DIGITAL IMAGE PROCESSING-1

Unit 2: Lecture 23-24

Dr. Shikha Tripathi
Department of Electronics &
Communication Engineering

DIGITAL IMAGE PROCESSING-1

Unit 2: Image Transforms

Dr. Shikha Tripathi
Department of Electronics & Communication Engineering

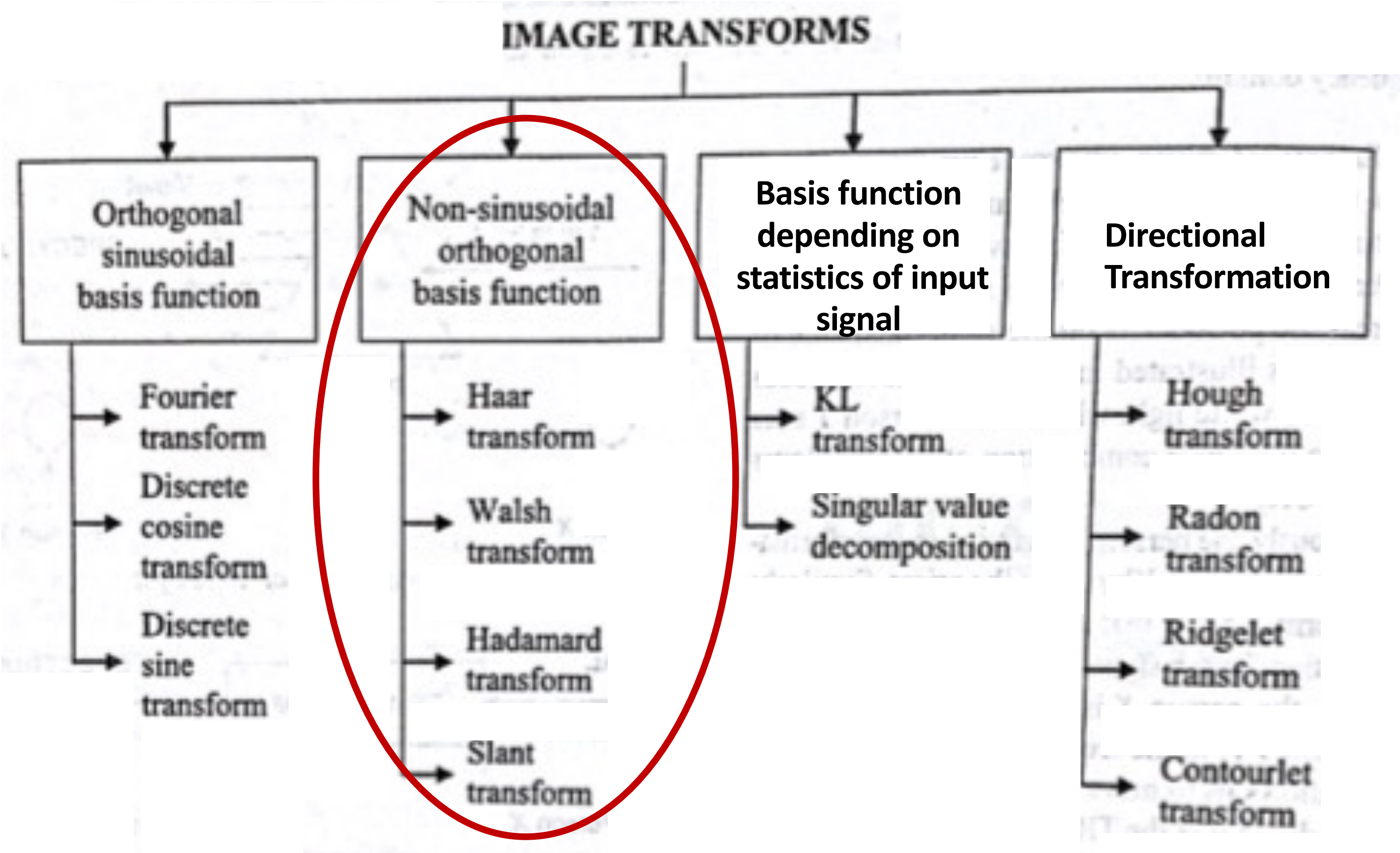
Last Session

- Discrete Cosine Transform (DCT) Cont..
- Discrete Sine Transform (DST)

