CS270: Digital Image Processing

Dr. Xavier LAGORCE

Email: xavier@shanghaitech.edu.cn

Office: SIST 1D-301B

ShanghaiTech University

Lecture 9
Wavelet and Other Image Transforms

Dr. Xavier LAGORCE
Email: xavier@shanghaitech.edu.cn
Office: 1D-301.B, SIST
ShanghaiTech University

上海科技大学
ShanghaiTech University

立志成才报国裕民

Outline

上海科技大学 ShanghaiTech University

- **2D Unitary transform**
- **Frequency Domain Extension**
  - Discrete Cosine Transform（余弦变换）
  - Hadamard Transform （哈德马变换）
- **Discrete Wavelet Transform (DWT, 小波变换)**
  - An example for 1D-DWT
  - Generalization of 1D-DWT
  - 2D-DWT

立志成才 报国裕民

Unitary Transform

- **Forward Transform:**

Matrix Vector

$$t = Af$$

$$t[k] = \sum_{n=0}^{N-1} A[k,n] f[n]$$

- **Inverse Transform:**

In that case A is a unitary transform

$$f = A^H t \quad \text{where } A^H = (A^T)^* \text{ and } AA^H = I$$

Conjugate transpose

Example for 1D Unitary Transform

上海科技大学 ShanghaiTech University

- Image rotation:

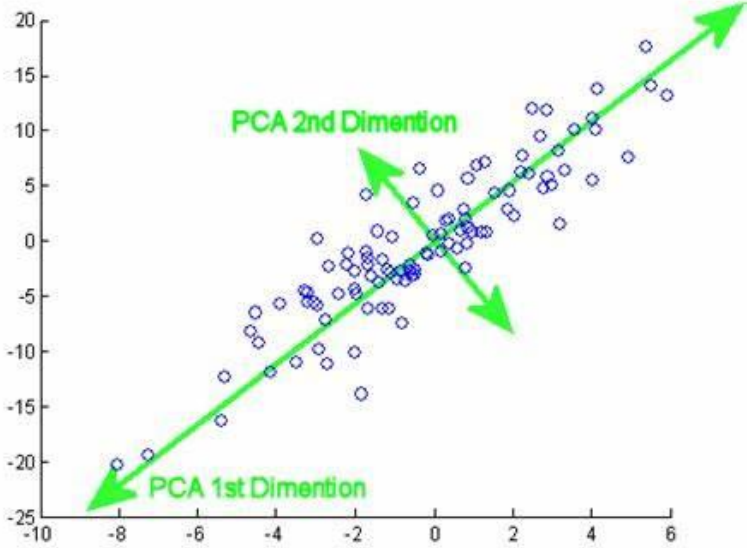
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

- Principle Component Analysis (PCA):

$$Y = PX$$

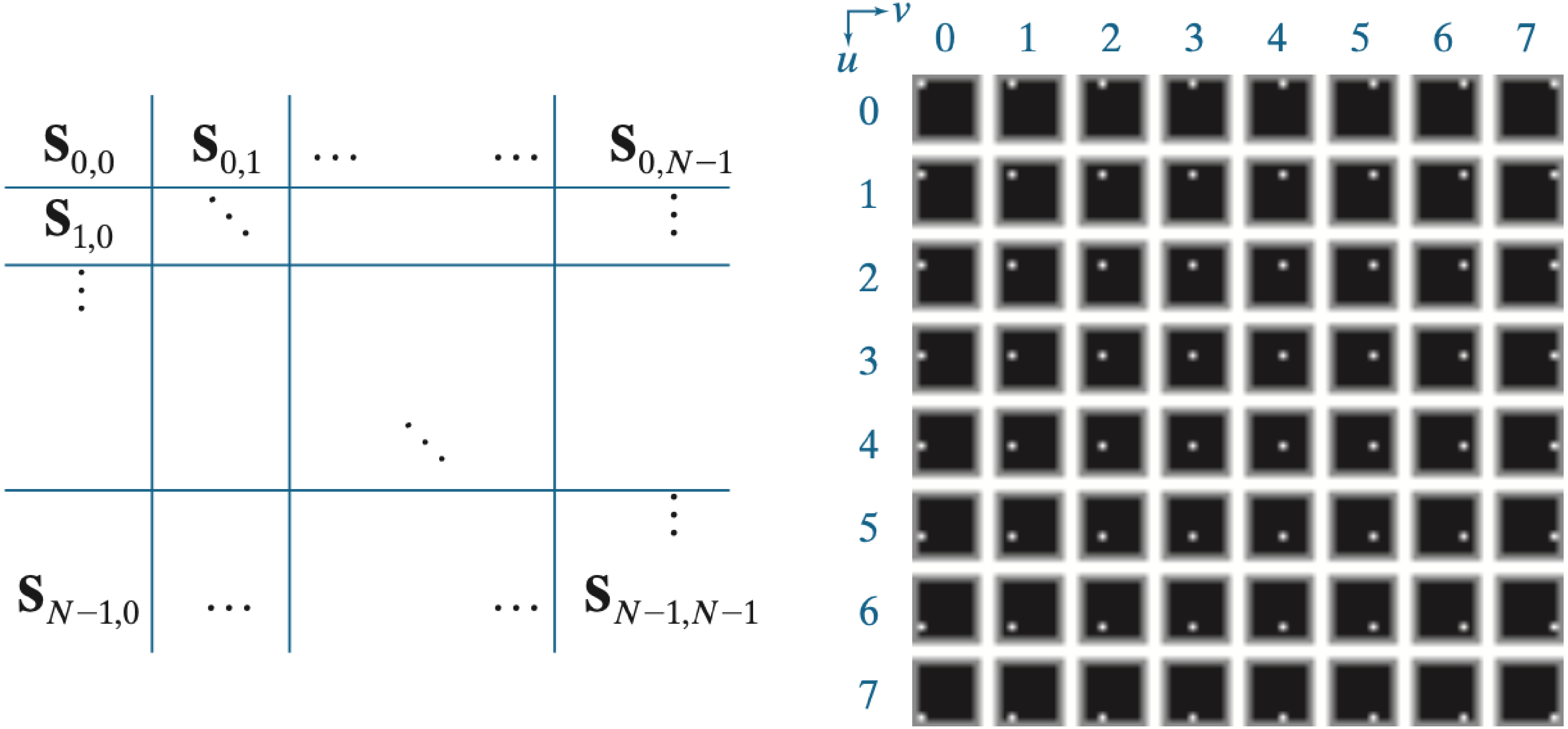
  - that satisfies:

$$C = XX^T, \qquad D = PCP^T, \qquad \text{and} \;\; PP^T = I$$



立志成才报国裕民

Goal: Basis Functions



a b

FIGURE 7.6
(a) Basis image organization and (b) a standard basis of size 8×8. For clarity, a gray border has been added around each basis image. The origin of each basis image (i.e., $x = y = 0$) is at its top left.

Discrete Fourier Transform

上海科技大学
ShanghaiTech University

- **Forward Transform:**

$$t = Af; \quad t[k] = \sum_{n=0}^{N-1} A[k,n] f[n]$$

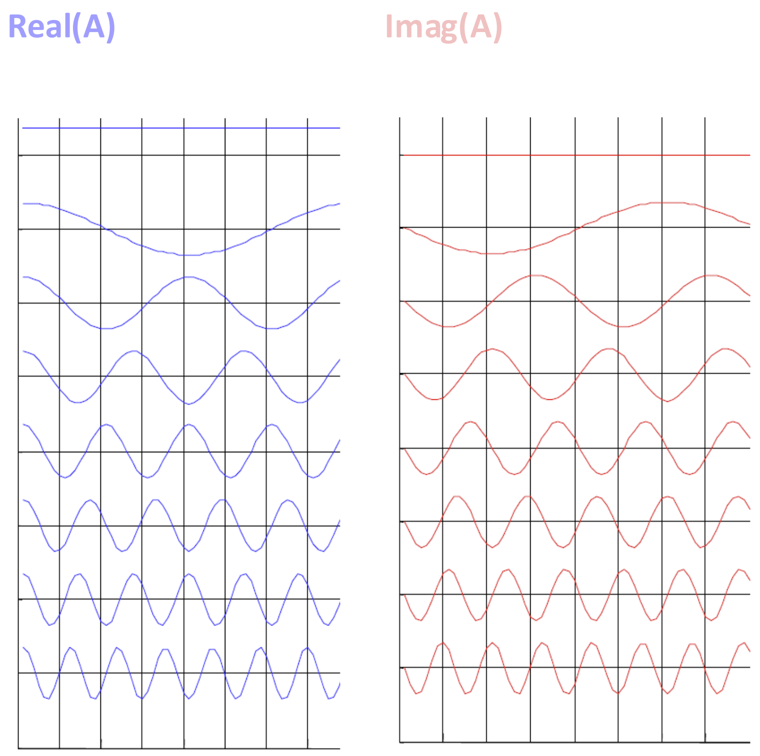
- **Inverse Transform：**

$$f = A^H t; \quad f[n] = \sum_{k=0}^{N-1} A^H[k,n] t[k]$$

- **1-D DFT**

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi kn}{N}}, \qquad (k = 1,2,\cdots,N)$$

$$A[k,n] = e^{-j\frac{2\pi kn}{N}} = cos\left(2\pi \frac{kn}{N}\right) - j.sin(2\pi \frac{kn}{N})$$



2D Unitary Transform

- Forward Transform:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

$$= A_M f A_N$$

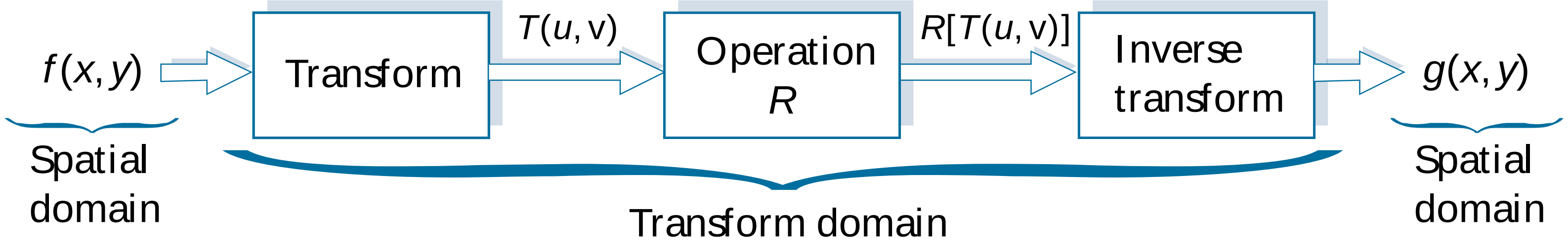
- Inverse Transform

$$f = A_M^H F A_N^H \qquad AA^H = I$$

Image Transform

上海科技大学
ShanghaiTech University

- The general approach for operating in linear transform domain



- The unitary transform satisfies

$$\sum_{x=0}^{M}\sum_{y=0}^{N}(\boldsymbol{f}[\boldsymbol{x},\boldsymbol{y}])^2 = \sum_{u=0}^{M}\sum_{v=0}^{N}(\boldsymbol{F}[\boldsymbol{u},\boldsymbol{v}])^2$$

  - i.e., the energy is preserved.

