# Digital Image Processing:

# Introduction to Fourier Transform

- Fourier Transform pair :

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du, \ j = \sqrt{-1}$$

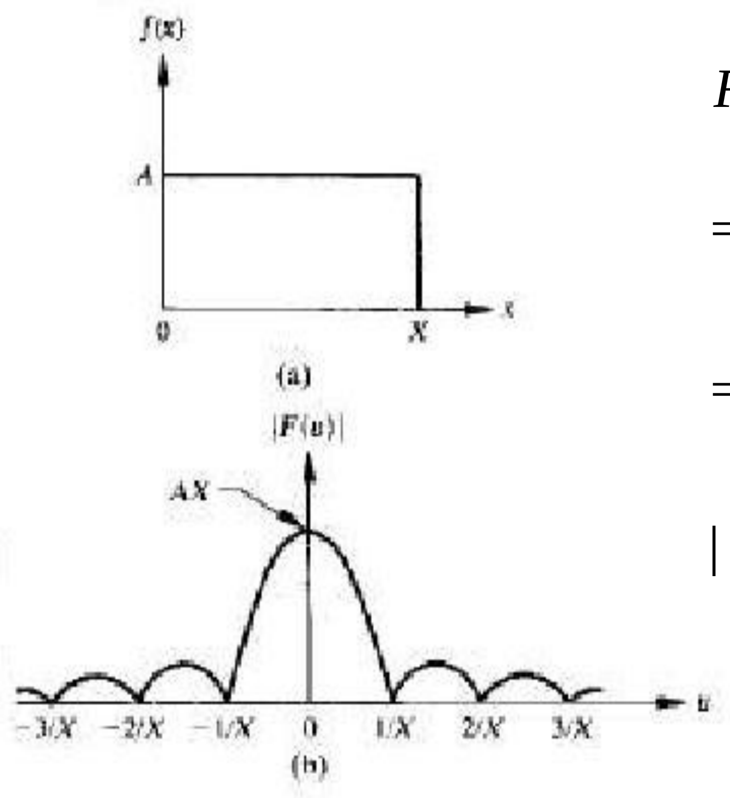
Fourier spectrum:

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2}$$

$$e^{-j2\pi ux} = \cos(2\pi ux) - j\sin(2\pi ux)$$

# Introduction to Fourier Transform(cont.)

- Example: f(x)=A, 0<x<X




A simple function and its Fourier spectrum.

$$F(u)=\int_{-\infty}^{\infty} f(x)e^{-j2\pi ux}dx=\int_{0}^{X} Ae^{-j2\pi ux}dx$$

$$=\frac{-A}{j2\pi u}[e^{-j2\pi ux}]_{0}^{X}=\frac{-A}{j2\pi u}(e^{-j2\pi uX}-1)$$

$$=\frac{A}{j2\pi u}(e^{j\pi uX}-e^{-j\pi uX})e^{-j\pi uX}=\frac{A}{\pi u}\sin(\pi uX)e^{-j\pi uX}$$

