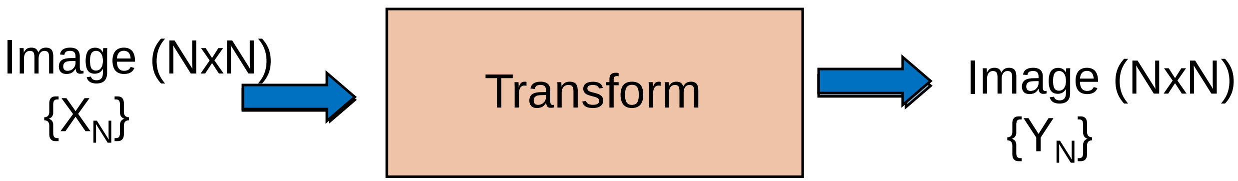
IMAGE TRANSFORMS

By
Dr. Ratnakar Dash
CSE Dept. NIT Rourkela

Image Transforms

- Image transforms can be simple arithmetic or complex mathematical operations on images which convert images from one representation to another.



$$Y=AX$$

$$X=A^{-1}Y$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Applications

- Preprocessing
  - Filtering
  - Enhancement

- Image Compression

- Feature Extraction
  - Edge detection
  - Corner detection

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Unitary Matrix

- Image transformation represents a given image into series sum of unitary matrices

- A matrix is called unitary if

$$A^{-1} = \underbrace{A^{*T} \equiv A^{H}}_{\text{Hermitian conjugate}}$$

- Unitary Matrices are basis images

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Properties of Unitary Transform y =Ax

- Energy Conservation
  - $\| \underline{y} \|^2 = \| \underline{x} \|^2$
    - $\| \underline{y} \|^2 = \| A\underline{x} \|^2 = (A\underline{x})^{*T} (A\underline{x}) = \underline{x}^{*T} A^{*T} A \, \underline{x} = \underline{x}^{*T} \underline{x} = \| \underline{x} \|^2$
- Rotation
  - A unitary transformation is a rotation of a vector in an N-dimension space, i.e., a rotation of basis coordinates

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

- Energy Compaction
  - Many common unitary transforms (like DCT, KLT etc.) tend to pack a large fraction of signal energy into just a few transform coefficients
- Decorrelation
  - Highly correlated input elements ➔ quite uncorrelated output coefficients
  - Covariance matrix $E[(\underline{y} - E(\underline{y}))(\underline{y} - E(\underline{y}))^{*T}]$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

What is orthogonal ?

- A set of real valued continuous functions $\{a_n(t)\} = \{a_o(t), a_1(t), \ldots\}$ is said to be orthogonal over (t_0, t_0+T) if

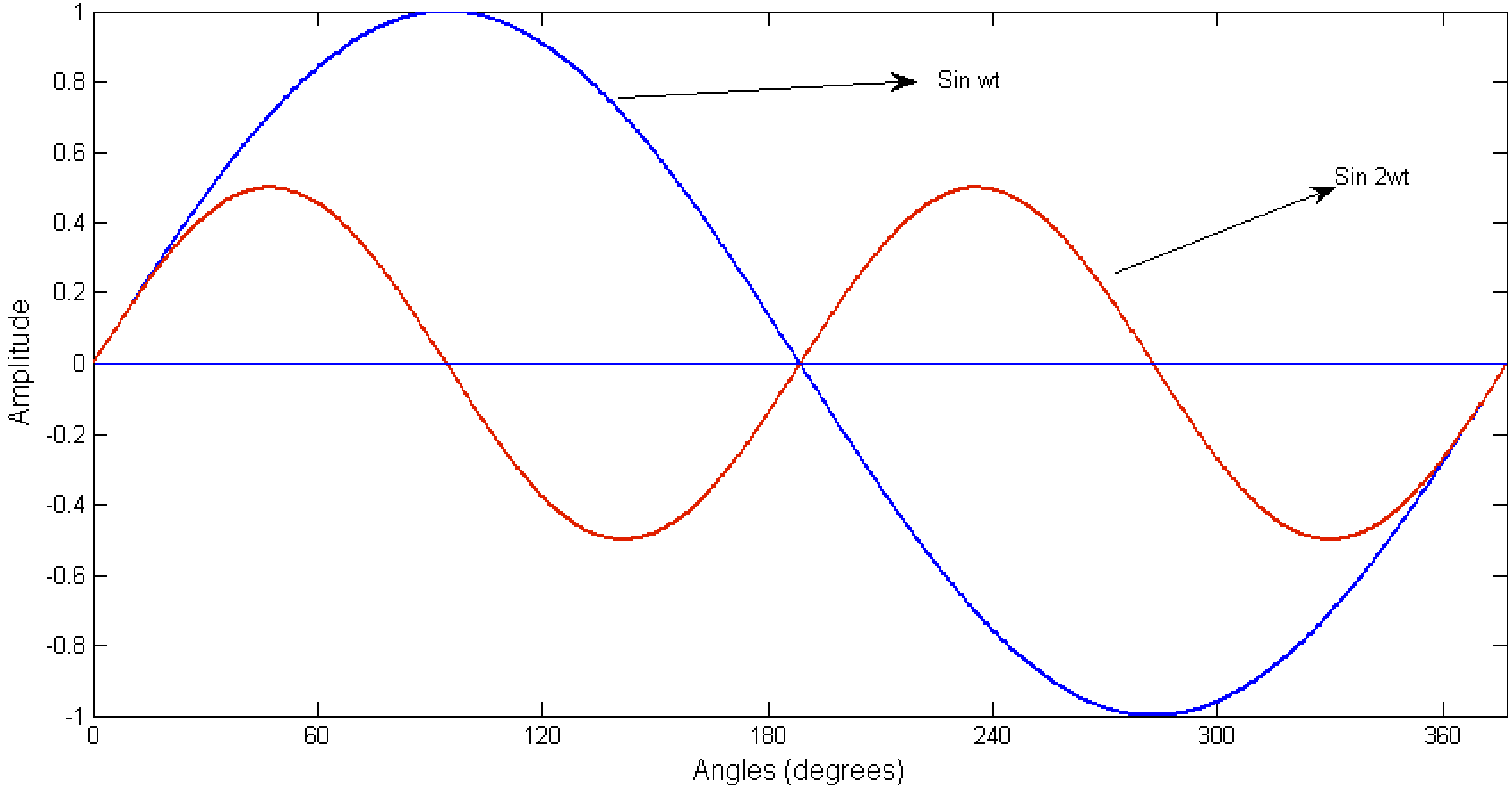
$$\int_T a_m(t).\, a_n(t)\, dt = \begin{cases} k \ \ if \ m = n \\ 0 \ \ if \ \ m \neq n \end{cases}$$

if $k = 1$ then $a_n(t)$ is orthonormal

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Example

- {sinωt, sin2ωt}



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Orthogonal Expansion

Any arbitrary signal *x(t); {t_0, t_0+T}*

Can be represented by series summation of a set of orthogonal basis functions.

$$x(t) = \sum_{n=0}^{\infty} c_n a_n(t)$$

$c_n \longrightarrow$ nth coefficient of expansion

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

$$x(t) = \sum_{n=0}^{\infty} c_n a_n(t), t_0 \le t < t_0 + T$$

$$\Rightarrow \int_T x(t) a_m(t) dt = \int_T \sum_{n=0}^{\infty} c_n a_n(t) a_m(t) dt$$

$$= c_0 \int_T a_0(t) a_m(t) dt + c_1 \int_T a_1(t) a_m(t) dt + \ldots + c_m \int_T a_m(t) a_m(t) dt + \ldots$$

$$\Rightarrow \int_T x(t) a_m(t) dt = k c_m \qquad \Rightarrow \int_T x(t) a_m(t) dt = c_m, \text{ if } k = 1$$

Orthonormal Set

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Complete, Closed(orthogonal functions)

- There is no signal x(t) with $\int_T x^2(t)dt < \infty$

such that

$$\int_T x(t)a_n(t)dt = 0, n = 0,1,..$$

- x(t) with $\int_T x^2(t)dt < \infty$

$$\hat{x}(t) = \sum_{n=0}^{N-1} c_n a_n(t)$$

Such that $\int |x(t) - \hat{x}(t)|^2 dt < \varepsilon \quad \varepsilon > 0$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Discrete Formulation

- Let the set of samples represented by $\{u(n): 0 \leq n \leq N-1\}$ is a vector of dimension N

- Premultiply **u** by a unitary matrix A of dimension N*N, we get another vector

$$v = Au$$

Transformed vector

Transformation matrix

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

In the form of series summation

$$v(k) = \sum_{n=0}^{N-1} a(k,n)u(n) \qquad 0 \le k \le N-1$$

where $\quad A^{-1} = A^{*T}$

As A is a unitary matrix we can get back u as

$$u(n) = \sum_{k=0}^{N-1} a^{*}(k,n)v(k) \qquad 0 \le n \le N-1$$

The columns of A^{*T}

i.e., $(a_k^{*} \approx \{a^{*}(k,n), 0 \le n \le N-1\}^{T})$ are called the basis vectors of A

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Example

- Let

$$U=\begin{bmatrix}u_0\\u_1\\u_2\end{bmatrix}\quad V=\begin{bmatrix}v_0\\v_1\\v_2\end{bmatrix}\quad A=\begin{bmatrix}a_{00}&a_{01}&a_{02}\\a_{10}&a_{11}&a_{12}\\a_{20}&a_{21}&a_{22}\end{bmatrix}$$

$$A^{-1}=A^{*T}=\begin{bmatrix}a_{00}&a_{10}&a_{20}\\a_{01}&a_{11}&a_{21}\\a_{02}&a_{12}&a_{22}\end{bmatrix}^*$$

$$U=A^{-1}V=A^{*T}V$$

$$U=\begin{bmatrix}u_0\\u_1\\u_2\end{bmatrix}=A^{-1}\begin{bmatrix}v_0\\v_1\\v_2\end{bmatrix}=\underbrace{\begin{bmatrix}a_{00}\\a_{01}\\a_{02}\end{bmatrix}^*}_{b_1}v_0+\underbrace{\begin{bmatrix}a_{10}\\a_{11}\\a_{12}\end{bmatrix}^*}_{b_2}v_1+\underbrace{\begin{bmatrix}a_{20}\\a_{21}\\a_{22}\end{bmatrix}^*}_{b_3}v_2$$

basis vectors $\longrightarrow$ b_1 b_2 b_3

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

In case of an image, an image can be represented by a set of basis images

Image $\longrightarrow$ $u(m,n) \qquad 0 \le m,n \le N-1$

$$v(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)u(m,n) \quad 0 \le k,l \le N-1$$

$a_{k,l}(m,n) \longrightarrow$ N × N matrix/ N^2 numbers of such matrices

Inverse Transform

$$u(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a_{k,l}^{*}(m,n)v(k,l) \quad 0 \le m,n \le N-1$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

- Set of complete orthonormal discrete basis function satisfies the following properties

$a_{k,l}(m,n) \longrightarrow$ Orthonormality

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)a^{*}_{k',l'}(m,n) = \delta(k-k', l-l')$$

Completeness

$$\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a_{k,l}(m,n)a^{*}_{k,l}(m', n') = \delta(m-m', n-n')$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Computational Complexity

- To Compute transform coefficient $v(k,l)$

$$v(k,l)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)u(m,n) \quad 0\le k,l\le N-1$$

- For each $v(k,l)$ no. of complex multiplications and additions ~ O(N^2), so for all $v(k,l)$~O(N^4) which is quite expensive for practical size Images.
- How to reduce computation time??
- Using separable unitary transforms

$$a_{k,l}(m,n)=a_k(m).b_l(n)\approx a(k,m)b(l,n)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Separable Transforms

where $\{a_k(m), k = 0,1,..N-1\}$ are 1D orthonormal sets of basis vectors
$\{b_l(n), l = 0,1,..N-1\}$

should be unitary matrices themselves

In most cases we choose A & B to be same. $A \approx \{a(k,m)\}$ *and* $B = \{a(l,n)\}$

$$v(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a(k,m)\,u(m,n)\,a(l,n);\ V = AUA^T$$

$$u(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a^*(k,m)\,v(k,l)\ a^*(l,n);\ U = A^{*T}VA^*$$

$$V = AU\,A^T$$

$$V^T = A[AU]^T$$

- These are called 2D separable transformation
- The 2D transform can be performed first by transforming each column of U and then transforming each row of the result to obtain rows of V complexity is $2N^3$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Concept of Basis Images

- Let a_k^* denotes the k[th] column of A^{*T}
- Define the matrices

$$A_{k,l}^* = a_k^* a_l^{*T}$$

- Define the inner product of two NxN matrices F and G

$$\langle F, G \rangle = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) g^*(m,n)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Concept of Basis Images

- Then the transformation equation can be written as

$$v(k,l)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)u(m,n) \approx \left\langle U, A_{k,l}^{*} \right\rangle$$

$$u(m,n)=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a_{k,l}^{*}(m,n)v(k,l)$$

$$\Rightarrow U = u(m,n)=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)A_{k,l}^{*}$$

- U is represented by linear combination of N^2 Matrices.
- These matrices are called basis images

$$A_{k,l}^{*} \quad k,\ l = 0,1,2..N-1$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Example

$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad U = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Transformed Image $V = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$= \frac{1}{2}\begin{pmatrix} 4 & 6 \\ -2 & -2 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$= \begin{pmatrix} 5 & -1 \\ -2 & 0 \end{pmatrix}$$

To get basis images, take outer product of the columns of A^{*T}

$$A^*_{0,0} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$A^*_{0,1} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} = A^{*T}_{1,0}$$

$$A^*_{1,1} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Contd..