立志成才报国裕民

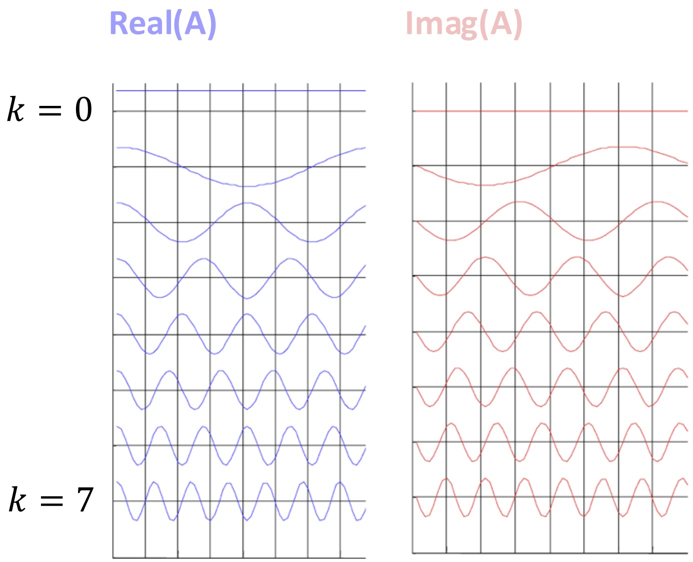
Good and Bad things about DFT

- ✅ **Positive：**
  - Energy is usually packed into low-frequency coefficients
  - Convolution property
  - Fast implementation
- ❌ **Negative：**
  - Transform is complex, even if image is real
  - The basis functions span the whole image height and width
  - Need to process stationary signals

DFT vs. DCT (Discrete Cosine Transform)

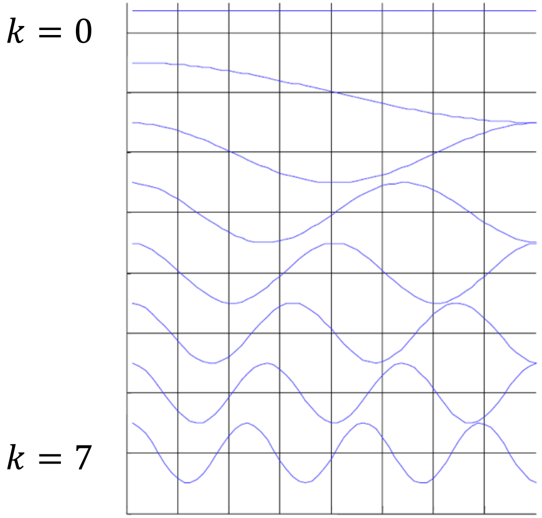
- **1D-DFT**

$$A[k,n] = e^{-j\frac{2\pi kn}{N}}$$
$$= cos\left(2\pi\frac{kn}{N}\right) + jsin\left(2\pi\frac{kn}{N}\right)$$



- **1D-DCT**

$$A[0,n] = \sqrt{\frac{1}{N}},\ A[k,n] = \sqrt{\frac{2}{N}}\, cos\frac{\pi(2n+1)k}{2N} \text{ for } k > 0$$



- Only real part
- Half-frequencies

What's the difference???

2D DCT

- Forward Transform

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

Still separable !

where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{for } u = 0 \\ \sqrt{\frac{2}{N}}, & \text{for } u = 1, 2, \dots, N-1 \end{cases}$$

- Inverse Transform

$$f(x,y) = \frac{1}{N}F(0,0)$$

$$+\frac{\sqrt{2}}{N}\sum_{u=1}^{N-1} F(u,0)\cos\frac{(2x+1)u\pi}{2N}$$

$$+\frac{\sqrt{2}}{N}\sum_{v=1}^{N-1} F(0,v)\cos\frac{(2y+1)v\pi}{2N}$$

$$+\frac{2}{N}\sum_{x=1}^{N-1}\sum_{y=1}^{N-1} F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

2D DCT – Requirements

上海科技大学 ShanghaiTech University

- Direct Cosine Transform assumes:
  - 2N-point periodicity
  - Even symmetry

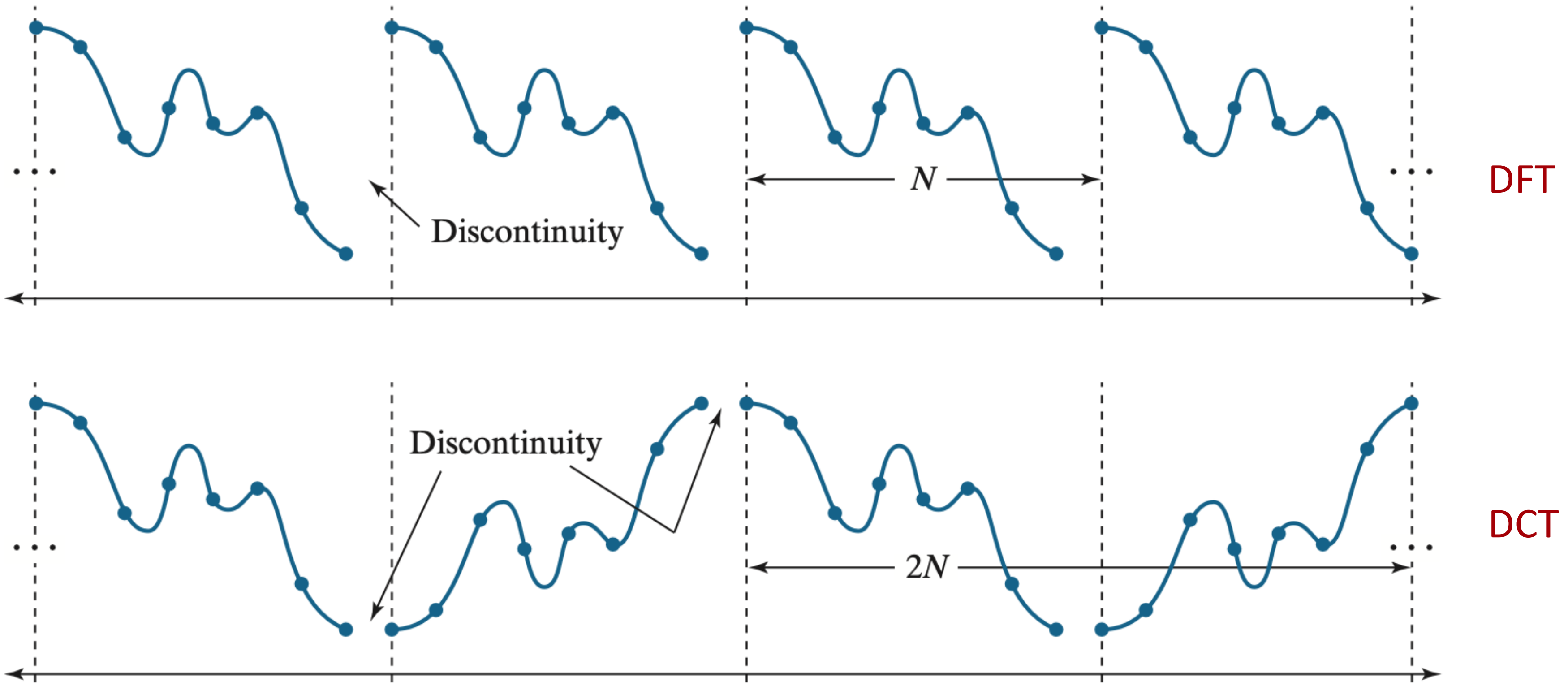
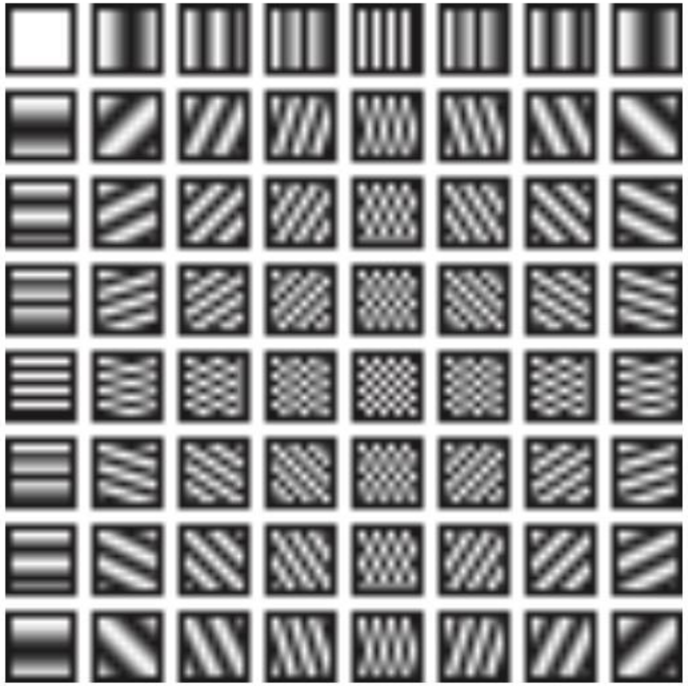


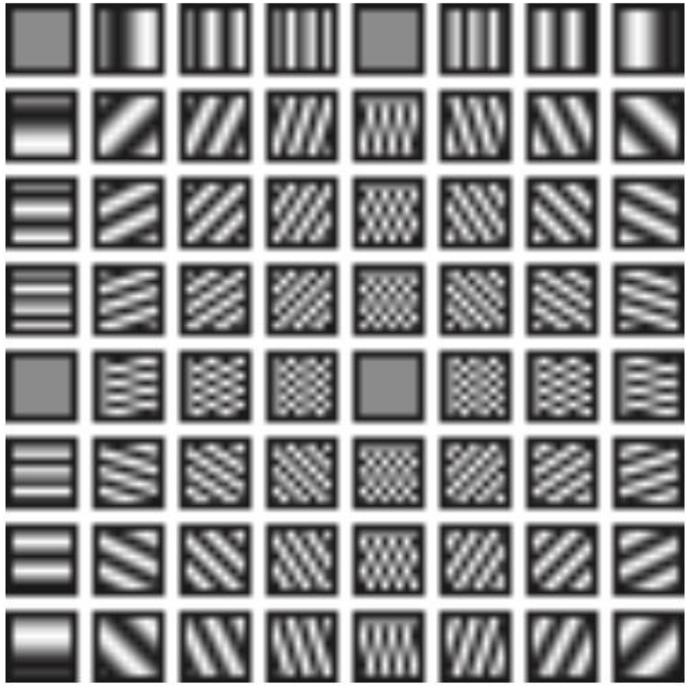
FIGURE 7.11 The periodicity implicit in the 1-D (a) DFT and (b) DCT.

