DIGITAL IMAGE PROCESSING-1

Unit 2: Lecture 21-22

Dr. Shikha Tripathi
Department of Electronics &
Communication Engineering

DIGITAL IMAGE PROCESSING-1

Unit 2: Image Transforms

Dr. Shikha Tripathi
Department of Electronics & Communication Engineering

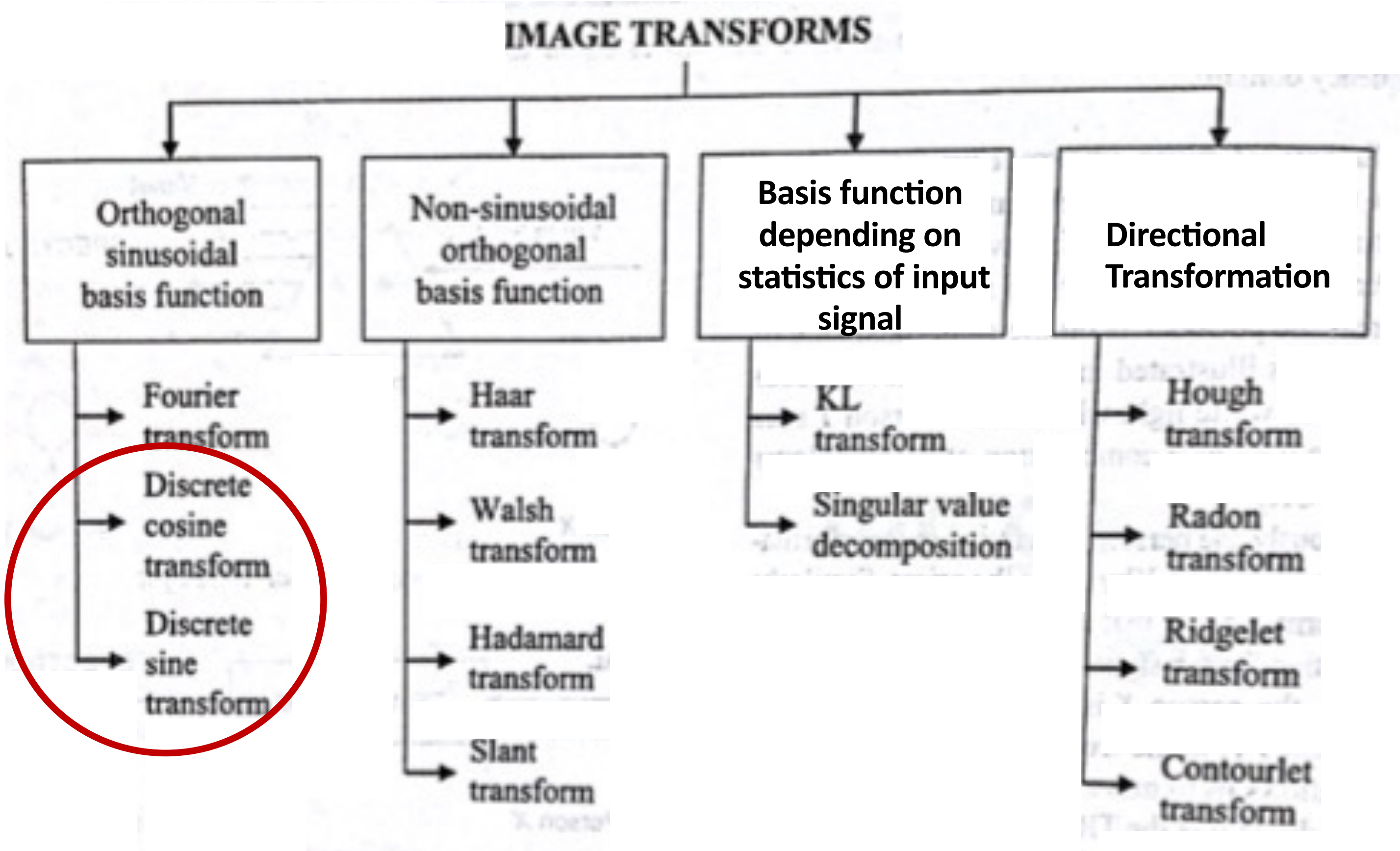
Last Session

- 1 D and 2D DFT Cont..
- Introduction to Discrete Cosine Transform (DCT)

Today's Session

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

Classification of Image Transforms



Discrete Cosine transform (DCT)

- DFT represents N-point sequence u(n) as a linear combination of complex exponentials (basis functions)
- In DFT basis sequences are complex sequences. Hence X(k) is in general complex even if x(n) is real
- **DCT is derived from DFT using its symmetry property**
- **The transformed sequence, v(k) is generally complex even if u(n) is real**

Objective of DCT:

- **To have orthogonal transforms that represent a real time-domain sequence u(n) by a real transform domain sequence v(k)**
- **This is achieved by incorporating symmetry in the input signal u(n)**

Aim: To develop a symmetric or antisymmetric periodic extension from a given finite length sequence