- Inverse Transform gives

$$A^{*T}VA^{*}=\frac{1}{2}\begin{pmatrix}1 & -1\\ 1 & -1\end{pmatrix}\begin{pmatrix}5 & -1\\ -2 & 0\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

$$=\frac{1}{2}\begin{pmatrix}3 & -1\\ 7 & -1\end{pmatrix}\begin{pmatrix}1 & -1\\ 1 & -1\end{pmatrix}=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$$

$$=U \Rightarrow \text{Original Image}$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Some Transformation Techniques

- Discrete Fourier Transform
- Discrete Cosine Transform
- Karhunen-Loeve Transform
- Haar Transform
- Walsh Transform
- Hadamard Transform

Fourier Transform (1D)

Continuous Fourier Transform (CFT)

$$W(f) = \mathcal{F}\{w(t)\} = \int_{-\infty}^{\infty} w(t)\, e^{-j2\pi ft}\, dt$$

$$W(f) = X(f) + j\,Y(f)$$

$$W(f) = |W(f)|\, e^{j\theta(f)}$$

Frequency, [Hz]

Phase Spectrum

Amplitude Spectrum

Inverse Fourier Transform (IFT)

$$w(t) = \mathcal{F}^{-1}\{W(f)\} = \int_{-\infty}^{\infty} W(f)\, e^{+j2\pi ft}\, df$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Discrete Fourier Transform (1D)

- Discrete Domains
  - Discrete Time: $n = 0, 1, 2, 3, \ldots\ldots\ldots\ldots, N\text{-}1$
  - Discrete Frequency: $k = 0, 1, 2, 3, \ldots\ldots\ldots\ldots, N\text{-}1$

Equal time intervals

Equal frequency intervals

- Discrete Fourier Transform

$$X[K] = \sum_{n=0}^{N-1} x[n]\, e^{-j\left(\frac{2\pi}{N}\right)nk}; \qquad K = 0, 1, 2, \ldots.., N\text{-}1$$

- Inverse DFT

$$x[n] = \frac{1}{N} \sum_{K=0}^{N-1} X[K]\, e^{j\left(\frac{2\pi}{N}\right)nk}; \qquad n = 0, 1, 2, \ldots.., N\text{-}1$$

2D – Discrete Fourier Transform

- Image: f(x, y), 0≤x≤N-1, 0≤y≤N-1
- Transform kernel: $g(x,y,u,v) = e^{-j2\pi\left(\frac{xu}{N}+\frac{yv}{N}\right)}$
- Discrete Fourier Transform of **f(x, y)**is

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

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

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

- In general, Transform kernel can be represented as:

$$w_N = e^{-j\frac{2\pi}{N}} \qquad w_N^{ux+vy} = e^{-j\frac{2\pi}{N}(ux+vy)}$$

- Discrete Fourier Transform of ***f(x, y)*** can be represented as

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

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

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

Let $R(u,v)$, $I(u,v)$ be real part and imaginary part of $F(u,v)$

Modulus:

$$|F(u,v)| = \left(R^2(u,v) + I^2(u,v)\right)^{1/2}$$

Phase:

$$\Phi(u,v) = \tan^{-1}\left(\frac{I(u,v)}{R(u,v)}\right)$$

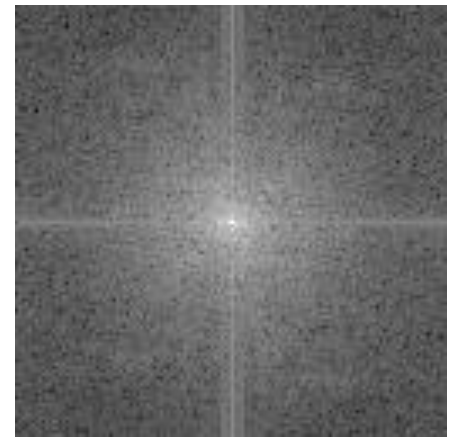
Power Spectrum:

$$E(u,v) = R^2(u,v) + I^2(u,v)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

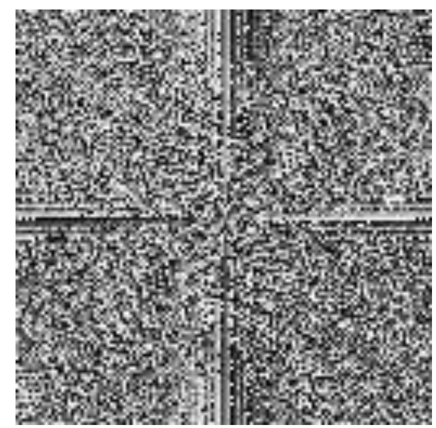
Amplitude and Phase

$$\left|F(x)\right|$$



amplitude

phase



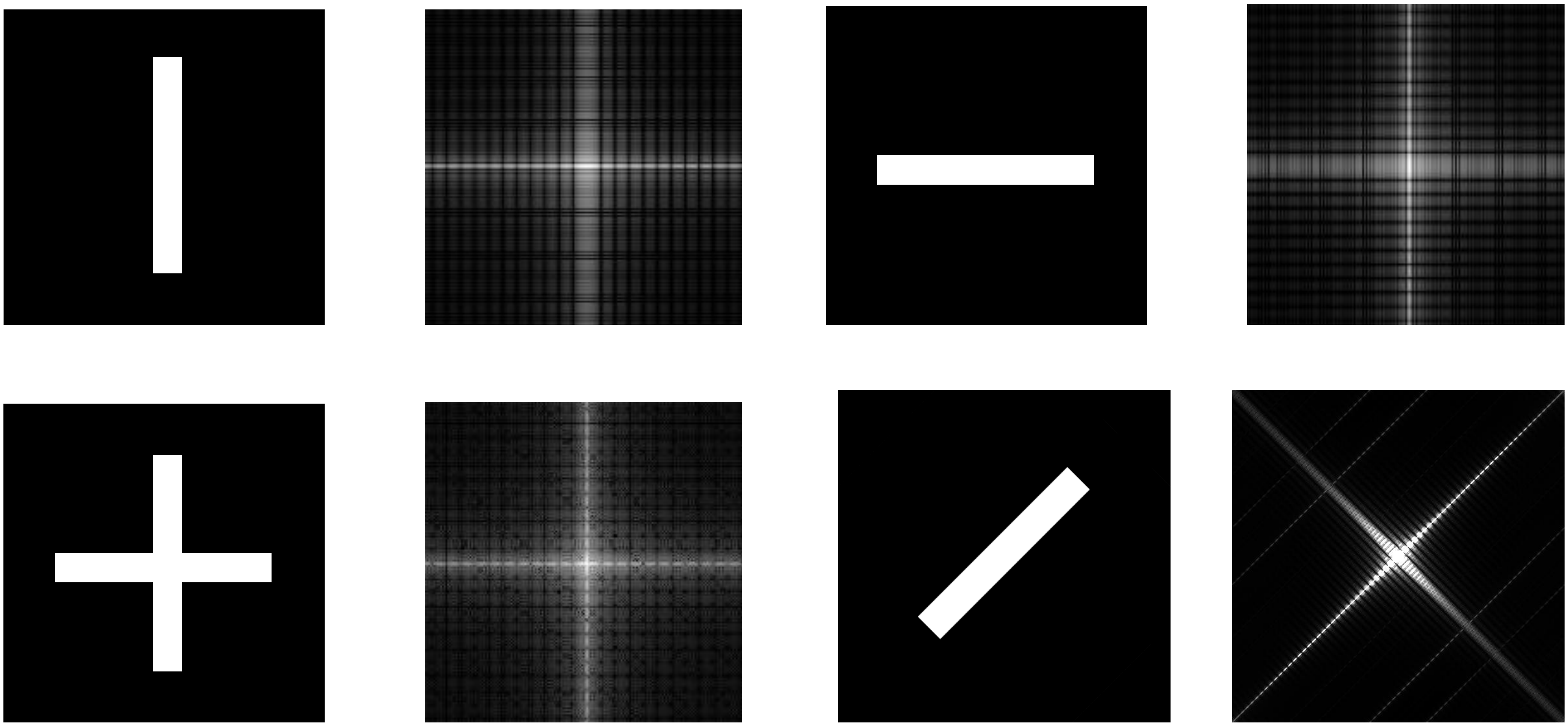
original

$$\angle F(x)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Images and their spectrums

Image domain | Frequency domain | Image domain | Frequency domain



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

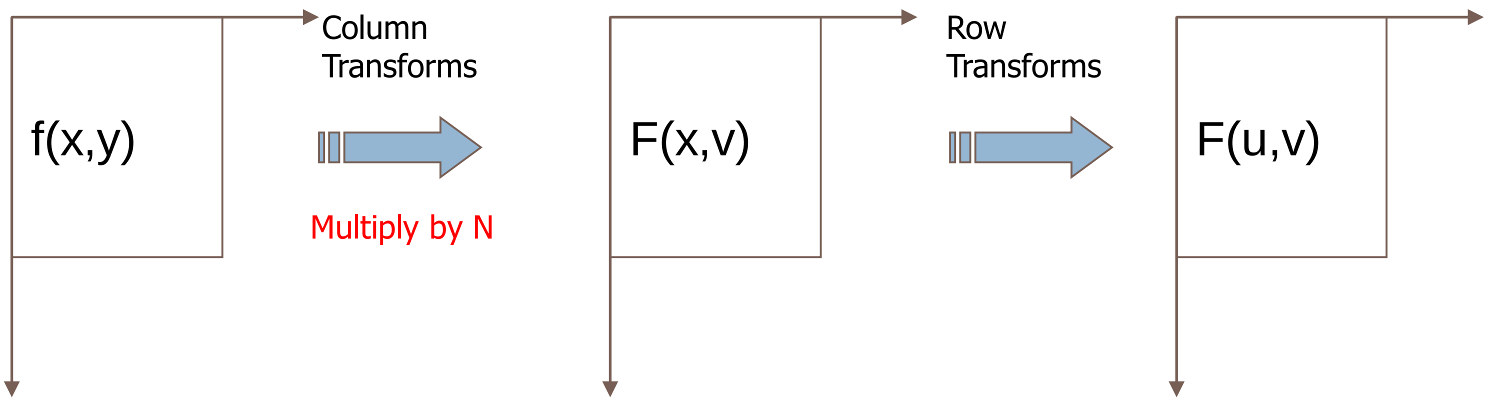
Properties of 2-D Fourier Transform

- Separability
- Linearity
- Scaling
- Shift Theorem
- Rotation
- Periodicity and conjugation
- Convolution
- Correlation
- Average