立志成才报国裕民

DFT vs DCT: Basis Functions

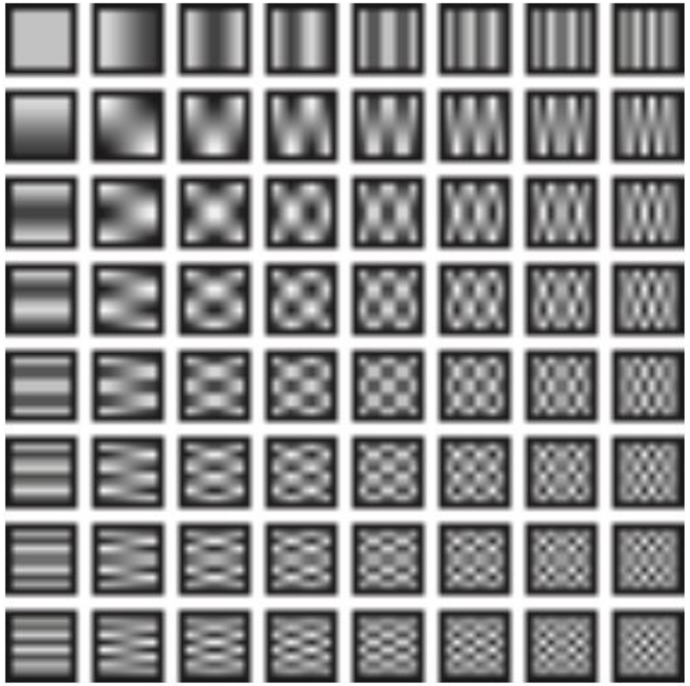


Real



Imag

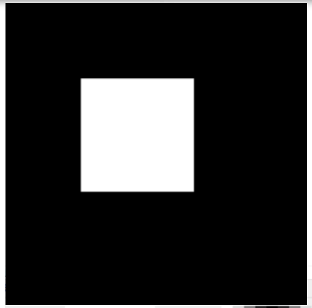
DFT



DCT

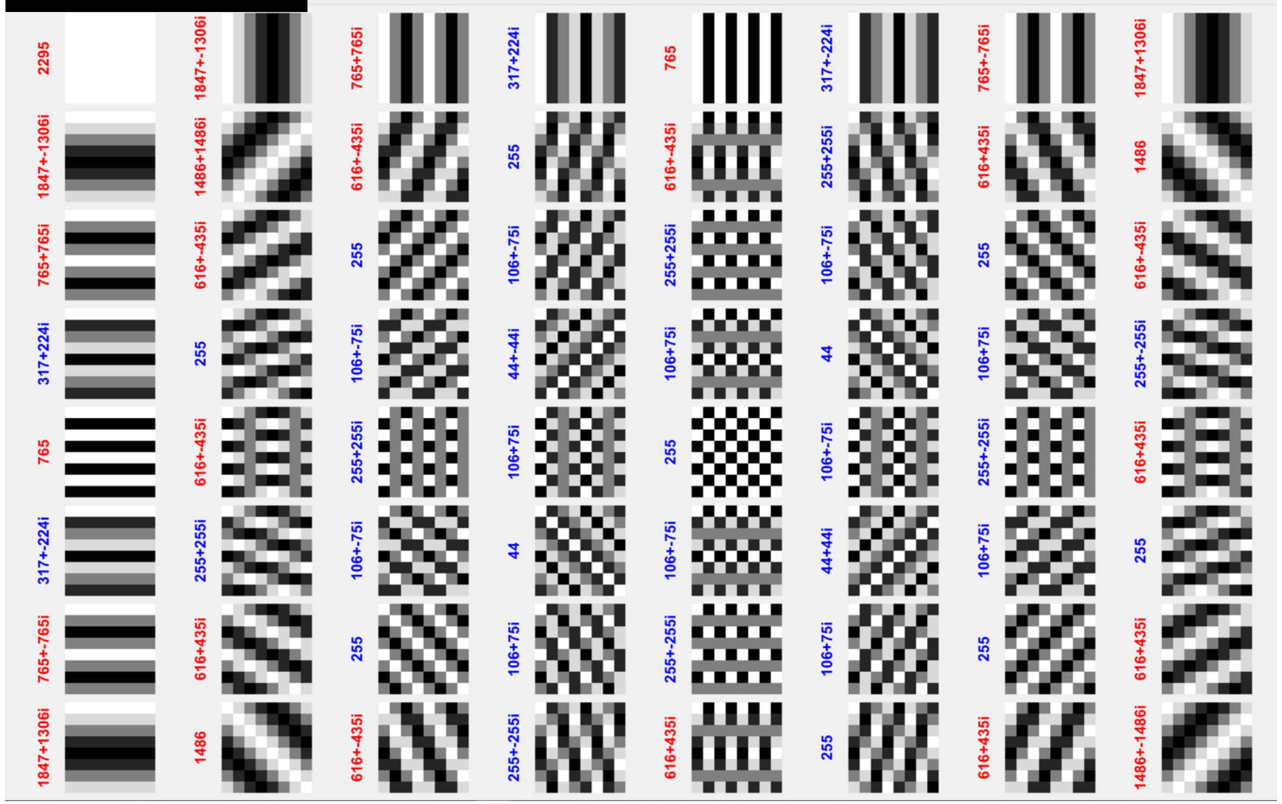
DFT example

- Complicated transform even if the signal is simple
  - Red: frequencies with weight above average
  - Blue: frequencies with weight below average
- Frequencies are going up and then down through the basis functions
- Large weights everywhere and large differences
- Kind of complexifies the problem



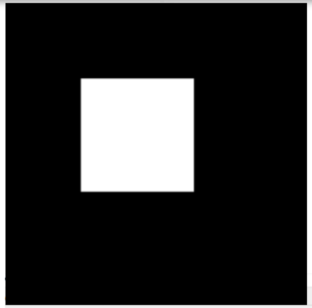
Input image

DFT coefficients



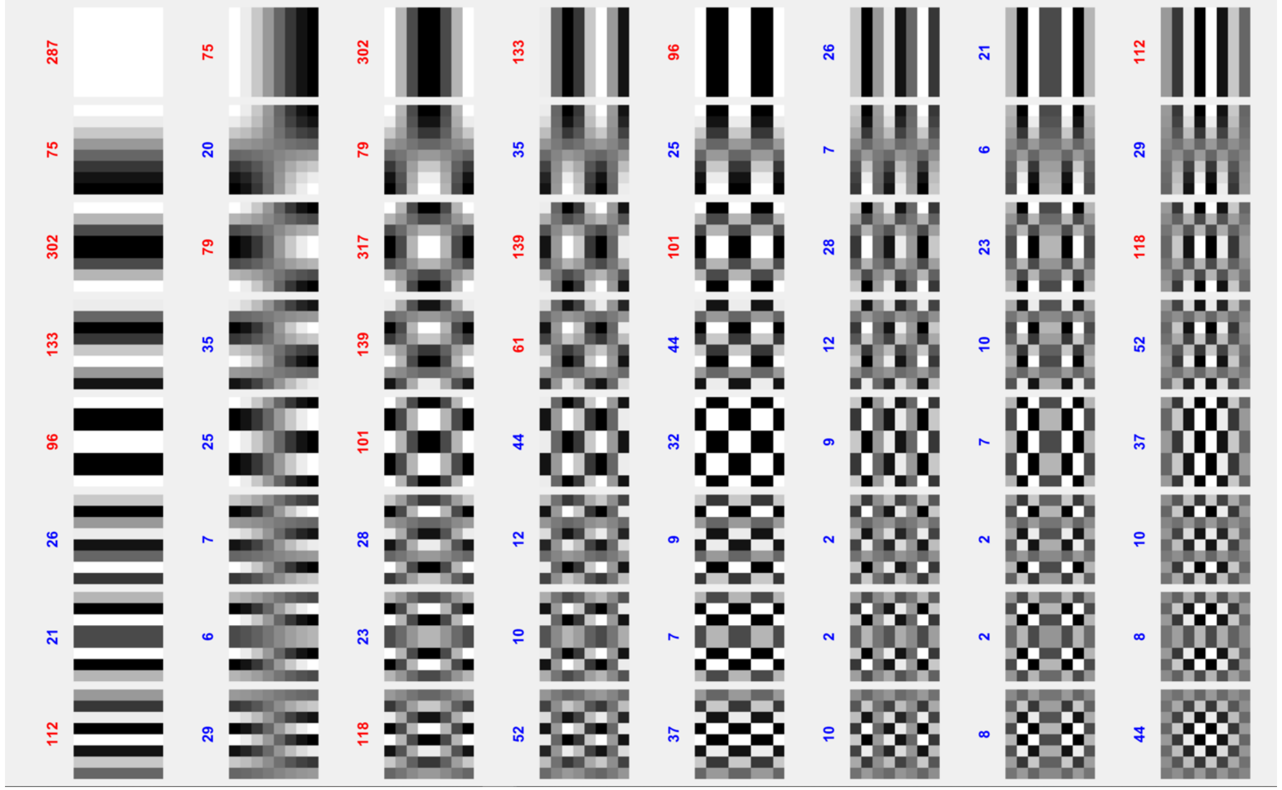
DCT example

- Same representation:
  - Red: frequencies with weight above average
  - Blue: frequencies with weight below average
- Large weights grouped in the top left corner ! (and more similar)
- No more complex numbers, more regular variation of frequencies in basis functions
- Still complexifies the problem



