Digital Image Processing (CSE/ECE 478)

Lecture # 17: Filter Banks and Wavelets I

Avinash Sharma

Center for Visual Information Technology (CVIT),

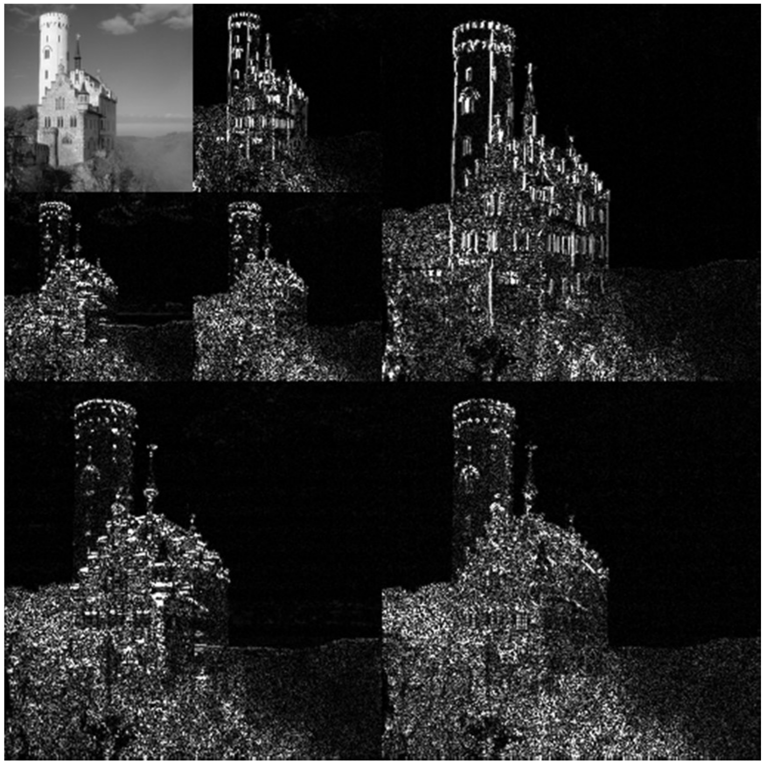
IIIT Hyderabad

Today's Lecture

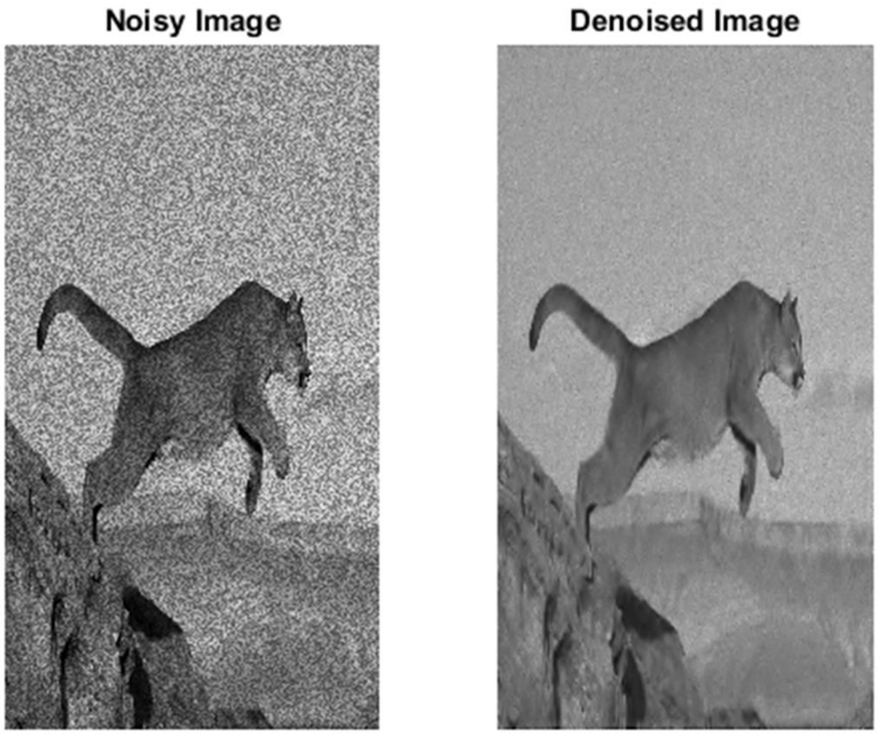
- Non Locality of DFT and how to solve that
- The Wavelet Transform and Haar filter bank

Multi resolution processing

- Wavelet is an approach for Multi Resolution Processing

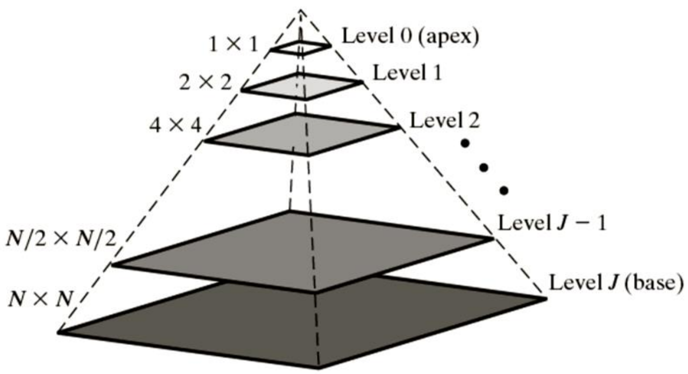


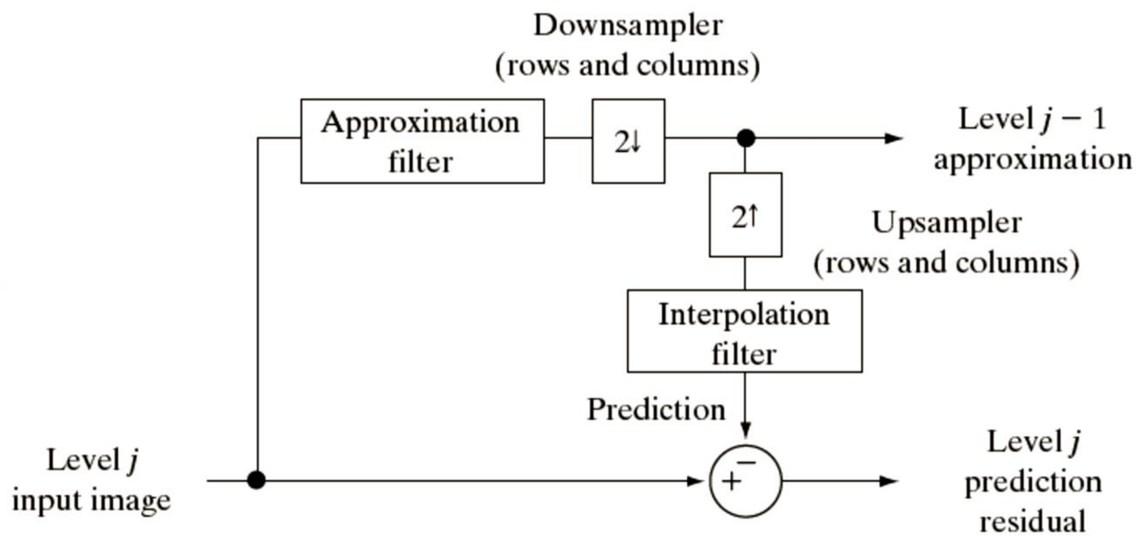
Application in denoising, compression…



Multi resolution processing (revision)

- Pyramids





Pyramids (revision)

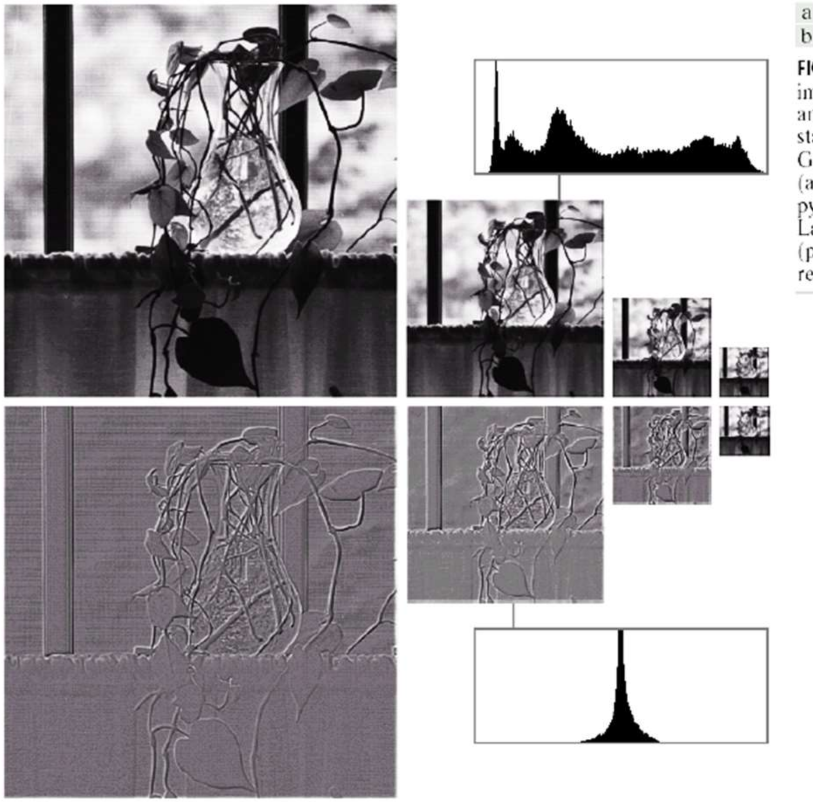


FIGURE 7.3 Two image pyramids and their statistics: (a) a Gaussian (approximation) pyramid and (b) a Laplacian (prediction residual) pyramid.