- Total combinations =

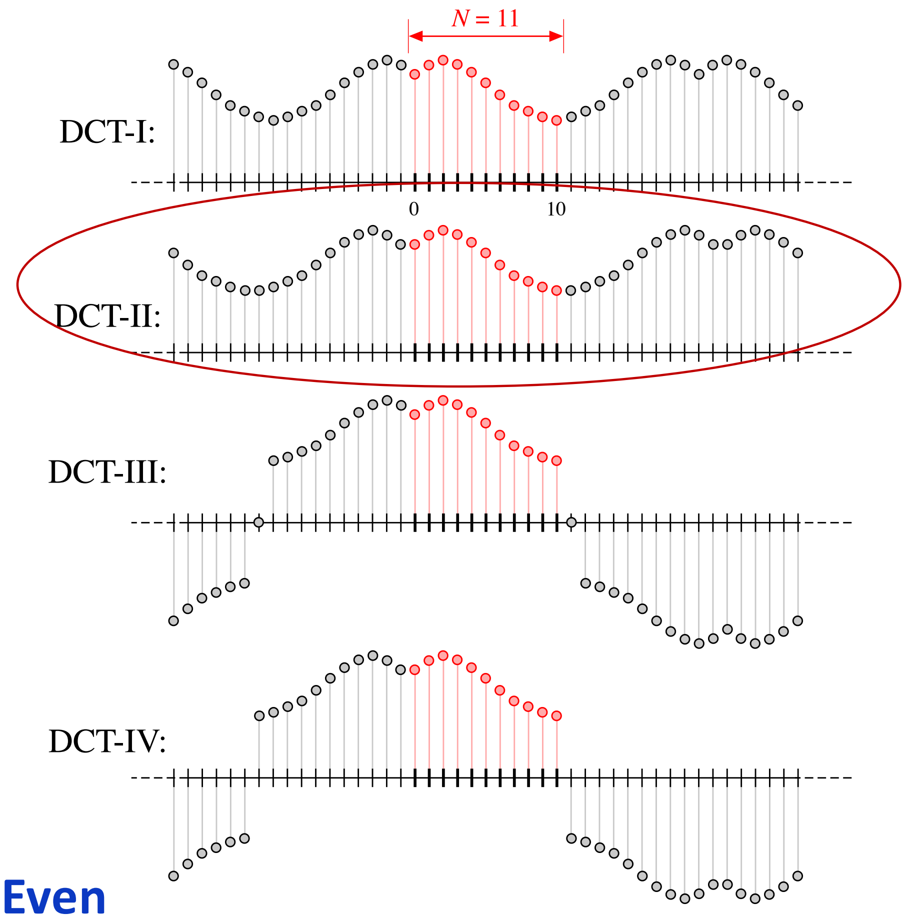
For even symmetry (4 for N even and 4 for N odd):

Type 1, Type 2, Type 3, Type 4, Type 5, Type 6, Type 7, Type 8

- There are eight standard DCT variants and they assume different symmetry conditions.
- **For example, the input could be assumed to be even about a sample or about a point halfway between two samples.**

Even symmetric extension: Discrete Cosine Transform (DCT);Type 1 -4(most common)

Periodic extension process: N Odd



Similarly we get 4 combinations for N Even

Discrete Cosine Transform (DCT)

- In DCT the objective is to find N x N orthogonal transform that expresses a real sequence x(n) as a linear combination of cosine sequences
- **This is possible if N-point sequence is real and even that is $x(n) = x(-n)$ and $x(n) = \pm x(N-n)$, $n = 0,1,2,\ldots.N-1$. Then $X(k)$ is also real and even**
- **To get a DCT for any N-point real sequence, we can find 2N-point DFT of an even extension of the sequence x(n)**
- **There are 8 different ways to perform this even extension giving rise to 8 definitions of DCT**
- **Also there are 8 ways to perform odd (anti symmetric) extensions leading to 8 types of DST**
- Type 2 DCT (DCT II) is widely used for speech and image processing applications as part of various standards (Rao & Huang, 1996). **DCT generally refers to DCT II**

Discrete Cosine Transform (DCT)

- In DCT basis functions are cosine. Since cosines are both periodic and have even symmetry the extension of x(n) (outside the range 0 to N-1) in the synthesis will be both periodic and symmetric
- Just like DFT involved implicit assumption of periodicity, DCT involves implicit assumption of both periodicity and even symmetry
- It is a powerful signal decorrelator. Amplitude of autocorrelation after DCT is small
- It’s a real valued function and thus can be effectively used in real time DSP operations

1D Discrete Cosine Transform (DCT)

- Consider a sequence $u(n)$ having N elements. DCT of $u(n)$ is defined as

$$v(k) = a(k)\sum_{n=0}^{N-1} u(n)cos\left[\frac{\pi(2n+1)k}{2N}\right]; k = 0,1,\dots N-1$$

Where$a(0) = \frac{1}{\sqrt{N}}$ and $a(k) = \sqrt{\frac{2}{N}}$

Inverse DCT is given by

$$u(n) = \sum_{k=0}^{N-1} a(k)v(k)cos\left[\frac{\pi(2n+1)k}{2N}\right]; \quad n = 0,1,\dots N-1$$

1D Discrete Cosine Transform (DCT)

$\boldsymbol{N \times N}$ cosine transform matrix is given by $C = \{c(k,n)\}$

$$\boldsymbol{c(k,n)} = \begin{cases} \dfrac{\mathbf{1}}{\sqrt{\boldsymbol{N}}} \quad ; & \boldsymbol{k = 0, 0 \le n \le N-1} \\ \sqrt{\dfrac{\mathbf{2}}{\boldsymbol{N}}}\ \boldsymbol{cos}\left[\dfrac{\boldsymbol{\pi(2n+1)k}}{\boldsymbol{2N}}\right] \quad ; & \boldsymbol{1 \le k \le N-1} \\ & \boldsymbol{0 \le n \le N-1} \end{cases}$$

1D Discrete Cosine Transform (DCT)

