Lecture 10 Unitary Transform

Yuyao Zhang PhD
zhangyy8@shanghaitech.edu.cn
SIST Building 2 302-F

Outline

– 2D Unitary transform（酉变换）

– Frequency Domain Extension

- Discrete Cosine Transform（余弦变换）
- Hadamard Transform （哈德马变换）
- Discrete Wavelet Transform （小波变换）

1-D Unitary Transform

- Forward Transform:

$$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{f}$$

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

- Inverse Transform:

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

Example for 1-D Unitary Transform

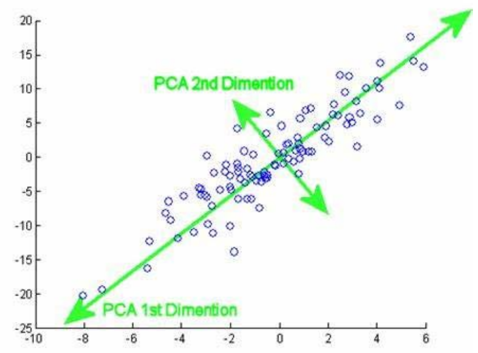
- Image rotation:

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

- Principle Component Analysis (PCA):

$Y = PX$ that satisfy $C = XX^T$ $D = PCP^T$

and $PP^T = I$



Discrete Fourier Transform （DFT）

- Forward Transform:

$$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{f}; \qquad \boldsymbol{t}[\boldsymbol{k}] = \sum_{n=1}^{N} \boldsymbol{A}[\boldsymbol{k},\boldsymbol{n}]\boldsymbol{f}[\boldsymbol{n}]$$

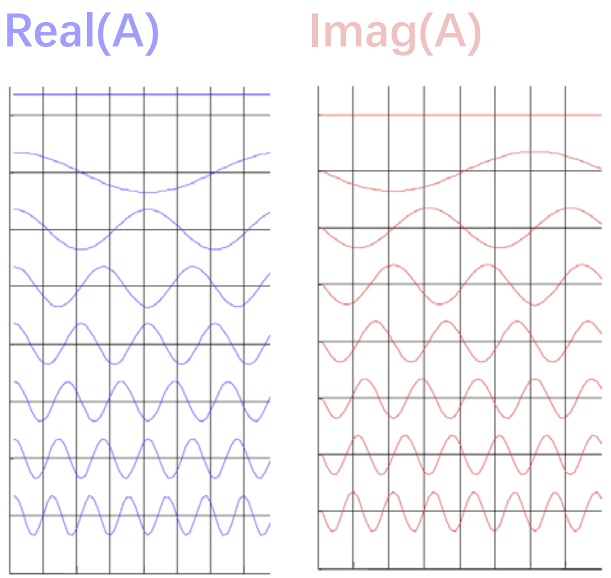
- Inverse Transform：

$$\boldsymbol{f} = \boldsymbol{A}^{H}\boldsymbol{t}; \quad \boldsymbol{f}[\boldsymbol{n}] = \sum_{k=1}^{K} \boldsymbol{A}^{H}[\boldsymbol{k},\boldsymbol{n}]\boldsymbol{t}[\boldsymbol{k}]$$

- 1-D DFT

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

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



2D Unitary Transform

- Forward Transform (2D-DFT for example)

$$F(u,v) = \sum_{x=0}^{M} \sum_{y=0}^{N} f[x,y]\, e^{-j\left(\frac{2\pi u x}{M} + \frac{2\pi v y}{N}\right)}$$