$$|F(u)|=AX\left|\frac{\sin(\pi uX)}{\pi uX}\right|$$

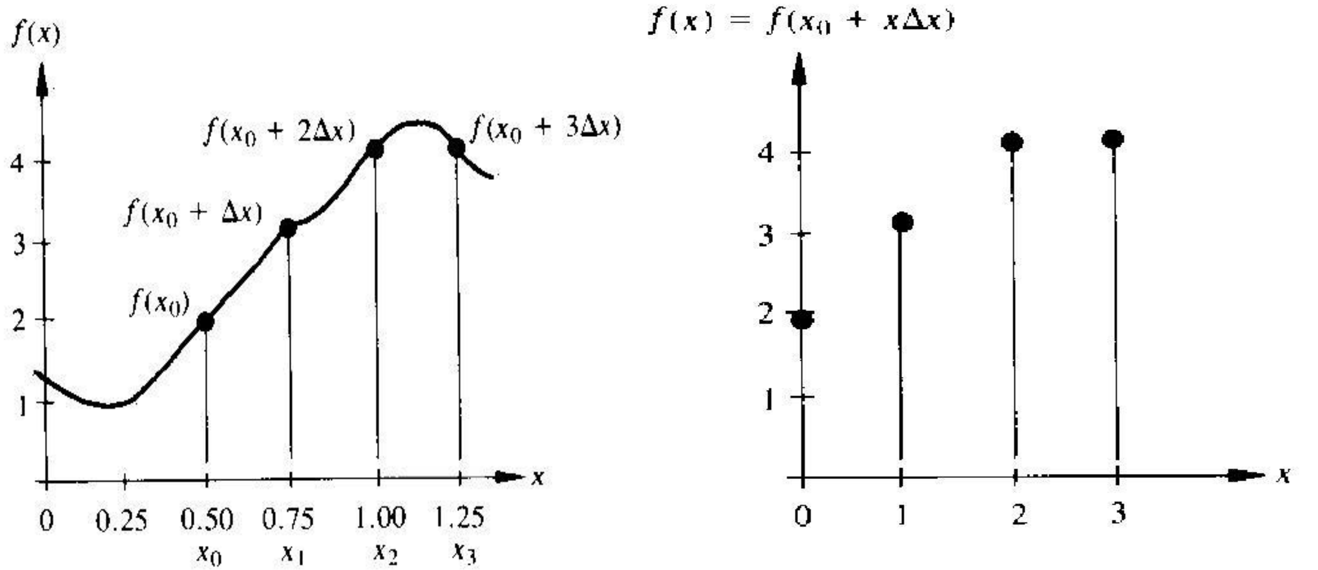
# Discrete Fourier Transform

- Discrete Fourier Transform pair

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N}, u = 0,1,2,\ldots, N-1$$

$$f(x) = \sum_{u=0}^{N-1} F(u)e^{j2\pi ux/N}, x = 0,1,2,\ldots, N-1$$

- Example:



# 2-Dimensional Discrete Fourier Transform

For an image of size MxN pixels

2-D DFT

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

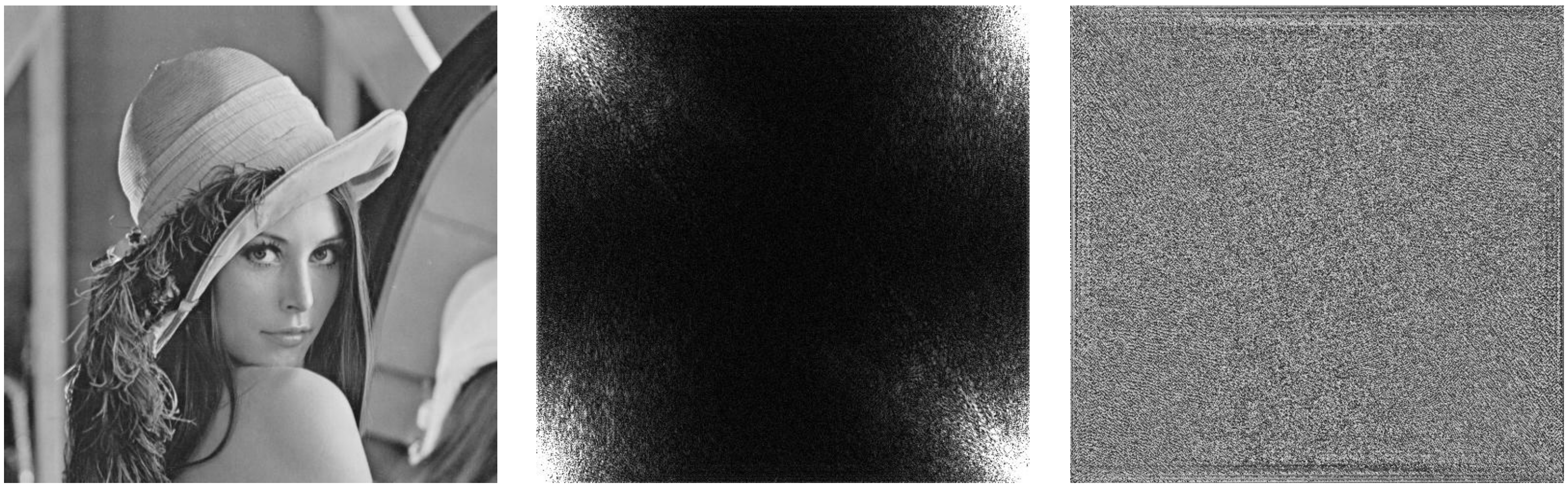
$u$ = frequency in $x$ direction, $u = 0, \ldots, M-1$
$v$ = frequency in $y$ direction, $v = 0, \ldots, N-1$

