Digital Image Processing (CSE/ECE 478)

Lecture # 18: Filter Banks and Wavelets II

Avinash Sharma

Center for Visual Information Technology (CVIT),

IIIT Hyderabad

Today's Lecture

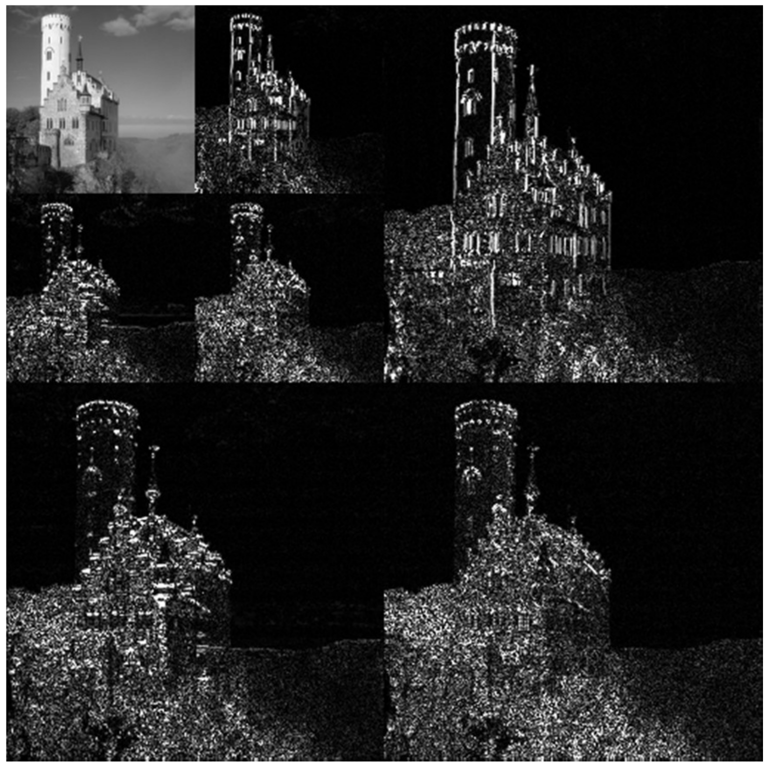
- 2D DWT
- Multi scale DWT

- Assignment 4 delayed due to project submission! Now coming on 26th Oct.

- Next class (23rd October) ? Yes

Multi resolution processing

- Wavelet is an approach for Multi Resolution Processing



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

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

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

- 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-DWT

- Let A be a M×N grayscale image
- The one stage, 2D-DWT is the linear mapping given by:

$$\mathcal{W}_a^1(A) = W_M^a A (W_N^a)^T$$

- W_M and W_N are M×M and N×N analysis matrices determined by (l_a, h_a)

Transform each column of A and then transform each row of resulting matrix
Or vice-versa

2D-DWT

- The inverse one stage transform is given by:

$$\mathcal{W}_s^1(\hat{A}) = W_M^s \hat{A} (W_N^s)^T$$

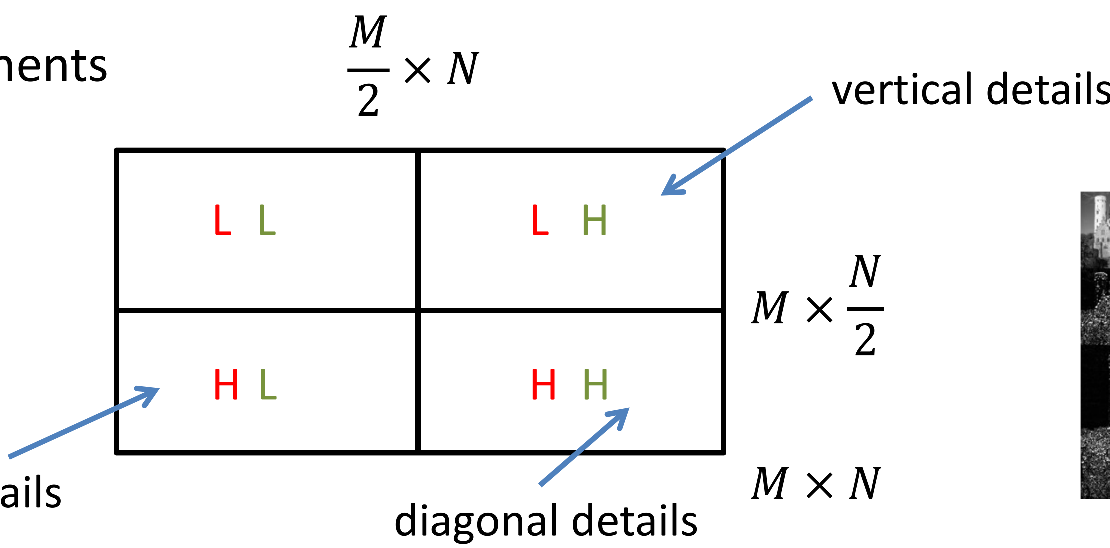
- W_M and W_N are M×M and N×N synthesis matrices determined by (l_s, h_s)

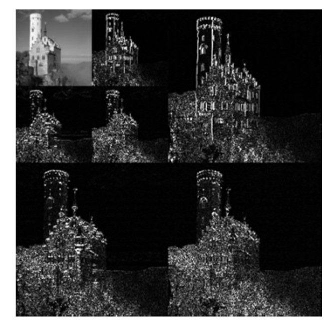
2D-DWT

- Transform each column of A and then transform each row of resulting matrix

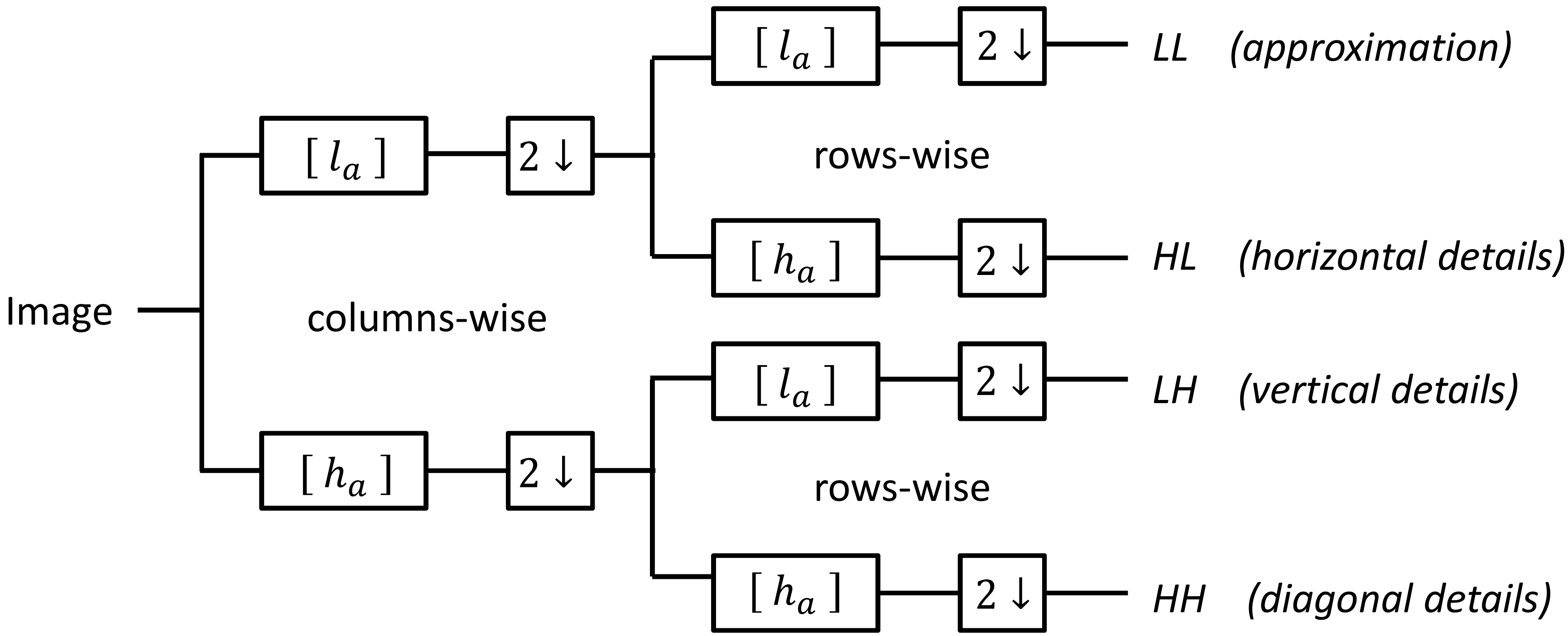
$$\mathcal{W}_a^1(A) = W_M^a A (W_N^a)^T$$