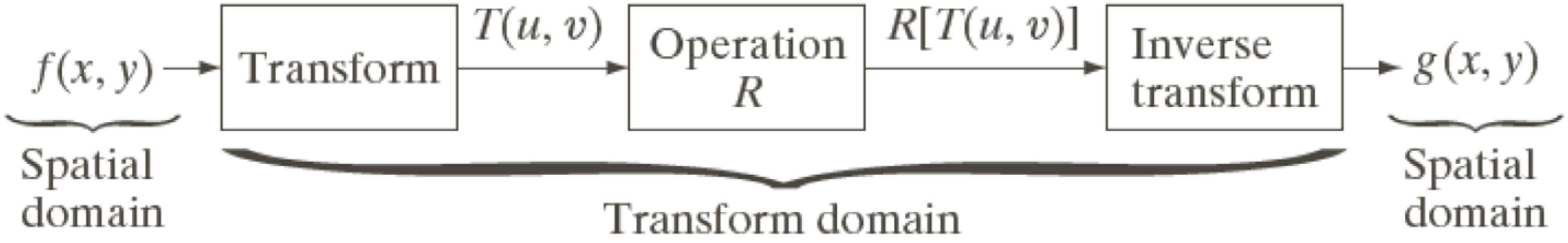
$$= A_M f A_N$$

Inverse Transform

$$f = A_M{}^T F A_N{}^T \qquad AA^T = I$$

Image Transform

- The general approach for operating in the 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. signal energy is preserved

Good and Bad things about DFT

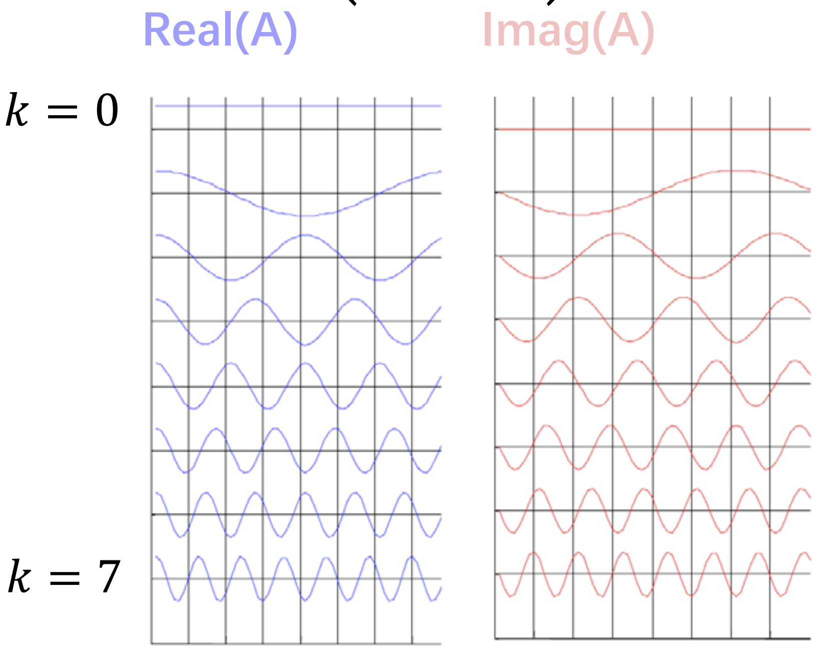
- Positive:
  - Energy is usually packed into low-frequency coefficients.
  - Convolution property.
  - Fast implementation.
- Negative:
  - Transform is complex, even if image is real.
  - The basis function span image height/width.

DFT vs. DCT (Discrete Cosine Transform)