N=4

$$\begin{array}{cc} & \begin{array}{cccc} n=0 & n=1 & n=2 & n=3 \end{array} \\ \begin{array}{c} k=0 \\ k=1 \\ k=2 \\ k=3 \end{array} & \boldsymbol{C} = \begin{bmatrix} \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} & \frac{1}{\sqrt{4}} \\ \sqrt{\frac{2}{4}}\cos\frac{\pi}{8} & \sqrt{\frac{2}{4}}\cos\frac{3\pi}{8} & \sqrt{\frac{2}{4}}\cos\frac{5\pi}{8} & \sqrt{\frac{2}{4}}\cos\frac{7\pi}{8} \\ \sqrt{\frac{2}{4}}\cos\frac{\pi}{4} & \sqrt{\frac{2}{4}}\cos\frac{3\pi}{4} & \sqrt{\frac{2}{4}}\cos\frac{5\pi}{4} & \sqrt{\frac{2}{4}}\cos\frac{7\pi}{4} \\ \sqrt{\frac{2}{4}}\cos\frac{3\pi}{8} & \sqrt{\frac{2}{4}}\cos\frac{9\pi}{8} & \sqrt{\frac{2}{4}}\cos\frac{15\pi}{4} & \sqrt{\frac{2}{4}}\cos\frac{21\pi}{8} \end{bmatrix} \end{array}$$

1D Discrete Cosine Transform (DCT)

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} \\ \mathbf{0.65} & \mathbf{0.27} & \mathbf{-0.27} & \mathbf{-0.65} \\ \mathbf{0.5} & \mathbf{-0.5} & \mathbf{-0.5} & \mathbf{0.5} \\ \mathbf{0.27} & \mathbf{-0.65} & \mathbf{0.65} & \mathbf{-0.27} \end{bmatrix}$$

In matrix notation

$$\left.\begin{array}{l} \text{Forward Transform } V = CU \\ \text{Inverse Transform } U = C^{*T}V \end{array}\right\} \text{For 1D DCT}$$

2D Discrete Cosine Transform (DCT)

- Consider a sequence u(m,n) having N x N elements. DFT of u(m,n) is defined as

$$v(k,l) = a(k)a(l)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)\cos\left[\frac{\pi(2m+1)k}{2N}\right]\cos\left[\frac{\pi(2n+1)l}{2N}\right];$$

$$0 \le k,l \le N-1$$

Where$a(0) = \frac{1}{\sqrt{N}}$ and $a(k) = a(l) = \sqrt{\frac{2}{N}}$

Inverse DCT is given by

$$u(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a(k)\,a(l)v(k,l)\cos\left[\frac{\pi(2m+1)k}{2N}\right]\cos\left[\frac{\pi(2n+1)l}{2N}\right];$$

$$m,n = 0,1,\ldots N-1$$

In matrix notation

$$\left.\begin{aligned}&\text{Forward 2D DC Transform } V = CUC^{T}\\&\text{Inverse 2D DC Transform } U = C^{*T}VC^{*}\end{aligned}\right\}\ \text{For 2D DCT}$$

Example on 1D DCT

1. Find forward and inverse DCT of the given matrix U.

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

w.k.t

In matrix notation

Forward Transform $V = CU$

Inverse Transform $U = C^{*T}V$

$$V = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Example on 1D DCT

$$= \begin{bmatrix} 0.5+0.5+0.5+0.5 \\ 0.65+0.27-0.27-0.65 \\ 0.5-0.5-0.5+0.5 \\ 0.27-0.65+0.65-0.27 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

* IDCT is given by.

$$U = C^T V$$

$$= \begin{bmatrix} 0.5 & 0.65 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.65 & 0.5 & -0.27 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+0+0+0 \\ 1+0+0+0 \\ 1+0+0+0 \\ 1+0+0+0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = U$$

Example on 2D DCT

2. Find forward and inverse DCT of the given matrix U.

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

* Forward DCT is given by

$$V = CUC^T$$

$$V = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0.65 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.65 & 0.5 & -0.27 \end{bmatrix}$$

Example on 2D DCT

$$V = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.65 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.65 & 0.5 & -0.27 \end{bmatrix}$$

$$V = \begin{bmatrix} 1+1+1+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} //$$

Example on 2D DCT

* Inverse DCT is given by : $U = C^T V C$

$$= \begin{bmatrix} 0.5 & 0.625 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.625 & 0.5 & -0.27 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & -0.65 & -0.27 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = U$$

Example on DCT Cont

3. (a) Let f = $[0\ 1\ 4\ 9]^T$. Create an 8-point extension of f with even symmetry and find the DCT using DFT of the extended signal.
   (b)Find forward DCT of the given signal and reconstruct using IDCT

Sol: g = $[0\ 1\ 4\ 9\ 9\ 4\ 1\ 0]^T$ is one period of an even symmetric function.
If we take DFT of this and consider the first 4 samples we get:

$$V = \begin{bmatrix} 7 \\ -6.69 \\ 2 \\ -0.48 \end{bmatrix}$$

Example on DCT Cont

1 (b)Find forward and inverse DCT of the given signal.

Sol: If we take DCT of this we get:

$$V = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 4 \\ 9 \end{bmatrix} = \begin{bmatrix} 7 \\ -6.69 \\ 2 \\ -0.48 \end{bmatrix}$$

Example on DCT Cont

- Signal Reconstruction using IDCT

Inverse DCT is given by

$$u(n) = \sum_{k=0}^{N-1} a(k)v(k)cos\left[\frac{\pi(2n+1)k}{2N}\right]; \quad n = 0,1, \ldots N-1$$