2-D IDFT

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

$x = 0, \ldots, M-1$
$y = 0, \ldots, N-1$

- 2-D DFT (cont.)
  - example



(a) Original Image (b) Magnitude (c) Phase

$$\boxed{2-\text{dim DFT of a } 512\times 512 \text{ Lena image}}$$

# 2-D FFT

$$D(u, v) = c \log[1 + |F(u, v)|]$$

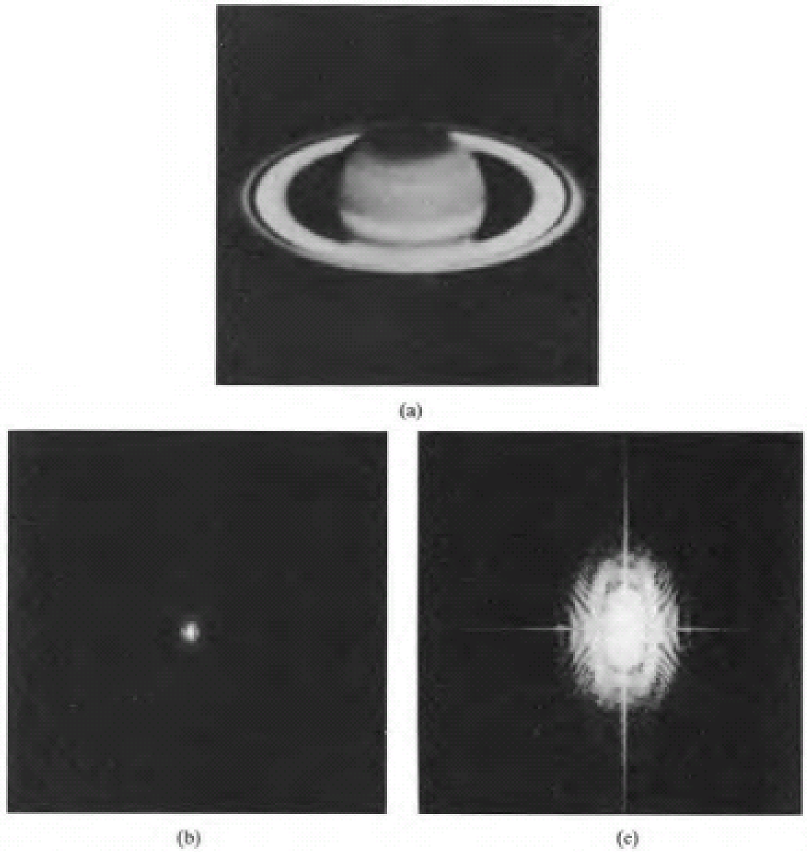



***Figure 3.6*** *(a) A picture of the planet Saturn; (b) display of |F(u, v)|; (c) display of log[1 + |F(u, v)|] scaled to 8 bits (i.e., 0 to 255 gray levels).*

# 2-D DFT (cont.)

- Properties of 2D DFT
  - **Separability**

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\exp[-j2\pi ux/N]\sum_{y=0}^{N-1}f(x,y)\exp[-j2\pi vy/N],u,v=0,\cdots,N-1$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\exp[j2\pi ux/N]\sum_{v=0}^{N-1}F(u,v)\exp[j2\pi vy/N],m,=xy=0,\cdots,N-1$$