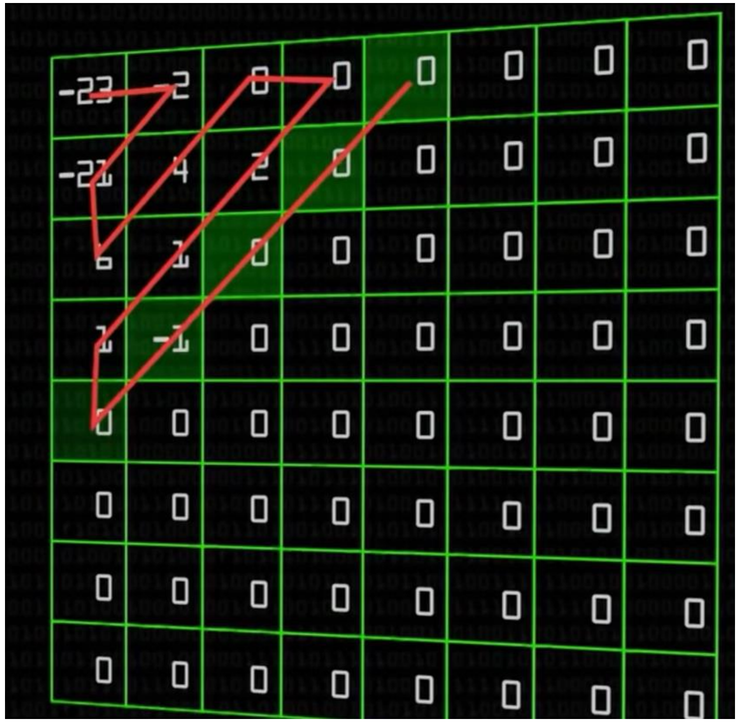
Input image

DCT coefficients



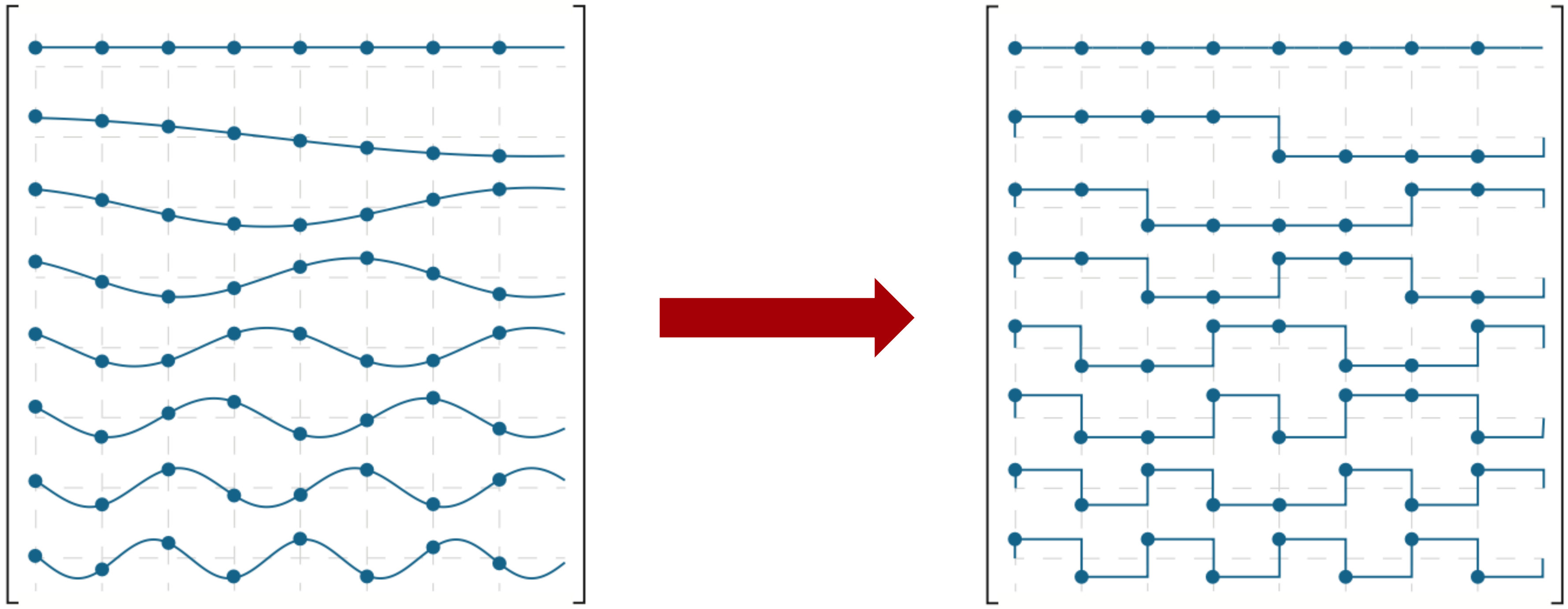
Good things about DCT

- ✅ **Positive：**
  - Transform is real, $C^{-1} = C^{T}$ (unitary transform).
  - Excellent energy compaction for nature images.
  - Fast transform.
  - JPEG algorithms.



How to simplify the transforms ?

- From DFT to DCT we already moved from complex numbers to real numbers
- But trigonometry functions are still complicated....



立志成才报国裕民

Walsh Transform

- Consist of ± 1 arranged in a checkerboard pattern.
- Transforms:

$$W(i) = \frac{1}{N}\sum_{t=0}^{N-1} f(t)\, Wal(i,t) \qquad f(t) = \sum_{t=0}^{N-1} W(i)\, Wal(i,t)$$

- Different types of $Wal(i,t)$ for different transforms:
  - Walsh Ordering (沃尔什定序)
  - Paley Ordering (佩利定序)
  - Hadamard Matrix Ordering (哈达玛矩阵定序)

Hadamard Matrix Ordering

Transformation matrix:
(contains all the basis vectors)

$$\mathbf{A}_W = \frac{1}{\sqrt{N}}\mathbf{H}_N$$

Where

$$\begin{cases} \mathbf{H}_{2N} = \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix} \\ \\ \mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{cases}$$

$$\mathbf{H}_4 = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Basis function

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Hadamard Matrix Ordering

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix}$$

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

Sequency: number of sign change along a row (like *frequency*)

Hadamard Matrix Ordering

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix}$$

$$= \begin{array}{cc} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} & \begin{array}{|c|} \hline 0 \\ \hline 7 \\ \hline 3 \\ \hline 4 \\ \hline 1 \\ \hline 6 \\ \hline 2 \\ \hline 5 \\ \hline \end{array} \end{array}$$

Sequency: number of sign change along a row (like *frequency*)

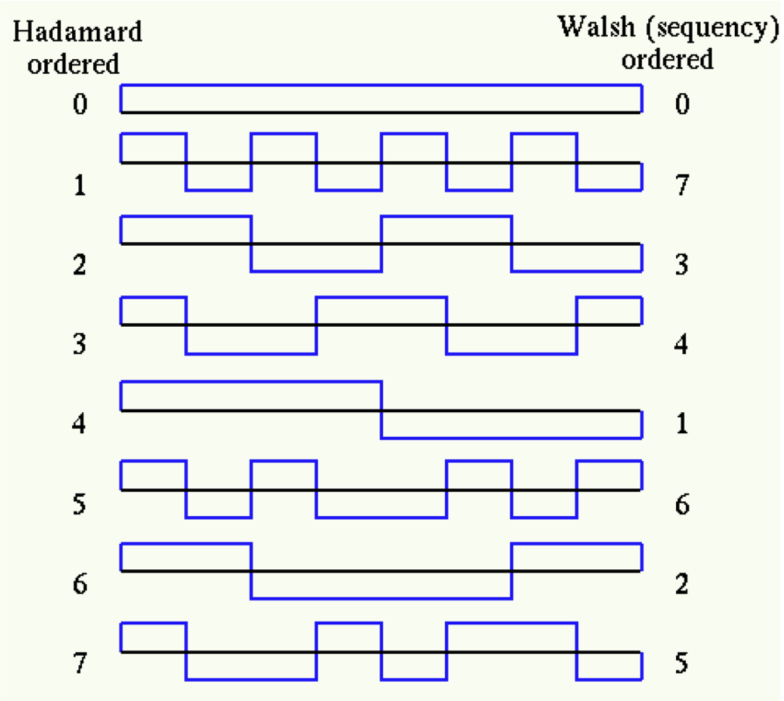
→ Measures the rate of change

Hadamard-ordered Walsh-Hadamard transform

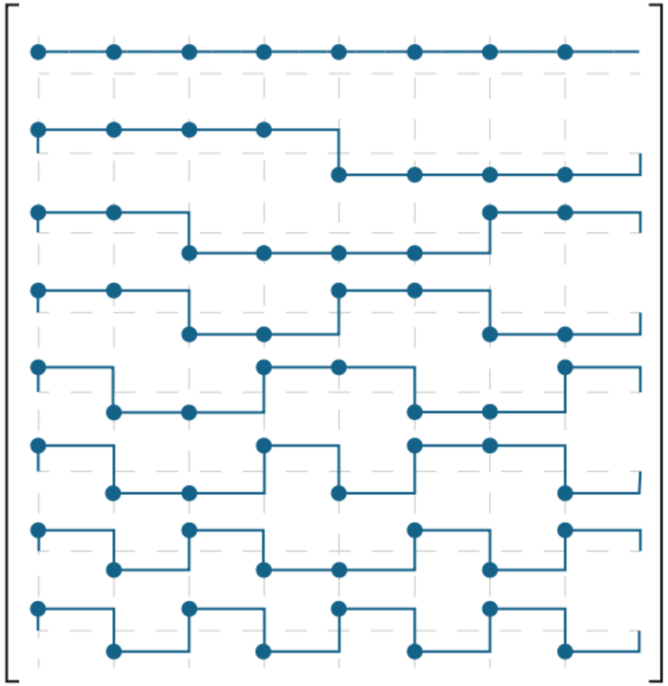
Hadamard Matrix Ordering

上海科技大学
ShanghaiTech University

- It is both common and desirable to sort the rows by increasing sequency



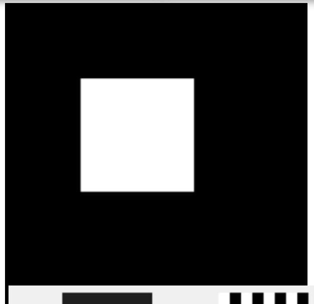
$$\mathbf{H}_8' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$



Sequency-ordered Walsh-Hadamard transform

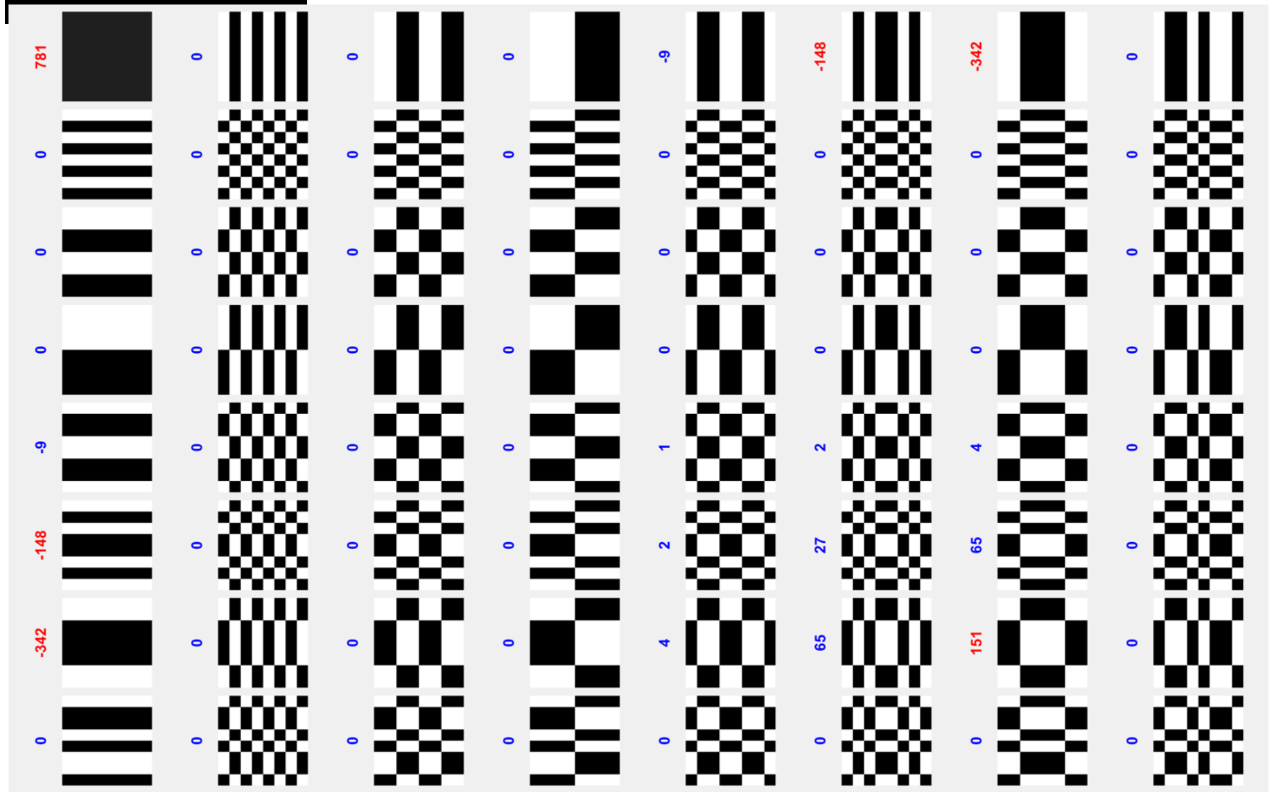
Hadamard example

- Same representation as before
  - Red: frequencies with weight above average
  - Blue: frequencies with weight below average
- Many zeroes → good for compression
- Energy is concentrated in a few weights



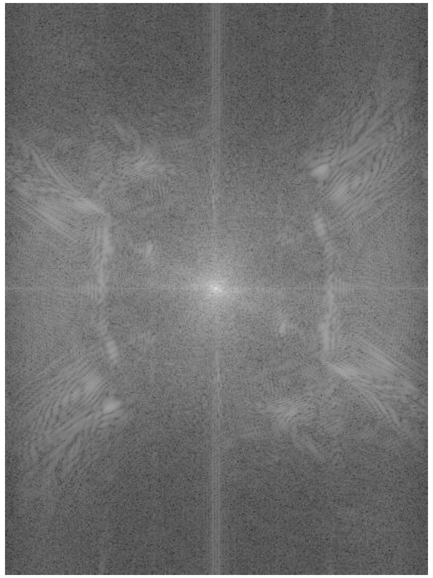
Input image

Hadamard coefficients



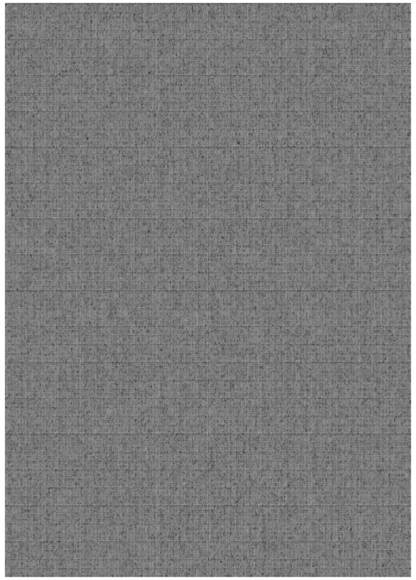
DFT, DCT, & Hadamard

DFT



DCT

Hadamard



Any connection between DFT and DCT?