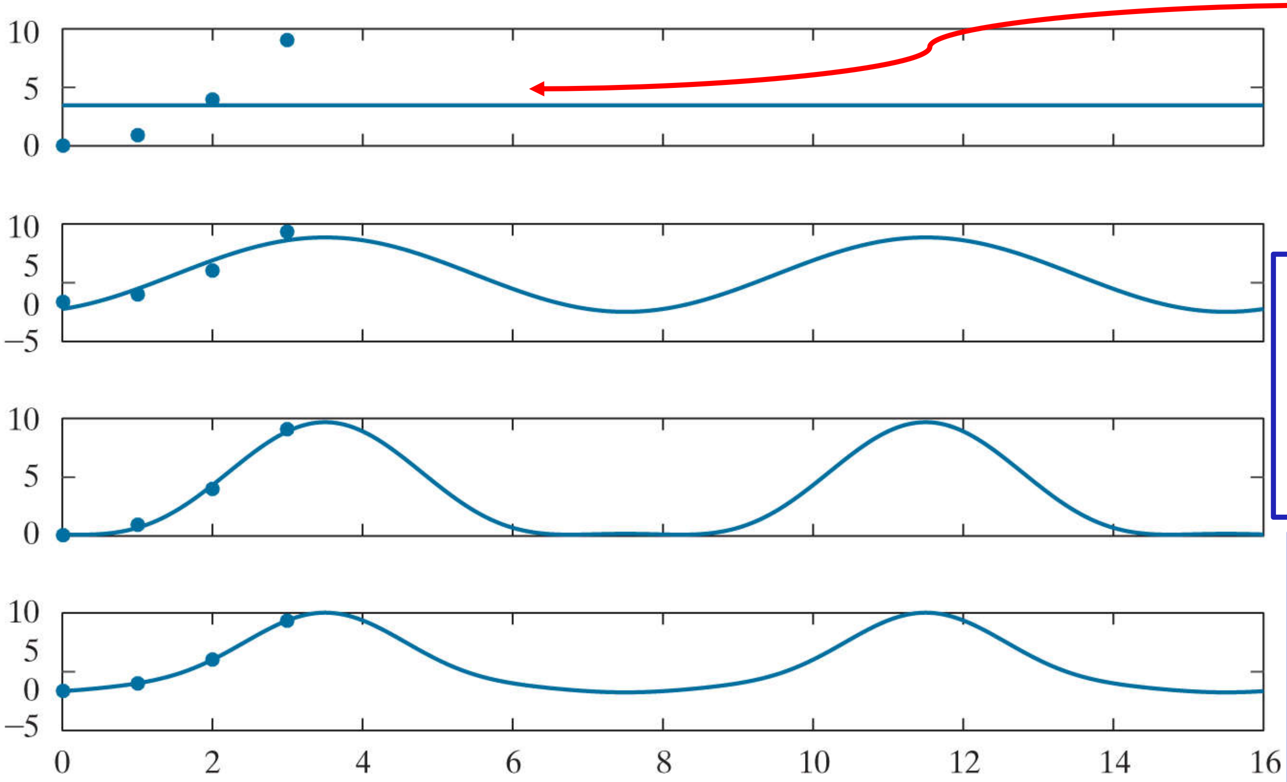
Where $a(0) = \frac{1}{\sqrt{N}}$ and $a(k) = \sqrt{\frac{2}{N}}$

* IDCT is given by.

$U = C^T V$

Example on 2D DCT

If we reconstruct the signal using IDCT we get



DC component is the average value of the discrete function—in this case, $(0+1+4+9)/4 = 3.5$. It is an initial but crude approximation of f.

As three additional cosines of increasing frequency are added in the figure, the accuracy of the approximation increases until a perfect reconstruction is achieved.

Note the x-axis has been extended to show that the resulting DCT expansion is indeed periodic with period 2N (in this case 8) and exhibits the even symmetry that is required of all discrete cosine transforms.

DCT reconstruction of a discrete function by the addition of progressively higher frequency components.
Note the 2N-point periodicity and even symmetry imposed by the DCT .

Properties of DCT

Property 1: C is real

$$\Longrightarrow C = C^*$$

$$C = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix}$$

Property 2: C is not symmetric

$$C^T = \begin{bmatrix} 0.5 & 0.65 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.65 & 0.5 & -0.27 \end{bmatrix} \neq C$$

Property 3: C is unitary

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

Properties of DCT

$$C^{*T} = \boldsymbol{C^T}$$

$$= \begin{bmatrix} 0.5 & 0.65 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.65 & 0.5 & -0.27 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix}$$

$$CC^{*T} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix} \begin{bmatrix} 0.5 & 0.65 & 0.5 & 0.27 \\ 0.5 & 0.27 & -0.5 & -0.65 \\ 0.5 & -0.27 & -0.5 & 0.65 \\ 0.5 & -0.65 & 0.5 & -0.27 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

Hence $CC^{*T} = I \Rightarrow C$ is Unitary.

Property 4: Sequency of DCT is in order (no. of sign changes in each row**)**

$$
\begin{array}{cc}
\boldsymbol{C} = \begin{bmatrix} \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} \\ \mathbf{0.65} & \mathbf{0.27} & \mathbf{-0.27} & \mathbf{-0.65} \\ \mathbf{0.5} & \mathbf{-0.5} & \mathbf{-0.5} & \mathbf{0.5} \\ \mathbf{0.27} & \mathbf{-0.65} & \mathbf{0.65} & \mathbf{-0.27} \end{bmatrix} & \begin{array}{l} \to 0 \\ \to 1 \\ \to 2 \\ \to 3 \end{array}
\end{array}
$$

Note :

Using above properties the DCT expressions in matrix notation can be expressed as follows:

1D Transform

$$V = C\,U$$
$$U = C^T V$$

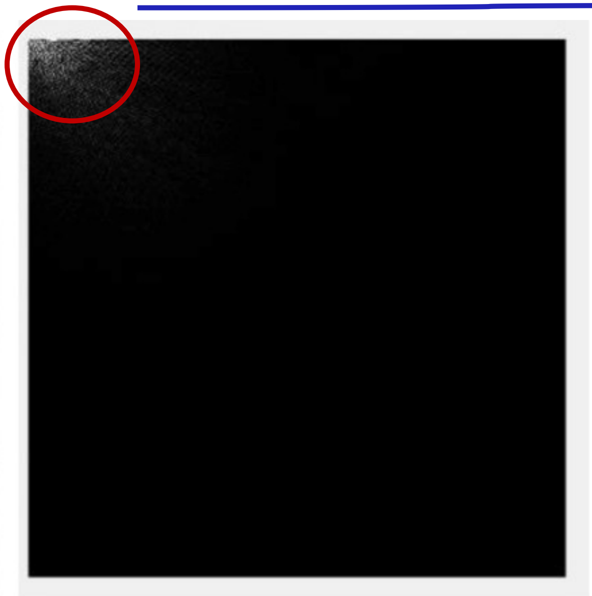
2D Transform

$$V = CU\,C^T$$
$$V = C^T V C$$

Property 5: **Has excellent energy compaction property: most of signal information tends to be concentrated in a few low frequency components of DCT**