Today's Session

- Walsh transforms
- Walsh - Hadamard transforms
- Slant Transform

Classification of Image Transforms



- The transforms discussed so far have been based on Sine and Cosine functions
- Transforms based on pulse like waveforms take only ± 1 $(simple\ \&\ fast\ to\ compute)$
  - More appropriate for representation of waveforms which contain discontinuities (images)
- Discrete Walsh transform is based on set of harmonically related rectangular waveforms known as **Walsh functions**

1D Discrete Walsh Transform (DWT)

- Forward transformation kernel for 1D DWT is given by:
- $g(x,u) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$

where

N=No. of samples

n=No. of bits needed to represent x/u

$b_k(z)$= k^{th} bit in digital/binary representation of z

For ex. $b_2(8)$= 2^{nd} bit of 8 i.e., 1000

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

1D Discrete Walsh Transform (DWT)

Inverse transformation Kernel is given by:

$h(x,u) = \prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$

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

- Also the kernel for forward and inverse transform is same except the scaling factor 1/N (positive aspect of Walsh transform)

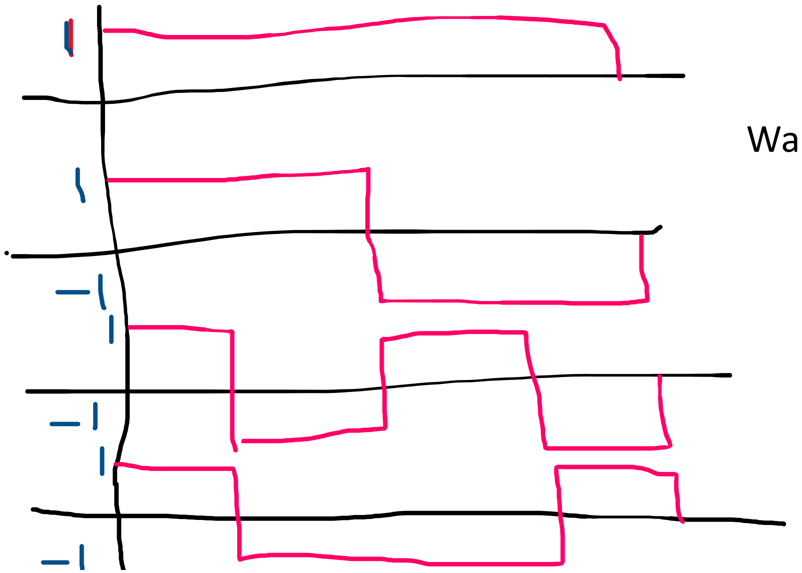
1D Discrete Walsh Transform (DWT)

Inverse Walsh transform kernel for N=4 and n=2 bits:

$$\mathrm{h}(x,u) = \prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

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

$x \downarrow$ \ $u \rightarrow$	0	1	2	3
0	1	1	1	1
1	1	1	-1	-1
2	1	-1	1	-1
3	1	-1	-1	1



Walsh Functions

1D Discrete Walsh Transform (DWT)

- Discrete Walsh transform is based on set of harmonically related rectangular waveforms known as **Walsh functions**

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

Also the kernel for forward and inverse transform is same except the scaling factor 1/N (positive aspect of Walsh transform)

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

$$\text{h}(x,u) = \prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

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

1D Discrete Walsh Transform (DWT)

- This transform also has energy compaction but not as strong as DCT
- It is separable & symmetric
  - Perform 1D DWT row-wise
  - Perform 1D DWT column-wise of the resultant

2D Discrete Walsh Transform (DWT)

- Forward transformation kernel for 2D DWT is given by:
- $g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{\{b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)\}}$
- $\therefore W(u, v) = \frac{1}{N} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x, y) \ \prod_{i=0}^{n-1} (-1)^{\{b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)\}}$
- Inverse transformation kernel is:

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

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

Discrete Hadamard Transform (DHT)

- The Hadamard transform is also called as Walsh-Hadamard transform (WHT) and is basically same a Walsh transform **but recursive (???)**
- Forward transform is given by

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

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

- Inverse transform is given by

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

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

2D Discrete Hadamard Transform (DHT)

- Forward transformation kernel for 2D DWT is given by:
- $g(x, y, u, v) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1}\{b_i(x)b_i(u)+b_i(y)b_i(v)\}}$
- Inverse transformation kernel is:

$h(x, y, u, v) = (-1)^{\sum_{i=0}^{n-1}\{b_i(x)b_i(u)+b_i(y)b_i(v)\}}$.

g and h are separable and symmetric and can be implemented row-wise and column-wise

1D Discrete Hadamard Transform (DHT)

Inverse Hadamard transform kernel for N=2 and n=1 bit:

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

$$H_2 = \begin{array}{c|cc} x \downarrow \backslash u \rightarrow & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & -1 \end{array}$$

h(0,0) = 1
h(0,1) = 1
h(1,0) = 1
h(1,1) = -1

1D Discrete Hadamard Transform (DHT)

Inverse Hadamard transform kernel for N=4 and n=2 bit:

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

$$H_2 = \begin{array}{c|cc} x \downarrow \backslash u \rightarrow & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & -1 \end{array}$$

$$H_4 = \begin{array}{c|cccc} x \downarrow \backslash u \rightarrow & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ 2 & 1 & 1 & -1 & -1 \\ 3 & 1 & -1 & -1 & 1 \end{array} = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

Hence H_4 can be computed from H_2 (hence recursive) or H_{2N} can be computed from H_N

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

Kronecker Product

- Kronecker product of two matrices A and B is defined as

$$A \otimes B = \{a(m,n)B\} = \begin{bmatrix} a(1,1)B \; a(1,2)B \ldots a(1,n)B \\ a(2,1)B \; a(2,2)B \ldots \; a(2,n)B \\ . \\ . \\ . \\ a(m,1)B \; a(m,2)B \ldots \; a(m,n)B \end{bmatrix}$$

1D Discrete Hadamard Transform (DHT)

- Basis functions of Hadamard transform are also non sinusoidal.
- Just like Walsh transform, Hadamard transform matrix also has only ± 1 in their basis functions.
- A $N \times N$ Hadamard transform matrix is generated by iterative rule (where $N=2^n$ or $n = log_2 N$)

$$H_n = H_1 \otimes H_{n-1}$$

Where $\otimes$ is Kronecker Product

$$H_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad n = log_2 N$$

$$H_n = H_1 \otimes H_{n-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes H_{n-1}$$

$$H_n = \frac{1}{\sqrt{2}}\begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}$$

Kronecker Product

Example: Prove that $A \otimes B \neq B \otimes A$

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$A \otimes B = \begin{bmatrix} 1\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} & 1\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \\ 1\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} & -1\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 & 3 \\ 2 & 1 & 2 & 1 \\ 4 & 3 & -4 & -3 \\ 2 & 1 & -2 & -1 \end{bmatrix}$$

$$B \otimes A = \begin{bmatrix} 4\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 3\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ 2\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 1\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 4 & 4 & 3 & 3 \\ 4 & -4 & 3 & -3 \\ 2 & 2 & -1 & -1 \\ 2 & -2 & -1 & 1 \end{bmatrix}$$

Therefore $A \otimes B \neq B \otimes A$

Hadamard Matrix

Ex. Calculate 4 × 4 Hadamard matrix

Sol:

Given $N = 4 \Longrightarrow n = log_2 N = log_2 4 \Longrightarrow n = 2$

$$H_2 = H_1 \otimes H_1$$

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

$$H_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Ex. Calculate 8 × 8 Hadamard matrix

Solution :

Given $N = 8 \Longrightarrow n = log_2 N = log_2 8 \Longrightarrow n = 3$

$$H_3 = H_1 \otimes H_2$$

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

$$H_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathrm{H}_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H_3 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Hadamard Transform

- 1D forward Hadamard Transform

$$V = HU$$

- 1D inverse Hadamard Transform

$$U = H^{*T}V$$

- 2D forward Hadamard Transform

$$V = HUH^{T}$$

- 2D inverse Hadamard Transform

$$U = H^{*T}VH^{*}$$

Properties of H

Property 1: H is real

$$H^{*} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = H \Rightarrow H = H^{*}$$

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

Property 2: H is symmetric

$H^{T} = H \Rightarrow$H is symmetric

Property 3: H is unitary

$$HH^{*T} = I$$

Properties of H

Property 3: H is unitary

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

$$H^{*T} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$HH^{*T} = \frac{1}{2}\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

$\because HH^{*T} = I \Rightarrow H$ is unitary matrix.

