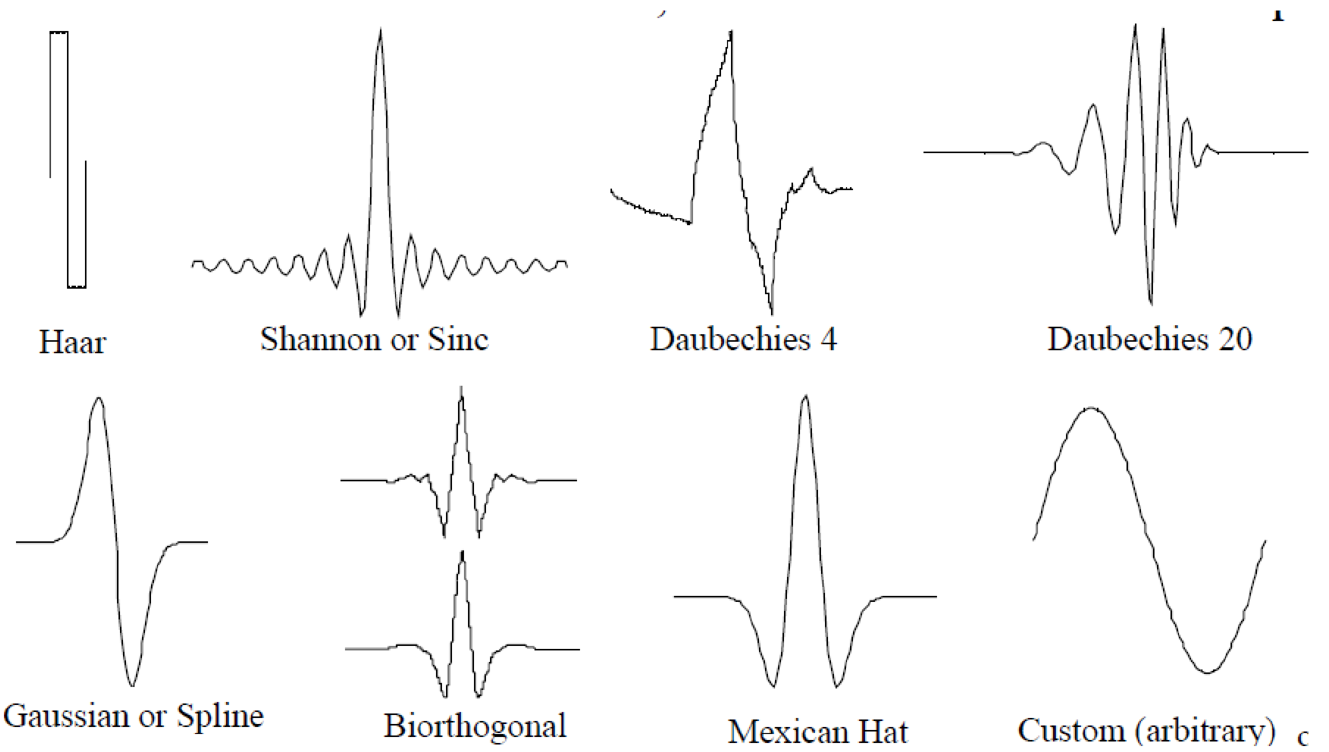
Motivation to use wavelet transform

- Wavelet transform allows variable size window
- Long time intervals can be used for more precise low frequency information
- Shorter intervals (giving more precise time intervals) for high frequency signals

What is Wavelet?