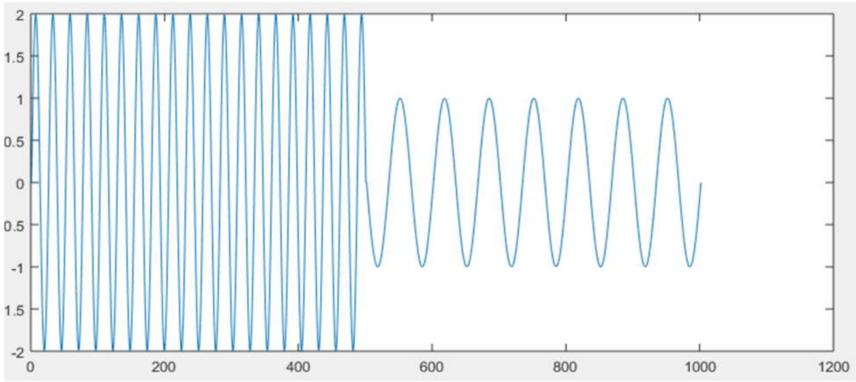
Today's Lecture

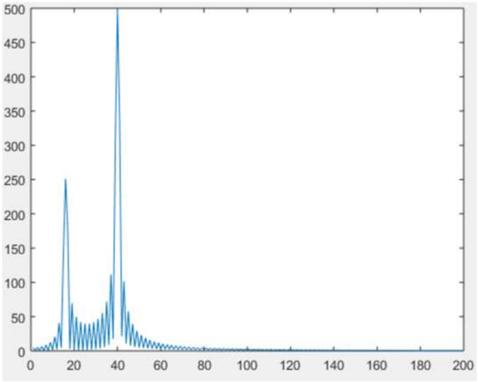
- Non Locality of DFT and how to solve that
- The Wavelet Transform and Haar filter bank

Non Locality of DFT

- Consider an audio signals on $t \in [0,1]$

$$g(t) = \begin{cases} 2 * \sin(2\pi \cdot 39t), 0 \leq t \leq 1/2 \\ \sin(2\pi \cdot 15t), 1/2 < t \leq 1 \end{cases}$$

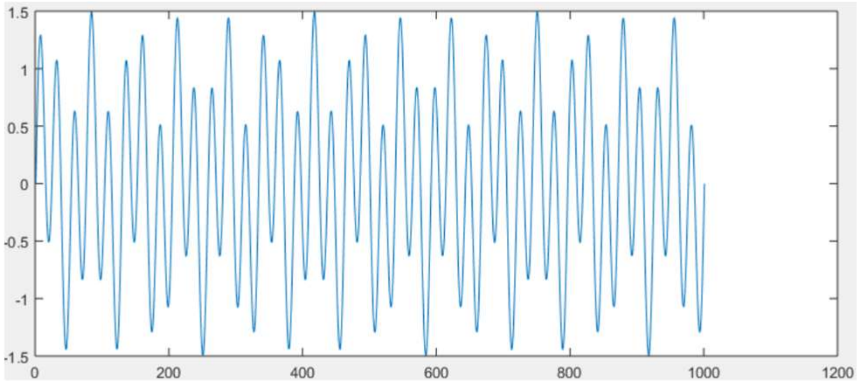




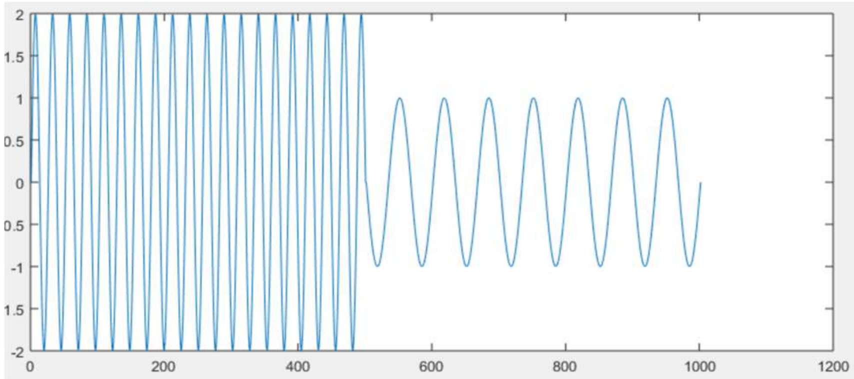
Non Locality of DFT

- Consider two different audio signals on $t \in [0,1]$

$$f(t) = \sin(2\pi \cdot 39t) + 0.5\sin(2\pi \cdot 15t)$$

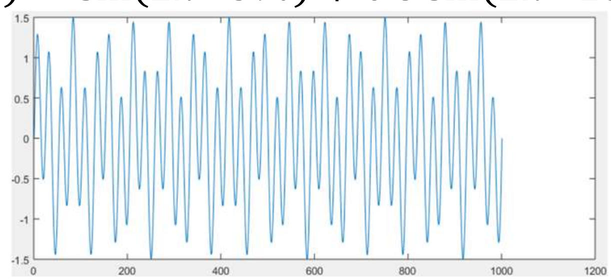


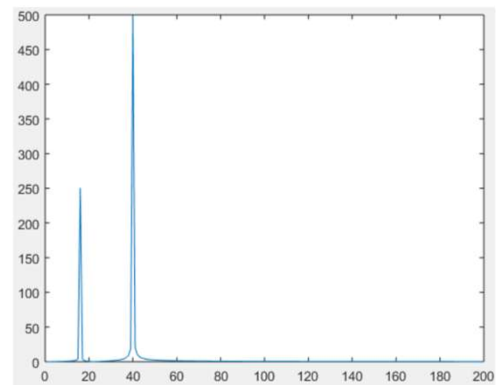
$$g(t) = \begin{cases} 2 * \sin(2\pi \cdot 39t), 0 \le t \le 1/2 \\ \sin(2\pi \cdot 15t), 1/2 < t \le 1 \end{cases}$$



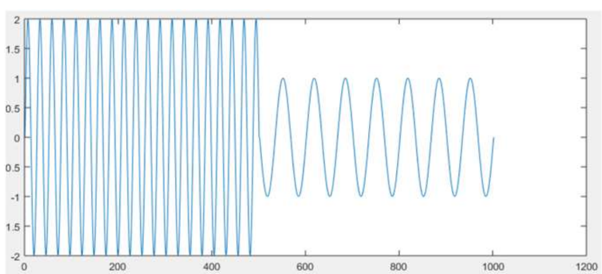
Non Locality of DFT

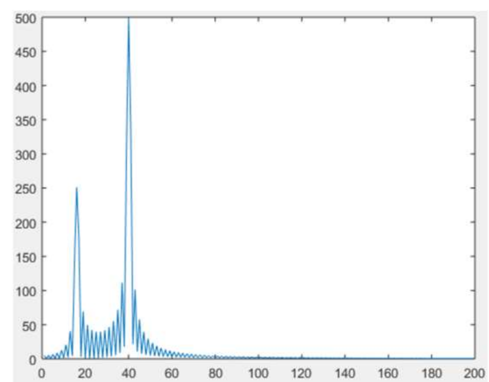
$$\mathrm{f}(t) = \sin(2\pi \cdot 39t) + 0.5\sin(2\pi \cdot 15t)$$





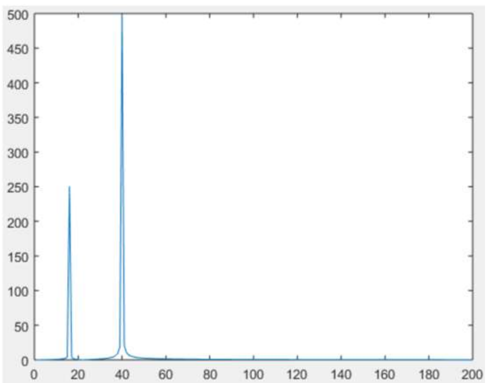
$$\mathrm{g}(t) = \begin{cases} 2 * \sin(2\pi \cdot 39t), 0 \le t \le 1/2 \\ \sin(2\pi \cdot 15t), 1/2 < t \le 1 \end{cases}$$

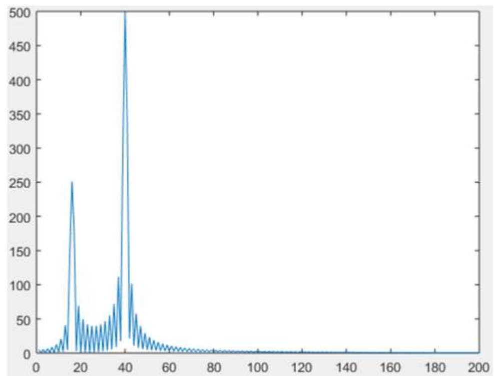




Non Locality of DFT

- Each signal contains dominant frequencies at 15 and 39 Hz
- Magnitudes of DFT are otherwise fairly similar





DFT is global in nature!

The fact that f(t) and g(t) are quite different in time domain is difficult to glean from the DFT graphs!