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}F(x,v)\exp[-j2\pi ux/N]$$

$$F(x,v)=N\left[\frac{1}{N}\sum_{y=0}^{N-1}f(x,y)\exp[-j2\pi vy/N]\right]$$

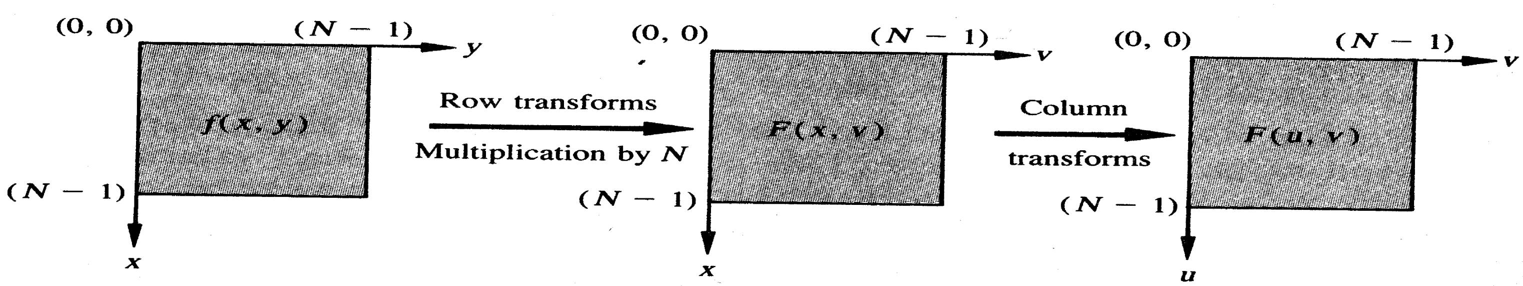
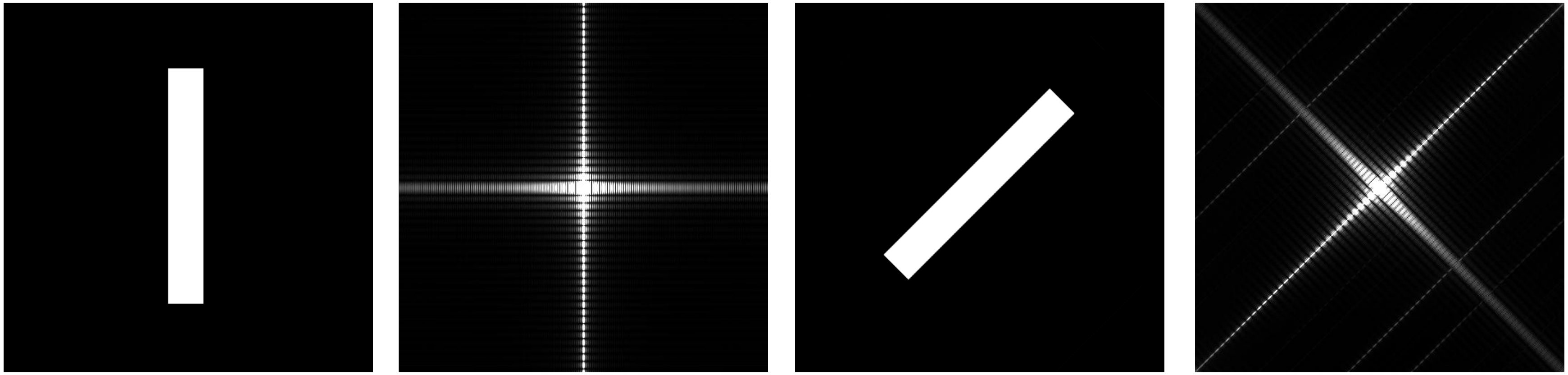



***Figure 3.7*** *Computation of the 2-D Fourier transform as a series of 1-D transforms.*

- 2-D DFT (cont.)
  - Properties of 2D DFT (cont.)
    - **Rotation**

$$m = r\cos\theta,\ n = r\sin\theta, \quad k = \omega\cos\phi,\ l = \omega\sin\phi$$

$$f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\phi+\theta_0)$$



(a) a sample image (b) its spectrum (c) rotated image (d) resulting spectrum

# 2-D DFT (cont.)

Convolution

$$f(x)*g(x)=\int_{-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha$$

$$f(x)*g(x)=\begin{cases} x/2 & 0\le x\le 1 \\ 1-x/2 & 1\le x\le 2 \\ 0 & otherwise \end{cases}$$

- Convolution Theorem

$$f(x)*g(x)\Leftrightarrow F(u)G(u)$$
$$f(x)g(x)\Leftrightarrow F(u)*G(u)$$

# Other separable image transforms:

A 2-dimensional transform with a separable kernel can be computed using a series of 1-dimensional transforms as follows:

$$T(x,v) = \sum_{y=0}^{N-1} f(x,y) g_2(y,v)$$

and then,

$$T(u,v) = \sum_{x=0}^{N-1} T(x,v) g_1(x,u)$$

This procedure can be done in reverse order which produces the same result.

Now realize that if the kernel is separable, we can write it as follows:

$$T(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} g_1(x,u)f(x,y)g_1(y,v)$$

also notice that $T$, $g_1$, and $f$ are two-dimensional $N*N$ matrices, therefore the above equation can be reduced to

$$\underline{T}=\underline{A}\,\underline{F}\,\underline{A}$$

[$A$ is a $N*N$ matrix with elements $a_{ij}=g_1(i,j)$]

# Other Separable Image Transforms:

- Regular Form :

$$T(u) = \sum_{x=0}^{N-1} f(x) g(x,u)$$

$$f(x) = \sum_{u=0}^{N-1} T(u) h(x,u)$$

g(x, u) is called forward transformation kernel, and h(x, u) is called inverse transformation kernel.

- Walsh Transform

$$g(x,u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}, N = 2^n$$

$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

$$n = 3, z = 6 = (110)_2$$

$$b_0(z) = 0$$

$$b_1(z) = 1$$

$$b_2(z) = 1$$

# Other Separable Image Transforms(cont.)

- Walsh Transform(cont.)
  - Example : N= 4= $2^2$

$$W(0)=\frac{1}{4}\sum_{x=0}^{3} f(x)\prod_{i=0}^{2-1}(-1)^{b_i(x)b_{2-1-i}(0)}$$

$$=\frac{1}{4}[f(0)+f(1)+f(2)+f(3)]$$

$$W(2)=\frac{1}{4}\sum_{x=0}^{3} f(x)\prod_{i=0}^{2-1}(-1)^{b_i(x)b_{2-1-i}(2)}$$