- Waveform of limited duration
- Has definite start and end (Unlike sinusoids that extend from minus infinity to plus infinity)
- Has positive and negative values.
- Average value is zero
- Wavelets are irregular, of limited duration and non symmetrical

Different wavelets



Continuous Wavelet Transform (CWT)

- To compute of signal, x(t), use wavelet $\varphi\left(\frac{t-\tau}{s}\right)$

  x(t) → $X(\tau, s)$

- $X(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\varphi\left(\frac{t-\tau}{s}\right) dt$
- Translation parameter is τ and scale parameters is s
- ϕ(t) is the transforming function and is called the mother wavelet
- The term mother implies that other wavelet functions are derived from mother wavelet
- Therefore daughter wavelets are $\varphi\left(\frac{t-\tau}{s}\right)$ for different values of τ and s
- Therefore mother wavelet is used as a prototype for generating the other window functions

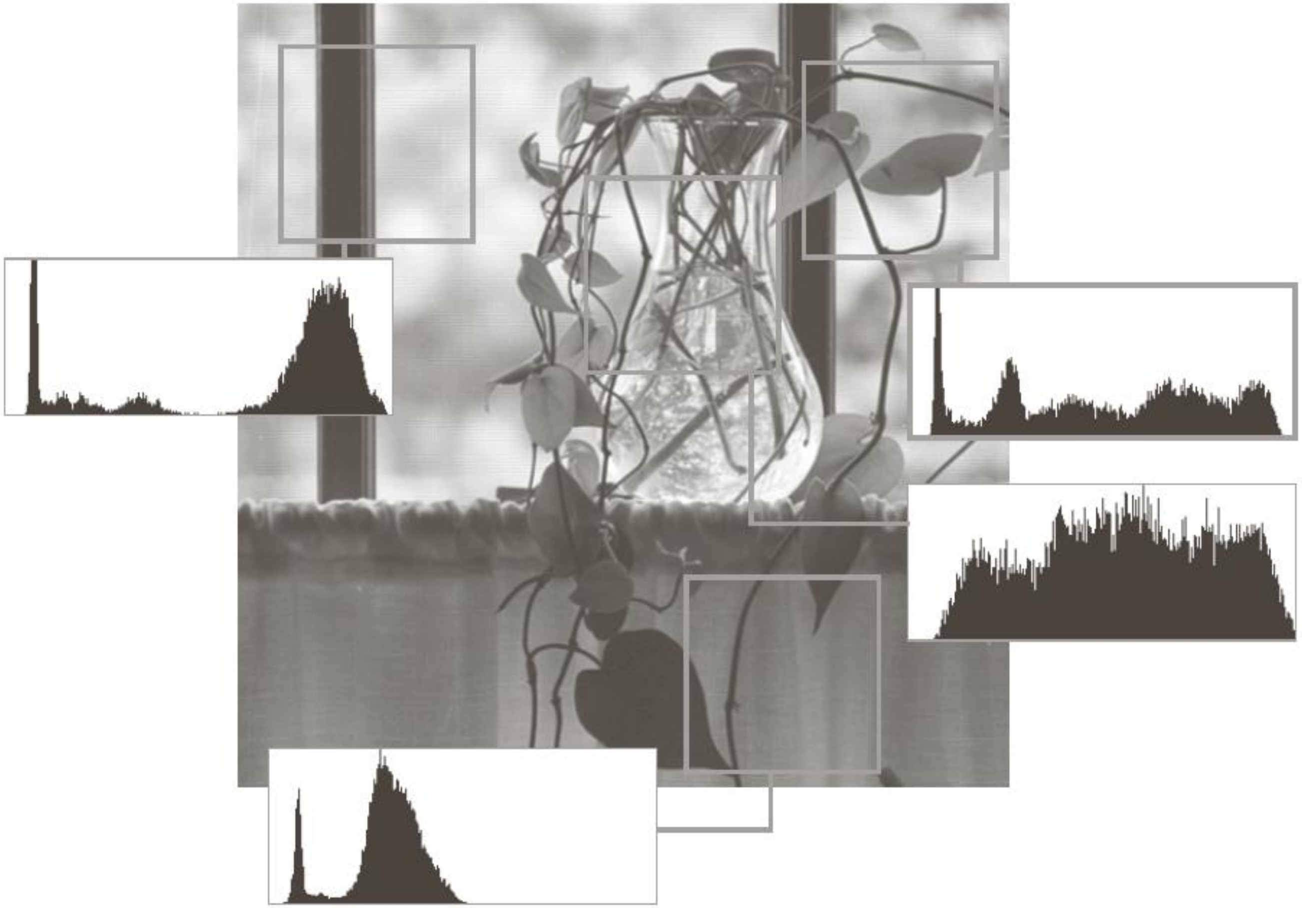
Translation and Scaling

- Translation - Related to the location of the window, as the window is shifted through the signal
- Scaling - Similar to the scale used in maps
  - high scales correspond to a non-detailed global view (of the signal)
  - low scales correspond to a detailed view