- More complex image → Energy is more scattered in Hadamard transform
- The relationship between the spectrum and the image is still complicated…

Take home message

- The key idea for unitary transform is to find a proper basis for data decomposition.
- DCT provides better frequency consistency than DFT.
- Hadamard transform is able to represent a simple image with simple coefficients. But can not keep energy compact for image full of details.

Wavelet transform Outline

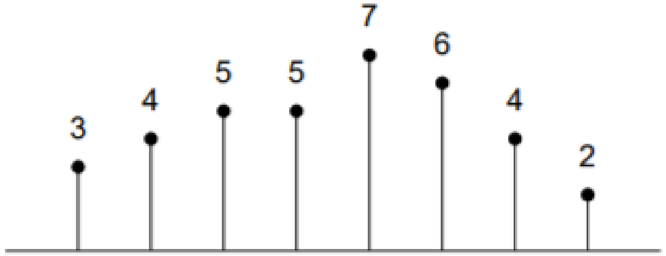
- Discrete Wavelet Transform (DWT)（小波变换）
  - An example for 1D-DWT
  - Generalization of 1D-DWT
  - 2D-DWT

Discrete Wavelet Transform (DWT)

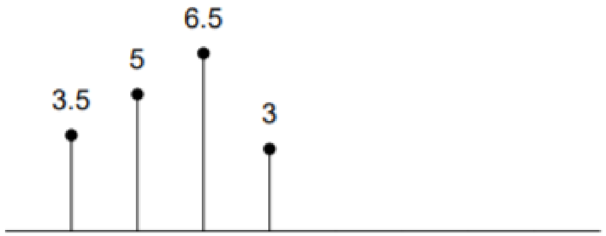
- Based on small waves called Wavelets:
  - Limited
  - Have oscillations
  - → small waves with band-limited spectra
- Key idea: Translation & Scaling.
- Localize both time/space and frequency.
- Efficient for noise reduction and image compression.
- Two types of DWT:
  - one for image processing (easily invertible)
  - one for signal processing (invertible but computationally expensive)

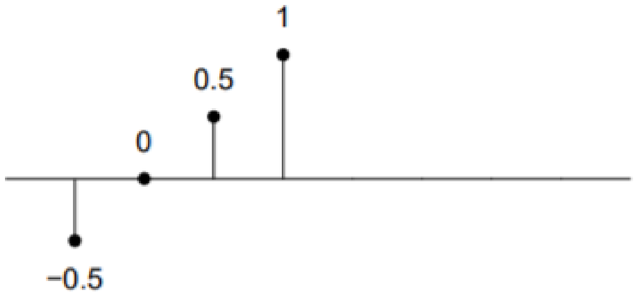
A simple example

- We can decompose an eight-point signal x(n):



into two four-point signals:





$$c(n) = 0.5\,x(2n) + 0.5\,x(2n+1)$$

Average of neighbors

$$d(n) = 0.5\,x(2n) - 0.5\,x(2n+1)$$

Difference of neighbors

A simple example

- **The above process can be represented by a block diagram:**

It is clear that this decomposition can be easily reversed:

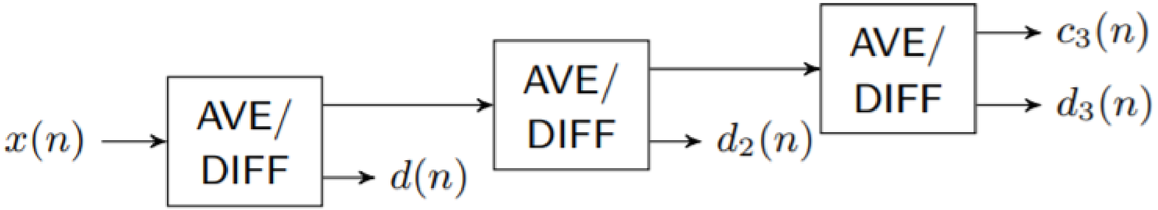
$$y(2n) = c(n) + d(n)$$
$$y(2n+1) = c(n) - d(n)$$

Which is also represented by a block diagram:

A simple example

上海科技大学 ShanghaiTech University

- When we repeat the simple AVE/DIFF signal decomposition:



$$x(n) = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8] = [3, 4, 5, 5, 7, 6, 4, 2]$$

Level 1

$c_1 =$ $\qquad$ $d = d_1 =$

Level 2

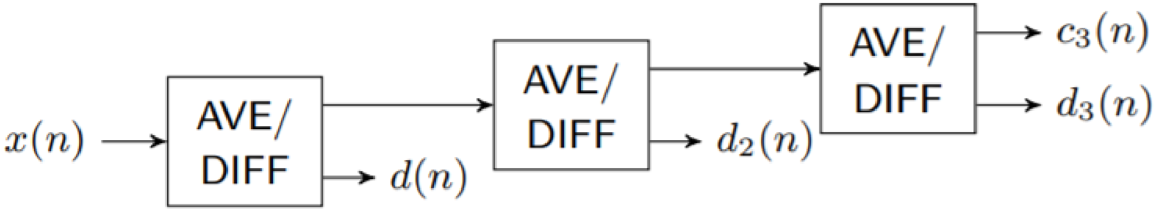
$c_2 =$ $\qquad$ $d_2 =$

Level 3

$c_3 =$ $\qquad$ $d_3 =$

A simple example

- When we repeat the simple AVE/DIFF signal decomposition:



$$x(n) = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8] = [3, 4, 5, 5, 7, 6, 4, 2]$$

Level 1

$$c_1 = \frac{1}{2}[x_1 + x_2, x_3 + x_4, x_5 + x_6, x_7 + x_8] \qquad d = d_1 = \frac{1}{2}[x_1 - x_2, x_3 - x_4, x_5 - x_6, x_7 - x_8]$$

Level 2

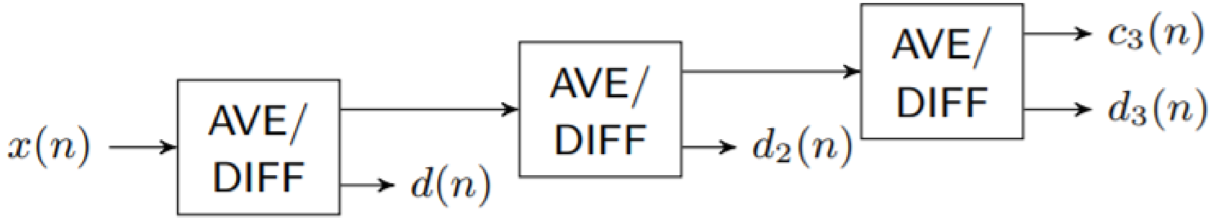
$$c_2 = \frac{1}{4}[x_1 + x_2 + x_3 + x_4, x_5 + x_6 + x_7 + x_8] \qquad d_2 = \frac{1}{4}[x_1 - x_3 + x_2 - x_4, x_5 - x_7 + x_6 - x_8]$$

Level 3

$$c_3 = \frac{1}{8}[x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8] \qquad d_3 = \frac{1}{8}[x_1 + x_2 + x_3 + x_4 - x_5 - x_6 - x_7 - x_8]$$

A simple example

- When we repeat the simple AVE/DIFF signal decomposition:



The Haar wavelet representation of the eight-point signal x[n] is simply the set of four output signals produced by this three-level operation :

$$
\begin{aligned}
c_3 &= [4.5] \\
d_3 &= [-0.25] \\
d_2 &= [-0.75, \quad 1.75] \\
d &= [-0.5, \quad 0, \quad 0.5, \quad 1]
\end{aligned}
$$

$$
\mathbf{H}_8 = \frac{1}{\sqrt{8}} \begin{array}{c} \left[\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
\sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\
2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -2
\end{array}\right] \end{array}
\begin{array}{l}
c_3 \\ d_3 \\ d_2(1) \\ d_2(2) \\ d(1) \\ d(2) \\ d(3) \\ d(4)
\end{array}
$$

Haar Transform matrix

- When N=2 we have:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- When N=4 we have:

$$\mathbf{H}_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

- When N=8 we have:

$$\mathbf{H}_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Haar Transform matrix