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

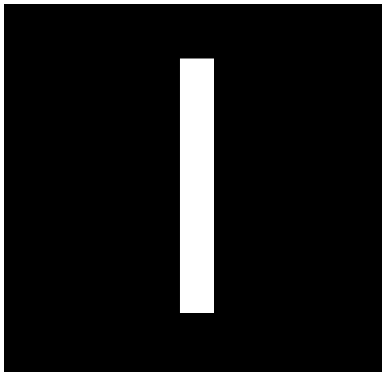
Separability

- The implementation steps for the two-dimensional DFT may be visualized as shown in the diagram below

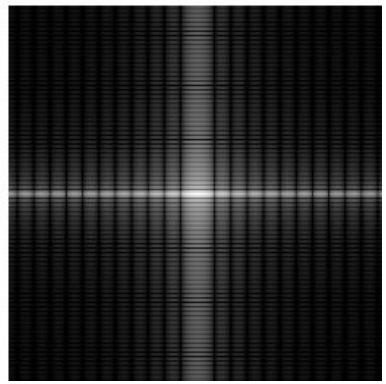


Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

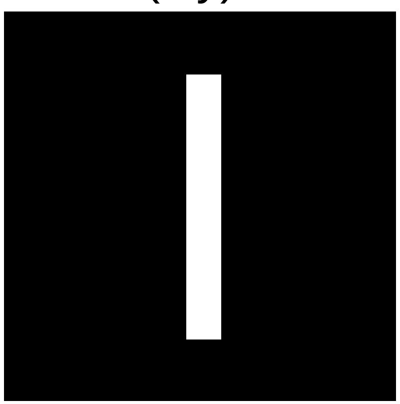
Separability (contd.)



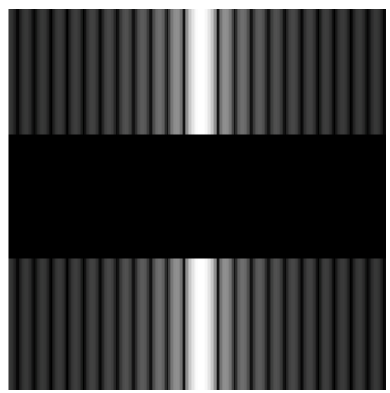
f(x,y)



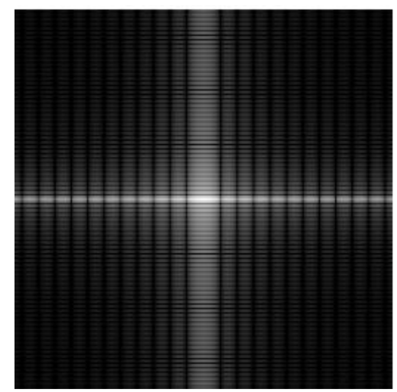
F(u,v)



f(x,y)



F(x,v)



F(u,v)

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

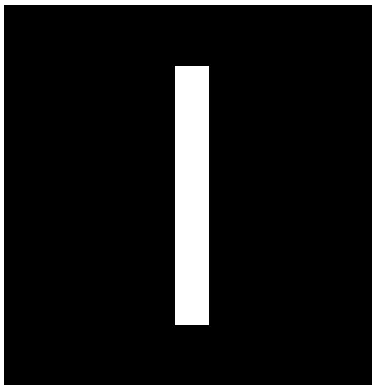
Linearity

- DFT is a linear operator

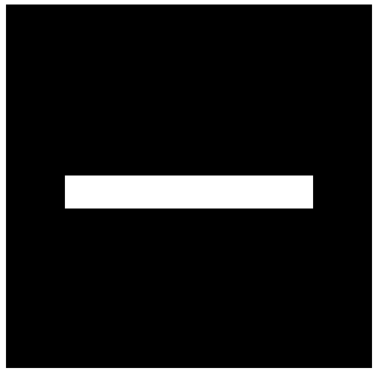
$$F[\ a\, f_1(x,y)+b\, f_2(x,y)] = aF_1(u,v)+bF_2(u,v)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

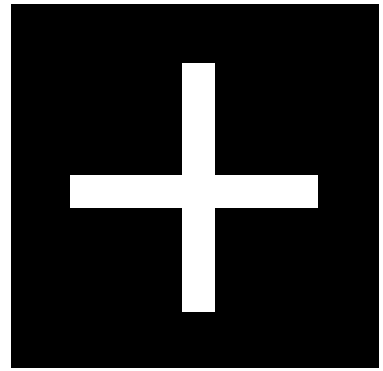
Linearity (contd..)



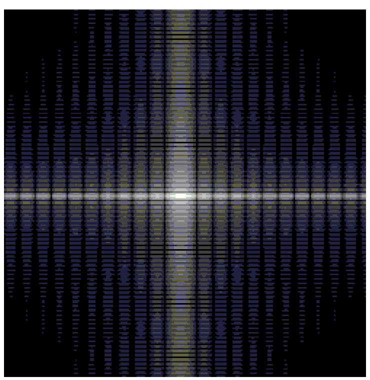
$f_1(x,y)$



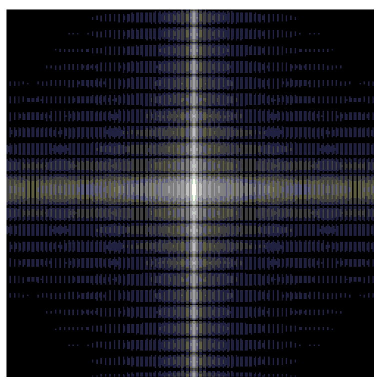
$f_2(x,y)$



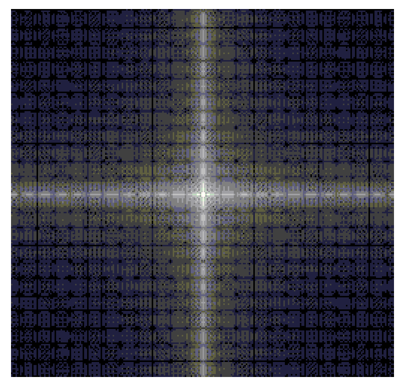
$2f_1(x,y)+3f_2(x,y)$



$F_1(u,v)$



$F_2(u,v)$



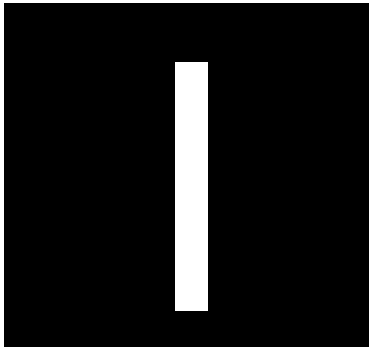
Phase Spectrum $(2F_1(u,v)+3F_2(u,v))$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

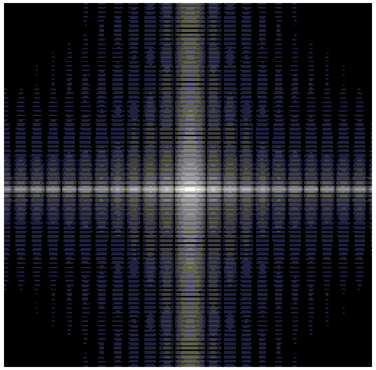
Scaling

$$f(ax, by) \xrightarrow{DFT} \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

Scaling (contd..)

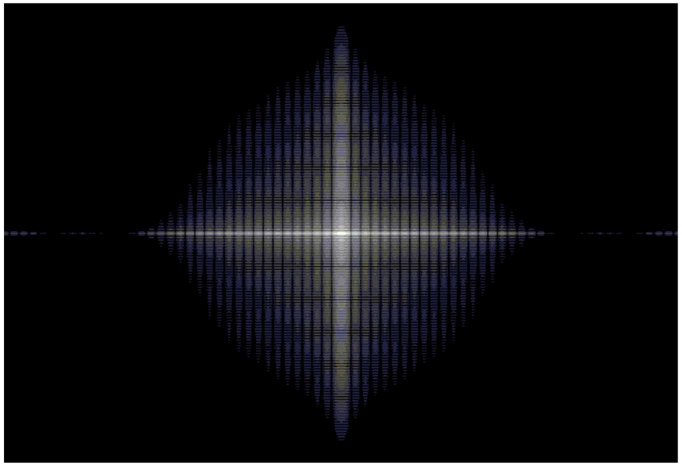


f(x,y)



F(u,v)

f(2x,3y)



F(u,v)

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Shift Theorem

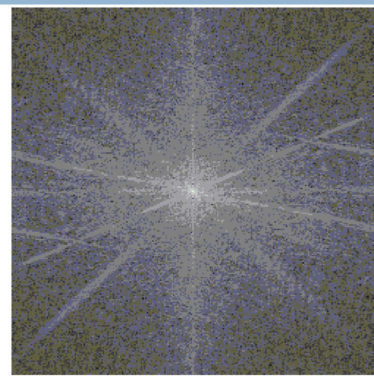
$$f(x-x_0, y-y_0) \Longleftrightarrow F(u,v)\exp[-j2\pi(ux_0+vy_0)/N]$$

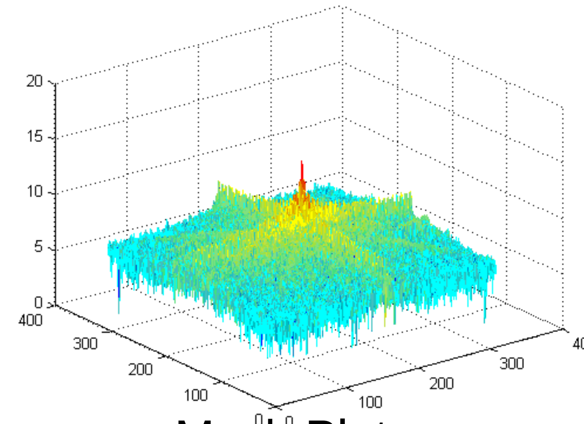
$$f(x,y)\exp[j2\pi(u_0x+v_0y)/N] \Longleftrightarrow F(u-u_0, v-v_0)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Shift Theorem (contd..)