  - Similarly, in terms of frequency, low frequencies (high scales) correspond to a global information of a signal
  - global information usually spans the entire signal
  - whereas high frequencies (low scales) correspond to a detailed information of a hidden pattern in the signal

Motivation for Multi-resolution Theory for images

- For images, objects can have similar texture and intensity
- Small and high contrast objects are examined at high resolution (high frequency)
- Big and low contrast objects are examined coarsely with low resolution (low frequency)
- If there are big and small, high and low contrast objects existing simultaneously then they are examined at several resolutions

Multi-resolution Theory for image



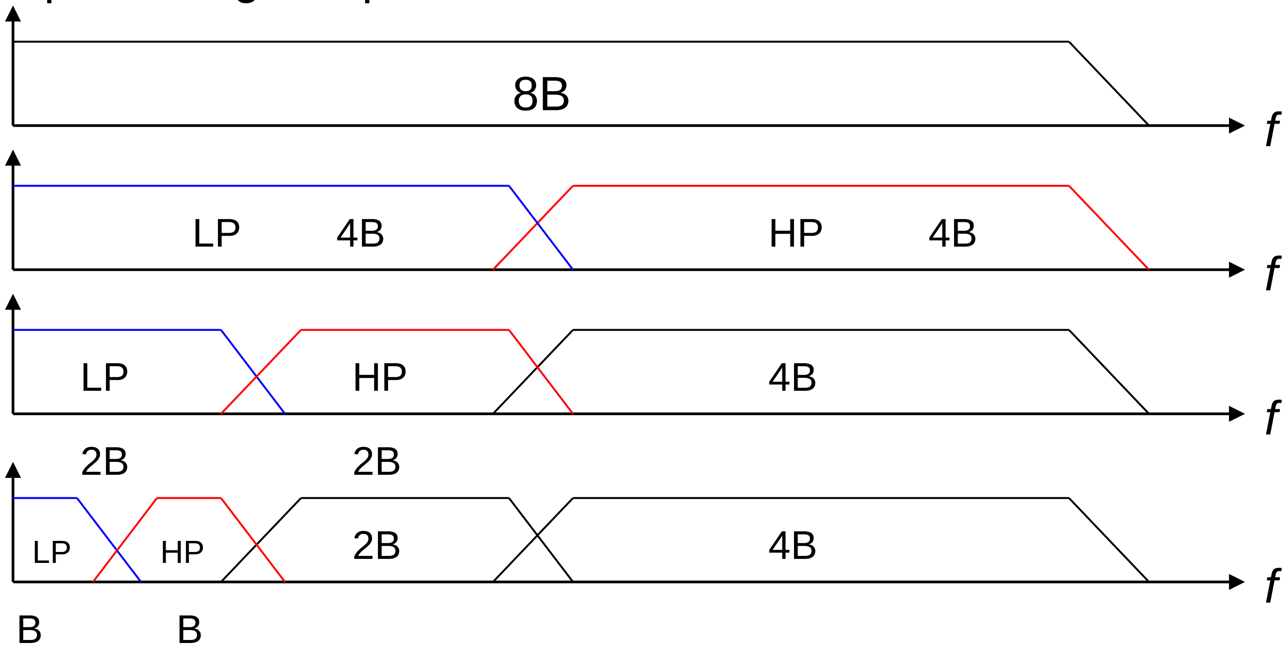
Local histogram of different areas in this image vary significantly from one part to another

Subband Coding

- Image is decomposed into a set of band limited components called sub bands
- Decomposition is performed such that subbands could be reassembled to reconstruct original image without error
- Signal is passed through a filter bank
- Outputs of the different filter stages are the wavelet-time and scaling function transform coefficients
- Wavelet coefficients lead to multiresolution called subband coding.