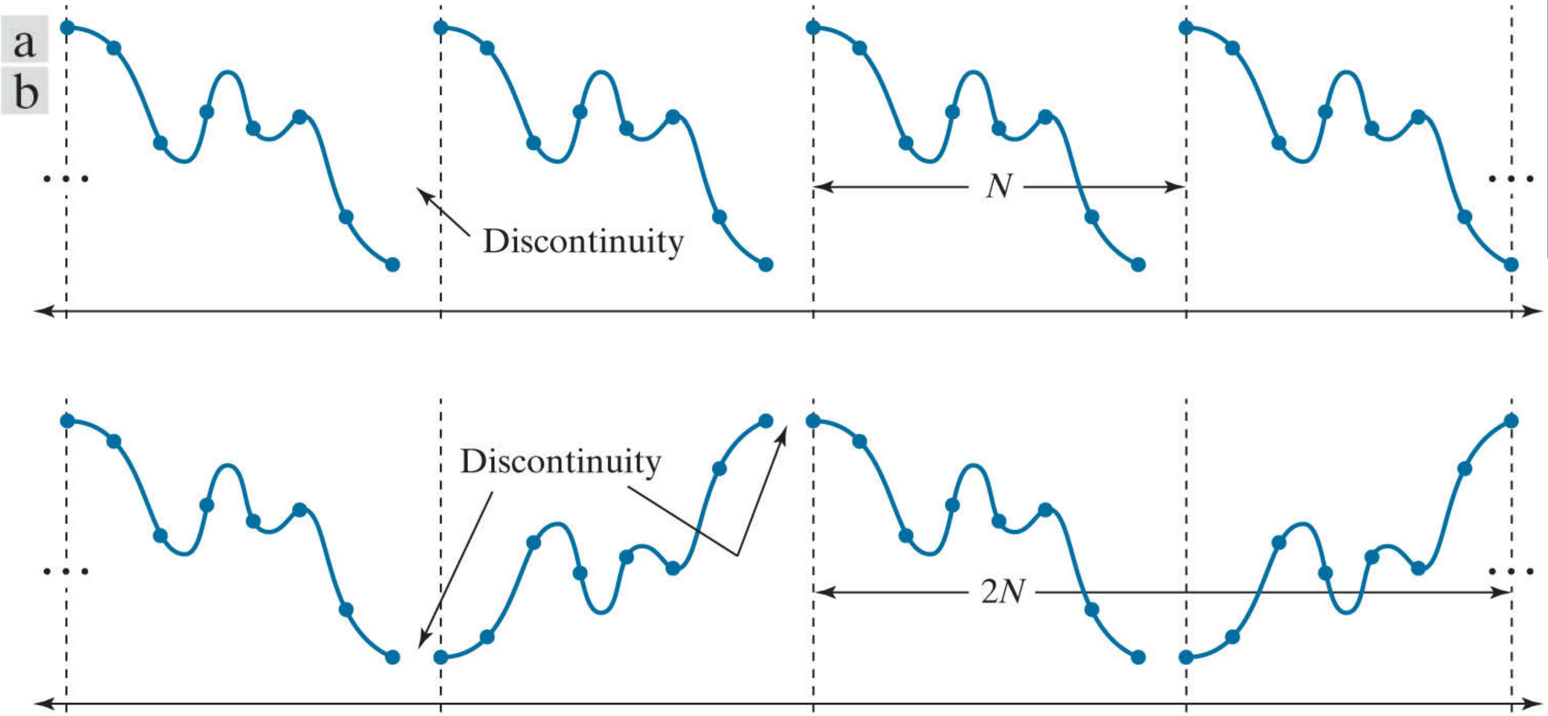
Most of energy concentrated in upper top region of matrix

(a) Original Lena



(b) 2D DCT of Lena

Advantages of DCT



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

Rather than N-point periodicity, (the underlying assumption of the DFT), the discrete cosine transform assumes 2N-point periodicity and even symmetry

while N-point periodicity can cause boundary discontinuities that introduce “artificial” high-frequency components into a transform, 2N-point periodicity and even symmetry minimize discontinuity, as well as the accompanying high-frequency artifact.

Advantages of DCT

- Time: Takes less time for computation (fast algorithms available)
- Space: Takes less volume
- Complexity: Has low complexity
- Feasibilty: Easily implemented in realtime
- Provides better energy compaction than DFT for image data
- Unlike DFT, DCT is real valued & provides better approximation of a signal with fewer coefficients