- Four components





2D-DWT



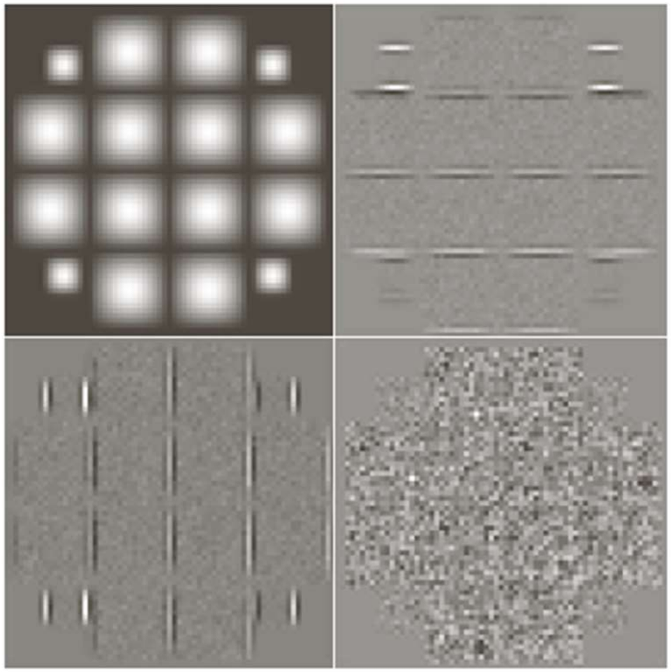
Convention here is first column-wise 1D transform followed by row-wise 1D transform

2D-DWT

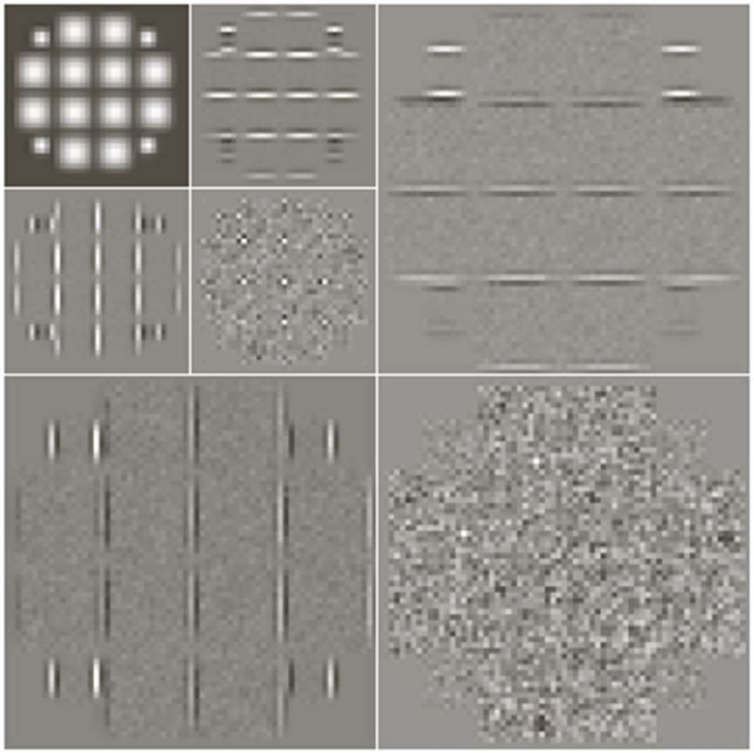
```
im=rgb2gray(imread('color_pencil.jpg'));
[LL,LH,HL,HH]=dwt2(im,'haar');

figure, subplot(2,2,1);imshow(LL,[]);title('LL band of image');
subplot(2,2,2);imshow(LH,[]);title('LH band of image');
subplot(2,2,3);imshow(HL,[]);title('HL band of image');
subplot(2,2,4);imshow(HH,[]);title('HH band of image');
```