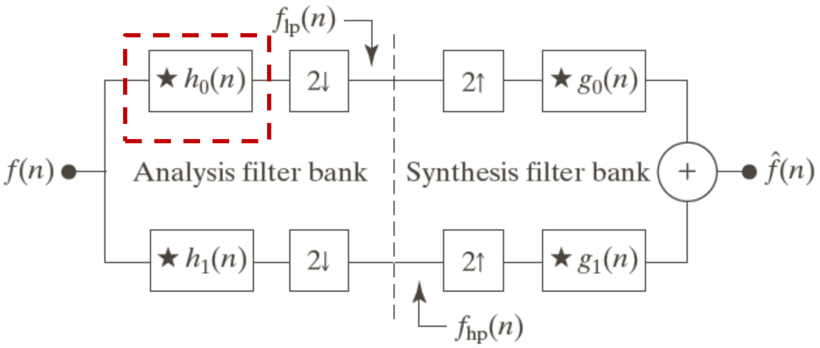
Discrete Wavelet transform as Subband coding

- Split the signal spectrum with an iterated filter bank.

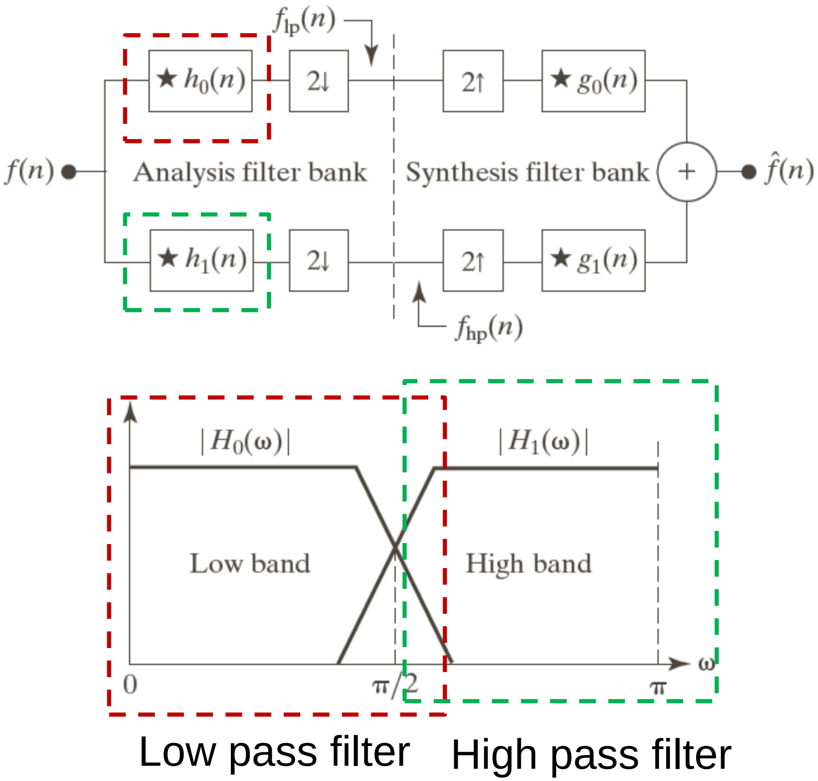


- Wavelet transform is iterative filter bank
- Computation does not require scaling and translation of window