Applications of DCT

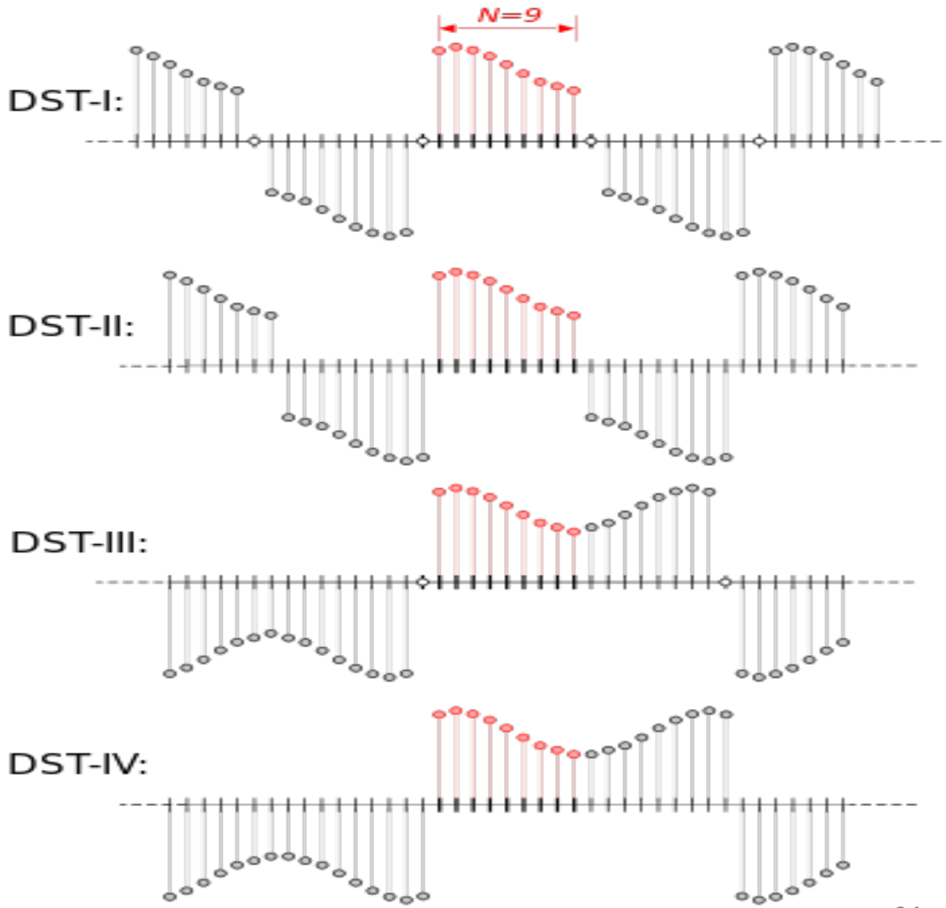
- It is becoming more powerful in real time because of speed and energy compaction property
- It is central to many kinds of signal processing but mainly used in image processing applications
- For non-stationary signals like speech, DCT provides good approximation of signal with fewer coefficients

Discrete Sine Transform (DST)

- To get a DCT for any N-point real sequence, we find 2N-point DFT of an even extension of the sequence x(n)
- There are 8 different ways to perform this even extension giving rise to 8 definitions of DCT
- **Also there are 8 ways to perform odd (anti symmetric) extensions leading to 8 types of DST**

Odd symmetric extension: Discrete Sine Transform (DST);Type 1 -4

Periodic extension process: N Odd



Similarly we get 4 combinations for N Even

1D Discrete Sine Transform (DST)

Forward Discrete Sine Transform for a one dimensional sequence u(n), $0 \leq n \leq N-1$

$$v(k) = \sqrt{\frac{2}{N+1}} \sum_{n=0}^{N-1} u(n) \frac{\sin\pi(k+1)(n+1)}{N+1}; 0 \leq k \leq N-1$$

- Inverse DST defined as

$$u(n) = \sqrt{\frac{2}{N+1}} \sum_{k=0}^{N-1} v(k) \frac{\sin\pi(k+1)(n+1)}{N+1}; 0 \leq n \leq N-1$$

$N \times N$ sine transform matrix $\psi(\mathrm{m,n})$ is defined as

$$\psi(m,n) = \sqrt{\frac{2}{N+1}} \frac{\sin\pi(k+1)(n+1)}{N+1}; 0 \leq k,n \leq N-1$$

1D Discrete Sine Transform (DST)

For $N = 4, 4 \times 4 \;\; \psi(m,n)$ matrix is given by

$$\psi = \sqrt{\frac{2}{5}}\begin{bmatrix} \sin\frac{\pi}{5} & \sin\frac{2\pi}{5} & \sin\frac{3\pi}{5} & \sin\frac{4\pi}{5} \\ \sin\frac{2\pi}{5} & \sin\frac{4\pi}{5} & \sin\frac{6\pi}{5} & \sin\frac{8\pi}{5} \\ \sin\frac{3\pi}{5} & \sin\frac{6\pi}{5} & \sin\frac{9\pi}{5} & \sin\frac{12\pi}{5} \\ \sin\frac{4\pi}{5} & \sin\frac{8\pi}{5} & \sin\frac{12\pi}{5} & \sin\frac{16\pi}{5} \end{bmatrix} = \begin{bmatrix} 0.372 & 0.602 & 0.60 & 0.372 \\ 0.602 & 0.372 & -0.372 & -0.602 \\ 0.602 & -0.372 & -0.372 & 0.602 \\ 0.372 & -0.602 & 0.602 & -0.372 \end{bmatrix}$$

In matrix form

$$V = \psi U$$

$$U = \psi^{*T} V$$

2D Discrete Sine Transform (DST)

Forward Discrete Sine Transform for 2 dimensional sequence u(m,n), $0 \leq m,n \leq \; N-1$
Is defined as

$$v(k,l) = \sqrt{\frac{2}{N+1}}\sqrt{\frac{2}{N+1}}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)\frac{sin\pi(k+1)(m+1)}{N+1}\frac{sin\pi(l+1)(n+1)}{N+1};$$

$$0 \leq k,l \leq N-1$$

- Inverse DST defined as

$$u(m,n) = \sqrt{\frac{2}{N+1}}\sqrt{\frac{2}{N+1}}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)\frac{sin\pi(k+1)(m+1)}{N+1}\frac{sin\pi(l+1)(n+1)}{N+1}$$

$$;0 \leq m,n \leq N-1$$

$N \times N$ sine transform matrix ψ is obtained from above

$$\psi = \begin{bmatrix} 0.372 & 0.602 & 0.60 & 0.372 \\ 0.602 & 0.372 & -0.372 & -0.602 \\ 0.602 & -0.372 & -0.372 & 0.602 \\ 0.372 & -0.602 & 0.602 & -0.372 \end{bmatrix}$$