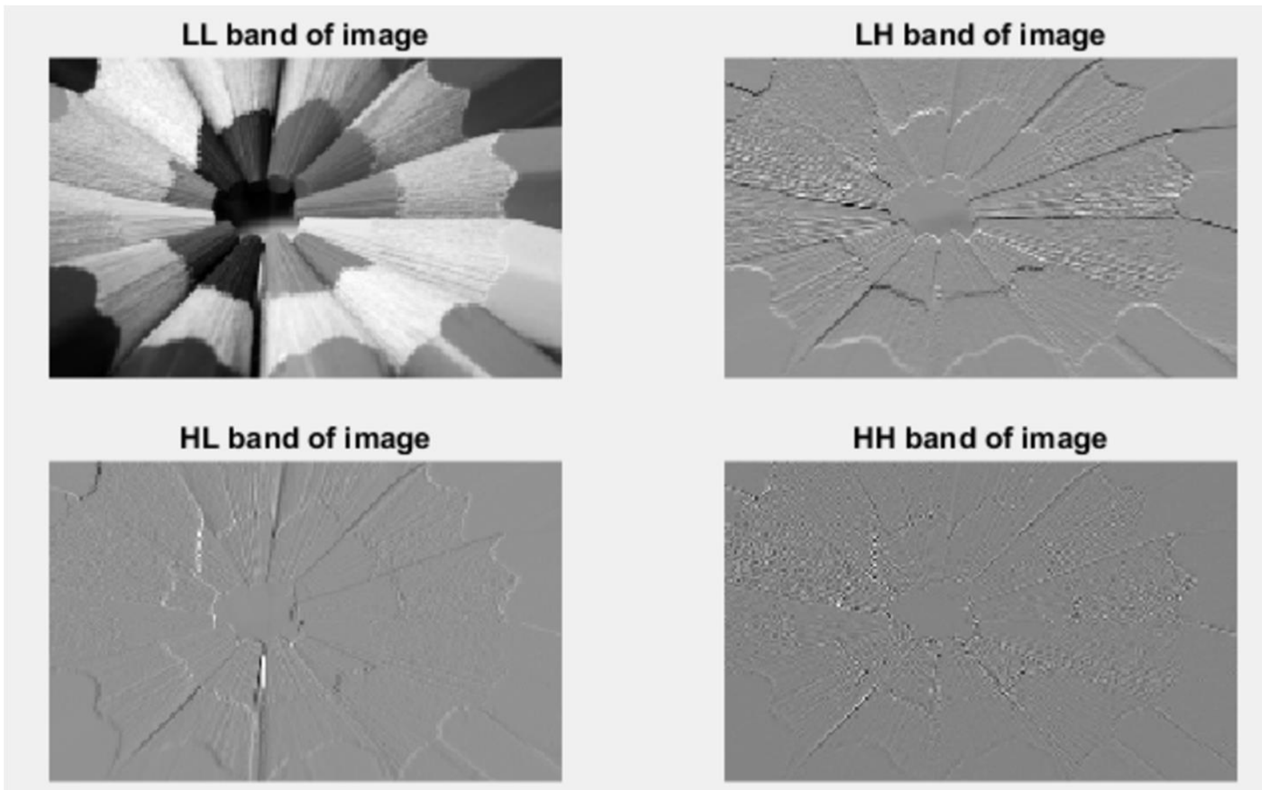
2D-DWT



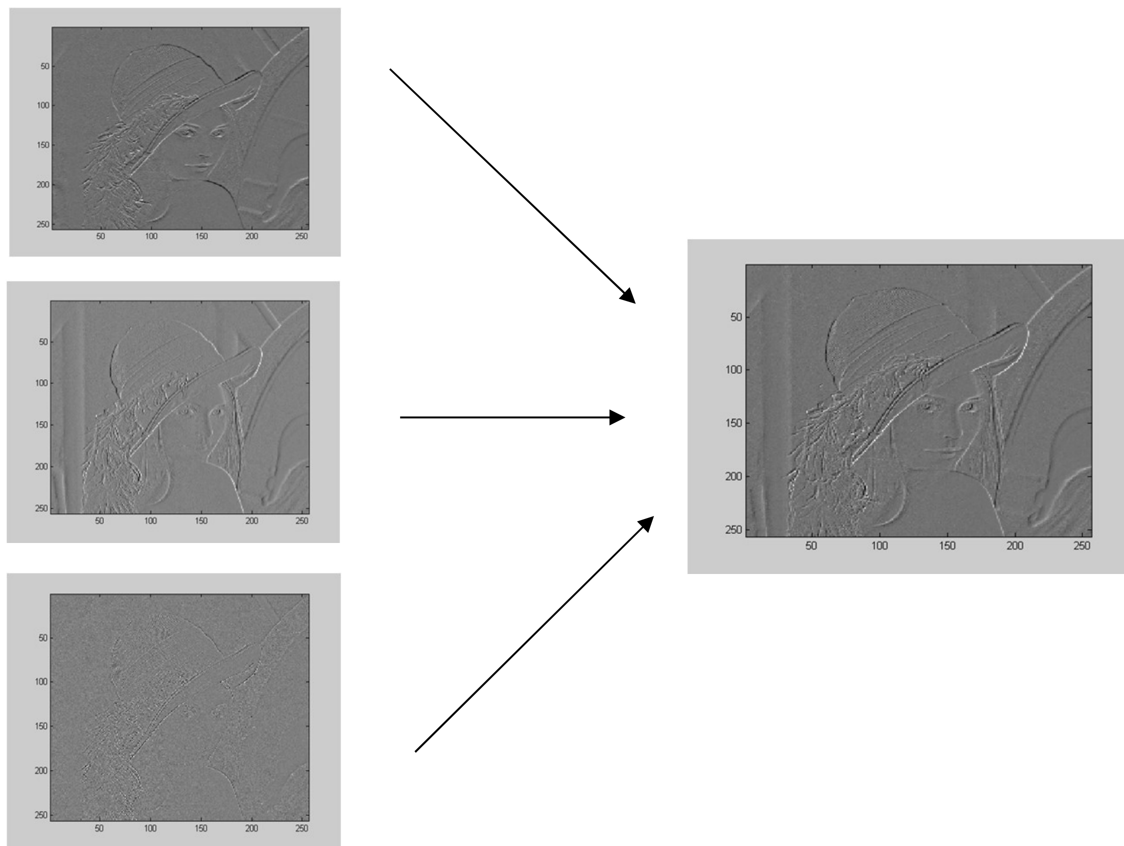
2D-DWT



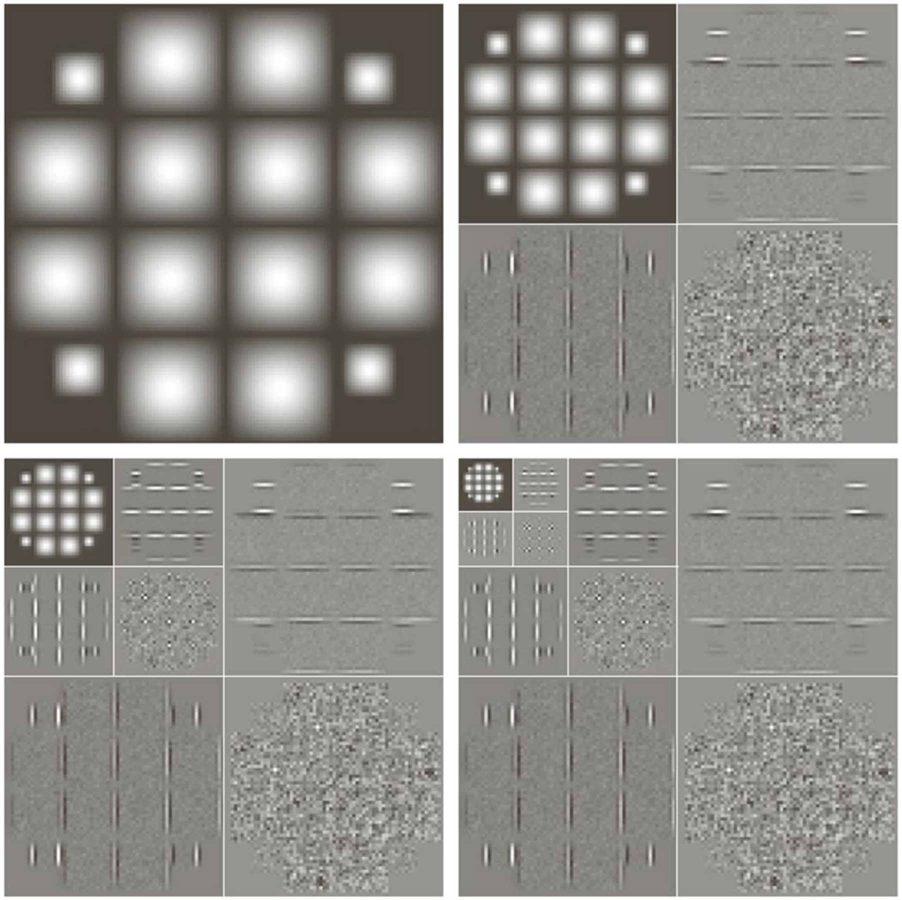
2D-DWT


LL band of image
LH band of image
HL band of image
HH band of image

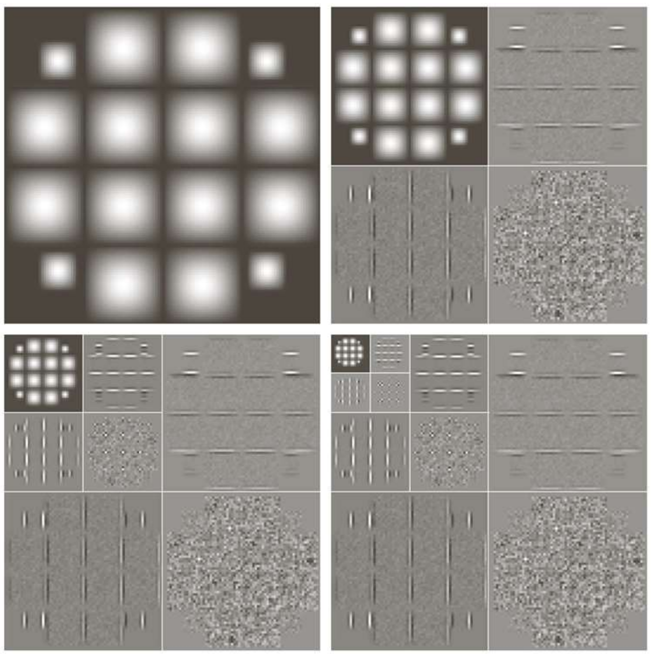
DWT applications: edge detection

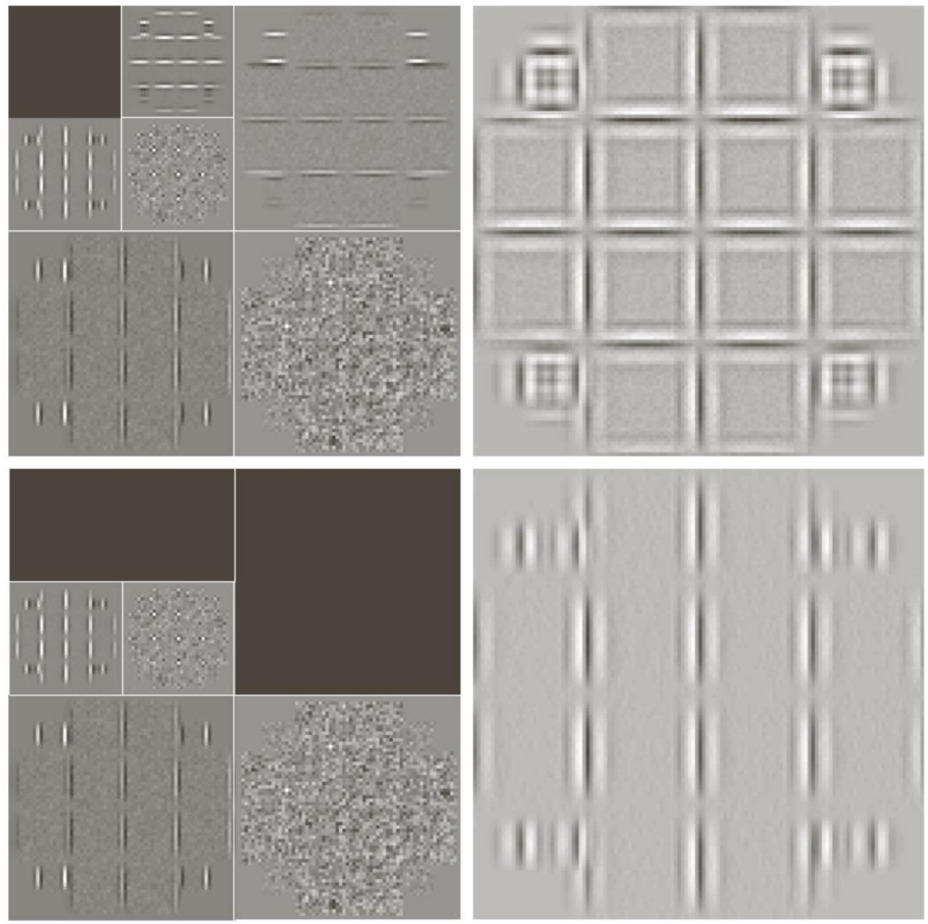


DWT applications: edge detection

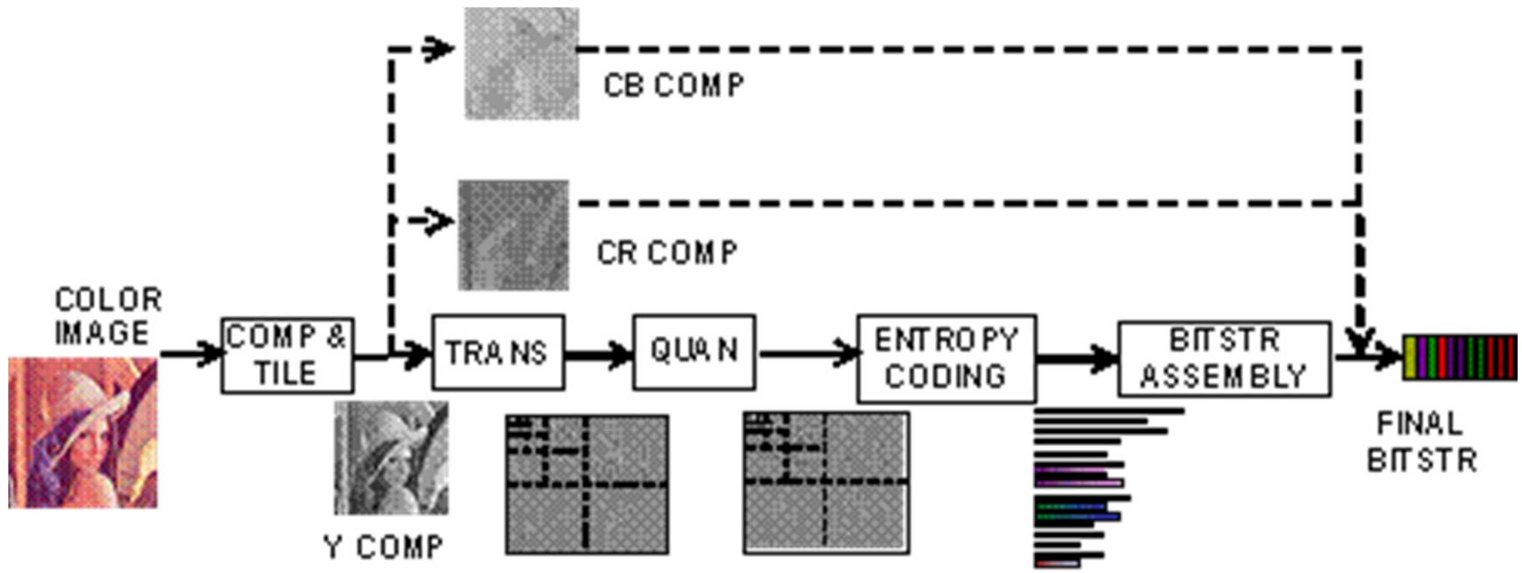