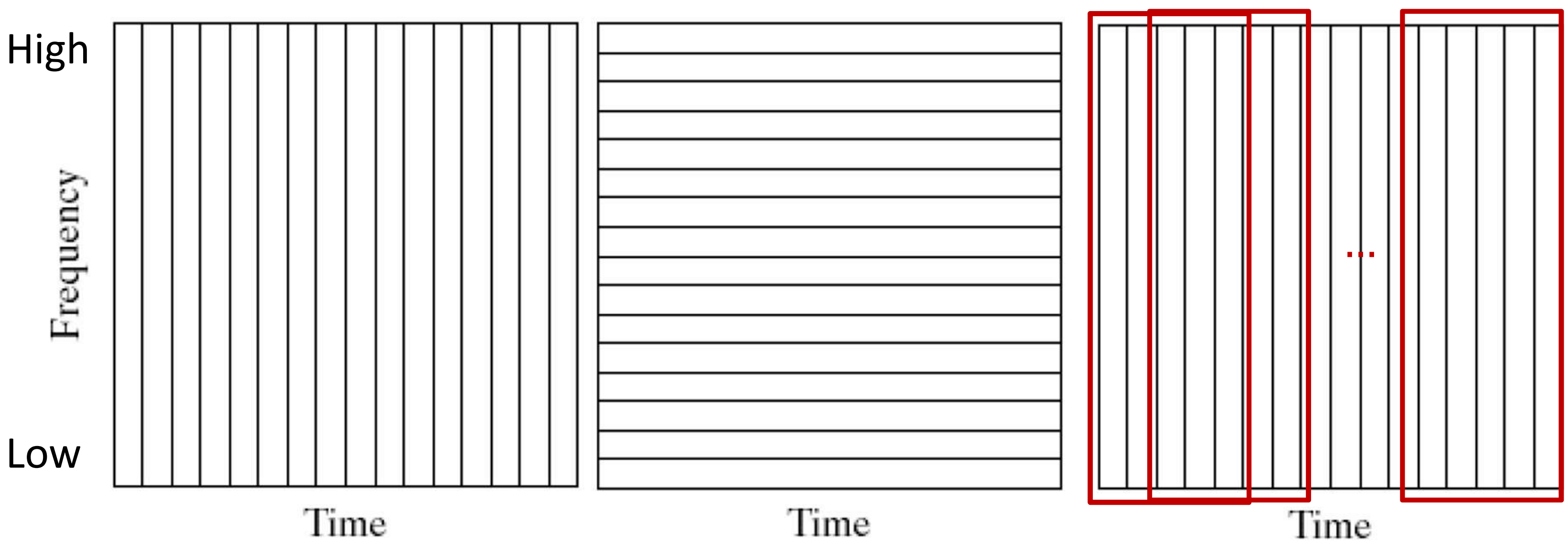
Non Locality of DFT

What can we do to solve this problem?

Two ways to solve this problems

1. Windowing and Short time Fourier Transform
2. Filter Banks and Discrete Wavelet Transform (DWT)
   – 1910 → Alfred Haar's thesis

Windowing



Signal

Global FFT

Windowed FFT

Windowing

- Break the signal into blocks “windows” in time domain
- Certain frequencies may be present in some blocks and not in others
- Select block size small enough so that frequency content is relatively stable over the window
- Apply DFT to each window independently
  - Adjacent blocks may overlap
- Represent the signal as a sequence of short-time DFT’s

Windowing

A rectangular window:

- Starting position m
- Length M Samples, m+M < N
- All samples x_j with j<m and j>m+M are zeroed out, others unchanged
- The resulting vector y has components $y_j = w_j x_j$
- Where the vector w defined as:

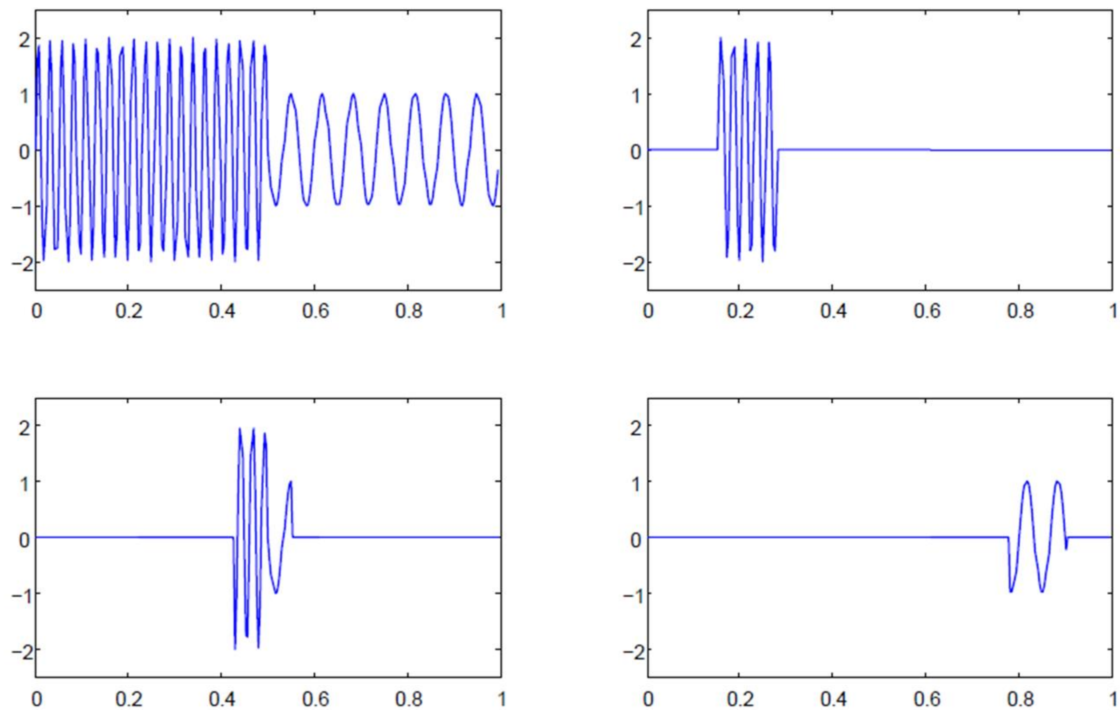
$$w_j = \begin{cases} 1, & m \leq j \leq m + M - 1 \\ 0, & otherwise \end{cases}$$

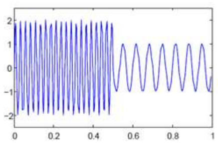
is called a (rectangular) window

Windowing

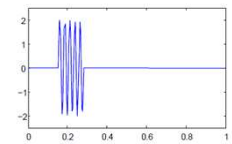
Original (upper left) and windowed signals with $N = 256$, $M = 32$

- $m = 40$
- $m = 110$
- $m = 200$



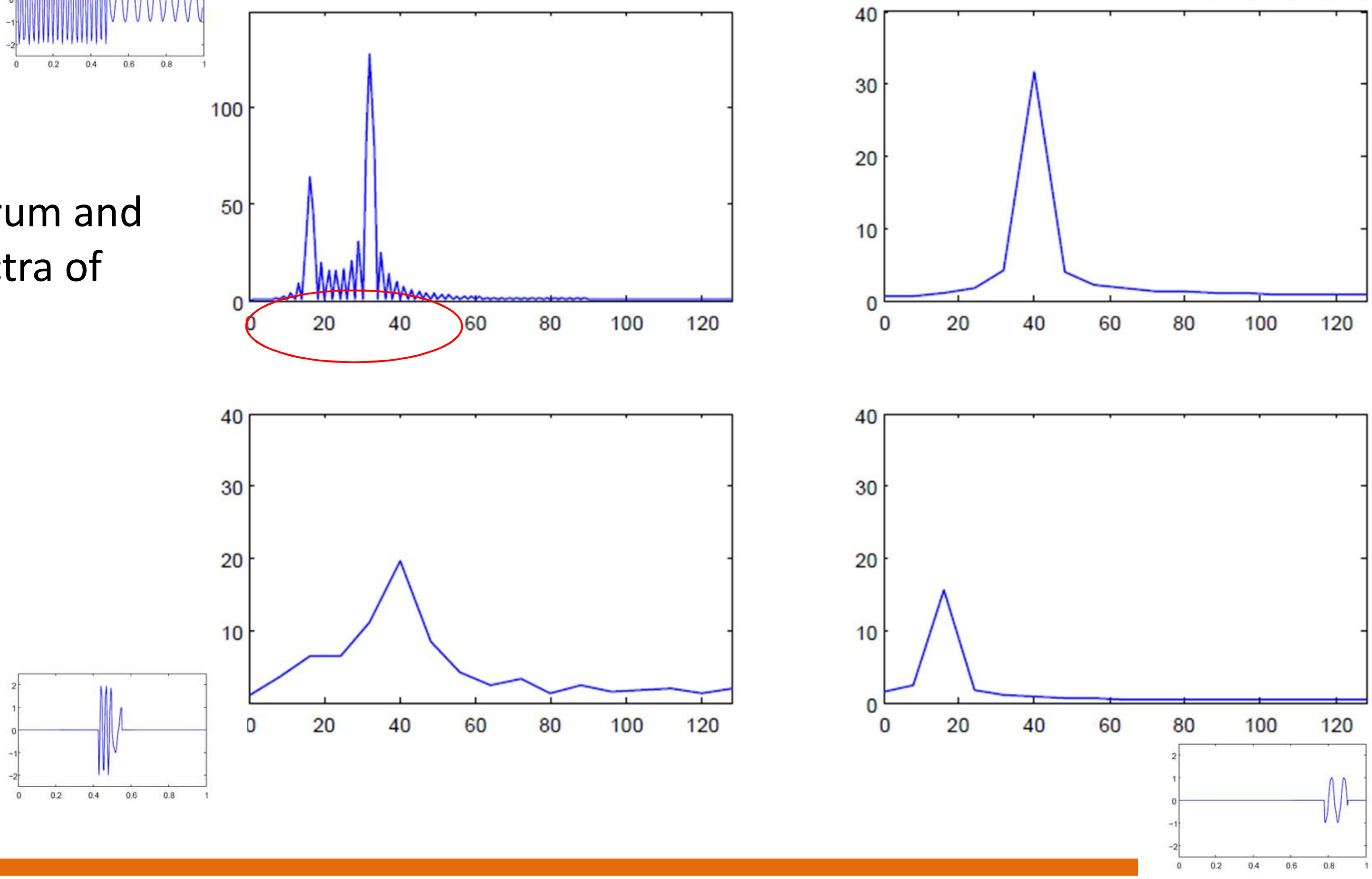


Windowing



Global DFT spectrum and M-point DFT spectra of windowed signal
N = 256, M = 32

- m = 40
- m = 110
- m = 200



Windowing

- Notice that frequency resolution of the DFT for a windowed signal is degraded
- A phenomenon related to the "Heisenberg uncertainty principle"
  - A signal localized in time has wide (coarse) DFT spectrum
  - And vice versa
- With windowing, we focus on local portion of the signal, but we lose the ability to distinguish closely spaced frequency

$$\Delta x \Delta p \ \geq h/4\pi$$

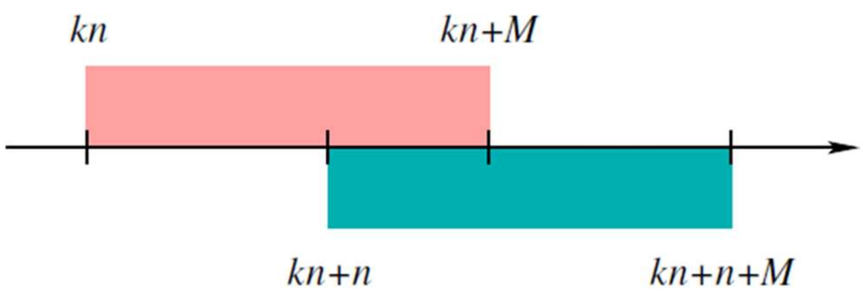
https://www.youtube.com/watch?v=PZIoFD_Z73M