- 1D-DFT

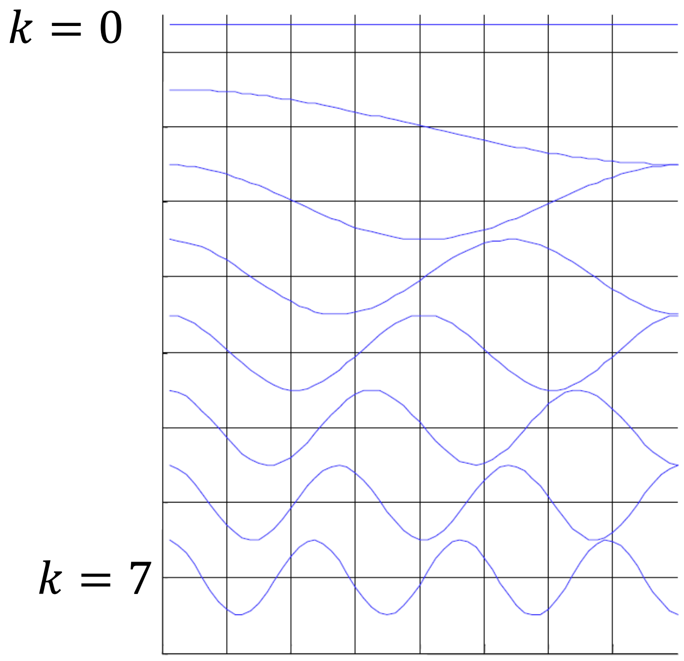
$$A[k,n] = e^{-j\frac{2\pi kn}{N}}$$

$$= cos\left(2\pi\frac{kn}{N}\right) + jsin\left(2\pi\frac{kn}{N}\right)$$



- 1D-DCT

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



2D DCT

Forward Transform:

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

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

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

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

2D IDCT

Inverse Transform:

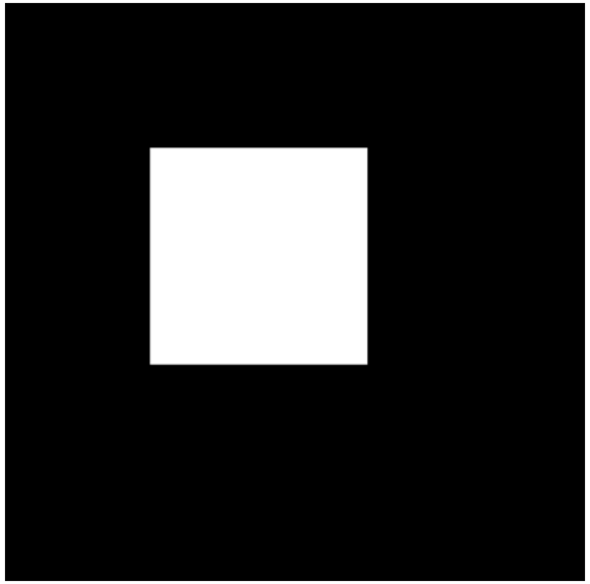
$$f(x,y) = \frac{1}{N} F(0,0)$$

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

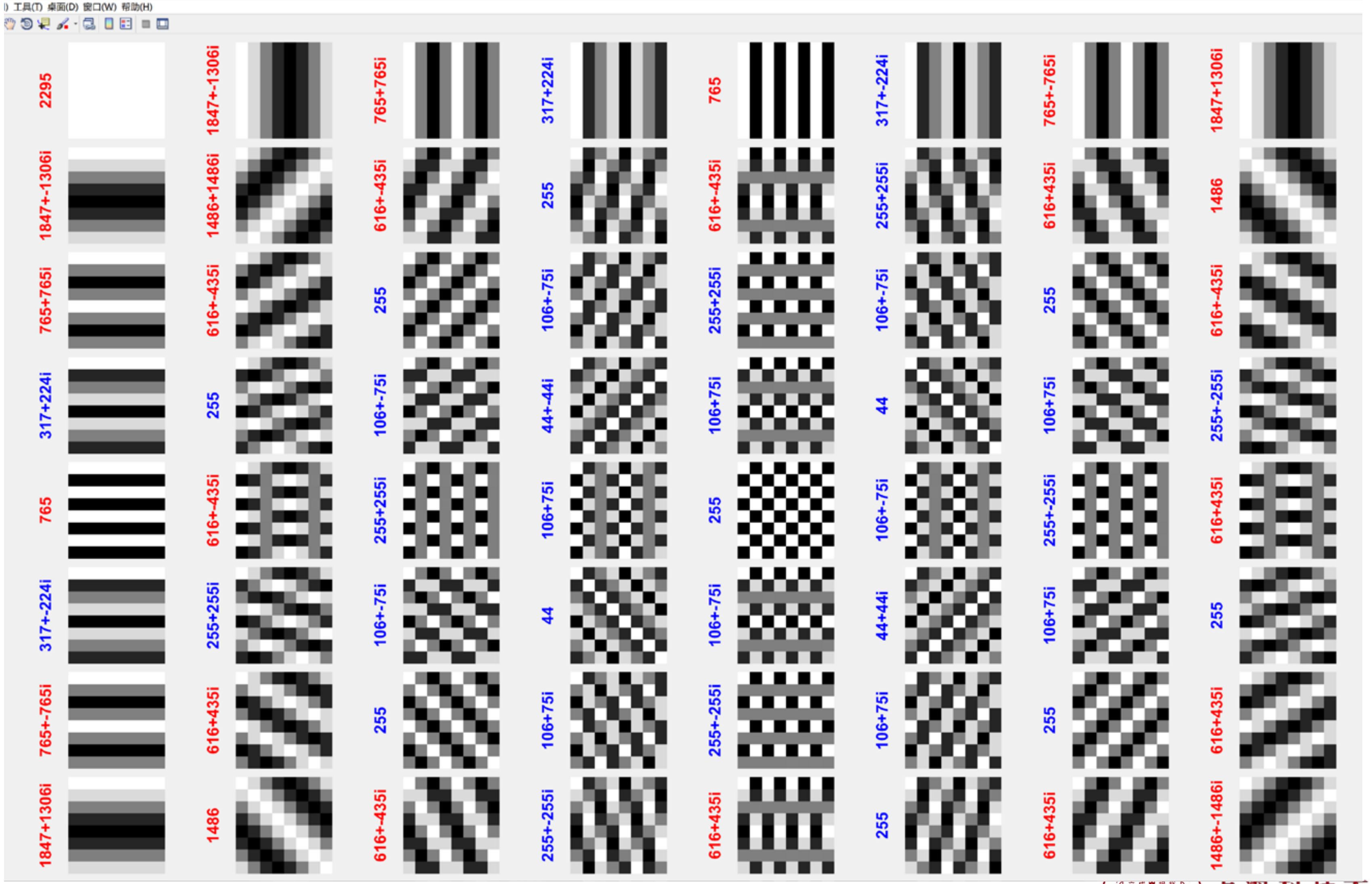
$$+\frac{\sqrt{2}}{N} \sum_{v=1}^{N-1} F(0,v) \cos \frac{(2y+1)v\pi}{2N}$$