Properties of DST

Property 1: ψ is real

$$\psi = \begin{bmatrix} 0.372 & 0.602 & 0.60 & 0.372 \\ 0.602 & 0.372 & -0.372 & -0.602 \\ 0.602 & -0.372 & -0.372 & 0.602 \\ 0.372 & -0.602 & 0.602 & -0.372 \end{bmatrix} = \psi^*$$

Property 2: ψ is symmetrical

$$\psi = \psi^T$$

Property 3: ψ is unitary

$$\psi^{*T}\psi = I$$

Property 3: ψ is unitary

$$\psi\psi^{*T} = \begin{bmatrix} 0.372 & 0.602 & 0.602 & 0.372 \\ 0.602 & 0.372 & -0.372 & -0.602 \\ 0.602 & -0.372 & -0.372 & 0.602 \\ 0.372 & -0.602 & 0.602 & -0.372 \end{bmatrix} \cdot \begin{bmatrix} 0.372 & 0.602 & 0.602 & 0.372 \\ 0.602 & 0.372 & -0.372 & -0.602 \\ 0.602 & -0.372 & -0.372 & 0.602 \\ 0.372 & -0.602 & 0.602 & -0.372 \end{bmatrix}$$

$$\therefore \psi\psi^{*T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I.$$

$\therefore \psi$ is unitary matrix.

Properties of DST

Property 4: Sequency of DST is in order (no. of sign changes in each row**)**

$$\begin{array}{cc} \begin{bmatrix} 0.372 & 0.602 & 0.60 & 0.372 \\ 0.602 & 0.372 & -0.372 & -0.602 \\ 0.602 & -0.372 & -0.372 & 0.602 \\ 0.372 & -0.602 & 0.602 & -0.372 \end{bmatrix} & \begin{array}{l} \rightarrow \; 0 \\ \rightarrow \; 1 \\ \rightarrow \; 2 \\ \rightarrow \; 3 \end{array} \end{array}$$

* Hence ψ is real, symmetrical and unitary.

Note :

$\psi = \psi^* = \psi^T = \psi^{-1} = \psi^{*T}$

Using above properties the DST expressions in matrix notation can be expressed as follows:

1D DST

$$V = \psi\, U$$
$$U = \psi\, V$$

FDST $\rightarrow$ $V = \psi U \psi^T$

$V = \psi U \psi \; [\because \psi^T = \psi]$

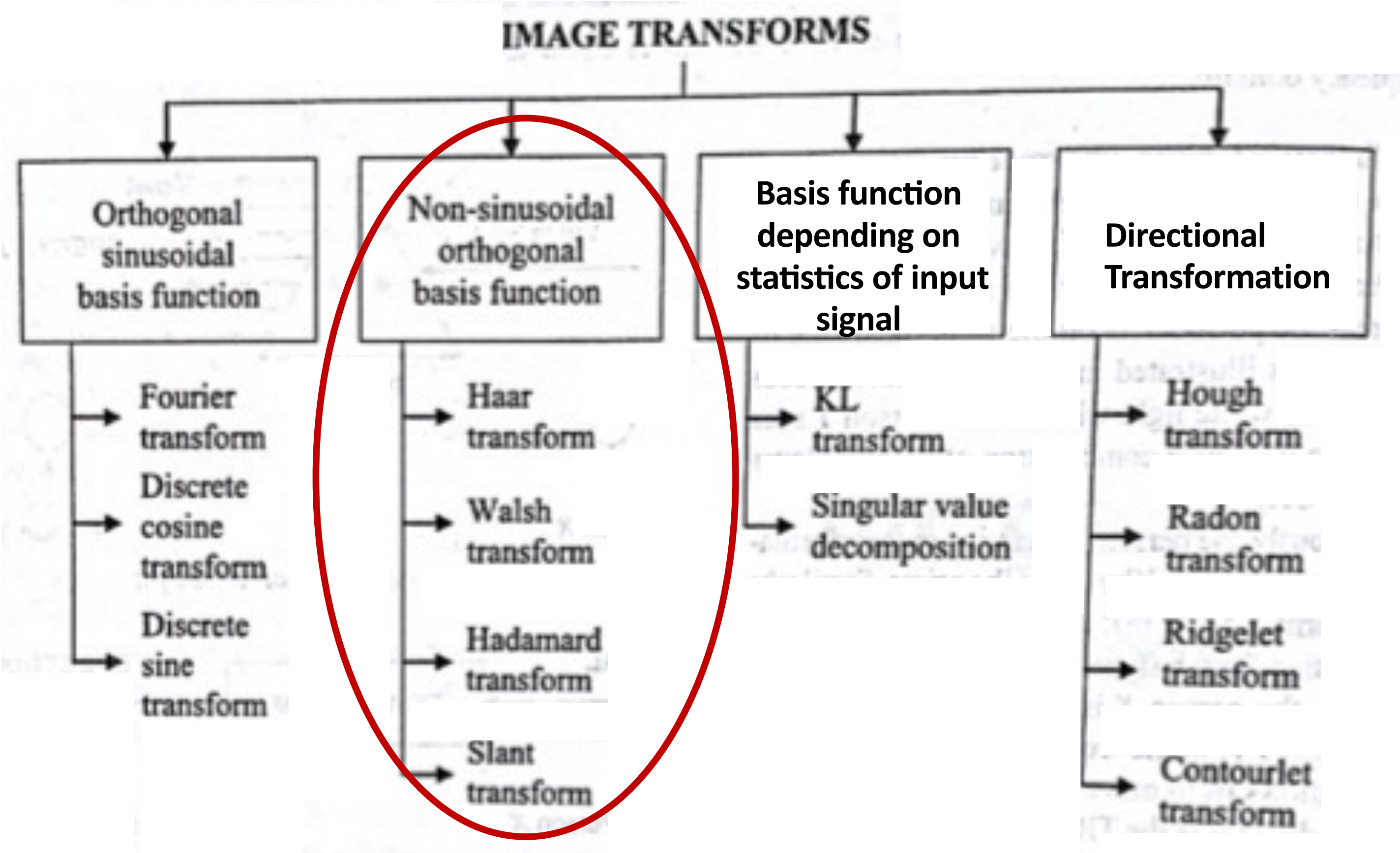
2D DST

IDST $\rightarrow$ $U = \psi^{*T} V \psi$

$U = \psi V \psi \; [\because \psi^* = \psi^T = \psi]$

$$V = \psi\, U\, \psi$$
$$U = \psi\, V\, \psi$$

Classification of Image Transforms



Discrete Walsh Transform (DWT)