DWT applications: edge detection



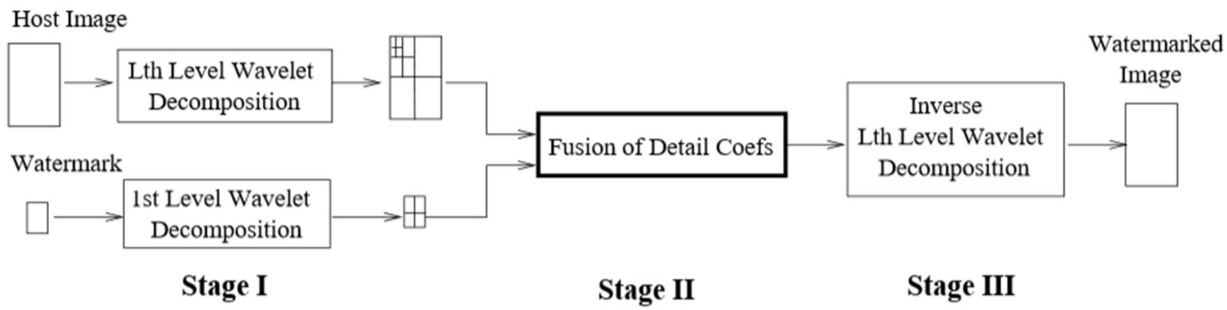


DWT applications: compression



JPEG 2000 Compression Pipeline

DWT applications: watermarking



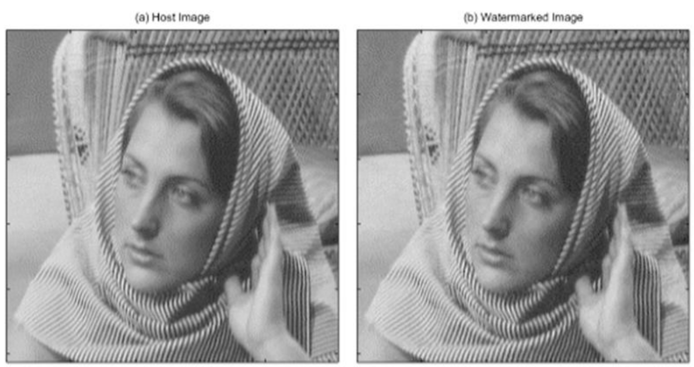


Figure 2: (a) Host Image (left), (b) Watermarked Image (right).

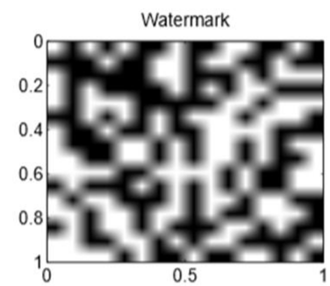
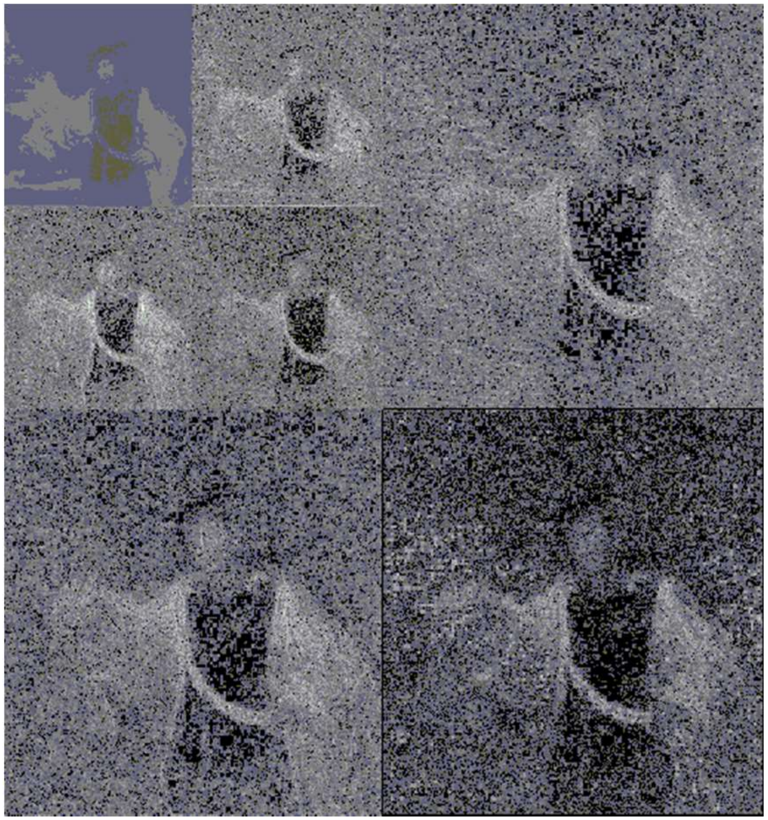


Figure 3: The 256 bit embedded watermark.