Property 4: Sequency of H is not in order (no. of sign changes in each row**)**

$$H_3 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{matrix} 0 \\ \to 7 \\ \to 3 \\ \to 4 \\ \to 1 \\ \to 6 \\ \to 2 \\ \to 5 \end{matrix}$$

Hadamard Transform

Note : From above properties matrix notation of Hadamard transform reduces to following form

$$H = H^* = H^T = H^{*T}$$

- 1D forward Hadamard Transform

$$V = HU$$

- 1D inverse Hadamard Transform

$$U = H^{*T}V \quad = H\,V$$

- 2D forward Hadamard Transform

$$V = HUH^T \; = HUH$$

- 2D inverse Hadamard Transform

$$U = H^{*T}VH^* \; = H\,V\,H$$

Hadamard Transform

- It is a fast transform
  - 1D transformation can be implemented in $O(N\log_2 N)$ additions and subtractions
- Since it contains only ± 1 values, no multiplications are required.
  - Useful in situations where minimizing amount of computation is very important
- Energy compaction of WHT is more than Walsh transform but less than DCT
  - Hence if sufficient computation power is available DCT is the choice

Example on Hadamard Transform

1. Find the forward Hadamard transform of the given matrix U.

$$U = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

1D Hadamard Transform is given by

$$V = HU = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1+1+1+1 \\ 1-1+1-1 \\ 1+1-1-1 \\ 1-1-1+1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example on Hadamard Transform

2. Find the forward Hadamard transform of the given matrix U.

$$\boldsymbol{U} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

2D Hadamard Transform is given by.

$$V = HUH = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Example on Hadamard Transform

$$V = \frac{1}{4}\begin{bmatrix} 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

- $$= \frac{1}{4}\begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Examples on Hadamard Transform

Ex 3. Find WHT for x(n) = [2,3,4,5]

$$H = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad V = HU$$

$$V = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1. & -1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 7 \\ -1 \\ -2 \\ 5 \end{bmatrix}$$

Slant Transform

- Many monochrome images have large areas of uniform intensity and areas of linearly increasing or decreasing brightness.
- With the exception of the discrete sine transform, all of the transforms that we have studied so far include a basis vector for representing efficiently constant gray level areas
- **None has a basis function that is targeted specifically at the representation of linearly increasing or decreasing intensity values.**
- **Slant transform includes such a basis function.**

Slant Transform

- Slant transform is widely used in image compression
- It is orthogonal and hence very fast
- Its kernel can be generated recursively like Hadamard transform
- Its basis function has sawtooth waveform or slant basis vectors
- A slant basis vector that is monotonically decreasing in consistent steps from maximum to minimum has the sequency property and a fast computational algorithm

Slant Transform

- Transformation matrix of the slant transform of order N×N where $N= 2^n$ is generated recursively using

$$\mathbf{A}_{\mathrm{Sl}} = \frac{1}{\sqrt{N}}\mathbf{S}_N$$

where *slant* matrix

$$\mathbf{S}_N = \begin{bmatrix} 1 & 0 & & 1 & 0 & \\ & & \mathbf{0} & & & \mathbf{0} \\ a_N & b_N & & -a_N & b_N & \\ & \mathbf{0} & \mathbf{I}_{(N/2)-2} & & \mathbf{0} & \mathbf{I}_{(N/2)-2} \\ 0 & 1 & & 0 & -1 & \\ & & \mathbf{0} & & & \mathbf{0} \\ -b_N & a_N & & b_N & a_N & \\ & \mathbf{0} & \mathbf{I}_{(N/2)-2} & & \mathbf{0} & -\mathbf{I}_{(N/2)-2} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{N/2} \end{bmatrix}$$

Here I_N , is the identity matrix of order N×N

I_1 is a 1×1 identity matrix [1] and I_0 is the empty matrix of size 0×0.

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

Slant Transform

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

and coefficients a_N and b_N are

$$a_N = \left[\frac{3N^2}{4(N^2-1)}\right]^{1/2} \quad \text{and} \quad b_N = \left[\frac{N^2-4}{4(N^2-1)}\right]^{1/2}$$

for $N > 1$. When $N \geq 8,$ matrix $\mathbf{S}_N$ is not sequency ordered, but can be made so