- The family of N Haar functions $h_u(x), (u = 0, \dots, N-1)$ is defined on the interval $0 \le x \le 1$. The shape of the specific function $h_u(x)$ of a given index u depends on two parameters p and q from the decomposition of u:

$$u = 2^p + q$$

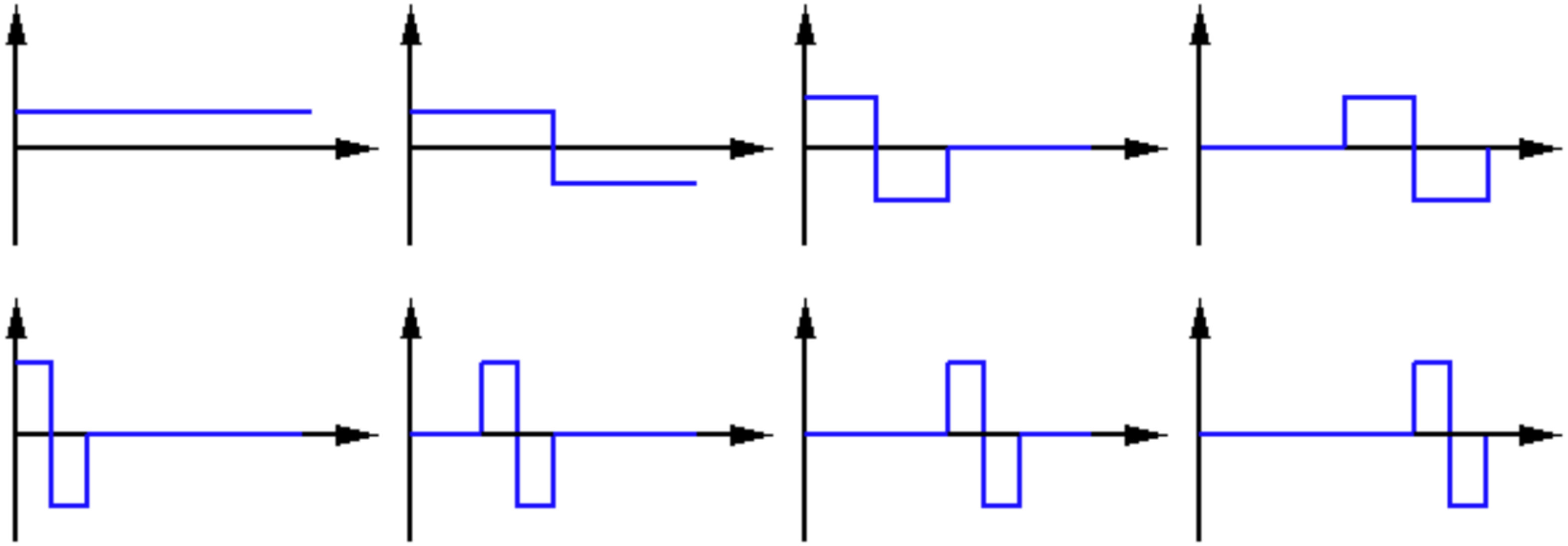
u	p	q
1	0	0
2	1	0
3	1	1

- The Haar basis functions are defined by:

$$h_u(x) = \begin{cases} 1 & u = 0 \text{ and } 0 \le x < 1 \\ 2^{p/2} & u > 0 \text{ and } q/2^p \le x < (q+0.5)/2^p \\ -2^{p/2} & u > 0 \text{ and } (q+0.5)/2^p \le x < (q+1)/2^p \\ 0 & \text{otherwise} \end{cases}$$

Haar Transform matrix

$$\mathbf{H}_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$



Generalization of 1D-DWT

- **Discrete Wavelet Transform (DWT):**

Scaling function (down-sampling)

Approximation coefficients

$$W_\varphi(j_0,k)=\frac{1}{\sqrt{M}}\sum_n f(n)\,\varphi_{j_0,k}(n)$$

Wavelets

Detail coefficients

$$W_\psi(j,k)=\frac{1}{\sqrt{M}}\sum_n f(n)\,\psi_{j,k}(n)\qquad j\geq j_0$$

- **Inverse Discrete Wavelet Transform (IDWT):**

$$f(n)=\frac{1}{\sqrt{M}}\sum_k W_\varphi(j_0,k)\,\varphi_{j_0,k}(n)+\frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_k W_\psi(j,k)\,\psi_{j,k}(n)$$

Where

$\varphi_{j_0,k}(n)$: scaling function (尺度函数) $\psi_{j,k}(n)$: Wavelet (小波)

$W_\varphi(j_0,k)$: Approximation coefficients (近似系数) $W_\psi(j,k)$: detail coefficients (细节系数)

2D-DWT

- **Define 2D wavelet function: Directionally sensitive wavelet**

$$\psi^H(x,y) = \psi(x)\varphi(y)$$

Variations along columns (i.e. horizontal edges)

$$\psi^V(x,y) = \varphi(x)\psi(y)$$

Variations along rows (i.e. vertical edges)

$$\psi^D(x,y) = \psi(x)\psi(y)$$

Variations along diagonals

- **2D-DWT**

$$W_\varphi(j_0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\,\varphi_{j_0,m,n}(x,y)$$

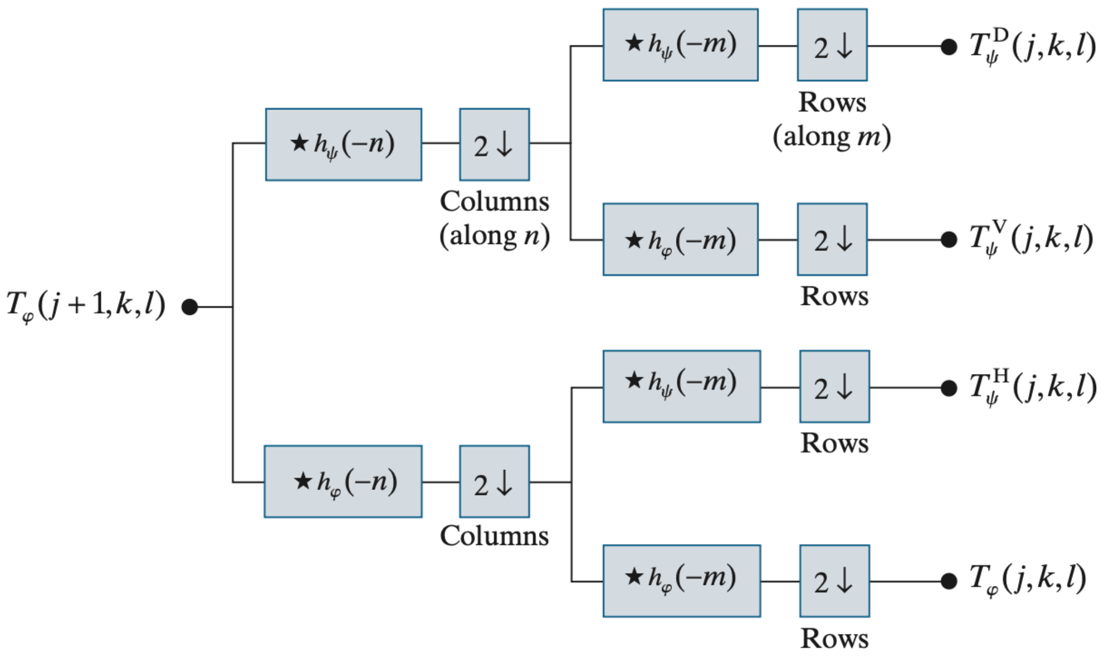
$$W_\psi(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\,\psi^i_{j,m,n}(x,y) \qquad i = \{H,V,D\}$$

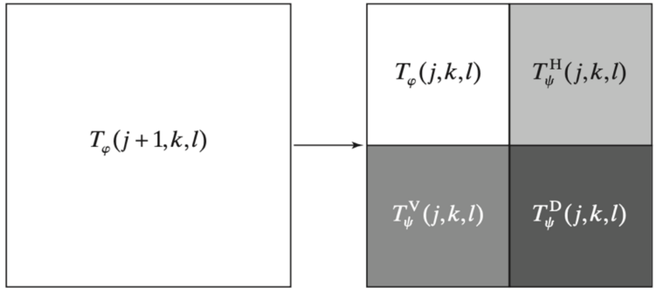
- **2D-IDWT**

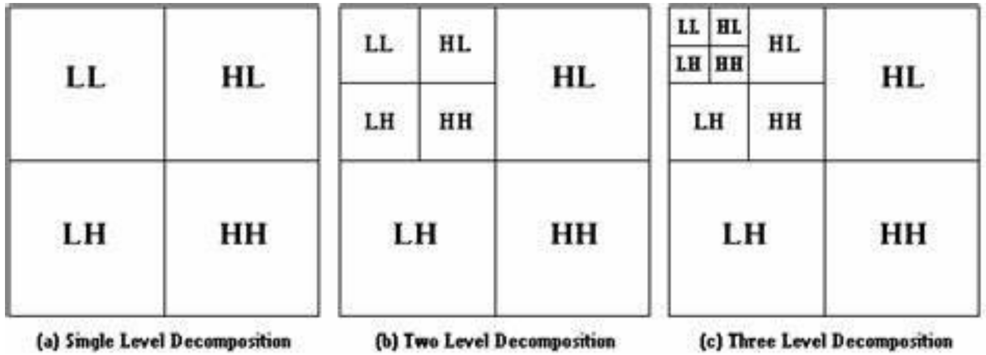
$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0,m,n)\,\varphi_{j_0,m,n}(x,y)$$

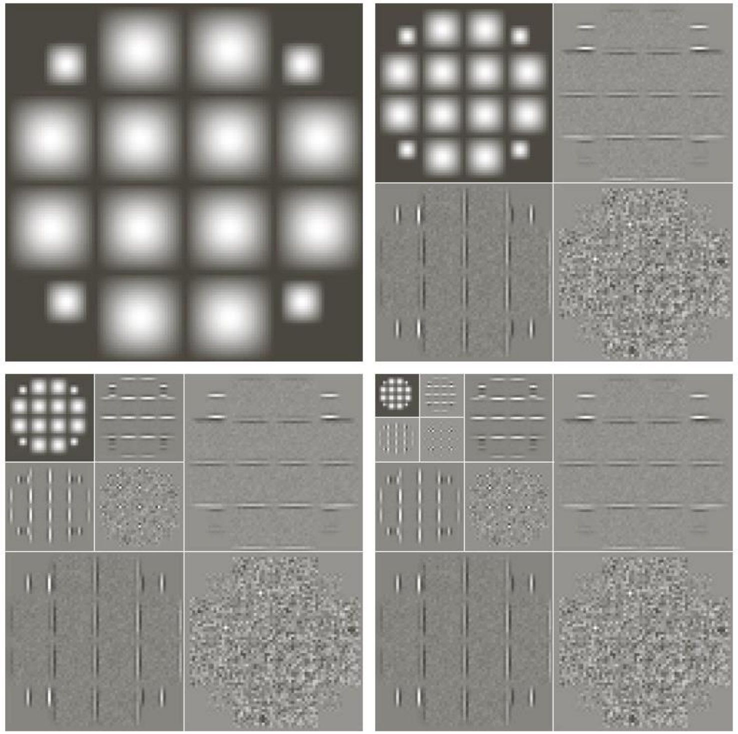
$$+ \frac{1}{\sqrt{MN}} \sum_{i=\{H,V,D\}} \sum_{j=j_0}^{\infty} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} W_\psi(j,m,n)\,\psi^i_{j,m,n}(x,y)$$

2D-DWT

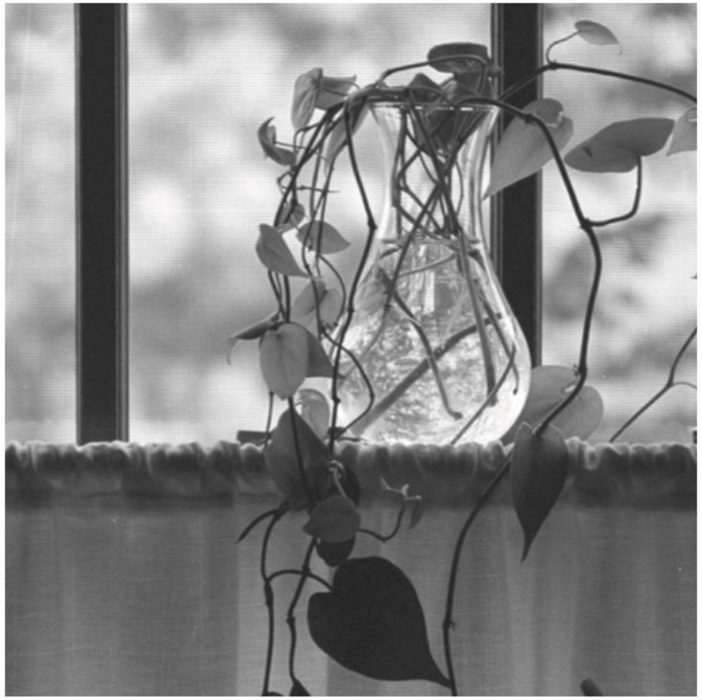

$T_\varphi(j+1,k,l)$
$\star h_\psi(-n)$
$2\downarrow$
Columns
(along n)
$\star h_\psi(-m)$
$2\downarrow$
Rows
(along m)
$T_\psi^D(j,k,l)$
$\star h_\varphi(-m)$
$2\downarrow$
Rows
$T_\psi^V(j,k,l)$
$\star h_\varphi(-n)$
$2\downarrow$
Columns
$\star h_\psi(-m)$
$2\downarrow$
Rows
$T_\psi^H(j,k,l)$
$\star h_\varphi(-m)$
$2\downarrow$
Rows
$T_\varphi(j,k,l)$