Demo of Heisenberg Uncertainty Principle

https://www.youtube.com/watch?v=a8FTr2qMutA

Windowing

- A collection of DFT's computed over windowed portions of the signal, is called a short-time Fourier transform
- Adjacent windows may overlap
- Let m = k . n be the starting point of the kth window
- The integer n controls in overlap
- No overlap for n=M



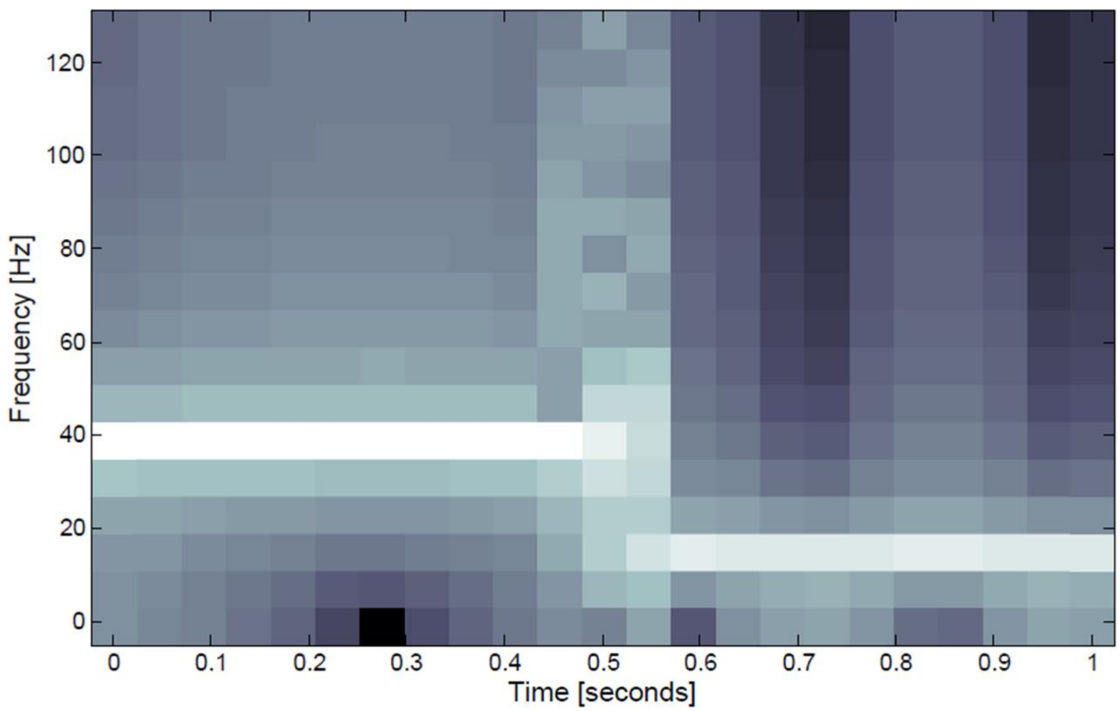
Windowing

- The kth block of data is: $(x_{kn}, x_{kn+1}, x_{kn+}\ , \dots, x_{kn+M-}\ )$, for $k = 0,1,\dots,[N-M]/n$

- The kth block of data is: $(x_{kn}, x_{kn+1}, x_{kn+2}, \dots, x_{kn+M-1})$, for $k = 0,1,\dots,[N-M]/n$
- Compute the M-point DFT of each block, plot its magnitude as a kth column of the intensity image
- The resulting plot is called a *spectrogram*
- In next page illustration, we take M=32 and n=10
- How many total blocks?

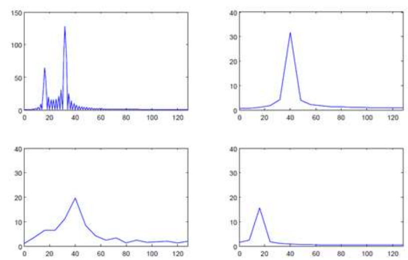
- Compute the M-point DFT of each block, plot its magnitude as a kth column of the intensity image
- The resulting plot is called a *spectrogram*
- In next page illustration, we take M=32 and n=10
- How many total blocks?

Windowing



Spectrogram of a piecewise monochromatic signal.

Lighter color → greater DFT magnitude

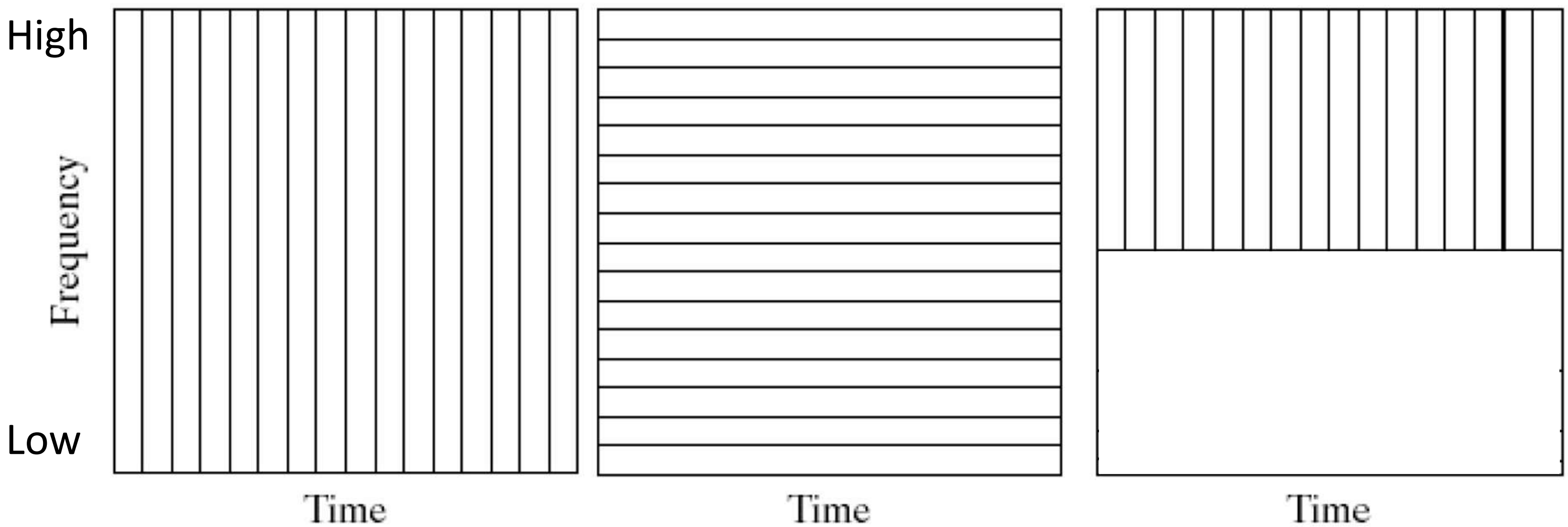


Today's Lecture

- Non Locality of DFT and how to solve that
- The Wavelet Transform and Haar filter bank

Wavelet Transform

Tradeoff b/w Time and Frequency



Signal Global FFT Wavelet Transform

The Haar filter bank

The filter bank method:

1. In filter bank method, the signal is split into two or more frequency bands
2. With each signal effectively being a down sampled version of the signal

The Haar filter bank

- Split the signal x into two bands by applying a low pass and a high pass filter
  - $x * l$ is the output of the low pass filter
  - $x * h$ is the output of the high pass filter

- As an example, we take l to be 2 point averaging filter

$$l_0 = \frac{1}{2},\ l_1 = \frac{1}{2},\ l_r = 0$$

- And h to be 2 point differentiating filter

$$h_0 = \frac{1}{2},\ h = -\frac{1}{2}, h_r = 0$$

The Haar filter bank

- Split the signal x into two bands by applying a low pass and a high pass filter
  - $x * l$ is the output of the low pass filter
  - $x * h$ is the output of the high pass filter

- As an example, we take l to be 2 point averaging filter

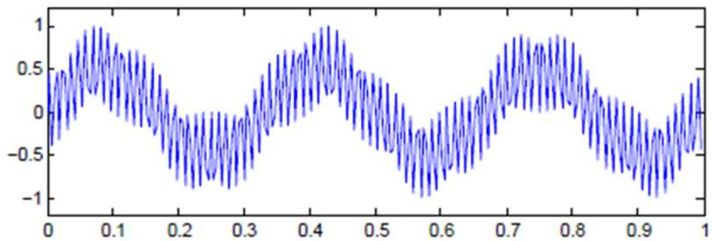
$$l_0 = \frac{1}{2},\ l_1 = \frac{1}{2},\ l_r = 0$$