$$=\frac{1}{4}[f(0)-f(1)+f(2)-f(3)]$$

$$W(1)=\frac{1}{4}\sum_{x=0}^{3} f(x)\prod_{i=0}^{2-1}(-1)^{b_i(x)b_{2-1-i}(1)}$$

$$=\frac{1}{4}[f(0)+f(1)-f(2)-f(3)]$$

$$W(3)=\frac{1}{4}\sum_{x=0}^{3} f(x)\prod_{i=0}^{2-1}(-1)^{b_i(x)b_{2-1-i}(3)}$$

$$=\frac{1}{4}[f(0)-f(1)-f(2)+f(3)]$$

## *Values of the 1-D Walsh Transformation Kernel for N=8*

$$\frac{1}{8}g(x,u)$$

| u \ x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | + | + | + | + | + | + | + | + |
| 1 | + | + | + | + | - | - | - | - |
| 2 | + | + | - | - | + | + | - | - |
| 3 | + | + | - | - | - | - | + | + |
| 4 | + | - | + | - | + | - | + | - |
| 5 | + | - | + | - | - | + | - | + |
| 6 | + | - | - | + | + | - | - | + |
| 7 | + | - | - | + | - | + | + | - |



Here we calculate the matrix of Walsh coefficients

# Two-Dimensional Walsh Transform

Note that the only difference between the forward and inverse transform is *1/N* constant. Now, we go to 2-dimensional transformation we will have:

$$g(x,y,u,v) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

and

$$h(x,y,u,v) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

Two-dimensional Walsh

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

and

Inverse Two-dimensional Walsh

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} W(u,v)\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