Hierarchical Filter bank for wavelet decomposition



Hierarchical Filter bank for wavelet decomposition

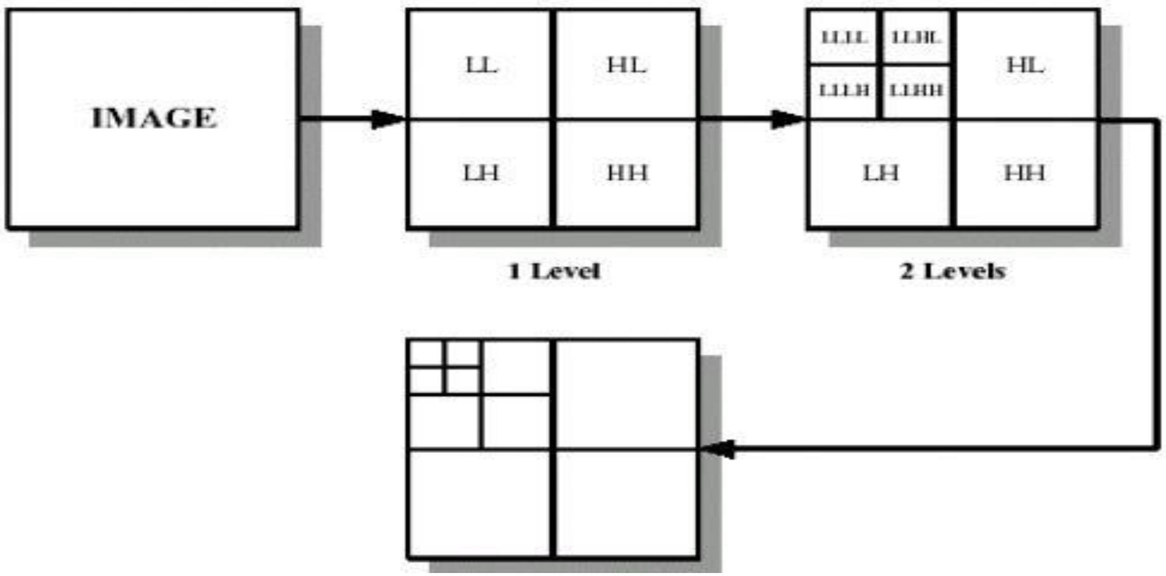


Example
f(n) has 500 samples
Output of h0(n) and h1(n) has 500 samples
Down sampling by 2 is down to reduce the size from 500 to 250