$T_\varphi(j+1,k,l)$
$T_\varphi(j,k,l)$
$T_\psi^H(j,k,l)$
$T_\psi^V(j,k,l)$
$T_\psi^D(j,k,l)$

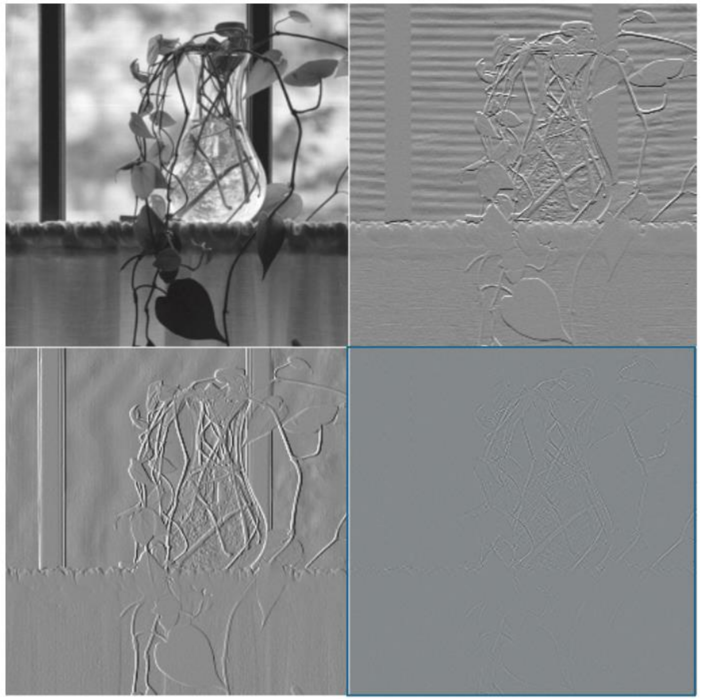

LL
HL
LH
HH
(a) Single Level Decomposition
(b) Two Level Decomposition
(c) Three Level Decomposition



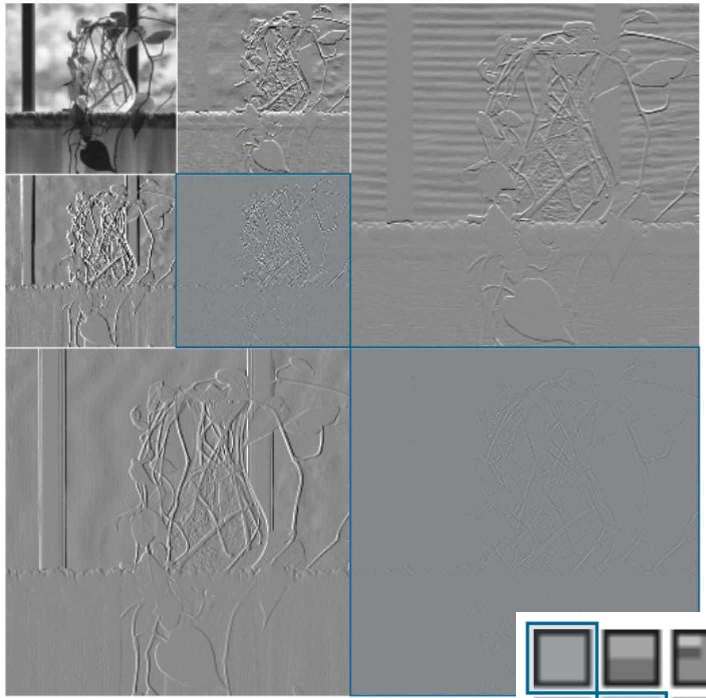
2D-DWT Example



Original image

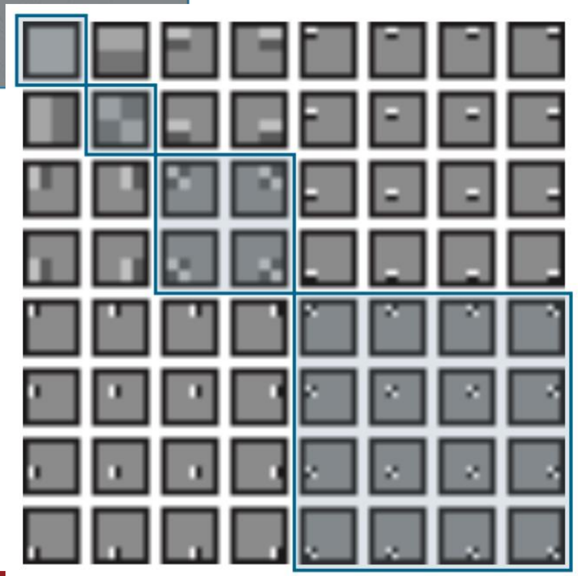


One scale DWT



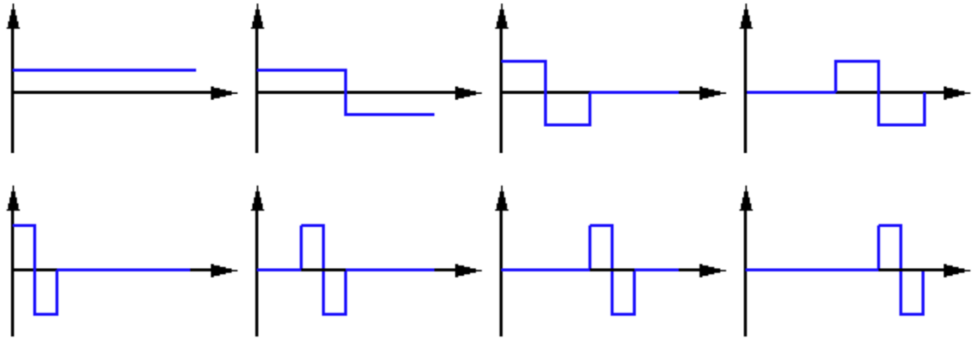
Two-scale DWT

Basis images

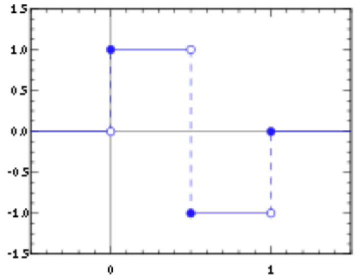


Mother Wavelet (母小波)

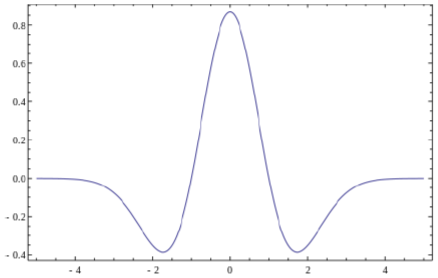
- **Mother Wavelet should satisfy:**
  - $\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1$
  - $\int_{-\infty}^{\infty} |\psi(t)| dt < \infty$
  - $\int_{-\infty}^{\infty} \psi(t) dt = 0$



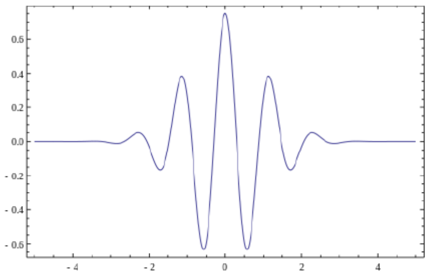
Haar



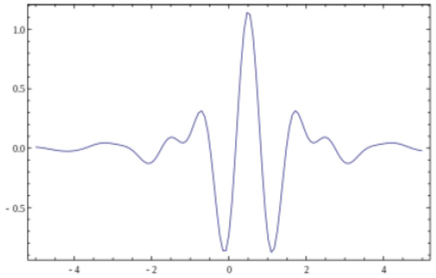
Mexican Hat



Morlet



Meyer



Using Wavelets in image processing

- The idea is similar to the Fourier domain processing:
  1. Compute the 2D-wavelet transform with respect to a selected wavelet basis
  2. Alter the computed transform to take advantage of the DWT's ability to:
     - Decorrelate image pixels
     - Reveal important frequency and temporal characteristics
     - Measure the image's similarity to the transform's basis images
     - Modifications designed for image smoothing, sharpening, noise reduction, edge detection, ... are possible
  3. Compute the inverse wavelet transform.

Example – Edge detection

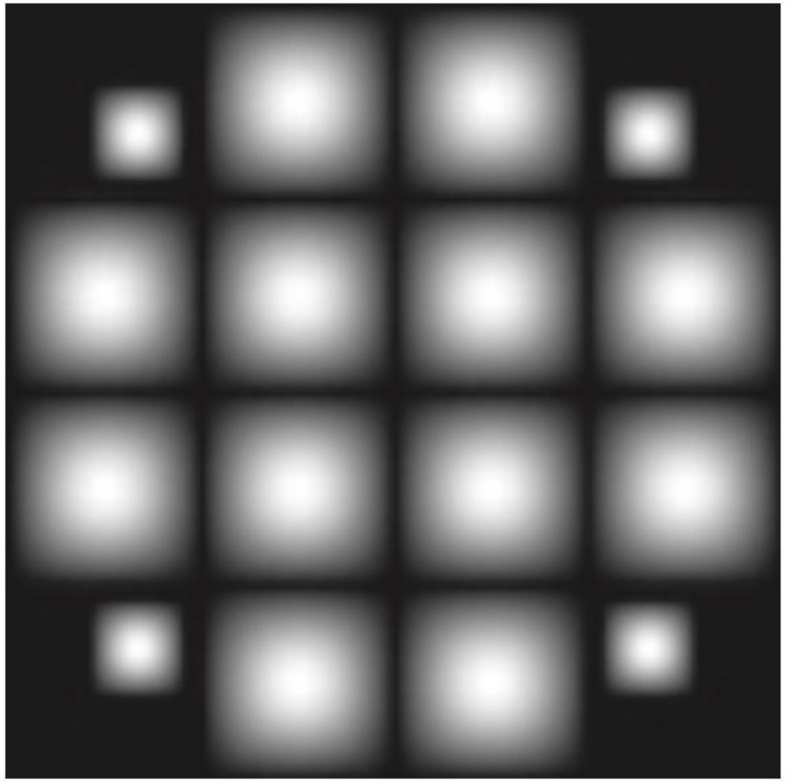
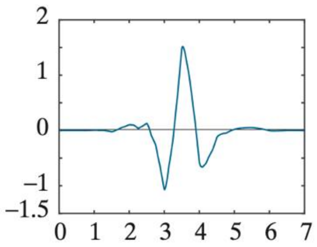
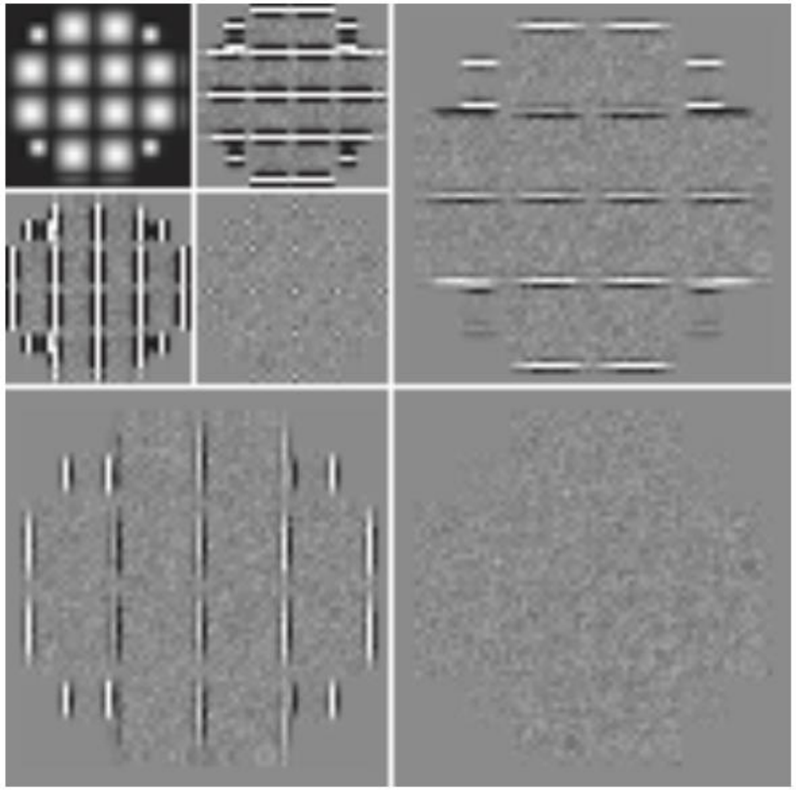