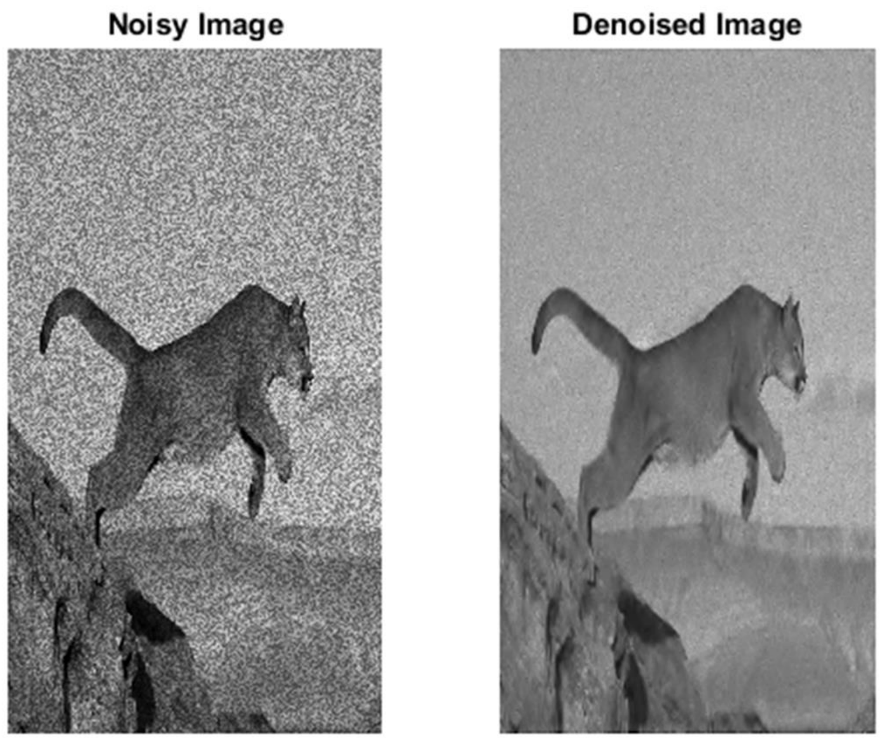
Courtesy: Kudur et al. ICIP 97

DWT applications: analysis

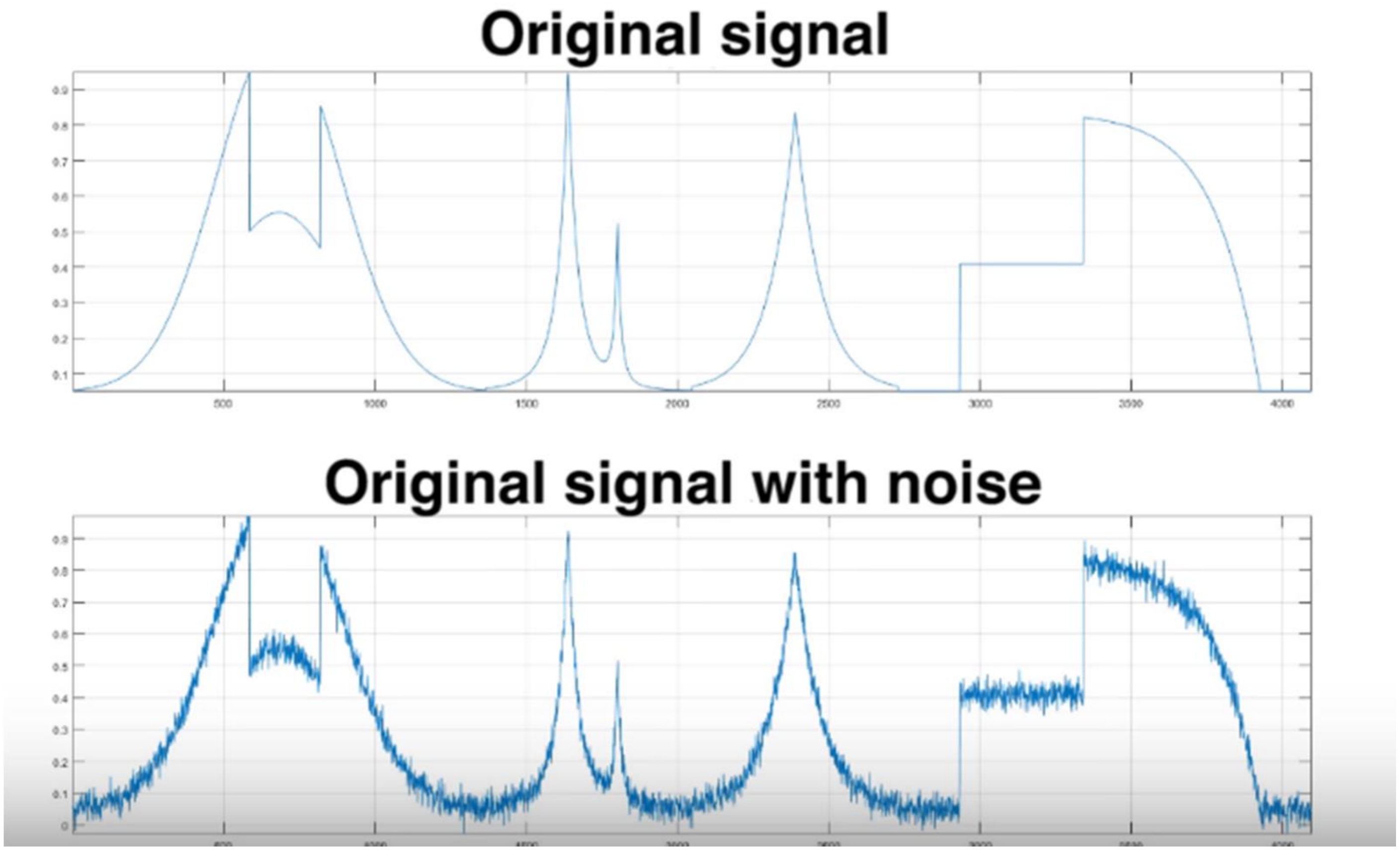


Courtesy: art spy

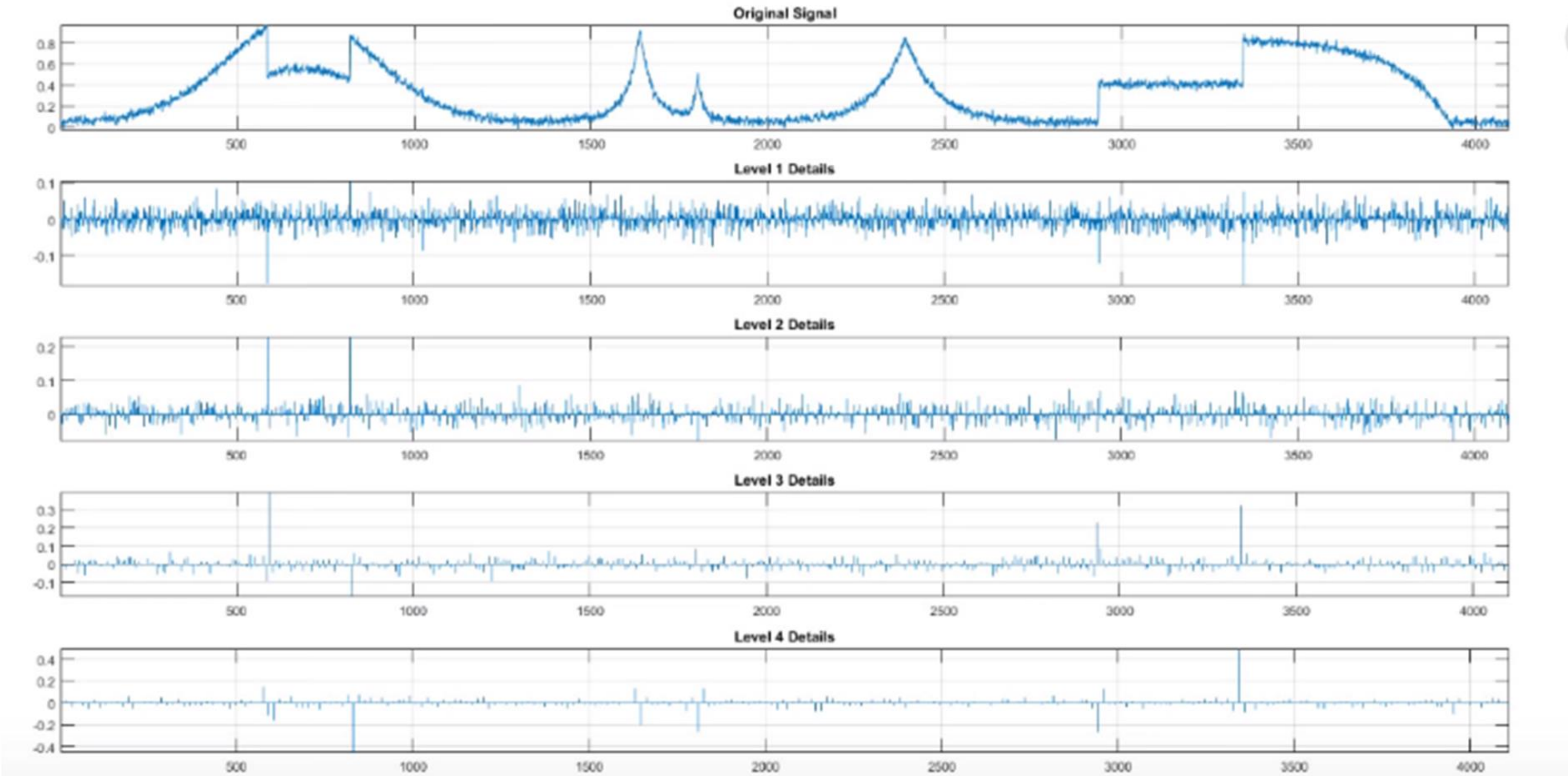
DWT applications: denoising



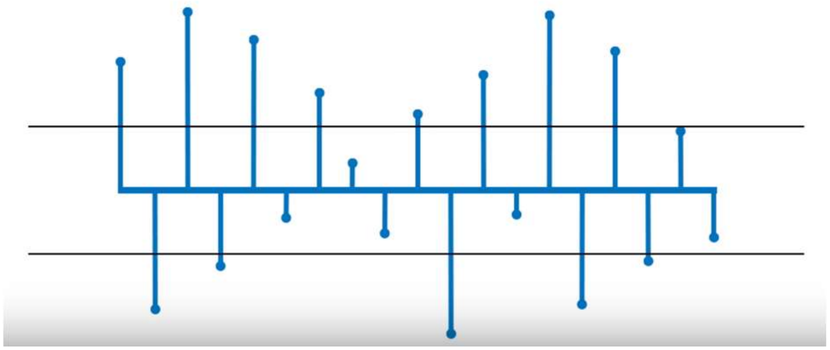
DWT applications: denoising

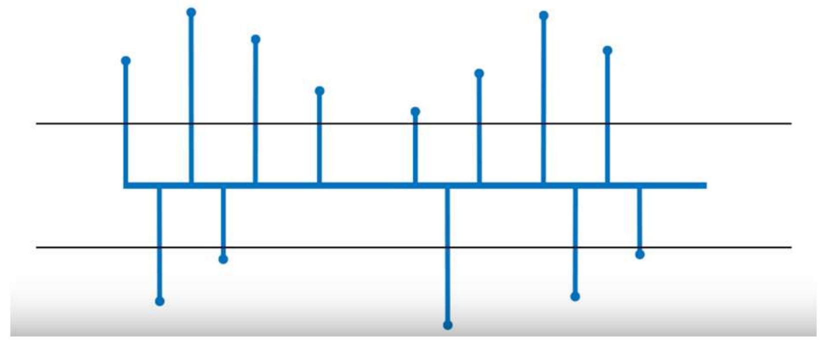


DWT applications: denoising



DWT applications: denoising



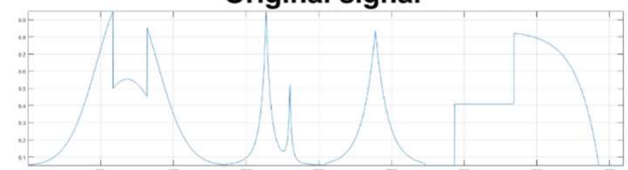


Hard Thresholding: Remove all coefficients below threshold

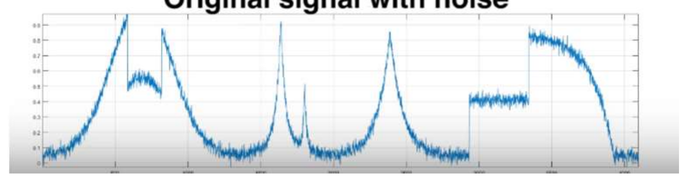
Soft Thresholding: Remove all coefficients below threshold and scale the other coefficients by threshold value