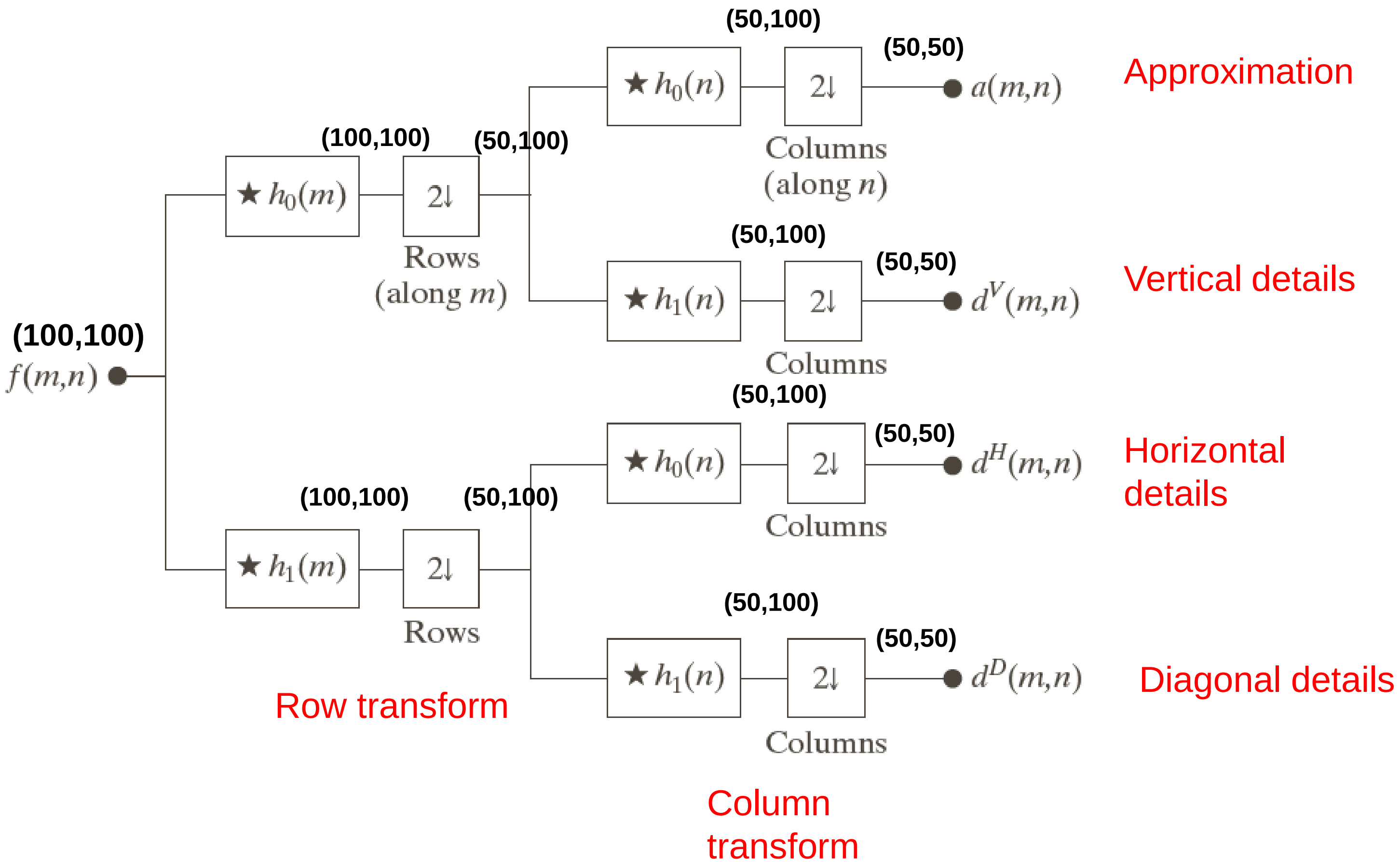
Hierarchical filter bank

- Low frequency contents is the most important part of the signal as it gives identity to the signal
- High frequency imparts finer details
- The process of filtering an down sampling can be repeated to get multilevel decomposition
- Number of stages of filtering and down sampling depends on the lowest frequency component desired
- Splitting of the signal into several frequency bands is known as subband or wavelet decomposition
- Original signal is recovered by reconstruction method known as inverse wavelet transform