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

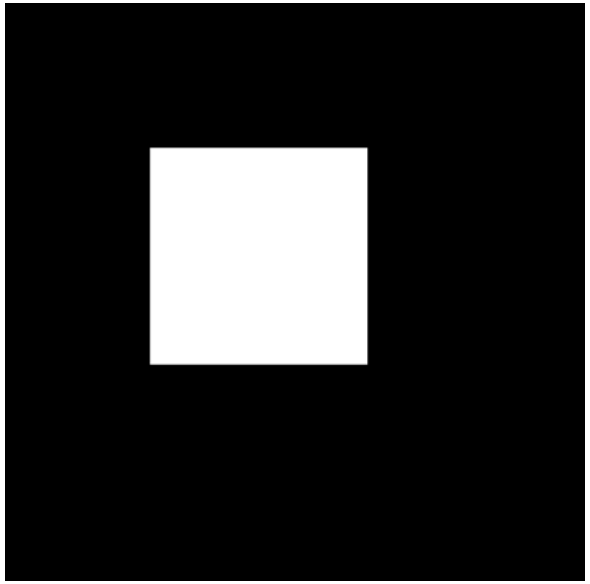
Input image



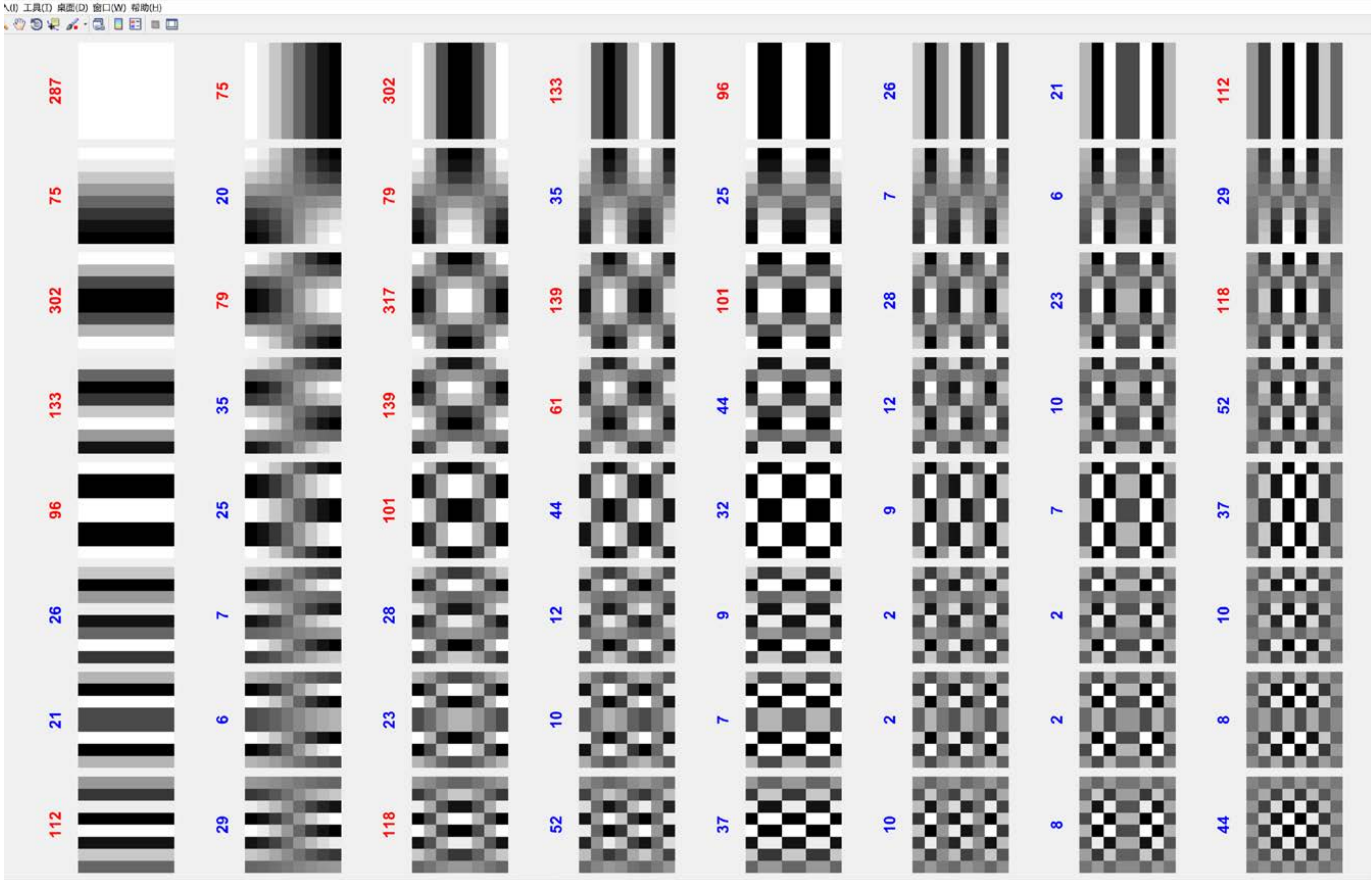
DFT coefficients



Input image



DCT coefficients



Good and Bad things about DCT

➢ Positive

- Transform is real, $C^{-1} = C^T$ (unitary transform).
- Excelent energy compaction for nature images.
- Fast transform.
- JPEG algorithm.