# Other Separable Image Transforms(cont.)

white:+1

black:-1

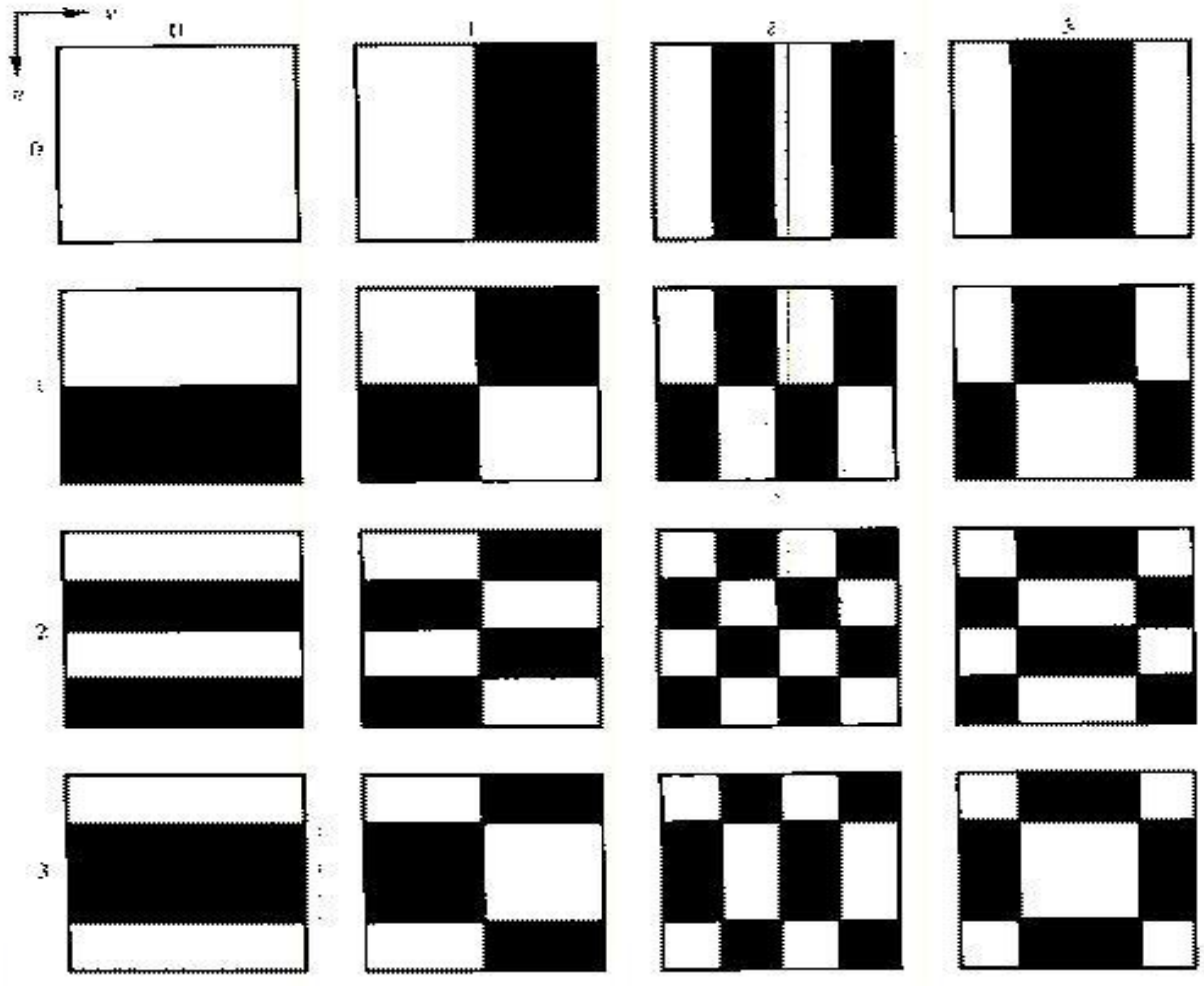


*Figure 3.25* Walsh basis functions for $N = 4$. Each block consists of $4 \times 4$ elements, corresponding to $x$ and $y$ varying from 0 to 3. The origin of each block is at its top left. White and black denote $+1$ and $-1$, respectively.

# Hadamard Transform

Transformation kernel is

$$g(x,u) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

$$Hence;$$

$$H(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

$$where \quad N = 2^n$$

The Hadamard kernel is an orthogonal matrix $\Rightarrow$

$$h(x,u) = (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)} \qquad and$$

$$f(x) = \sum_{u=0}^{N-1} H(u)(-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

Two-dimensional Hadamard transforms are

$$H(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1} [b_i(x) b_i(u) + b_i(y) b_i(v)]}$$

*and*,

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} H(u,v)(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_i(u)+b_i(y)b_i(v)]}$$

Again notice that coefficient $1/N^2$ is divided between the pair so, one can use the transform for inverse transform and also just like other orthogonal functions kernel is separable.

$$g(x,y,u,v) = g_1(x,u)g_1(y,v) = h_1(x,u)h_1(y,v)$$

$$= \left[\frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}\right]\left[\frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{n-1} b_i(y)b_i(v)}\right]$$

## *Values of the 1-D Hadamard Transformation Kernel for N=8*

$$\frac{1}{\sqrt{8}} g_H(x,u)$$

| u \ x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | + | + | + | + | + | + | + | + |
| 1 | + | - | + | - | + | - | + | - |
| 2 | + | + | - | - | + | + | - | - |
| 3 | + | - | - | + | + | - | - | + |
| 4 | + | + | + | + | - | - | - | - |
| 5 | + | - | + | - | - | + | - | + |
| 6 | + | + | - | - | - | - | + | + |
| 7 | + | - | - | + | - | + | + | - |

The Hadamard matrix of lowest order (N = 2) is

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

Then letting $\mathbf{H}_N$ represent the matrix of order N, the recursive relationship is given by the expression:

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

Using matrix format $\boldsymbol{H} = \boldsymbol{AFA}$ suggests that $A = \frac{1}{\sqrt{N}} H_N$ .

***Fact!*** The number of sign changes along a column of the Hadamard matrix is often called the ***sequency*** of that column. For instance, $\mathrm{H_8}$ has a sequency of

0, 7, 3, 4, 1, 6, 2, 5

- It is important to order the sequency such that it increase as $u$ increases.
- For one-dimensional Hadamard this ordering can be accomplished through the following relation:

$$g(x,u) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x) p_i(u)}$$

Where,

$$p_0(u) = b_{n-1}(u)$$

$$p_1(u) = b_{n-1}(u) + b_{n-2}(u)$$

$$p_2(u) = b_{n-2}(u) + b_{n-3}(u)$$

$$\vdots$$

$$p_{n-1}(u) = b_1(u) + b_0(u)$$

$$\textit{and recursive formula is} \Rightarrow \quad p_i(u) = b_{n-i}(u) + b_{n-i-1}(u)$$

$$\begin{cases} H(u) = \dfrac{1}{N} \displaystyle\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(u)\, p_i(u)} \\ f(x) = \dfrac{1}{N} \displaystyle\sum_{x=0}^{N-1} H(u)(-1)^{\sum_{i=0}^{n-1} b_i(u)\, p_i(u)} \end{cases}$$

In the same manner, the 2-D transform pair can be written as :

$$
\begin{cases}
H(u,v) = \dfrac{1}{N} \displaystyle\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1} [b_i(x)p_i(u) + b_i(y)p_i(v)]} \\
f(x,y) = \dfrac{1}{N} \displaystyle\sum_{u=0}^{N-1} \sum_{v=0}^{N-1} H(u,v)(-1)^{\sum_{i=0}^{n-1} [b_i(x)p_i(u) + b_i(y)p_i(v)]}
\end{cases}
$$

How does the ordering look for one dimensional kernel?