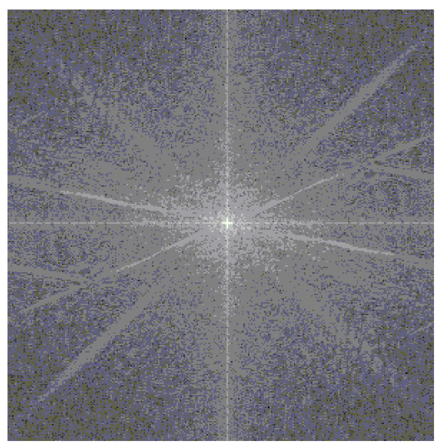

Original Image

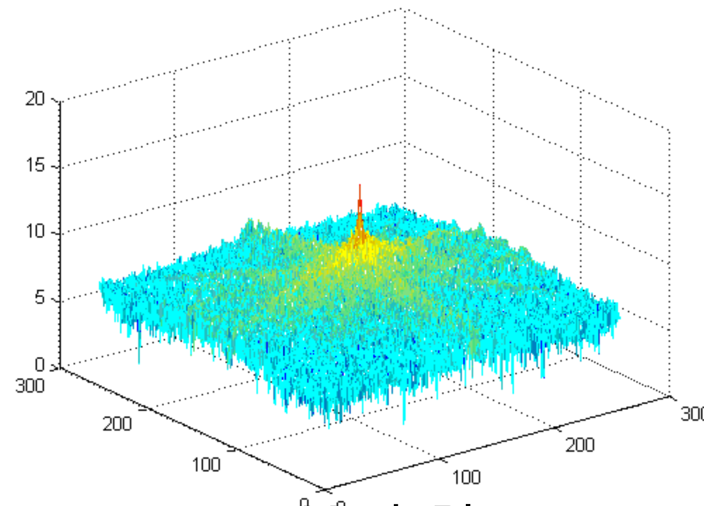

Spectrum Image



Mesh Plot


$T_x = 50, T_y = 50$


Spectrum Image



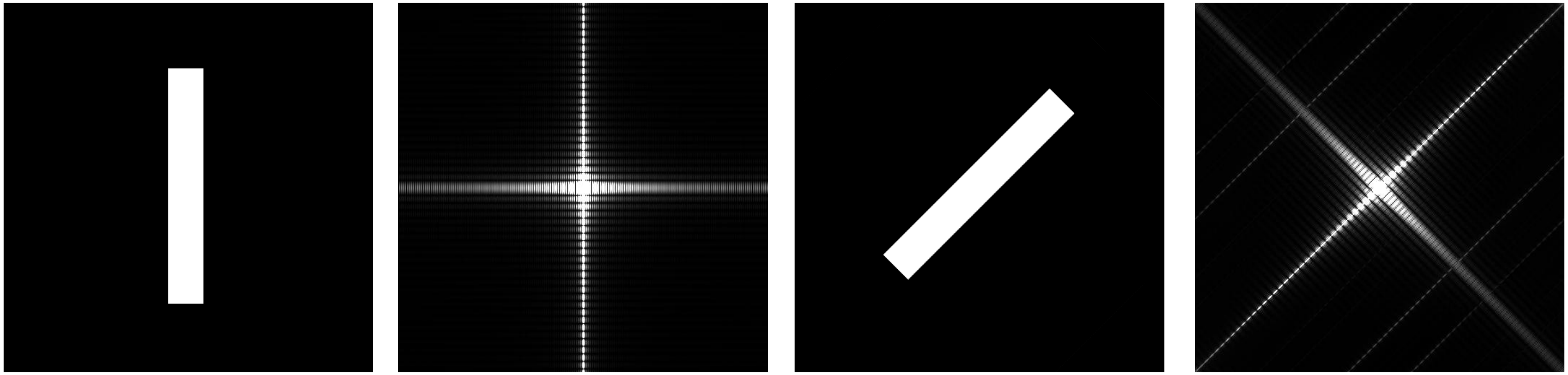
Mesh Plot

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Rotation

In polar coordinates, $f(x,y)$ and $F(u,v)$ can be represented by $f(r, \theta)$ and $F(w,\varphi)$ alternatively. Then

$$f(r, \theta + \theta_0) \Longleftrightarrow F(w, \varphi + \theta_0)$$



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

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Periodicity and conjugation

Periodicity：

$$F(u,v) = F(u+aN, v+bN)$$

$$f(x,y) = f(x+aN, y+bN)$$

Where a, b = 0, $\pm$1, $\pm$ 2,……

Conjugation：

$$F(u,v) = F^*(-u,-v)$$

$$|F(u,v)| = |F(-u,-v)|$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Convolution

2-D convolution is defined as:

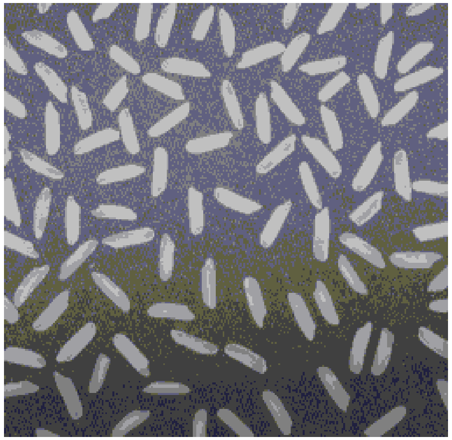
$$f_e(x,y) * g_e(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)g_e(x-m,y-n)$$

Then: $f(x, y)*g(x, y) \Leftrightarrow F(u, v)G(u, v)$

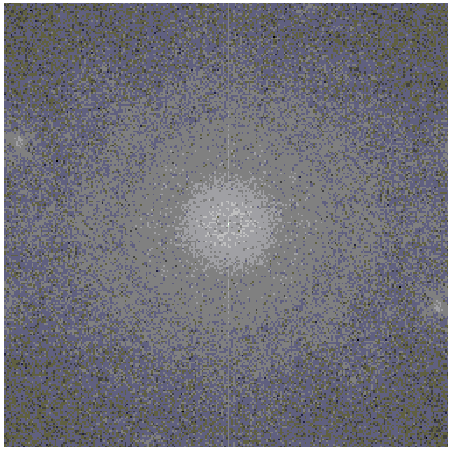
$$f(x, y)g(x, y) \Leftrightarrow F(u, v)*G(u, v)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

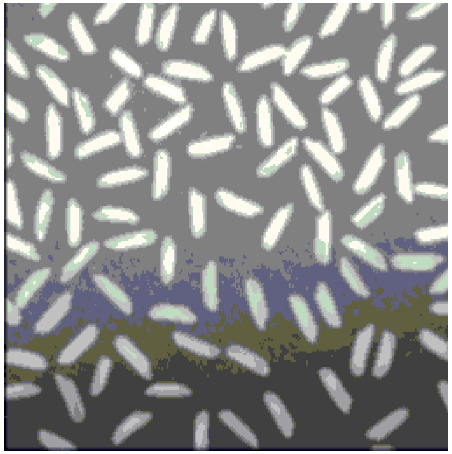
Convolution (contd..)



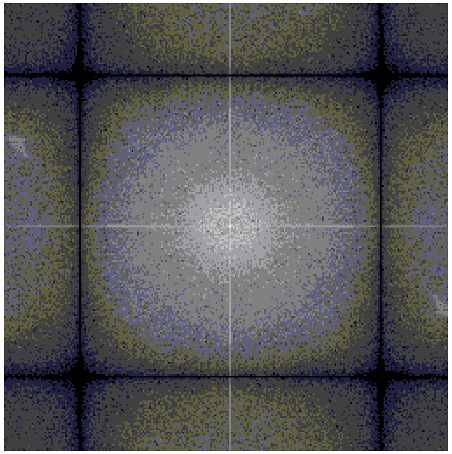
Original image



Spectrum



Convolved image(convolved with mean filter)



Spectrum

Correlation

2-D correlation is defined as：

$$f_e(x,y)\circ g_e(x,y)=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f_e^*(m,n)g_e(x+m,y+n)$$

Correlation Theorems：

$$f(x,y)\circ g(x,y)\Leftrightarrow F^*(u,v)G(u,v)$$

$$f^*(x,y)g(x,y)\Leftrightarrow F(u,v)\circ G(u,v)$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Average

Average of 2D image *f* (x, y) is defined as:

$$\bar{f}(x,y) = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)$$

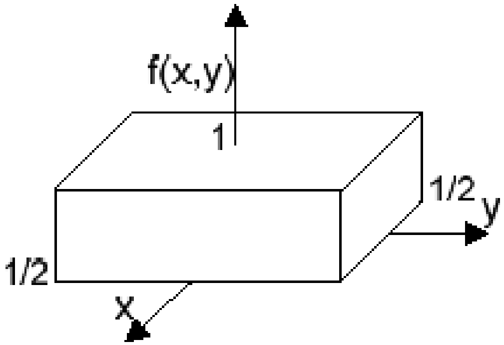
Let $u=v=0$, then:

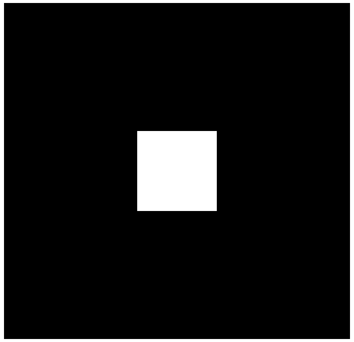
$$F(0,0) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)$$

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

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

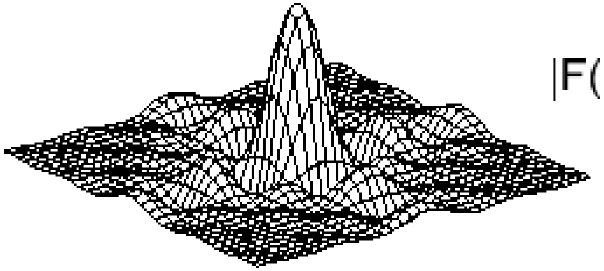




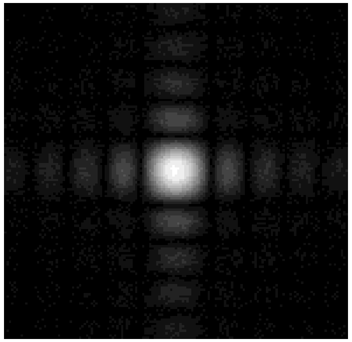
image

$$f(x,y) = \mathrm{rect}(x,y) = \begin{cases} 1 & |x| \quad 1/2,\ |y| \quad 1/2 \\ 0 & \text{otherwise} \end{cases}$$