DWT applications: denoising

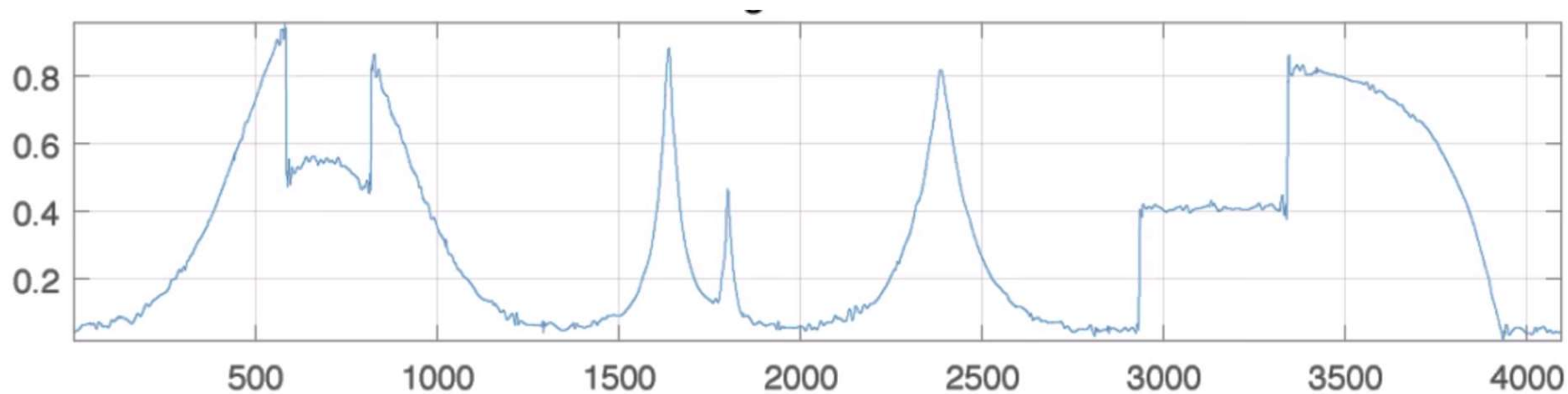
Original signal



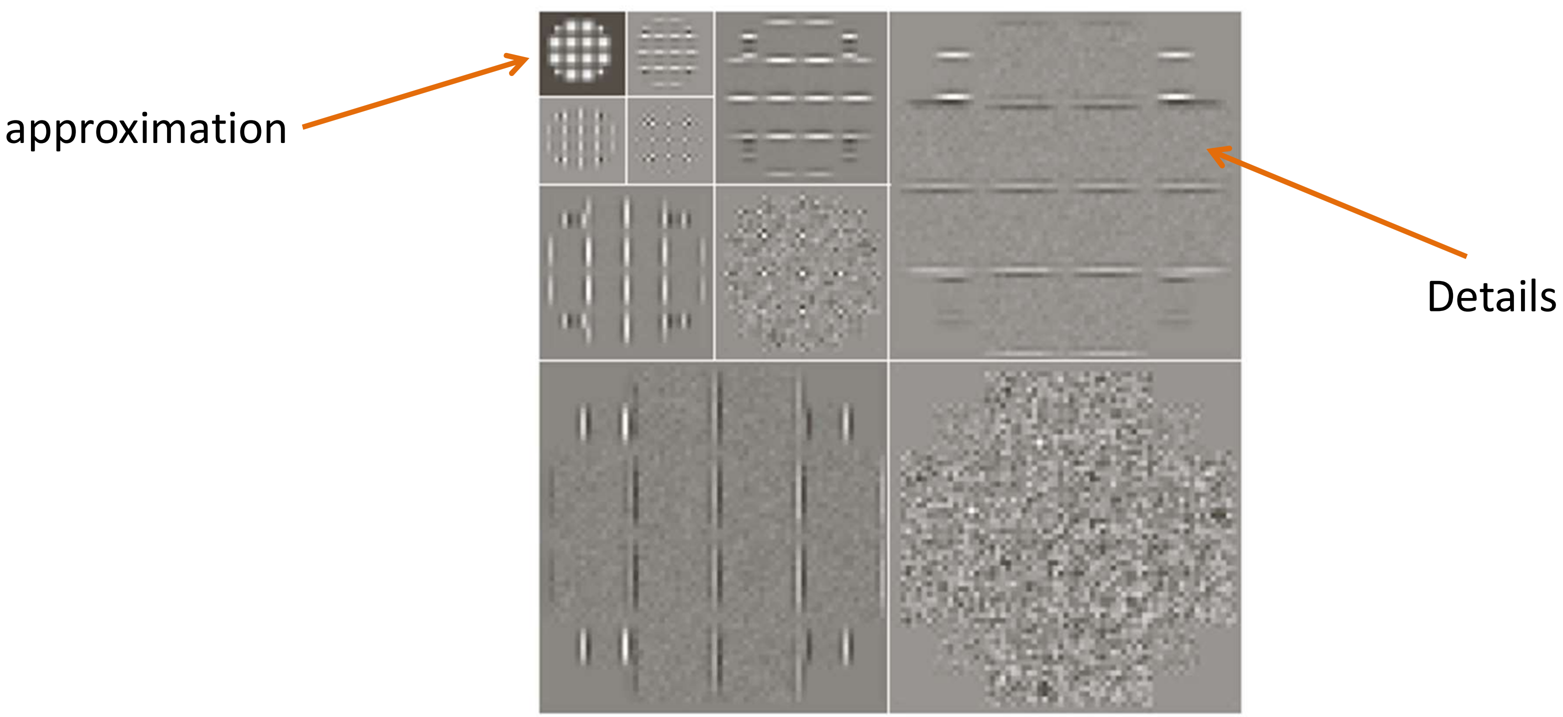
Original signal with noise



Denoised signal with soft thresholding



Re look: Multi scale-DWT



Multi scale-DWT

- Suppose we reconstruct the signal from $\mathbf{X}_l$ only

$$\text{IDWT}(\mathbf{X}_\ell || \mathbf{0}) = \alpha_1(\mathbf{x}) \in \mathbb{C}^N$$

- $\alpha_1(\mathrm{x})$ is the stage one approximation of x

- Signal reconstruction from detail coefficients:

$$\text{IDWT}(\mathbf{0} || \mathbf{X}_h) = \delta_1(\mathbf{x}) \in \mathbb{C}^N$$

- $\delta_1(\mathrm{x})$ is the stage one approximation of x

- Stage 1 representation of signal: $\boxed{\mathrm{x} = \alpha_1(\mathrm{x}) + \delta_1(\mathrm{x})}$

Multi scale-DWT

- At each stage, the sequence of detail representations $\delta_1(\mathrm{x}), \delta_2(\mathrm{x}), \dots \dots, \delta_{m-1}(\mathrm{x})$ is extended by one term, $\delta_m(\mathrm{x})$

$$
\begin{array}{ll}
\text{stage 1:} & \mathbf{x} = \alpha_1(\mathbf{x}) + \delta_1(\mathbf{x}) \\
\text{stage 2:} & \mathbf{x} = \alpha_2(\mathbf{x}) + \delta_2(\mathbf{x}) + \delta_1(\mathbf{x}) \\
\text{stage 3:} & \mathbf{x} = \alpha_3(\mathbf{x}) + \delta_3(\mathbf{x}) + \delta_2(\mathbf{x}) + \delta_1(\mathbf{x}) \\
 & \vdots \\
\text{stage m:} & \mathbf{x} = \alpha_m(\mathbf{x}) + \delta_m(\mathbf{x}) + \delta_{m-1}(\mathbf{x}) + \cdots + \delta_1(\mathbf{x})
\end{array}
$$

Let me invert the notations!

Multiscale representation of a signal

Multi scale- 1D Haar Wavelet transform

l_a	1/2	1/2

h_a	-1/2	1/2

9	7	3	5	6	10	2	6

Approx α_2

8	4	8	4

1	-1	-2	-2

Detail coeff δ_2

Approx α_1

6	6

2	2

Detail coeff δ_1

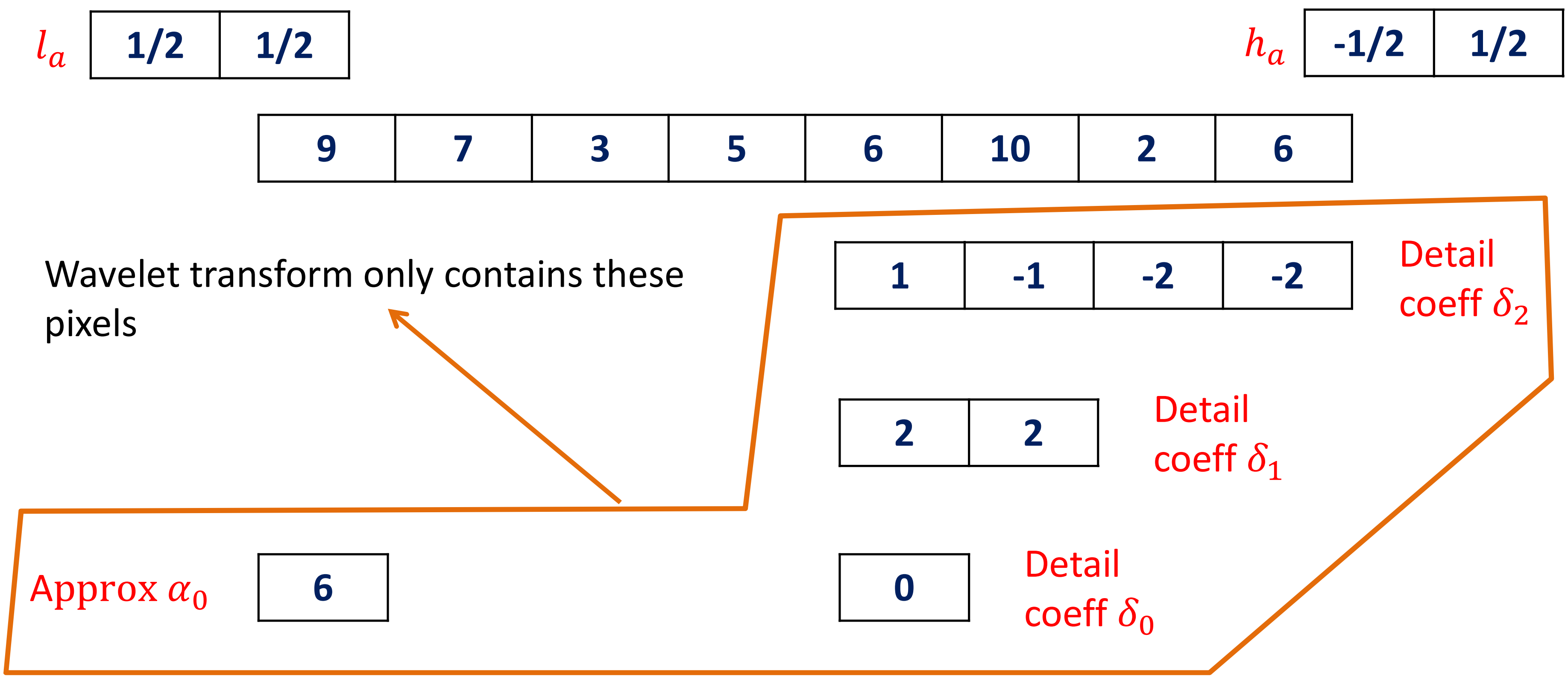
Approx α_0

6

0

Detail coeff δ_0

Multi scale- 1D Haar Wavelet transform



Multi scale- 1D Haar Wavelet transform

l_a

1/2	1/2

h_a

-1/2	1/2

9	7	3	5	6	10	2	6

Wavelet transform only contains these pixels

1	-1	-2	-2

Detail coeff δ_2

2	2

Detail coeff δ_1

Approx α_0

6

0

Detail coeff δ_0

X (3 stage) =

6	0	2	2	1	-1	-2	-2

Multi scale- 1D Haar Wavelet transform

$$
\begin{array}{c}
\alpha \\ \delta_0 \\ \delta_1 \\ \\ \delta_2 \\ \\ \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{array}{c}
\\ \\ \\ \\ \\ \\ \frac{1}{2} \\ \\
\end{array}
\begin{bmatrix}
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