Walsh Transform

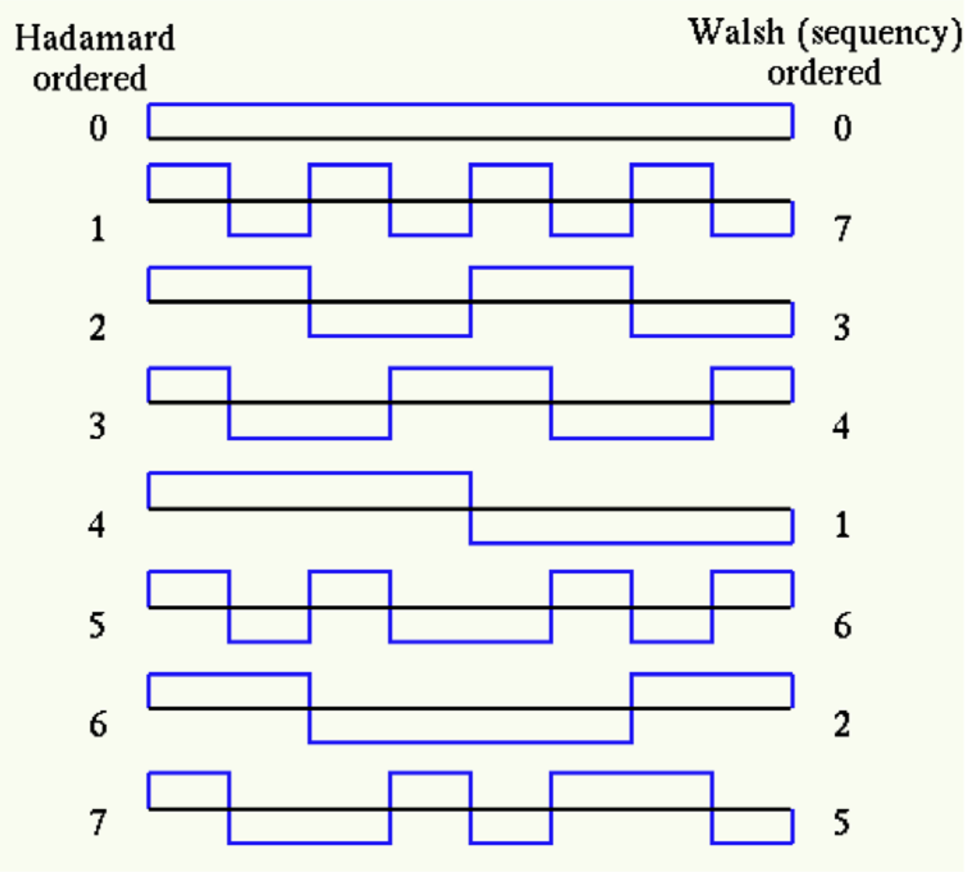
- **Consist of ±1 arranged in a checkerboard pattern**
- **Transform:**

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

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

- **Types of** $\mathrm{Wal}(i,t)$
  - Walsh Ordering (沃尔什定序)
  - Paley Ordering （佩利定序）
  - Hadamard Matrix Ordering（哈达玛矩阵定序）