$$F(u,v) = \ \mathrm{sinc}(u) \quad \mathrm{sinc}(v) = \mathrm{sinc}(u,v)$$

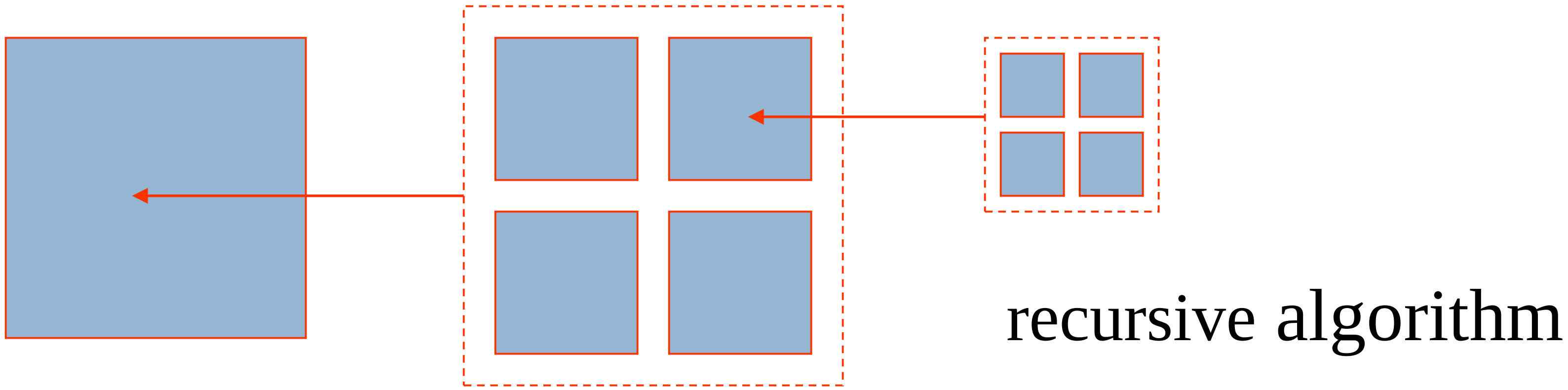


|F(u,v)|



and its
spectrum

Fast Fourier Transform



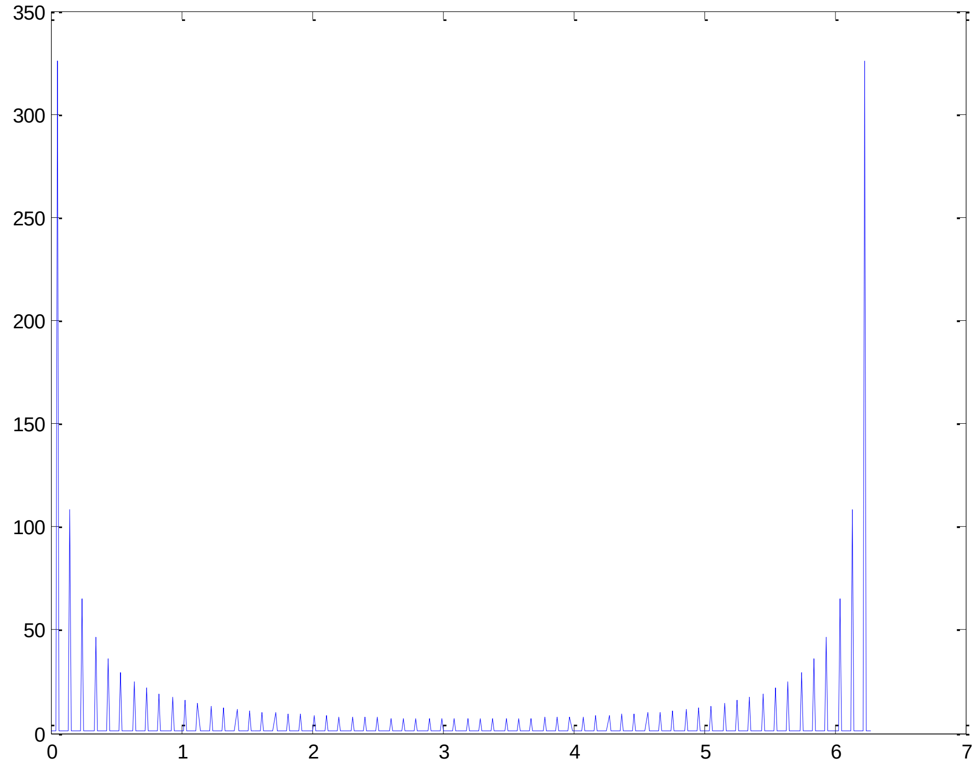
- decimation in time = odd even in freq. domain
- decimation in freq. domain = odd even in time

$$N^4 \rightarrow N^2 \log N$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Visualizing the DFT

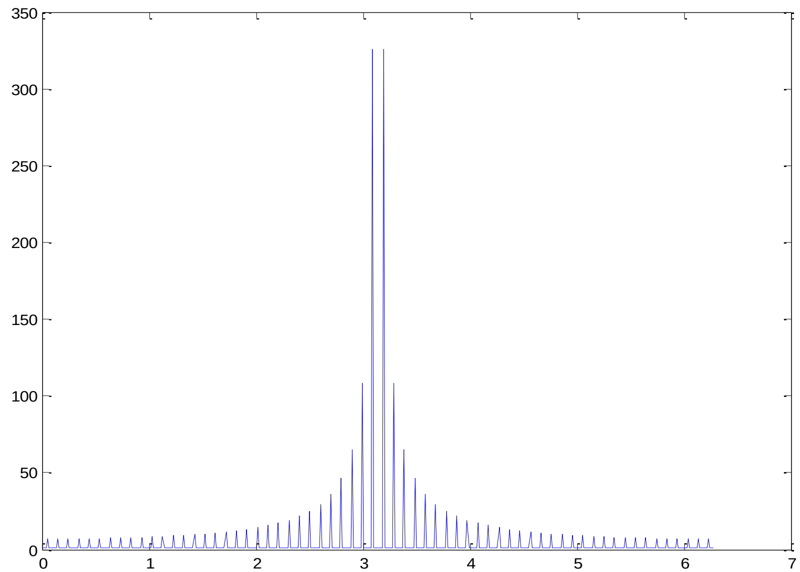
- Consider the power spectrum of the 1D square wave



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

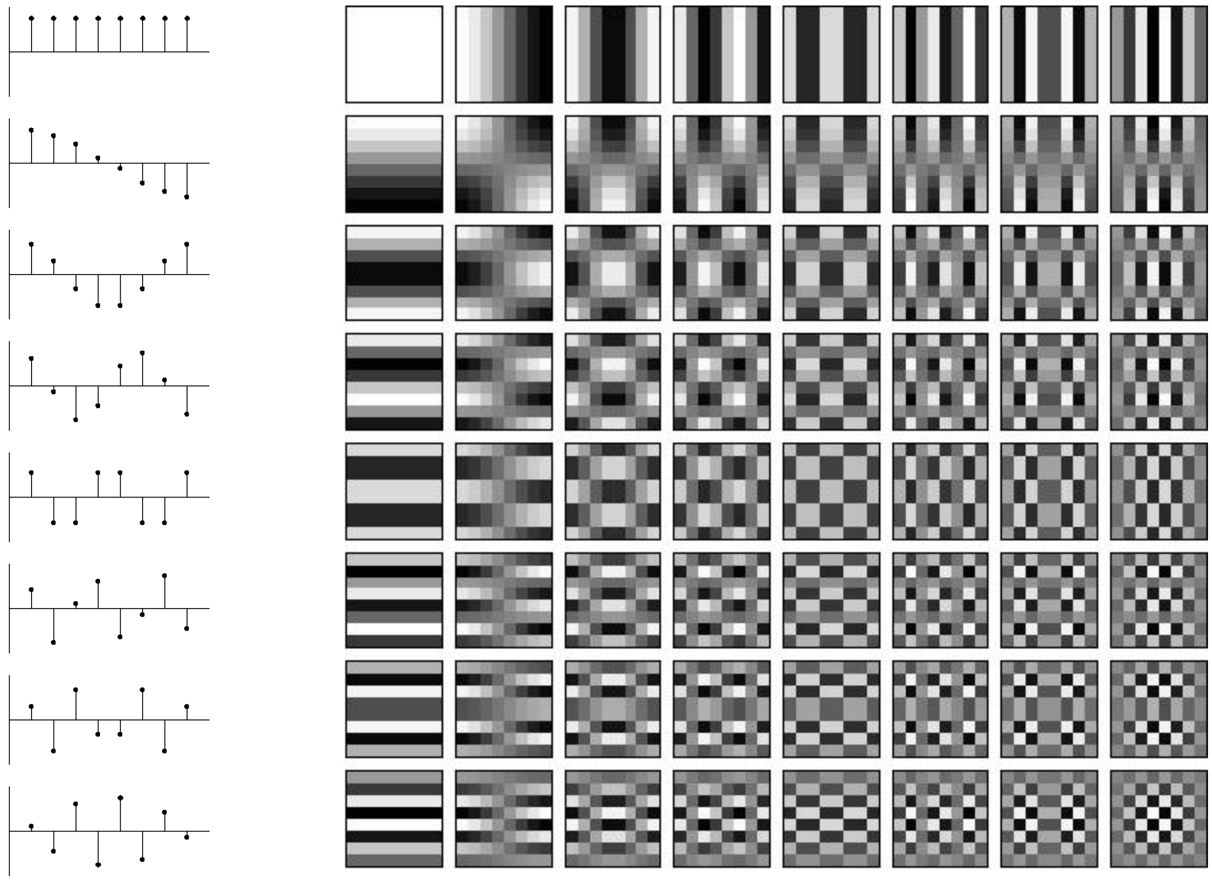
Visualizing the DFT(circular shifting)

- The FT is centered about the origin
- But the DFT is centered about N/2
- We need to correct with a circular shift operation
- Or, multiply by $(-1)^k$ prior to taking the transform



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

DFT basis images



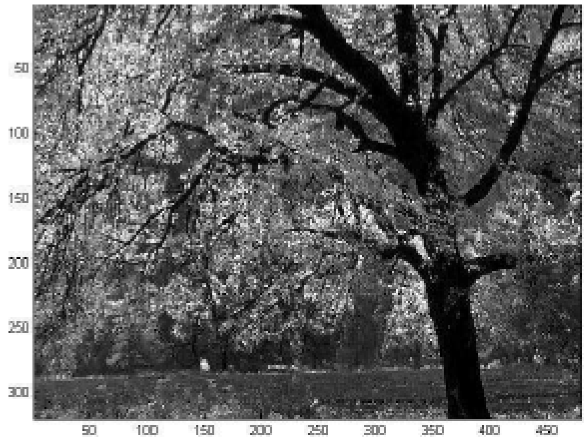
Procedure for Filtering in the Frequency Domain

1. Multiply the input image by $(-1)^{x+y}$ to center the transform
2. Compute the DFT *F(u,v)* of the resulting image
3. Multiply F(u,v) by a filter G(u,v)
4. Computer the inverse DFT transform h*(x,y)
5. Obtain the real part h(x,y) of 4
6. Multiply the result by $(-1)^{x+y}$

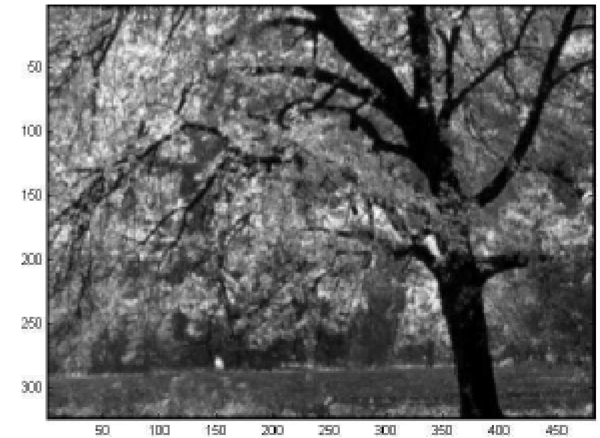
Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Filtering Example
Smooth an Image with a Gaussian Kernel

- Traditionally, we would just convolve the image with the a gaussian kernel



$$* \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} /256 =$$

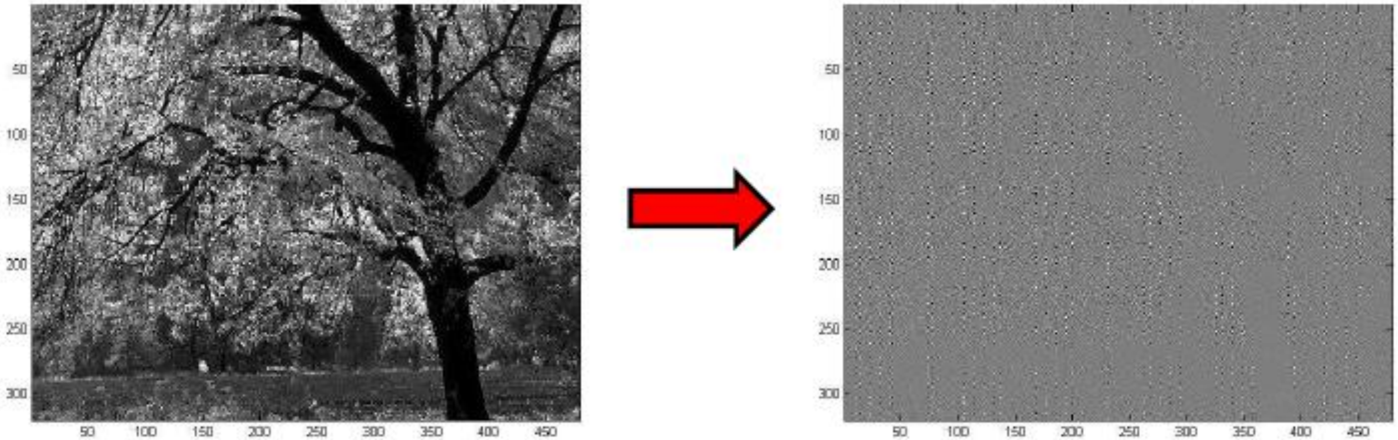


- Instead, we will perform multiplication in the frequency domain to achieve the same effect