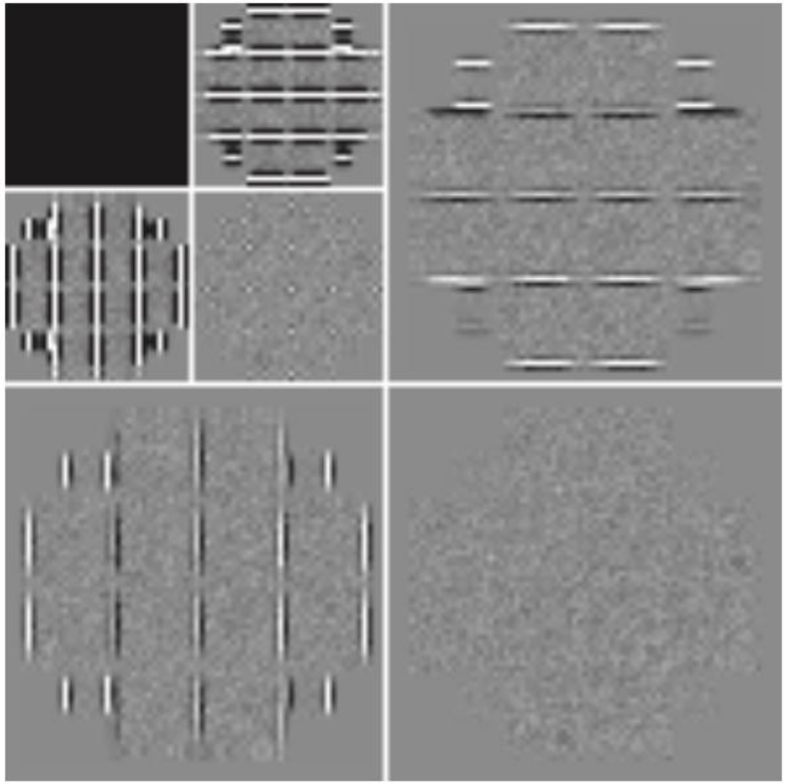
Image with 2D sine-shaped pulses



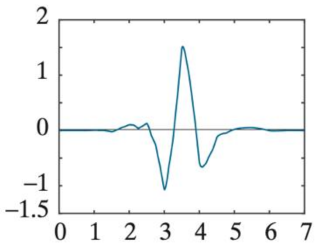
Two-scale DWT
with 4th-order symlets
(*symmetrical wavelets*)



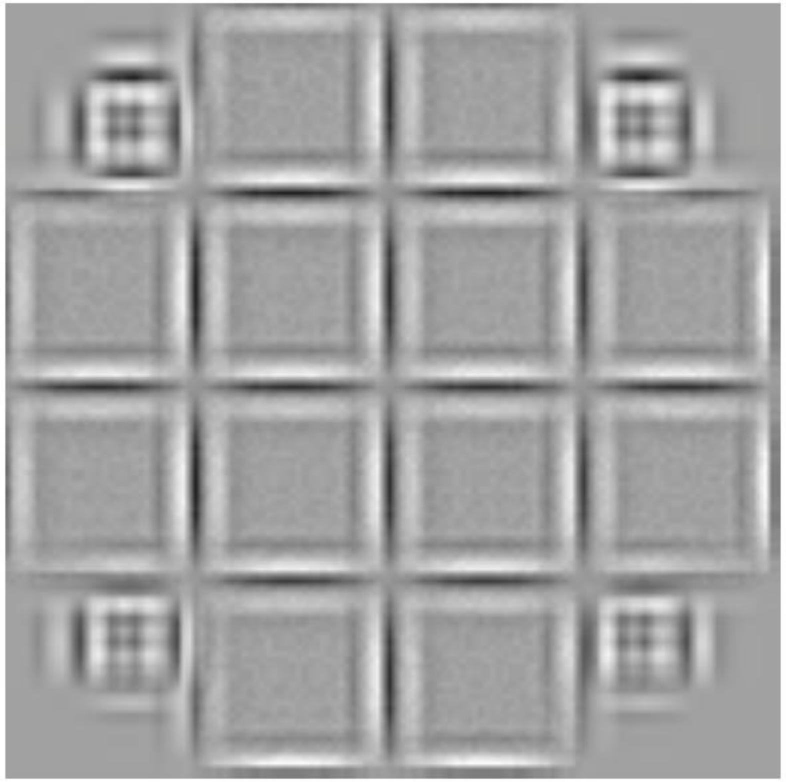
Example – Edge detection



Setting approximation to zero

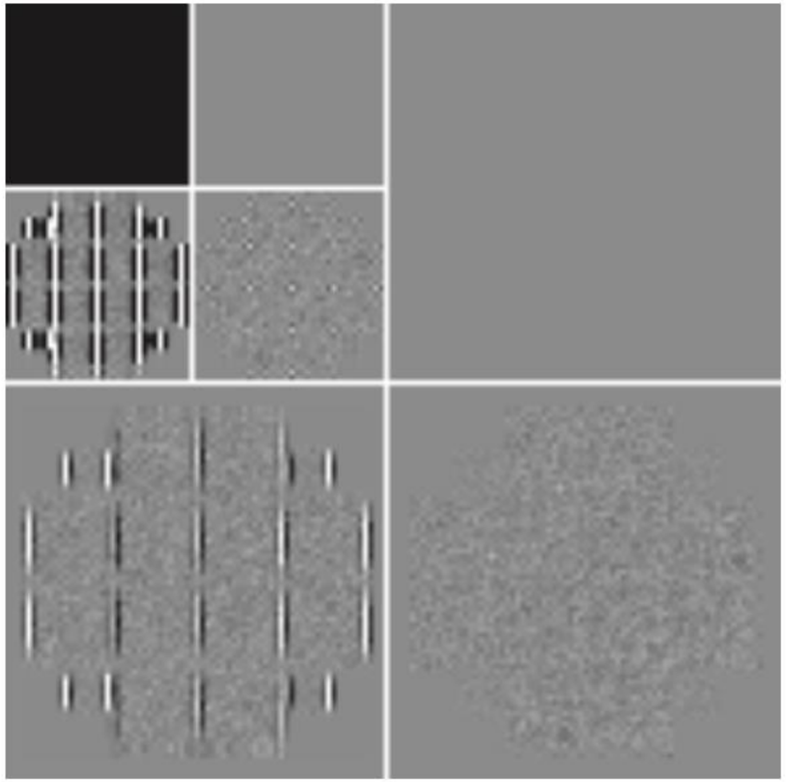


Inverse DWT

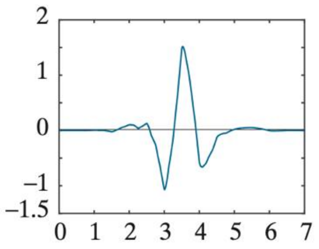


Edge enhancement

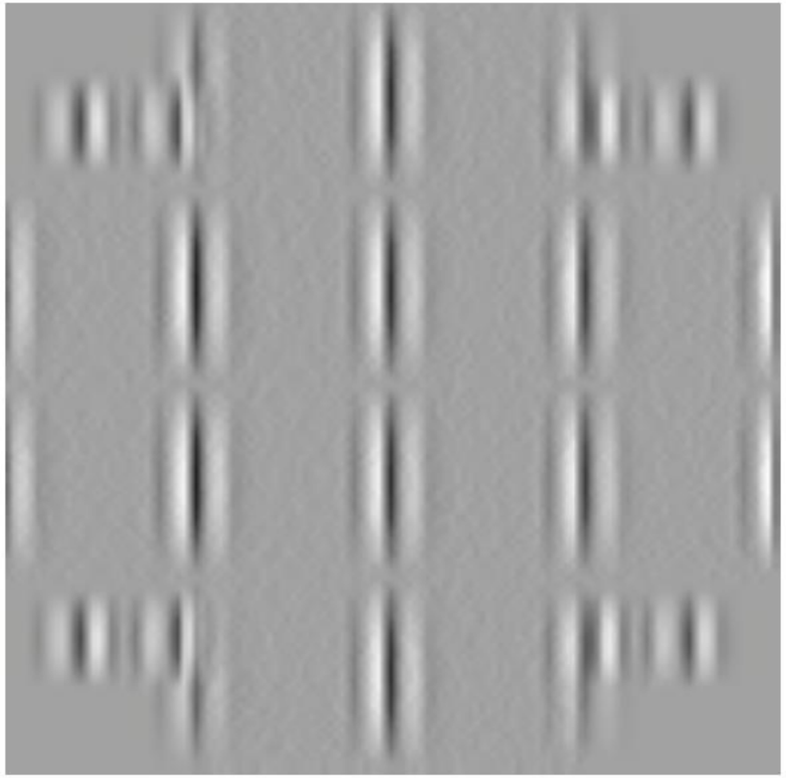
Example – Edge detection



Setting approximation to zero
+ horizontal details to zero



Inverse DWT



Isolates vertical edges

Take home message

- ❑ Based on small waves called Wavelets-1) limited; 2) oscillation.
- ❑ Key idea: Translation & Scaling.
- ❑ Localized both time/space and frequency.
- ❑ Efficient for noise reduction and image compression.
- ❑ JPEG2000, FBI finger printing database (for its digitization and compression).