Multi scale- 1D Haar Wavelet transform

9	7	3	5	6	10	2	6

8	4	8	4	1	-1	-2	-2

$$
\begin{array}{c} \alpha_0 \\ \delta_0 \\ \delta_1 \\ \\ \delta_2 \\ \\ \\ \\ \end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{bmatrix}
 & & & & & & & \\
 & & & & & & & \\
\frac{1}{2}\,1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\cdot
\frac{1}{2}
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

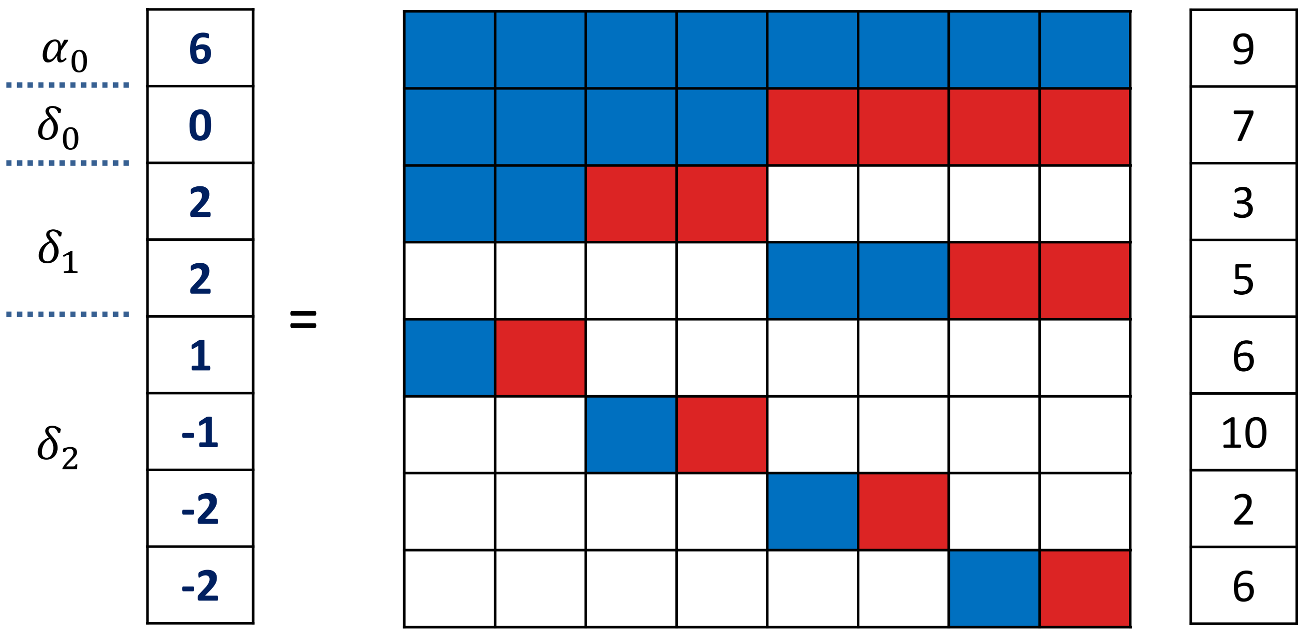
Multi scale- 1D Haar Wavelet transform

$$\begin{array}{c} x_0 \\ D_0 \\ \\ D_1 \\ \\ D_2 \\ \\ \\ \end{array} \begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix} = \begin{array}{c} \\ \\ \\ \frac{1}{4} \\ \\ \\ \frac{1}{2} \\ \\ \end{array} \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}$$

Multi scale- 1D Haar Wavelet transform

$$
\begin{array}{c}
\alpha_0 \\ \delta_0 \\ \\ \delta_1 \\ \\ \\ \delta_2 \\ \\
\end{array}
\begin{bmatrix} 6 \\ 0 \\ 2 \\ 2 \\ 1 \\ -1 \\ -2 \\ -2 \end{bmatrix}
=
\begin{array}{c}
\frac{1}{8} \\ \\ \frac{1}{4} \\ \\ \\ \frac{1}{2} \\ \\ \\
\end{array}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix} 9 \\ 7 \\ 3 \\ 5 \\ 6 \\ 10 \\ 2 \\ 6 \end{bmatrix}
$$

Multi scale- 1D Haar Wavelet transform



Multi scale- Haar Basis

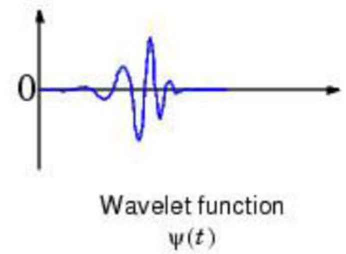
$$\psi(x) \equiv \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} < x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

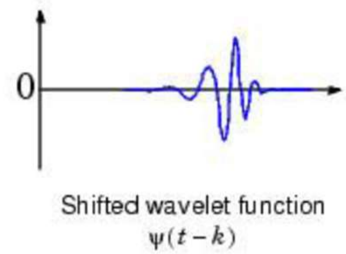
$$\psi_{jk}(x) \equiv \psi\left(2^j x - k\right)$$

j is non negative integer and $0 \leq k \leq 2^j - 1$

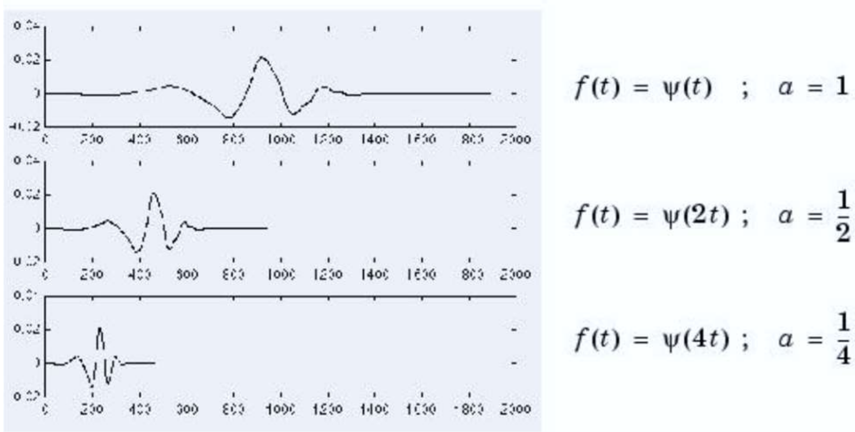
Scaling and Shifting of Wavelets

- Shifting a wavelet





- Scaling a wavelet



Multi scale- 1D Haar Wavelet transform

- Two stage transform

$$\mathcal{W}_2^a = \begin{pmatrix} W_{N/2}^a & 0 \\ 0 & \mathbf{I}_{N/2} \end{pmatrix} W_N^a$$

Multi scale- 1D Haar Wavelet transform

- An r stage DWT is obtained by iteration:

$$\mathcal{W}_r^a = \begin{pmatrix} W^a_{N/2^{r-1}} & 0 \\ 0 & \mathbf{I}_{N(1-1/2^{r-1})} \end{pmatrix} \cdots \begin{pmatrix} W^a_{N/2} & 0 \\ 0 & \mathbf{I}_{N/2} \end{pmatrix} W^a_N$$

- The inverse DWT is governed by the matrix

$$\mathcal{W}_r^s = W^s_N \begin{pmatrix} W^s_{N/2} & 0 \\ 0 & \mathbf{I}_{N/2} \end{pmatrix} \cdots \begin{pmatrix} W^s_{N/2^{r-1}} & 0 \\ 0 & \mathbf{I}_{N(1-1/2^{r-1})} \end{pmatrix}$$