Slant Transform

Example: For N=4 determine the slant matrix A_{sl}

using

$$\mathbf{A}_{\mathrm{Sl}} = \frac{1}{\sqrt{N}}\mathbf{S}_N \qquad \text{For N=4} \qquad \mathbf{A}_{\mathrm{Sl}} = \frac{1}{\sqrt{4}}\mathbf{S}_4$$

$$\mathbf{S}_N = \begin{bmatrix} 1 & 0 & & 1 & 0 & \\ & & \mathbf{0} & & & \mathbf{0} \\ a_N & b_N & & -a_N & b_N & \\ & \mathbf{0} & \mathbf{I}_{(N/2)-2} & & \mathbf{0} & \mathbf{I}_{(N/2)-2} \\ 0 & 1 & & 0 & -1 & \\ & & \mathbf{0} & & & \mathbf{0} \\ -b_N & a_N & & b_N & a_N & \\ & \mathbf{0} & \mathbf{I}_{(N/2)-2} & & \mathbf{0} & -\mathbf{I}_{(N/2)-2} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{N/2} \end{bmatrix}$$

Using this determine S_4

Slant Transform

$$\mathbf{S}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad a_N = \left[\frac{3N^2}{4(N^2-1)}\right]^{1/2} \qquad b_N = \left[\frac{N^2-4}{4(N^2-1)}\right]^{1/2}$$

Step 1: Determine a_4 and b_4

$$a_4 = \left[\frac{3N^2}{4(N^2-1)}\right]^{1/2} = \left[\frac{3(4^2)}{4(4^2-1)}\right]^{1/2} = \frac{2}{\sqrt{5}}$$

$$b_4 = \left[\frac{N^2-4}{4(N^2-1)}\right]^{1/2} = \left[\frac{4^2-4}{4(4^2-1)}\right]^{1/2} = \frac{1}{\sqrt{5}}$$

Step 2: Determine S_4

Slant Transform

Step 2: Determine S_4

$$a_4 = \frac{2}{\sqrt{5}} \qquad b_4 = \frac{1}{\sqrt{5}} \qquad \mathbf{A}_{\mathrm{Sl}} = \frac{1}{\sqrt{4}}\mathbf{S}_4$$

$$S_4 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ 0 & 1 & 0 & -1 \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$A_{sl} = \frac{1}{2}S_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

$$\mathbf{S}_N = \begin{bmatrix} 1 & 0 & \mathbf{0} & 1 & 0 & \mathbf{0} \\ a_N & b_N & & -a_N & b_N & \\ \mathbf{0} & & \mathbf{I}_{(N/2)-2} & \mathbf{0} & & \mathbf{I}_{(N/2)-2} \\ 0 & 1 & \mathbf{0} & 0 & -1 & \mathbf{0} \\ -b_N & a_N & & b_N & a_N & \\ \mathbf{0} & & \mathbf{I}_{(N/2)-2} & \mathbf{0} & & -\mathbf{I}_{(N/2)-2} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{N/2} \end{bmatrix}$$

I_0= Empty Matrix of size 0

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

Since N= 4, the basis vectors of A_{Sl} (and the rows of slant matrix of S_4) are sequency ordered

Slant Transform

1D Slant transform:-

Forward Slant Transform

$$V = SU$$

Inverse Slant Transform

$$U = S^{*T}V \qquad (\because S^* = S)$$
$$= S^T V$$

2D Slant Transform:-

Forward Slant Transform

$$V = SUS^T$$

Inverse Slant Transform

$$U = S^{*T} V S^* \qquad (\because S^* = S)$$
$$U = S^T V S$$

Slant Transform

Example: Determine the slant transform the slant transform of function f= [2 3 4 5]^T

$$t_{sl} = A_{sl}\, f = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$\mathbf{t}_{\mathrm{Sl}} = \mathbf{A}_{\mathrm{Sl}}\mathbf{f} = [7 \quad -2.24 \quad 0 \quad 0]^T$$

- The transform contains only two nonzero terms, while the Walsh-Hadamard transform in the previous example had three nonzero terms.
- The slant transform represents f more efficiently because f is a linearly increasing function—that is, f is highly correlated with the slant basis vector of sequency one.
- Thus, there are fewer terms in a linear expansion using slant basis functions as opposed to Walsh basis functions.

Slant Transform

- The slant transform reproduces the linear variations very well
- Its performance at edges is not as optimal as KLT or DCT
- Because of slant nature of the lower order coefficients its effect is to smear the edges

Next Session

- Haar Transform
- KL Transform
- SVD

THANK YOU

Dr. Shikha Tripathi
Department of Electronics & Communication Engineering

shikha@pes.edu
+91 9482219115