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

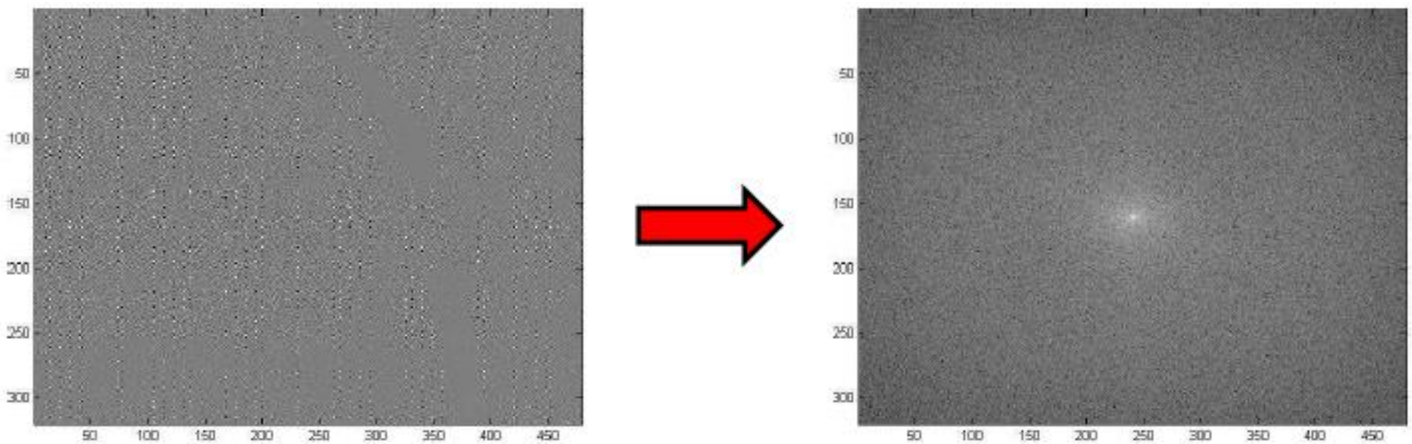
Filtering Example
Smooth an Image with a Gaussian Kernel

1. Multiply the input image by $(-1)^{x+y}$ to center the transform



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

2. Compute the DFT *F(u,v)* of the resulting image



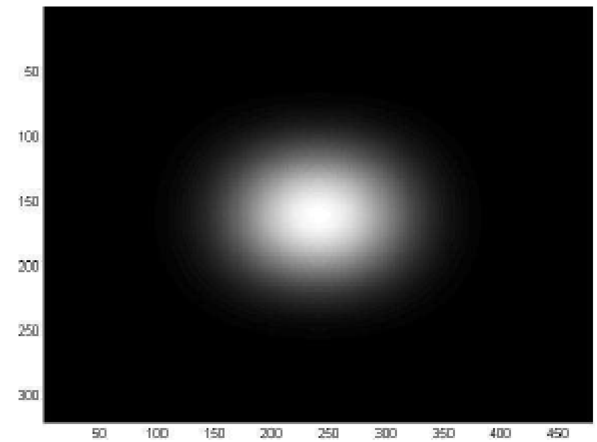
log transform

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

3. Multiply F(u,v) by a filter G(u,v)

$$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} / 256$$

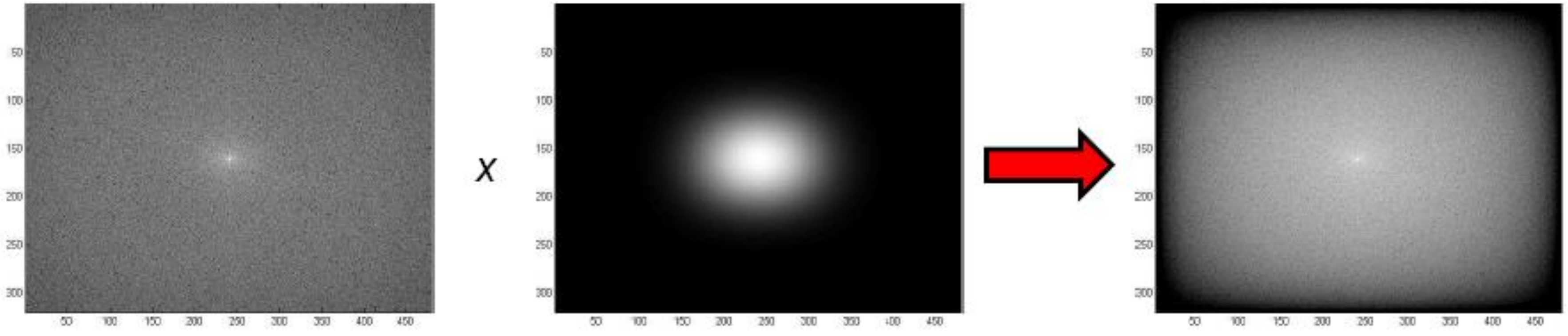
g(x,y)



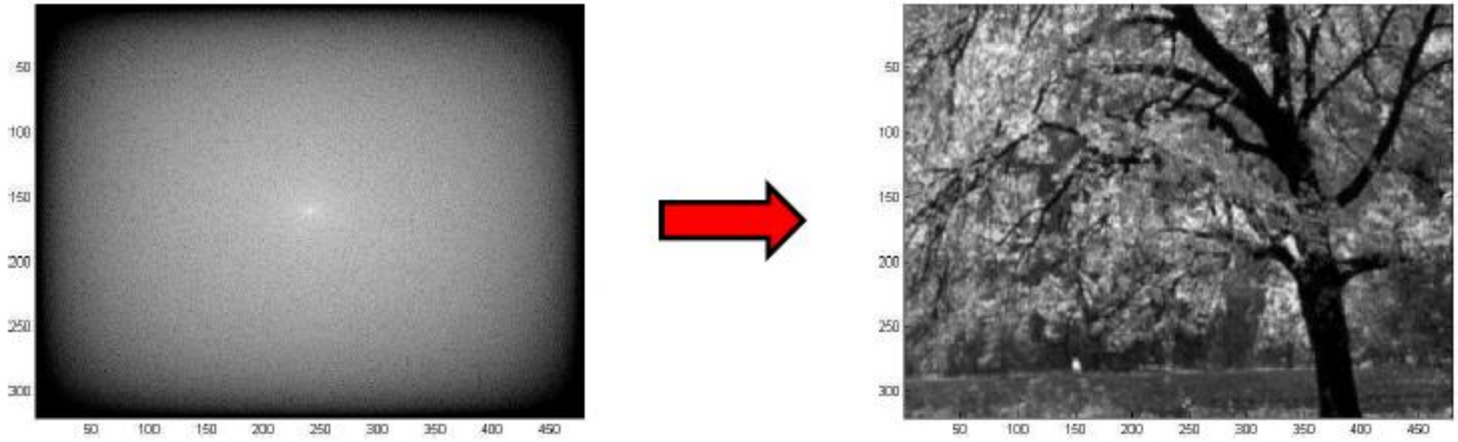
G(u,v)

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

3. Multiply F(u,v) by a filter G(u,v)



4. Computer the inverse DFT transform $h^*(x,y)$
5. Obtain the real part $h(x,y)$ of 4
6. Multiply the result by $(-1)^{x+y}$

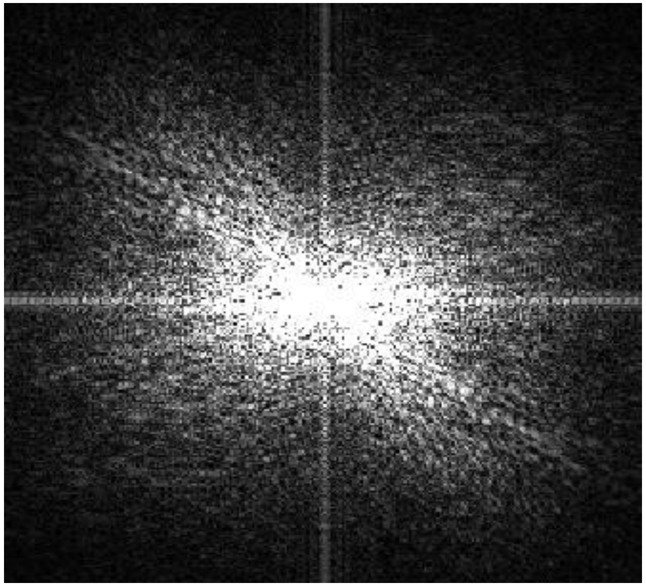


Example


```
img=imread('lena.bmp','bmp');
subplot(121);imshow(img);
title('original image ')
fimg=fftshift(fft2(img));
subplot(122); imshow(abs(fimg)/10000)
title(' transformed image ')
```

original image

transformed image



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Discrete Cosine Transform (DCT)

- DCT expresses a sequence of finite data points in terms of a sum of cosine functions oscillating at different frequencies.
- These cosine functions are treated as its basis function.
- Fast algorithm exists.
- Most popular in image compression application because of its energy compaction property.
- Adopted in JPEG.
- The periodicity implied by DCT implies that it causes less blocking effect than DFT.
- Can be implemented by 2*n* points FFT.

Contd..

Transform kernel of two-dimensional DCT is

$$C(x, y, u, v) = \alpha(u) \cdot \alpha(v) \cdot \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cdot \cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

where

$$\alpha(0) = \sqrt{\frac{1}{N}}, \qquad \alpha(i) = \sqrt{\frac{2}{N}}, \; 1 \le i \le N$$

Clearly, the kernel for the DCT is both separable and symmetric, and hence the DCT may be implemented as a series of one dimensional DCTs.

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

Forward transform

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

Inverse Transform

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

where $u, v, x, y = 0, 1, 2, \ldots, N\text{-}1$

Forward and inverse transformations are same

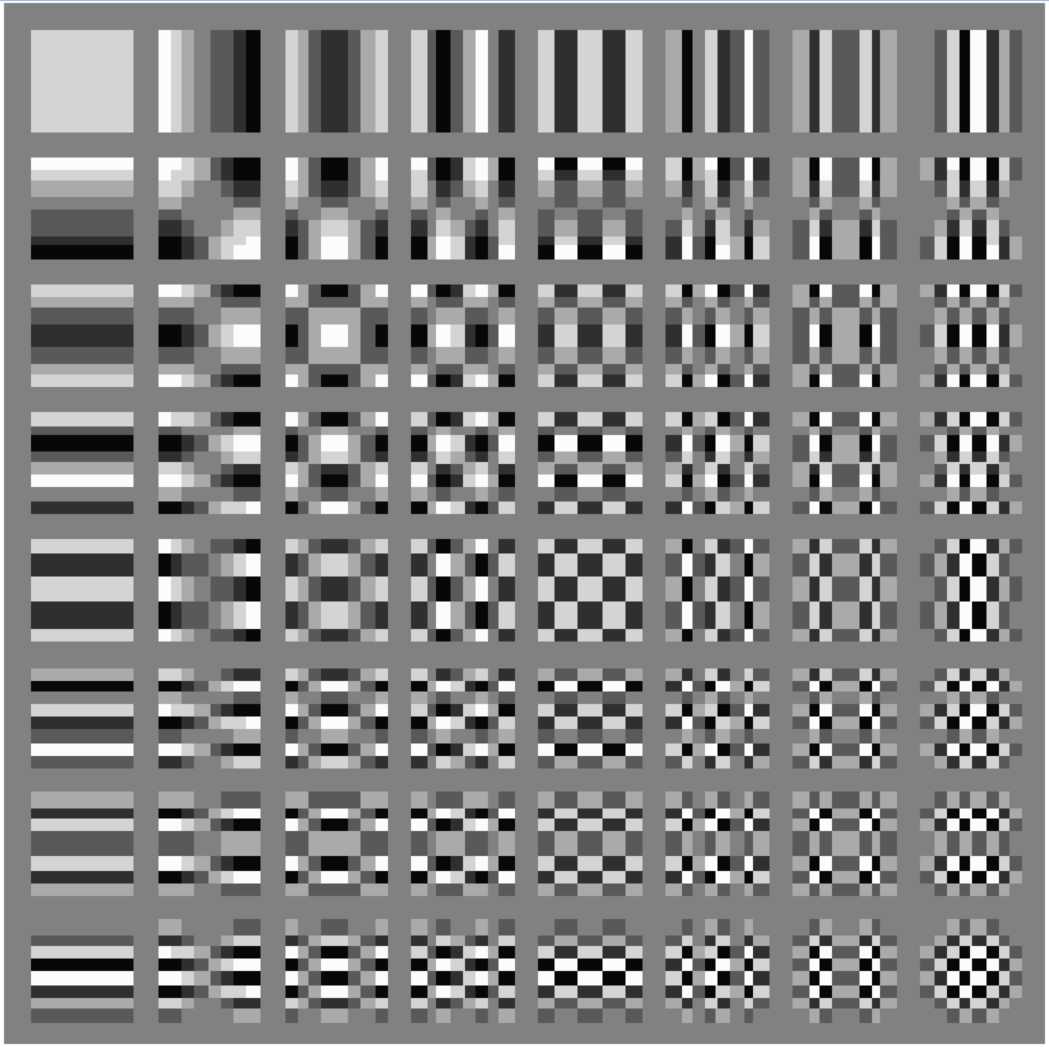
Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

```
clc;
clear
N=8
x=(0:N-1)';
C=cos((2*x+1)*x'*pi/(2*N))*sqrt(2/N);
C(:,1)=C(:,1)/sqrt(2);
```