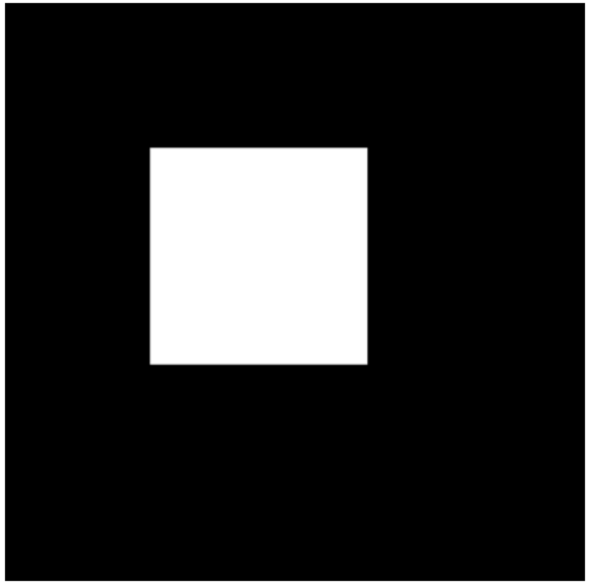
Hadamard Matrix Ordering



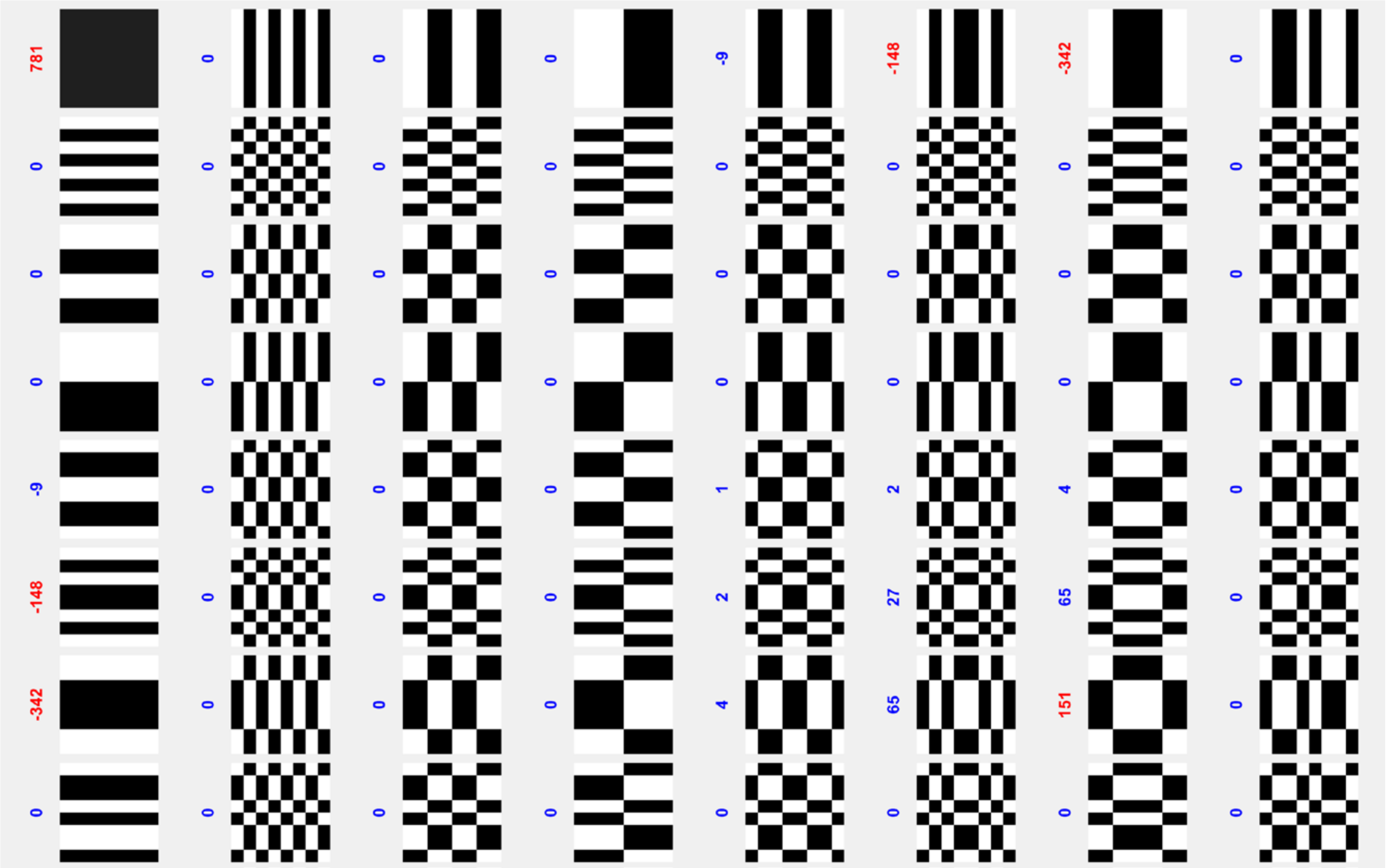
$$V_8 = \begin{pmatrix} W_4 & W_4 \\ W_4 & -W_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