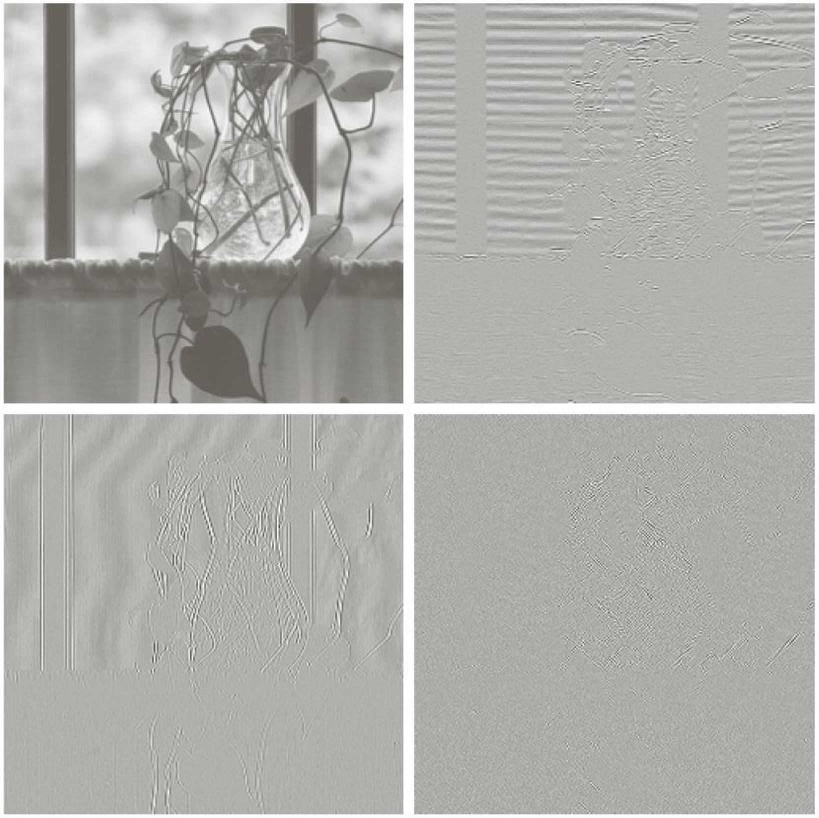
2-D Discrete Wavelet Transform



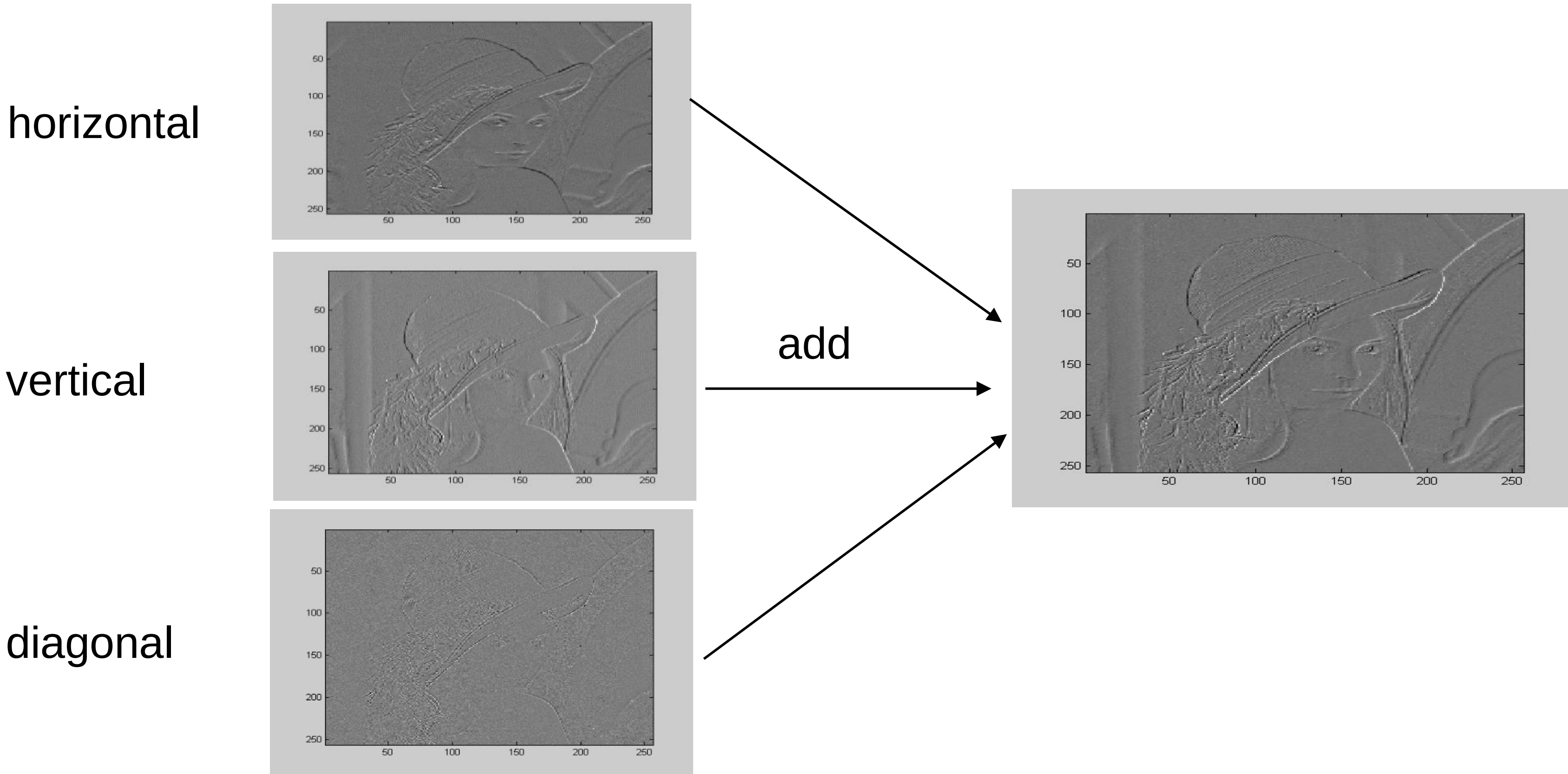
2-D Discrete Wavelet Transform



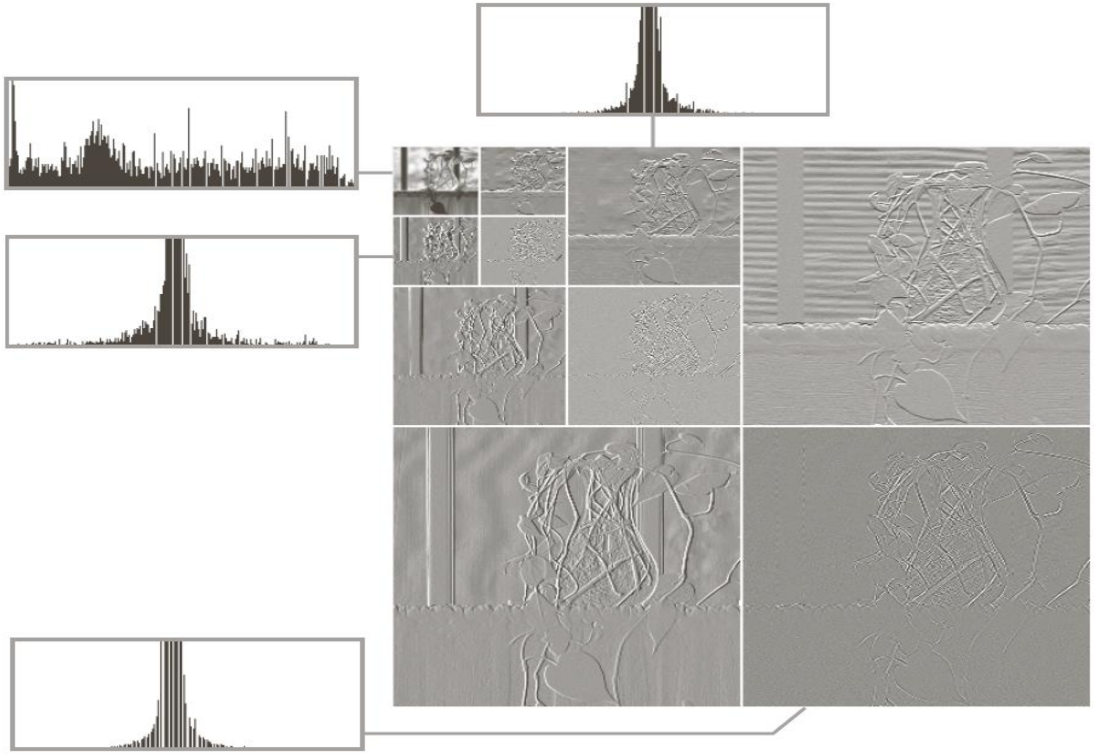
Two dimensional Single level Decomposition



Edge Detection in Images (1 level)



Edge Detection in Images(3 levels)



approximations (64 × 64, 128 × 128, and 256 × 256) that can be obtained from (a).

Application of Wavelet Transform

- Image Denoising
- Edge Detection
- Image Compression
- Image Enhancement