- And h to be 2 point differentiating filter

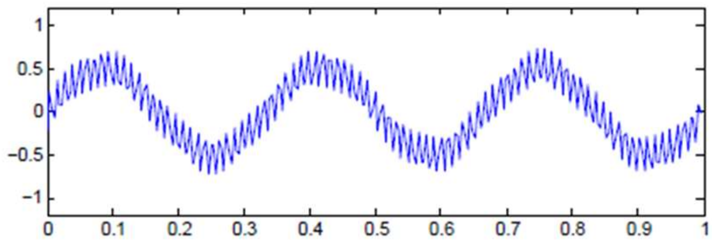
$$h_0 = \frac{1}{2}, h_1 = -\frac{1}{2}, h_r = 0$$

The Haar filter bank

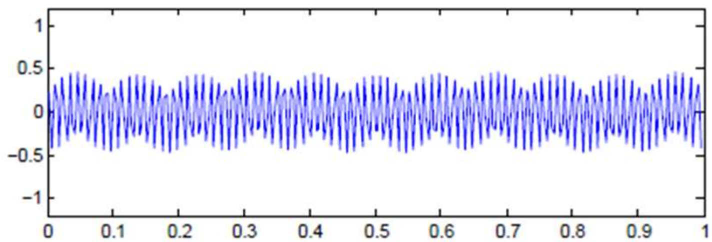
x



$x * l$



$x * h$



Example: Haar filters

- Let x(t) be an analog signal
- $\mathrm{x}(\mathrm{t}) = 0.5\sin(2\pi \cdot 3t) + 0.5\,\sin(2\pi \cdot 89t)$
- Sampled a 256 Hz for t ϵ [0 1]

The Haar filter bank

$$\mathbf{x} = \begin{pmatrix} \vdots \\ x_{-2} \\ x_{-1} \\ x_0 \\ x_1 \\ x_2 \\ \vdots \end{pmatrix} \; ; \quad \mathbf{x} * \ell = \frac{1}{2} \begin{pmatrix} \vdots \\ x_{-2} + x_{-3} \\ x_{-1} + x_{-2} \\ x_0 + x_{-1} \\ x_1 + x_0 \\ x_2 + x_1 \\ \vdots \end{pmatrix} \; ; \quad \mathbf{x} * \mathbf{h} = \frac{1}{2} \begin{pmatrix} \vdots \\ x_{-2} - x_{-3} \\ x_{-1} - x_{-2} \\ x_0 - x_{-1} \\ x_1 - x_0 \\ x_2 - x_1 \\ \vdots \end{pmatrix}$$

- Observe that $(x * l + x * h)_k = x_k$ and $(x * l - x * h)_k = x_{k-1}$
- The transformation $x \rightarrow (x * l + x * h)$ is invertible
- We can drop away every other component of $x * l$ and $x * h$ and still be able to reconstruct x

The Haar filter bank

Down-sampling operator:

- $(D(x))_k = x_{2k}$
- i.e. every odd-indexed element of x is dropped away

$$x = (\dots\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots\dots)$$
$$D(x) = (\dots\dots, x_{-4}, x_{-2}, x_0, x_2, x_4, \dots\dots)$$

Up-sampling operator:

- $(U(x))_k = x_{k/2}$ when k is even, $(U(x))_k = 0$ when k is odd
- i.e. inserting zeros between every pair of adjacent elements

$$x = (\dots\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots\dots)$$
$$U(x) = (\dots, x_{-2}, 0, x_{-1}, 0, x_0, 0, x_1, 0, x_2, 0 \dots)$$

The Haar filter bank

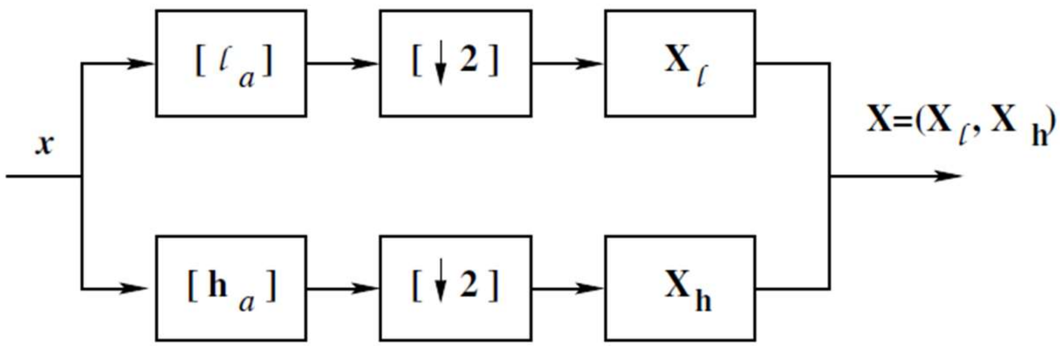
- Down-sampled versions of filtered signal

$$D(\mathbf{x} * \ell) = \frac{1}{2}\begin{pmatrix} \vdots \\ x_{-4} + x_{-5} \\ x_{-2} + x_{-3} \\ x_0 + x_{-1} \\ x_2 + x_1 \\ x_4 + x_3 \\ \vdots \end{pmatrix} \; ; \; D(\mathbf{x} * \mathbf{h}) = \frac{1}{2}\begin{pmatrix} \vdots \\ x_{-4} - x_{-5} \\ x_{-2} - x_{-3} \\ x_0 - x_{-1} \\ x_2 - x_1 \\ x_4 - x_3 \\ \vdots \end{pmatrix}$$

- Let $\mathbf{X}_{\boldsymbol{l}} = D(x * l), \mathbf{X}_{\boldsymbol{h}} = D(x * h)$
- The invertible transform $\mathrm{W(x)} = (\mathbf{X}_{\boldsymbol{l}}, \mathbf{X}_{\boldsymbol{h}}) = \mathbf{X}$ is called the Haar filter bank transform

The Haar filter bank

- The transform W is an example of analysis filter bank
- Coefficients of $\mathbf{X}_{\boldsymbol{l}}$ are the approximation coefficients
- Coefficients of $\mathbf{X}_{\boldsymbol{h}}$ are the detail coefficients



One stage, two channel analysis filter bank

The Haar filter bank

Inverse filter bank transform:

1. Upsample $\mathbf{X}_{\boldsymbol{l}}$ and $\mathbf{X}_{\boldsymbol{h}}$
2. Convolve $U(\mathbf{X}_l)$ and $U(\mathbf{X}_l)$ with appropriate synthesis filters'

$$(l_s)_{-1} = 1, (l_s)_0 = 1, (l_s)_r = 0, \text{ otherwise}$$
$$(h_s)_{-1} = -1, (h_s)_0 = 1, (h_s)_r = 0, \text{ otherwise}$$

3. Combine the resulting vector to recover x

The transformation $\mathbf{X} = (\mathbf{X}_{\boldsymbol{l}}, \mathbf{X}_{\boldsymbol{h}}) \rightarrow x$ is called the synthesis filter transform

The Haar filter bank

- Upsampling operation yields

$$U(\mathbf{X}_\ell) = \frac{1}{2}\begin{pmatrix} \vdots \\ 0 \\ x_{-2}+x_{-3} \\ 0 \\ x_0+x_{-1} \\ 0 \\ x_2+x_1 \\ 0 \\ \vdots \end{pmatrix} \; ; \; U(\mathbf{X}_h) = \frac{1}{2}\begin{pmatrix} \vdots \\ 0 \\ x_{-2}-x_{-3} \\ 0 \\ x_0-x_{-1} \\ 0 \\ x_2-x_1 \\ 0 \\ \vdots \end{pmatrix}$$

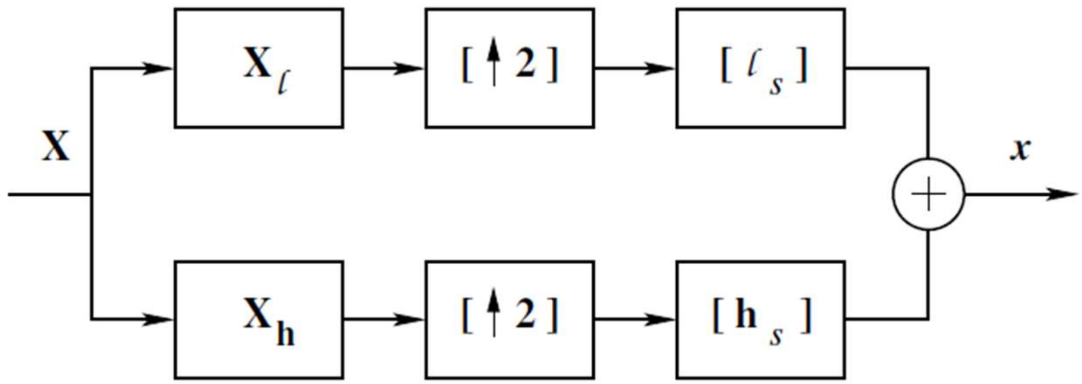
The Haar filter bank

- Let $\mathrm{v}_l = U(\mathbf{X}_l) * l_s$ and $\mathrm{v}_h = U(\mathbf{X}_h) * l_h$
- The resulting vectors are

$$\mathbf{v}_\ell = \frac{1}{2}\begin{pmatrix} \vdots \\ x_{-2} + x_{-3} \\ x_{-2} + x_{-3} \\ x_0 + x_{-1} \\ x_0 + x_{-1} \\ x_2 + x_1 \\ x_2 + x_1 \\ x_4 + x_3 \\ \vdots \end{pmatrix} ; \quad \mathbf{v}_h = \frac{1}{2}\begin{pmatrix} \vdots \\ x_{-3} - x_{-2} \\ x_{-2} - x_{-3} \\ x_{-1} - x_0 \\ x_0 - x_{-1} \\ x_1 - x_2 \\ x_2 - x_1 \\ x_3 - x_4 \\ \vdots \end{pmatrix}$$