上海科技大学 ShanghaiTech University

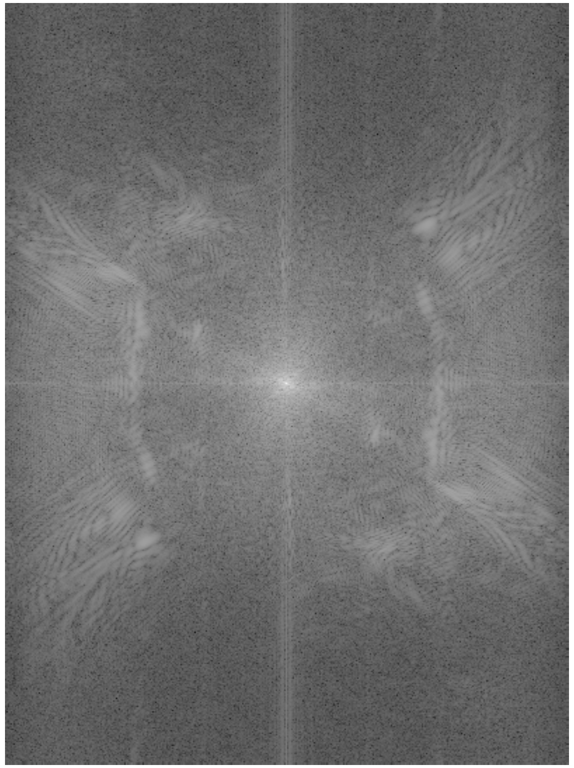
Input image



Hadarmad coefficients

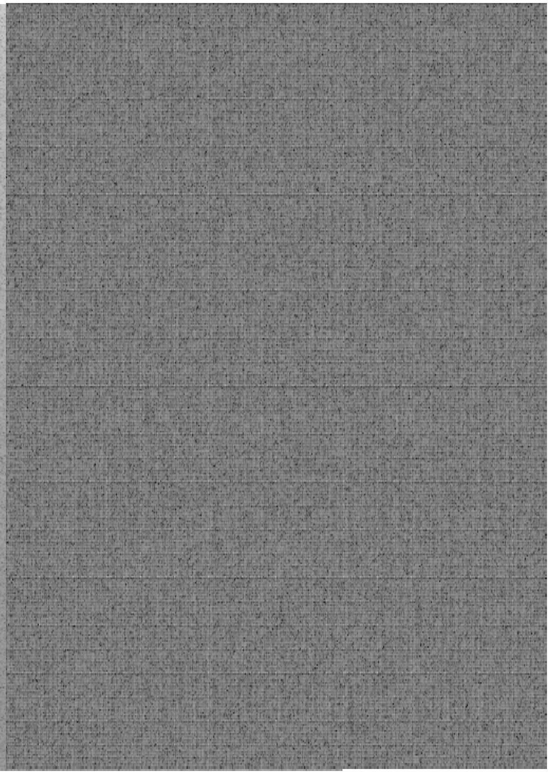


DFT



DCT

Hadamard



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 present a simple image with simple coefficients. But can not keep energy compact for image full of details.