- 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 = 0

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

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)

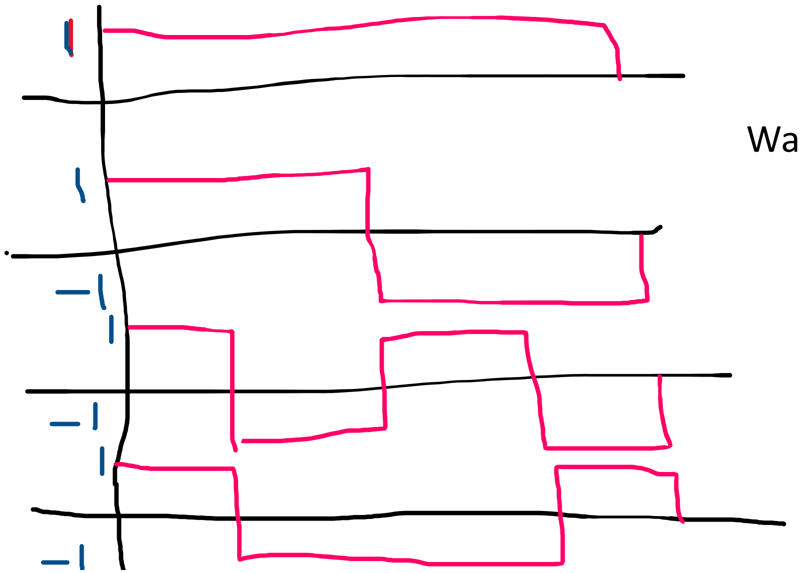
1D Discrete Walsh Transform (DWT)

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

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

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

$x \downarrow$ \ $u \rightarrow$	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**

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

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

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

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

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)

- 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 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:

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

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

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

1D Discrete Hadamard Transform (DHT)

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

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

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

Hence H_4 can be computed from H_2 (hence recursive)

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 non sinusoidal.
- Hadamard transform matrix has only ± 1 in their basis functions.
- A $N \times N$ Hadamard transform matrix is generated by iterative rule

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

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

$$H_2 = H_1 \otimes H_1$$

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

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

Hadamard Matrix

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

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

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

$H^T = H$ ⇒H is symmetric

Property 3: H is unitary

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

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**)**

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

Note : From above properties matrix notation of Hadamard transform reduced 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

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

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.

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

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

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

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

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

Ex 4. Consider the following image

$$\begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) Obtain the 2D DWHT by first taking the 1D transform of rows then taking column wise
(b) Repeat the same by first taking column and then row
(c) Compare & comment

Example on Hadamard Transform

(a) Taking 1D transform of rows:

$$\begin{bmatrix} A_0 & A_1 & A_2 & A_3 \\ B_0 & B_1 & B_2 & B_3 \\ C_0 & C_1 & C_2 & C_3 \\ D_0 & D_1 & D_2 & D_3 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 10 & 2 & 4 & 0 \\ 7 & 1 & 3 & 1 \\ 5 & 1 & 1 & 1 \\ 4 & 0 & 0 & 0 \end{bmatrix}$$

rowwise:
$X = xH$
column wise
$X = Hx$

Example on Hadamard Transform

Taking 1D transform of columns:

$$\begin{bmatrix} w_0 & x_0 & y_0 & z_0 \\ w_1 & x_1 & y_1 & z_1 \\ w_2 & x_2 & y_2 & z_2 \\ w_3 & x_3 & y_3 & z_3 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 10 & 2 & 4 & 0 \\ 7 & 1 & 3 & 1 \\ 5 & 1 & 1 & 1 \\ 4 & 0 & 0 & 0 \end{bmatrix}$$

$$= \frac{1}{16}\begin{bmatrix} 26 & 4 & 8 & 2 \\ 4 & 2 & 2 & 0 \\ 8 & 2 & 6 & 0 \\ 2 & 0 & 0 & -2 \end{bmatrix}$$

Example on Hadamard Transform

(b) Taking 1D transform of columns first:

$$\begin{bmatrix} A_0 & B_0 & C_0 & D_0 \\ A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_3 & B_3 & C_3 & D_3 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 10 & 7 & 5 & 4 \\ 2 & 1 & 1 & 0 \\ 4 & 3 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

row wise

$$\begin{bmatrix} W_0 & W_1 & W_2 & W_3 \\ X_0 & X_1 & X_2 & X_3 \\ Y_0 & Y_1 & Y_2 & Y_3 \\ Z_0 & Z_1 & Z_2 & Z_3 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 10 & 7 & 5 & 4 \\ 2 & 1 & 1 & 0 \\ 4 & 3 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 26 & 4 & 8 & 2 \\ 4 & 2 & 2 & 0 \\ 8 & 2 & 6 & 0 \\ 2 & 0 & 0 & -2 \end{bmatrix}$$

Next Session

- Haar Transform

THANK YOU

Dr. Shikha Tripathi
Department of Electronics & Communication Engineering

shikha@pes.edu
+91 9482219115