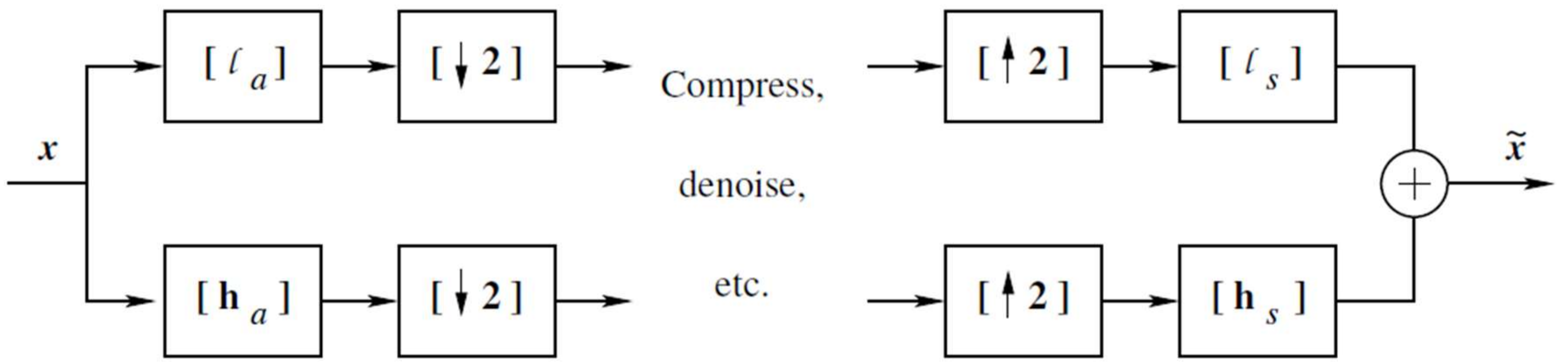
- Note that $\mathrm{v}_l + \mathrm{v}_h = x$

The Haar filter bank



One-stage two-channel synthesis filter bank

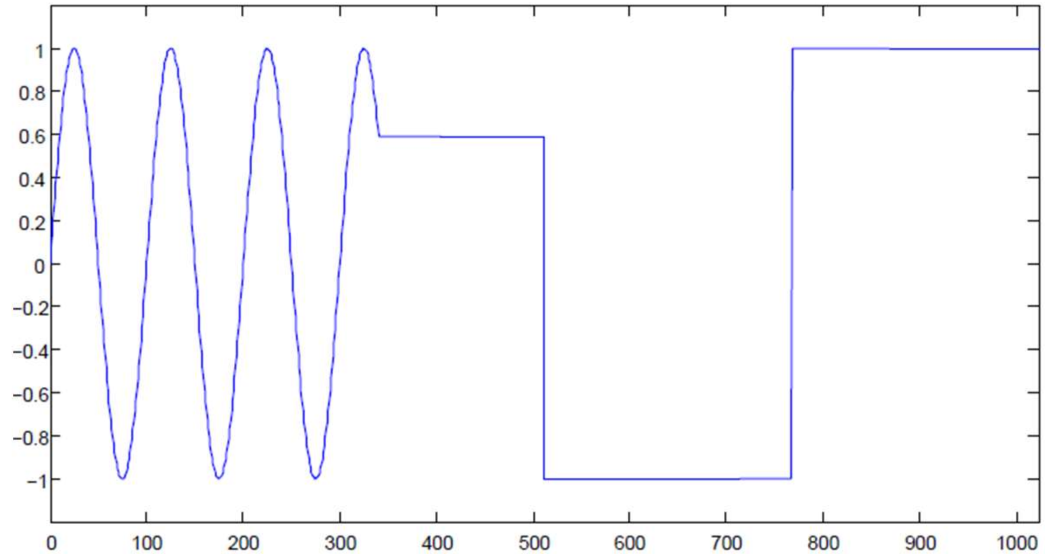
The Haar filter bank



Analysis/synthesis filter bank. $\tilde{x}$ is an altered version of x

The Haar filter bank

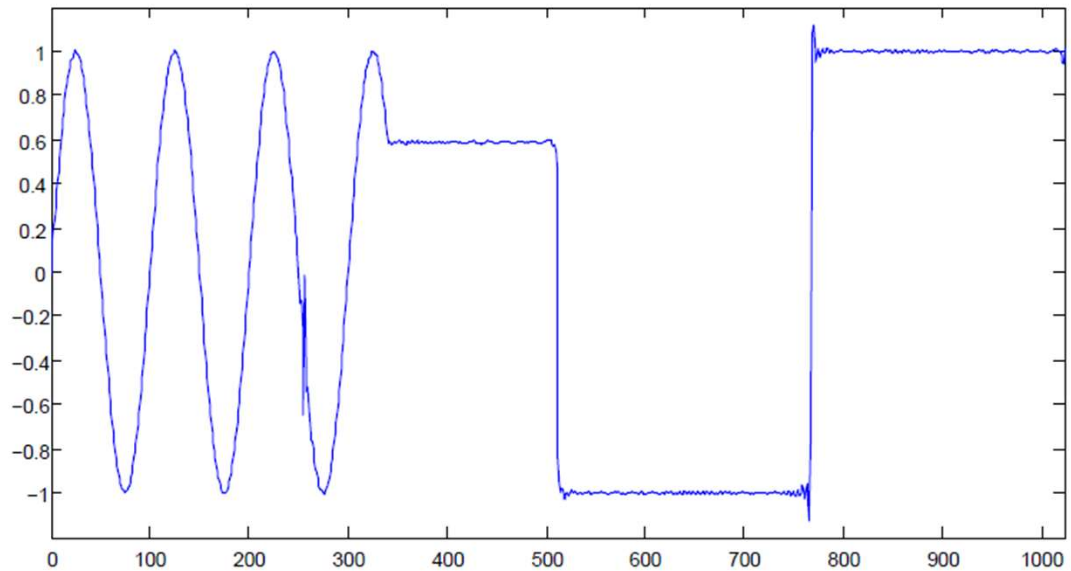
- DFT compression via Haar transform



Signal to be compressed

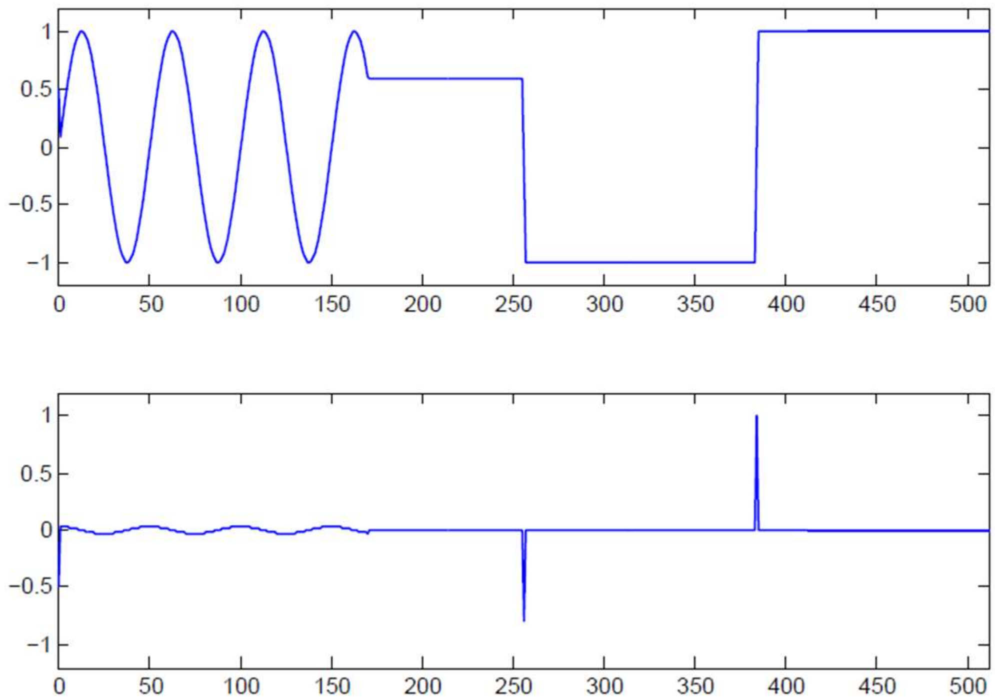
The Haar filter bank

- DFT compression performs poor at the jumps



DFT thresholding compression (half of the frequencies removed)

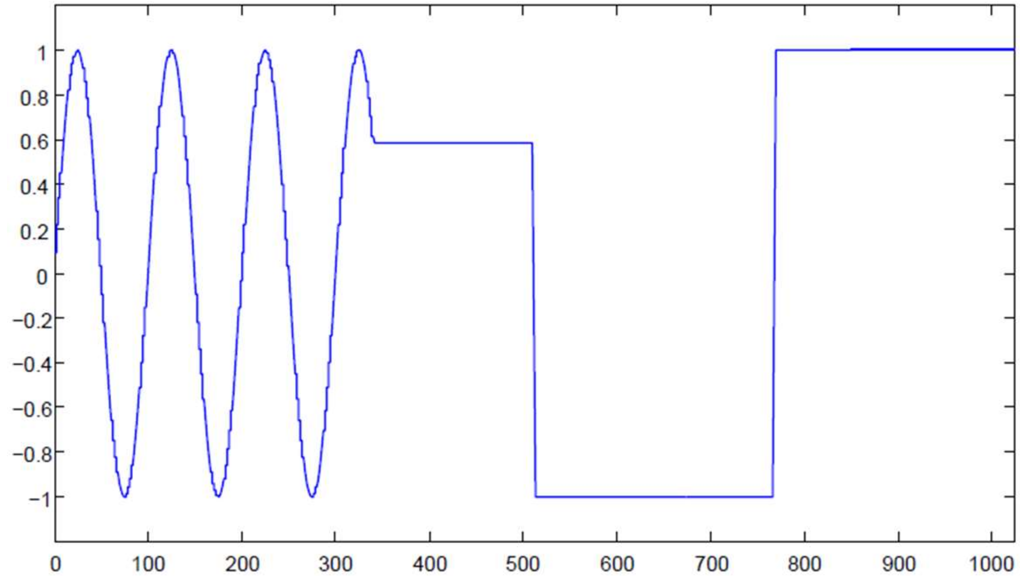
The Haar filter bank



Approximation (top) and detail (bottom) coefficients

The Haar filter bank

- Compression: zero out all detail coefficients $\mathbf{X}_h$
- Apply a synthesis filter bank transformation to the compressed signal