# Standard Trivial Functions for Hadamard

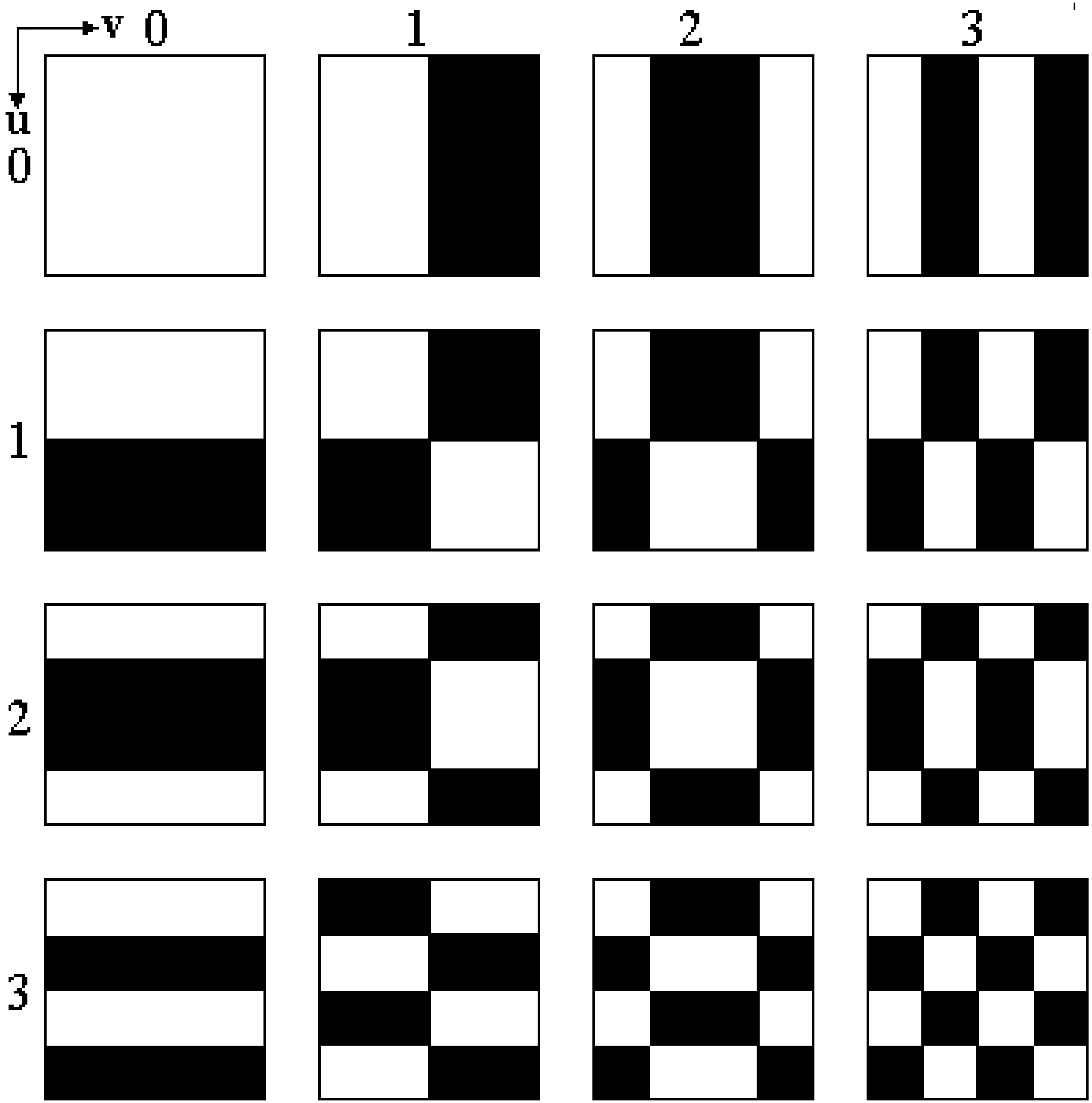
***Values of the 1-D Ordered Hadamard Kernel for N=8***

| u \ x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | + | + | + | + | + | + | + | + |
| 1 | + | + | + | + | - | - | - | - |
| 2 | + | + | - | - | - | - | + | + |
| 3 | + | + | - | - | + | + | - | + |
| 4 | + | - | - | + | + | - | - | + |
| 5 | + | - | - | + | - | + | + | - |
| 6 | + | - | + | - | - | + | - | + |
| 7 | + | - | + | - | + | - | + | - |