Kernel matrix to generate DCT Basis Images

0.353553	0.490393	0.46194	0.415735	0.353553	0.277785	0.191342	0.097545
0.353553	0.415735	0.191342	-0.09755	-0.35355	-0.49039	-0.46194	-0.27779
0.353553	0.277785	-0.19134	-0.49039	-0.35355	0.097545	0.46194	0.415735
0.353553	0.097545	-0.46194	-0.27779	0.353553	0.415735	-0.19134	-0.49039
0.353553	-0.09755	-0.46194	0.277785	0.353553	-0.41573	-0.19134	0.490393
0.353553	-0.27779	-0.19134	0.490393	-0.35355	-0.09755	0.46194	-0.41573
0.353553	-0.41573	0.191342	0.097545	-0.35355	0.490393	-0.46194	0.277785
0.353553	-0.49039	0.46194	-0.41573	0.353553	-0.27779	0.191342	-0.09755

Basis images of an 8x8 DCT



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Properties of DCT

- Separability

  2D DCT/ IDCT expression shows that they are separable.
  So they can be implemented as two 1-D DCT/IDCT

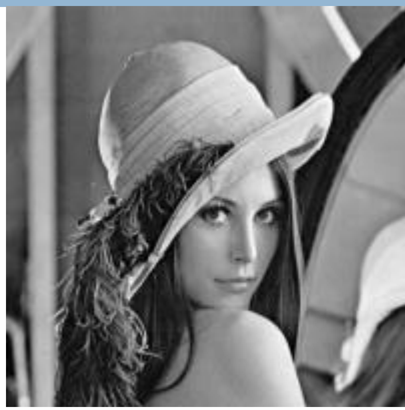
- Fast DCT

  FDCT is possible in the same manner as FFT

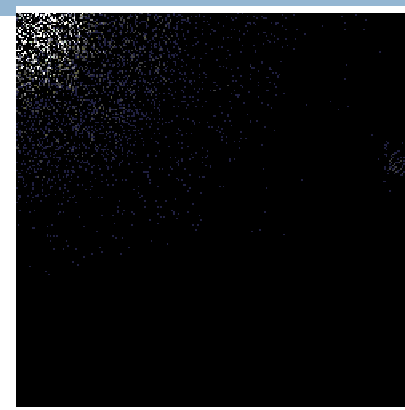
- Periodicity

  Magnitude of DCT coefficient is periodic with period 2N

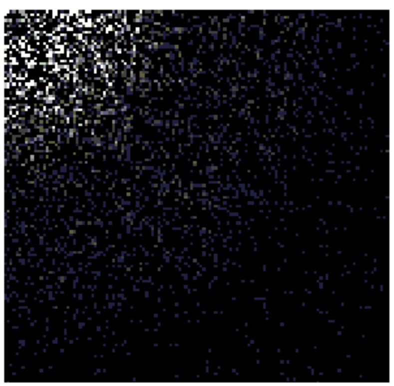
Example



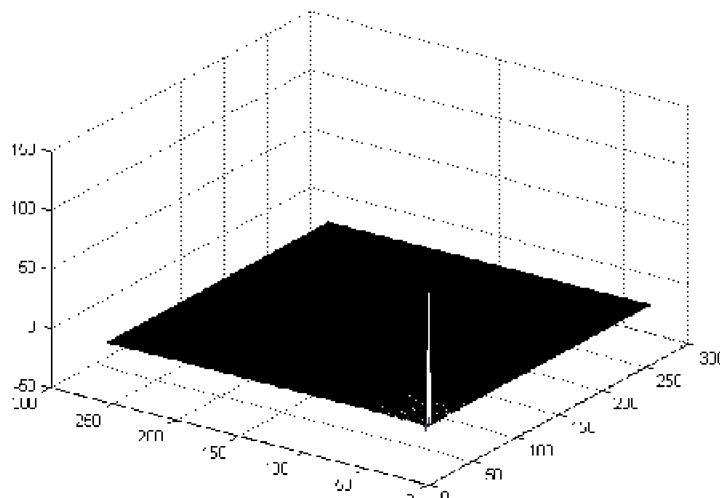
Original Image



DCT Coefficients



DCT Coefficients (Zoomed)



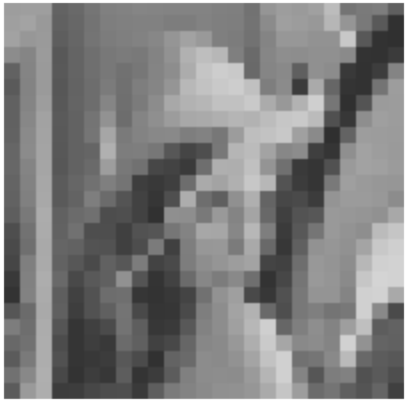
Mesh Plot

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Image Reconstruction in DCT

Original image

Reconstructed image



Using only 1 DCT coefficient

Using 3 DCT coefficients

Coefficients are selected out of 8x8 window

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

Using 6 DCT coefficients

Using 15 DCT coefficients

Using 10 DCT coefficients

Using 64 DCT coefficients

Karhunen-Loeve Transform

- K-L transform is a linear combination of orthogonal functions of ensembles of input images.
- Basis functions are image dependent
- No fast algorithm exists
- Usually used for comparison
- Transform kernel is not fixed unlike other transforms
- Based on second order statistical properties of image
- Also called Hotelling transform or method of principal components

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

- Covariance matrix of a population of vectors

$$X = \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix}$$

$$C_x = E\left\{ (X - \mu_x)(X - \mu_x)^T \right\}$$

$$\mu_x = \frac{1}{M} \sum_{k=1}^{M} X_k$$

$$C_x \Rightarrow \text{real \& symmetric}$$

So, Orthogonal vectors of C_x can be calculated

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

- Calculate eigenvectors and eigen values of C_x
- Sort eigen vectors depending on decreasing values of eigen values
- Select some of the eigenvectors
- These eigenvectors are considered as transformed matrix
- Calculate the transformed images

$$Y = A\left(X - \mu_x\right)$$

- Y is called as principal component coefficients.
- This transformation is called K-L transform

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Properties of Y

$$\mu_y = 0$$

$$C_y = AC_xA^T$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Example

Input matrix

$$x = \left\{ \begin{pmatrix} 3 \\ 4 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \end{pmatrix} \begin{pmatrix} 4 \\ 5 \end{pmatrix} \begin{pmatrix} 5 \\ 4 \end{pmatrix} \begin{pmatrix} 5 \\ 5 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix} \begin{pmatrix} 6 \\ 5 \end{pmatrix} \right\}$$

Mean Vector

$$\mu_x = \begin{pmatrix} 4.5 \\ 4.5 \end{pmatrix}$$

Contd..

$$C_x = \begin{pmatrix} 0.75 & 0.375 \\ 0.375 & 0.75 \end{pmatrix}$$

Compute Eigen values

$$(0.75-\lambda)^2 = (0.375)^2$$

$$\Rightarrow 0.75-\lambda = \pm 0.375$$

$$\Rightarrow \lambda = 0.75 \pm 0.375$$

$$\lambda_1 = 1.125$$

$$\lambda_2 = 0.375$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

We get two Eigen vectors

$$e_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$e_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Transform Matrix

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

Contd.

- Reconstructed $$X = A^T Y + \mu_X$$

- K no of Eigen vectors

$$Y = A_k \left(X - \mu_X \right)$$

$$\hat{X} = A_k^T Y + \mu_X$$

$$A_k = k \times n$$

$$Y = k \times 1$$

$$X = n \times 1$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

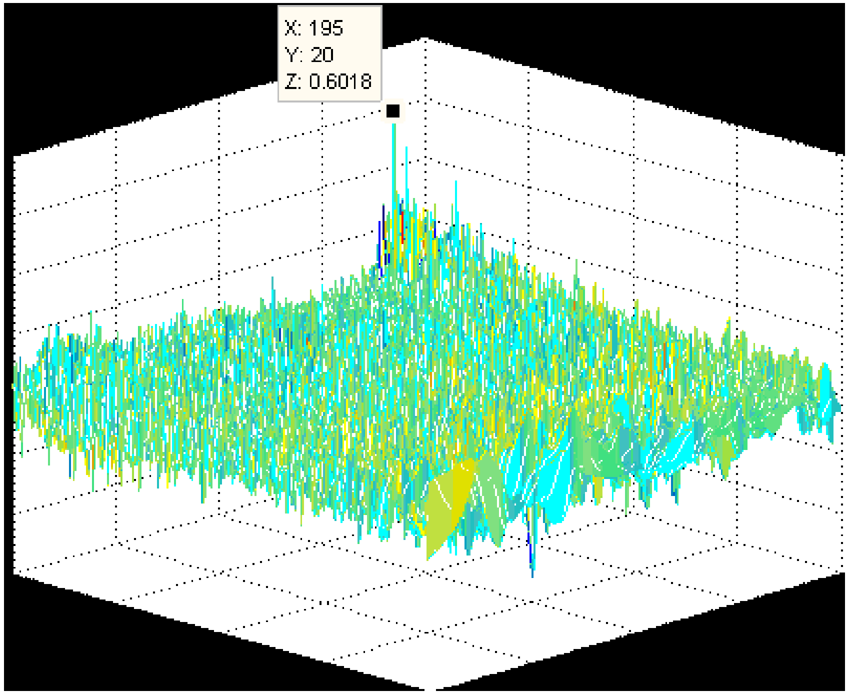
Properties of K-L Transform

- Decorrelation
  - $E[\,\underline{y}\,\underline{y}^H\,] = E[\,(U^H x)\,(U^H x)^H\,] = U^H\,E[\,x\,x^H\,]\,U = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$
  - Note: Other matrices (unitary or nonunitary) may also decorrelate the transformed sequence
- Minimizing MSE under basis restriction
  - If only allow to keep *m* coefficients for any $1 \le m \le N$, what's the best way to minimize reconstruction error?
  - → *Keep the coefficients w.r.t. the eigenvectors of the first m largest Eigen values*

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Principal Component Coefficients

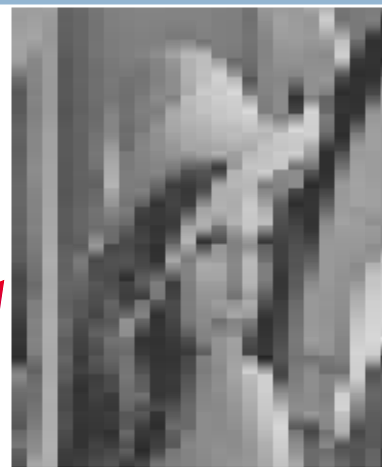
Lena image



Energy compaction in lena image

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Image Reconstruction in K-L Transform