The signal compressed via the Haar filter bank transform

Filter bank transform vs DFT

- Filter bank is local in nature
  - Each approximation and detail coefficient depends on the relatively few neighbouring samples
- The DFT is global
  - Each DFT coefficient depends on all samples in x

General one-stage two-channel filter bank transform

- Analysis filters l_a, h_a need not be the 2-point averaging/difference filters
- Any pair of low/high pass finite impulse response filters will do

Analysis filter bank:

- $x \rightarrow \mathbf{X} = (\mathbf{X}_{\boldsymbol{l}}, \mathbf{X}_{\boldsymbol{h}})$
- Where, $\mathbf{X}_{\boldsymbol{l}} = D(x * l_a)$; $\mathbf{X}_{\boldsymbol{h}} = D(x * h_a)$

Synthesis filter bank:

- $x' = l_s * U(\mathbf{X}_{\boldsymbol{l}}) + h_s * U(\mathbf{X}_{\boldsymbol{h}})$

Any analysis/synthesis filters could be used as long as x' is perfect reconstruction of x

General one-stage two-channel filter bank transform

- Example: Le Gall 5/3 filters

Analysis filters:

$$l_a = \left(\ldots, 0, -\frac{1}{8}, \frac{1}{4}, \frac{3}{4}, \frac{1}{4}, -\frac{1}{8}, 0, \ldots\right)$$

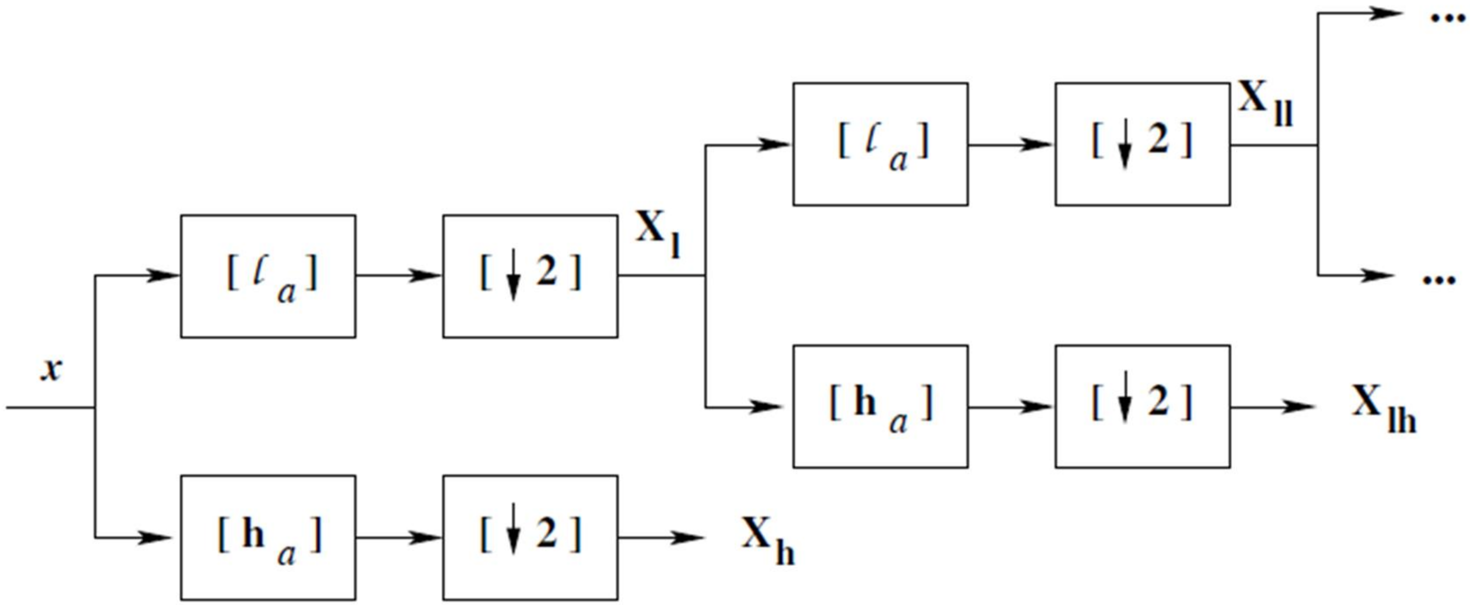
$$h_a = \left(\ldots, 0, -\frac{1}{2}, 1, -\frac{1}{2}, 0, \ldots\right)$$

Synthesis filters:

$$l_s = \left(\ldots, 0, \frac{1}{2}, 1, , \frac{1}{2}, 0, \ldots\right)$$

$$h_s = \left(\ldots, 0, -\frac{1}{8}, -\frac{1}{4}, \frac{3}{4}, -\frac{1}{4}, -\frac{1}{8}, 0, \ldots\right)$$

Multi stage filter bank



Multi stage filter bank

- Multi stage analysis filter bank transform

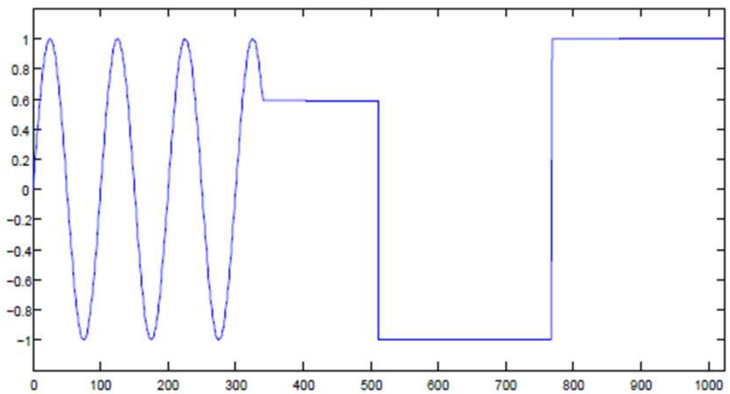
$$\mathbf{x} \longrightarrow \begin{pmatrix} \mathbf{X}_\ell \\ \mathbf{X}_h \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{X}_{\ell\ell} \\ \mathbf{X}_{\ell h} \\ \mathbf{X}_h \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{X}_{\ell\ell\ell} \\ \mathbf{X}_{\ell\ell h} \\ \mathbf{X}_{\ell h} \\ \mathbf{X}_h \end{pmatrix} \longrightarrow \ldots$$

- Multi stage synthesis is applied in reverse
  - First to $\mathbf{X}_{lll}$ and $\mathbf{X}_{llh}$ to produce $\mathbf{X}_{ll}$, and so on

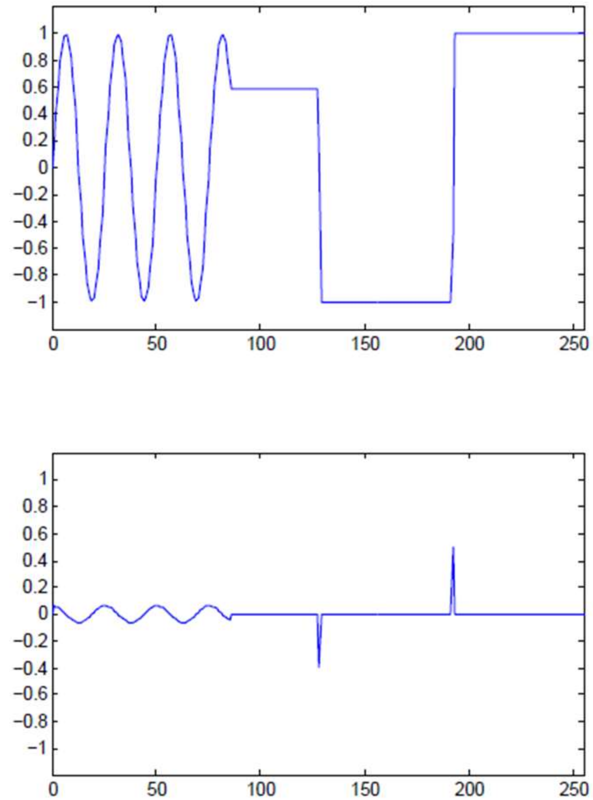
Multi stage filter bank

- 2 Stage

$$\mathbf{x} \longrightarrow \begin{pmatrix} \mathbf{X}_{\ell\ell} \\ \mathbf{X}_{\ell h} \\ \mathbf{X}_{h} \end{pmatrix}$$



Input signal $\mathbf{x}$

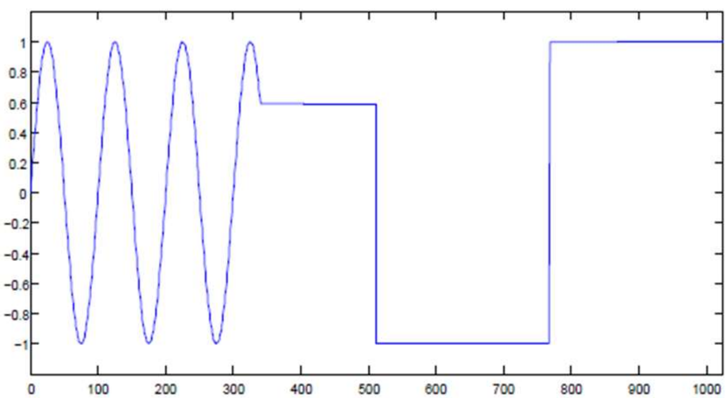


$\mathbf{X}_{\ell\ell}$ (top), $\mathbf{X}_{\ell h}$ (bottom)