One change (row 1)

two changes (row 2)

**_Ordered Hadamard basis functions for N = 4. Each block consist of 4 x 4 elements, corresponding to x and y varying from 0 to 3._**



# Haar Transform:

- **Haar transform**
  - Haar function (1910, Haar) : periodic, orthonormal, complete

$$h_0(z) \stackrel{\Delta}{=} h_{00}(z) = \frac{1}{\sqrt{N}}, \qquad z \in [0,1]$$

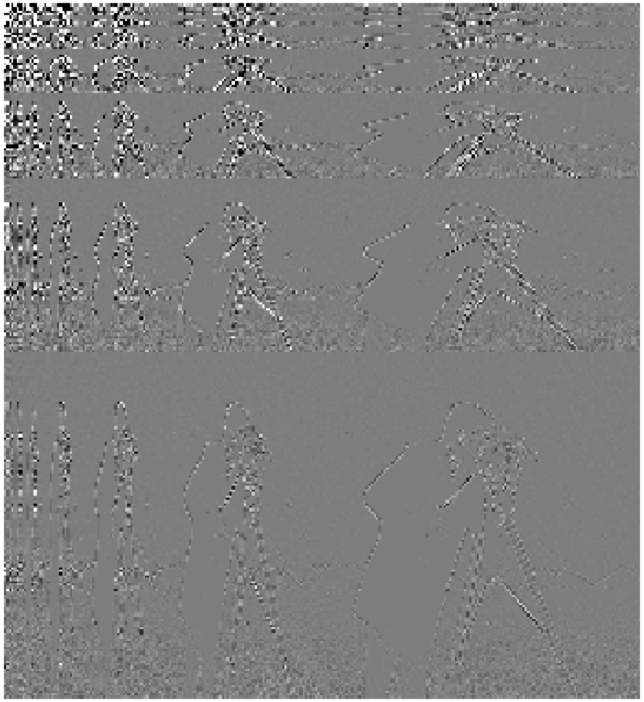
$$h_k(z) \stackrel{\Delta}{=} h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \le z < \frac{q-\frac{1}{2}}{2^p} \\ -2^{p/2} & \frac{q-\frac{1}{2}}{2^p} \le z < \frac{q}{2^p} \\ 0 & \text{elsewhere for } z \in [0,1] \end{cases}$$

$$H_4 = \frac{1}{\sqrt{4}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

$$H_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

# Haar transform example:

Original *Cameraman* *256x256*



*256x256 Haar transform* of *Cameraman*

# Slant transform:

1971, Enomoto and Shibata

$$S_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$S_2 = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ a_2 & b_2 & -a_2 & b_2 \\ \hline 0 & 1 & 0 & -1 \\ -b_2 & a_2 & b_2 & a_2 \end{array}\right]\left[\begin{array}{c|c} S_1 & 0_2 \\ \hline 0_2 & S_1 \end{array}\right]$$

# Slant transform contd..:

$$S_N = \frac{1}{2^{\frac{1}{2}}} \left[ \begin{array}{cc|c|cc|c} 1 & 0 & & 1 & 0 & \\ a_N & b_N & 0 & -a_N & b_N & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & I_{(N/2)-2} \\ \hline 0 & 1 & & 0 & -1 & \\ -b_N & a_N & 0 & b_N & a_N & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & -I_{(N/2)-2} \end{array} \right] \left[ \begin{array}{c|c} S_{N/2} & 0 \\ \hline 0 & S_{N/2} \end{array} \right]$$

$$a_N = \left( \frac{3N^2}{4N^2 - 1} \right)^{\frac{1}{2}}$$

$$b_N = \left( \frac{N^2 - 1}{4N^2 - 1} \right)^{\frac{1}{2}}$$

# Slant transform contd..:

$$S_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix}$$

$\underline{\text{Properties}}$

1. The Slant transform is real and orthogonal

$$S = S^*, \qquad S^{-1} = S^t$$

2. fast algorithm : $O(N \log_2 N)$ for $N \times 1$ vector

3. very good energy compaction

# DCT:

- Discrete Cosine Transform

$$C(u) = a(u)\sum_{x=0}^{N-1} f(x)\cos[(2x+1)u\pi / 2N]$$

$$u = 0,1,2,\ldots, N-1$$

$$f(x) = \sum_{u=0}^{N-1} a(u)C(u)\cos[(2x+1)u\pi / 2N]$$

$$x = 0,1,2,\ldots, N-1$$

$$a(u) = \begin{cases} \frac{1}{\sqrt{N}} & u = 0 \\ \frac{2}{\sqrt{N}} & u = 1,2,\ldots, N-1 \end{cases}$$