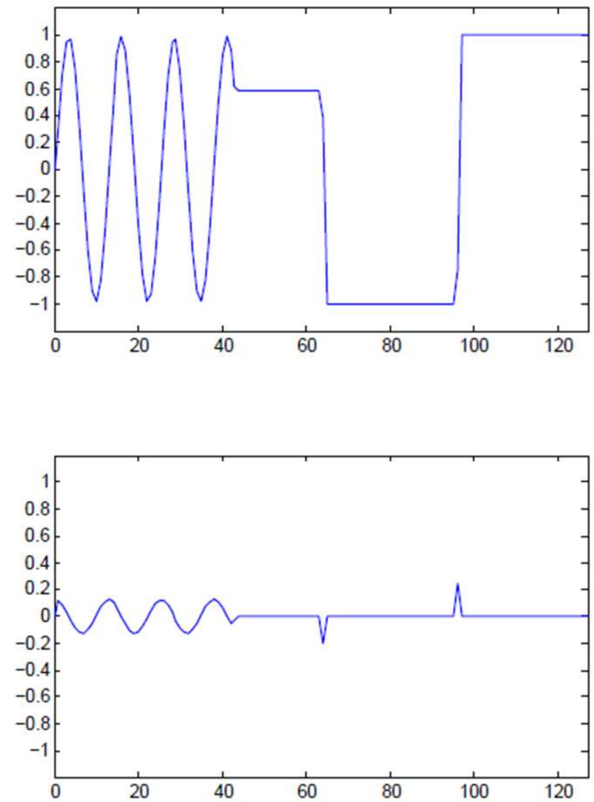
Multi stage filter bank

- 3 Stage

$$\mathbf{x} \longrightarrow \begin{pmatrix} \mathbf{X}_{\ell\ell\ell} \\ \mathbf{X}_{\ell\ell h} \\ \mathbf{X}_{\ell h} \\ \mathbf{X}_{h} \end{pmatrix}$$



Input signal $\mathbf{x}$



$\mathbf{X}_{\ell\ell\ell}$ (top), $\mathbf{X}_{\ell\ell h}$ (bottom)

Multi stage filter bank

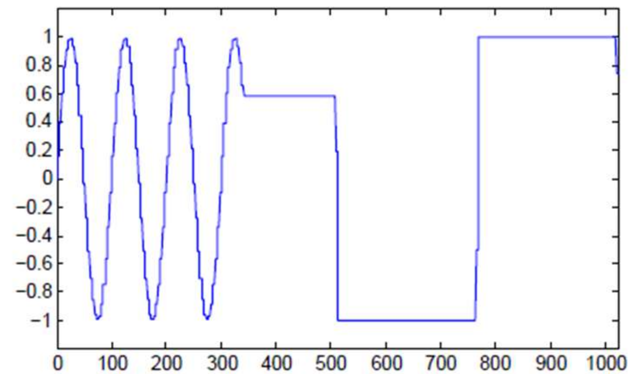
- Zero out detailed coefficients

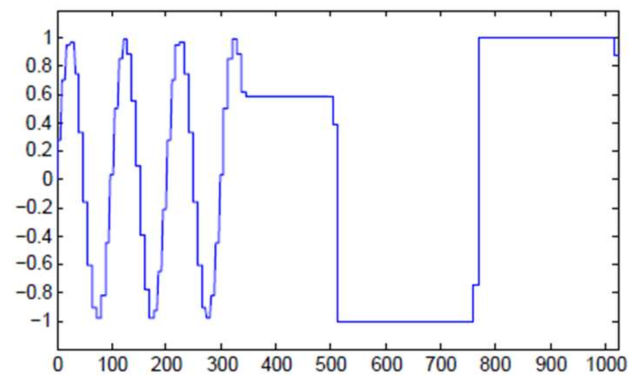
4 fold compression

$$x \rightarrow \begin{pmatrix} \mathbf{X}_{ll} \\ \mathbf{X}_{lh} \\ \mathbf{X}_{h} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{X}_{ll} \\ 0 \\ 0 \end{pmatrix} \rightarrow \tilde{x}$$

8 fold compression

$$x \rightarrow \begin{pmatrix} \mathbf{X}_{lll} \\ \mathbf{X}_{llh} \\ \mathbf{X}_{lh} \\ \mathbf{X}_{h} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{X}_{lll} \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow \tilde{x}$$





Filter bank transform for finite signals

Adapting filter transform to finite signals

1. Extend x to $\tilde{x}$ (zero padding, periodic extension, half-point symmetric extension)
2. Perform filtering on extended signal
3. Truncate back to desired length

Such a filter bank transform for finite signals → **DWT**

Discrete Wavelet Transform (DWT)

Example: $\mathrm{x} = (a, b, c, d)$; $l_a = \left(\frac{1}{2}, \frac{1}{2}, 0,0\right)$ and $h_a = \left(\frac{1}{2}, -\frac{1}{2}, 0,0\right)$

$$\mathbf{x} * l_a = \tfrac{1}{2}(a+d, b+a, c+b, d+c), \text{ ext. periodically}$$
$$\mathbf{x} * \mathbf{h}_a = \tfrac{1}{2}(a-d, b-a, c-b, d-c), \text{ ext. periodically}$$

- Truncating and Downsampling

$$\mathbf{X}_l = \frac{1}{2}(a+d, c+b) \quad \mathbf{X}_h = \frac{1}{2}(a-d, c-b)$$

- The DWT of x is

$$\mathbf{X} = \frac{1}{2}(a+d, c+b, a-d, c-b)$$

Discrete Wavelet Transform (DWT)

Matrix view of DWT

$$\mathbf{X} = W_4^a \mathbf{x},$$

$$\mathbf{X} = \frac{1}{2}(a+d, c+b, a-d, c-b)$$

$$W_4^a = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{pmatrix}$$

Inverse Discrete Wavelet Transform (IDWT)

Example: $\mathbf{X} = (A, B, C, D)$; $l_s = (1,0,0,1)$ and $h_s = (1,0,0,-1)$

- Up-sampling: $U(\mathbf{X}_\ell) = (A, 0, B, 0),\ U(\mathbf{X}_h) = (C, 0, D, 0)$

- Convolving with synthesis filters

$$\begin{aligned} U(\mathbf{X}_\ell) * \ell_s &= (A, B, B, A) \\ U(\mathbf{X}_h) * \mathbf{h}_s &= (C, -D, D, -C) \end{aligned}$$

$$\boxed{\mathbf{X} = \frac{1}{2}(\overset{A}{a+d}, \overset{B}{c+b}, \overset{C}{a-d}, \overset{D}{c-b})}$$

- IDFT of $\mathbf{X}$ is

$$\mathbf{x} = (A + C, B - D, B + D, A - C)$$

Inverse Discrete Wavelet Transform (IDWT)

Matrix view of IDWT

$$\mathbf{x} = W_4^s \mathbf{X},$$

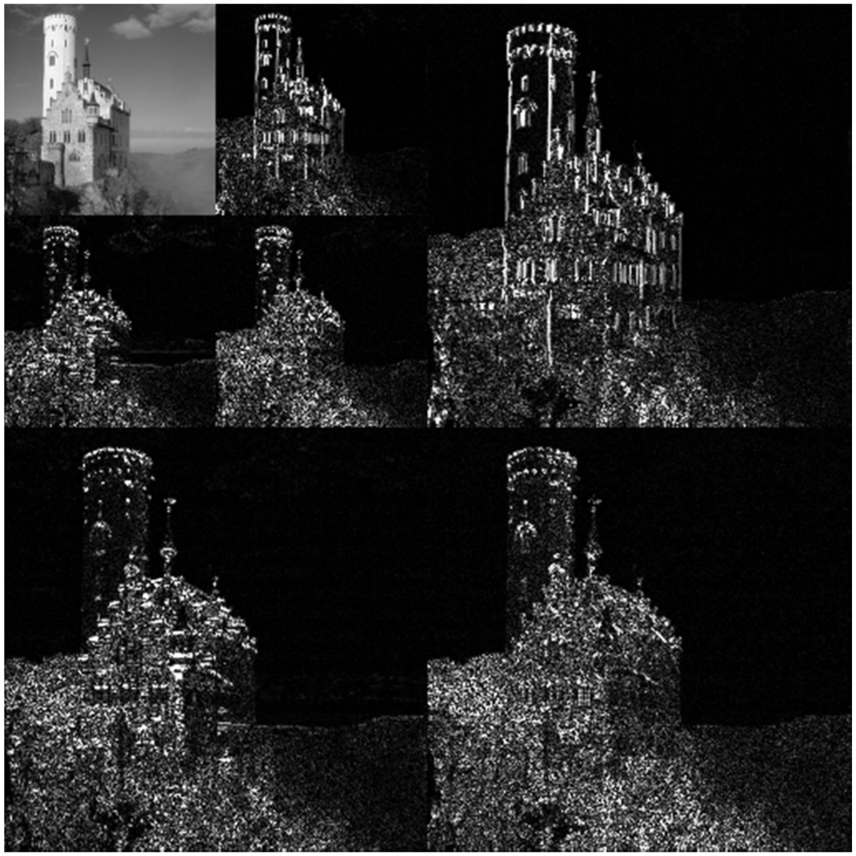
$$\mathbf{x} = (A + C, B - D, B + D, A - C)$$

$$W_4^s = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{pmatrix}$$

Verify

$$W_4^s \cdot W_4^a = I$$

2D example : DWT



References

- Broughton and Bryan, **Discrete Fourier Analysis and Wavelets**, *John Wiley and Sons*