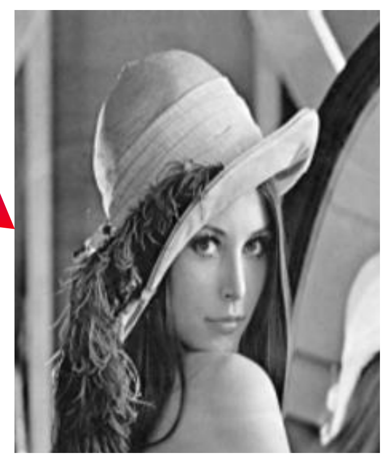


1 PC

10 PCs

20 PCs

Original image



50 PCs

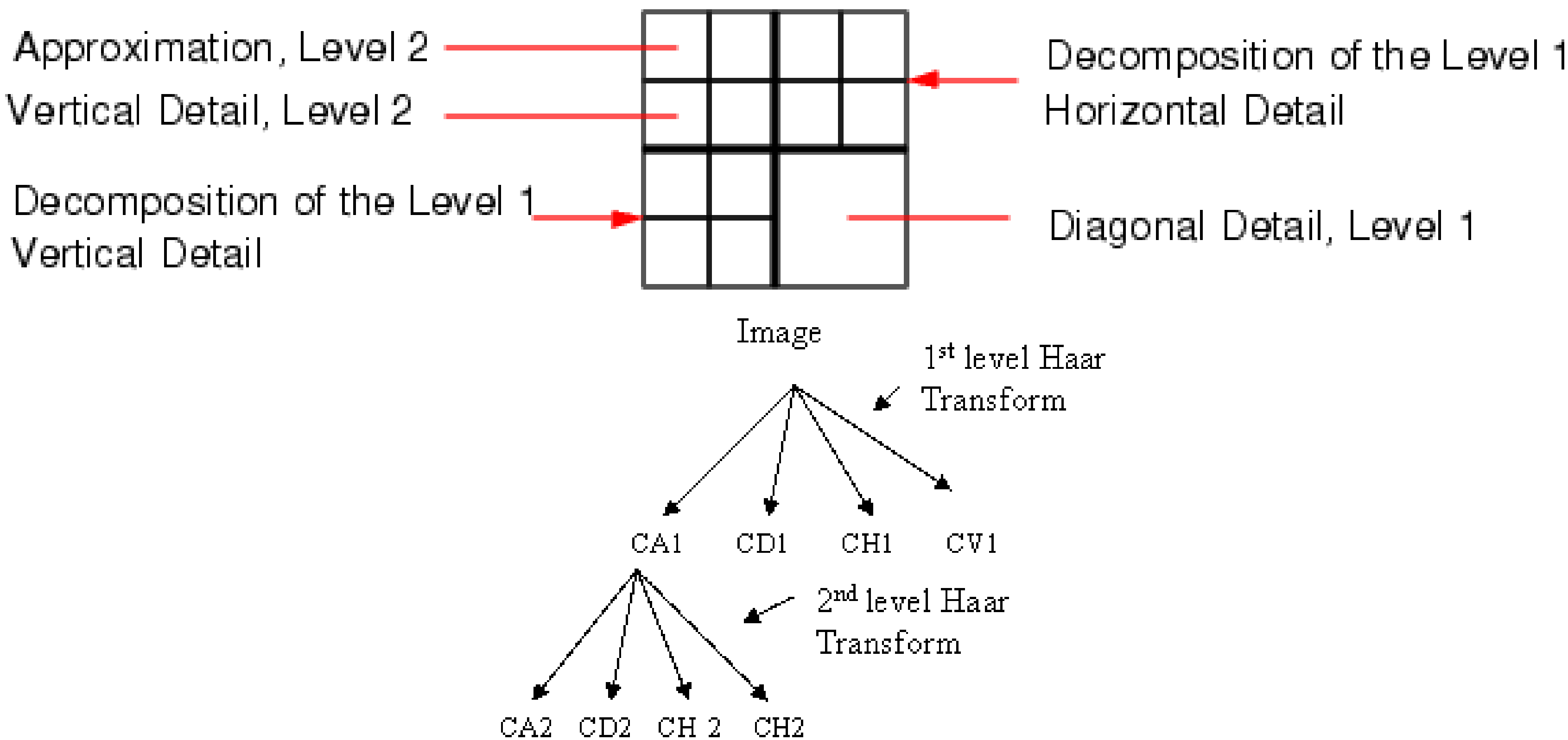
All PCs

Haar Wavelet Transform

- One dimensional transformation on each row followed by one dimensional transformation of each column.
- Extracted coefficients would be
  - Approximation
  - Vertical
  - Horizontal
  - Diagonal
- Approximation coefficients are further decomposed into the next level
- 4 level decomposition is used

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..



Graphical Representation - Wavelet decomposition
(level = 2)

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Approach

- For a 2X2 matrix

$$x = \overrightarrow{\begin{pmatrix} a & b \\ c & d \end{pmatrix}} \qquad x = \begin{pmatrix} a+b & a-b \\ c+d & c-d \end{pmatrix}\Big\downarrow$$

$$y = \frac{1}{2}\begin{pmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b-c+d \end{pmatrix}$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Example

2	2	9	5
2	0	5	2
6	7	4	7
3	4	8	8

Column wise Summation

4	14	0	4
2	7	2	3
13	11	-1	-3
7	16	-1	0

Row wise Summation

6	21	2	7
20	27	-2	-3
2	7	-2	1
6	-5	0	-3

Finding Average

Approximation | Horizontal

3	10.5	1	3.5
10	13.5	-1	-1.5
1	3.5	-1	0.5
3	-2.5	0	-1.5

Vertical | Diagonal

Properties of Haar transform

- The Haar transform is real and orthogonal. Therefore,

  $\mathbf{H_r = H_r^*}$

- The Haar transform is a very fast transform.
- On an Nx1 vector it can be implemented in O(N) operations
- The Haar transform has poor energy compaction for Images

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

- %Initialization
- H=[1]; NC=1/sqrt(2);%normalization constant
  LP=[1 1]; HP=[1 -1]; % iteration from H=[1]
- for i=1:Level
  H=NC*[kron(H,LP);kron(eye(size(H)),HP)];
- end
- HaarTransformationMatrix=H;

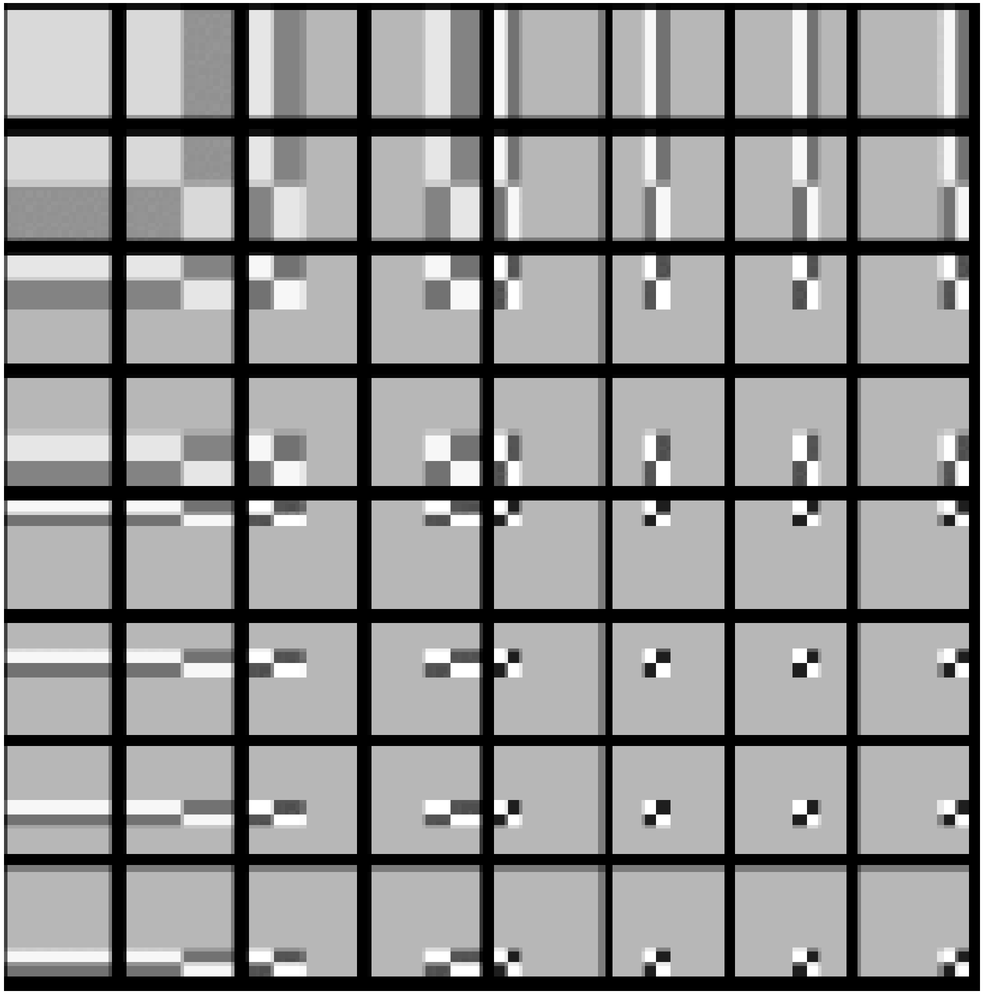
Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

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

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

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Basis images of Haar Transform



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

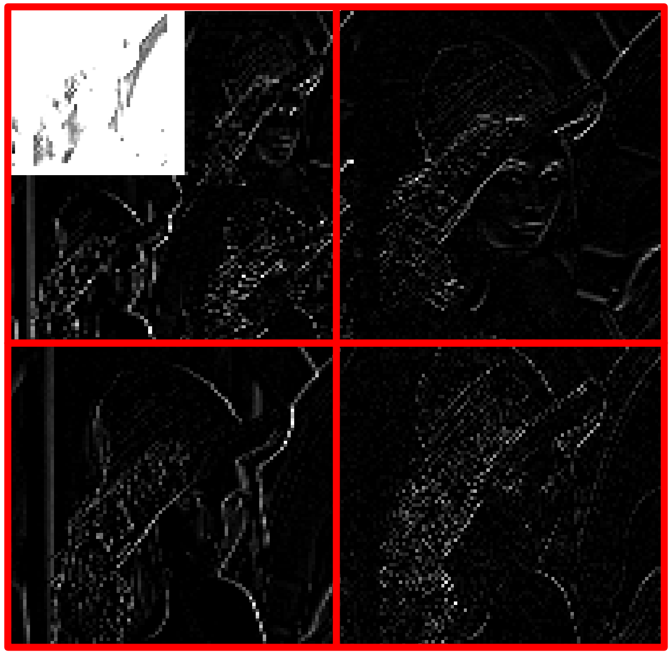
1-step Haar Decomposition

Original image

Haar transformed image

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

2-step Haar Decomposition



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Discrete Walsh Transform (1-D)

$$\underbrace{g(x,u) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}}_{1-D\ Forward\ \text{Kernel}}$$

where,

$N \rightarrow$ No. of Samples

n $\rightarrow$ No. of bits needed to represent x and u

$\mathrm{b_k}(z) \rightarrow k^{th}$ bit in digital representation of z

$$\underbrace{W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}}_{1-D\ \ \text{Forward}\ \ \text{Transform}}$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

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

$$\underbrace{f(x) = \sum_{x=0}^{N-1} W(u) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}}_{1-D \text{ Inverse Transform}}$$

Difference lies in Multiplicative Factor

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Discrete Walsh Transform (2-D)

$$\underbrace{g(x,y,u,v)=\frac{1}{N}\prod_{i=0}^{n-1}(-1)^{\{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)\}}}_{2-D\ Forward\ \text{Kernel}}$$

$$\underbrace{W(u,v)=\frac{1}{N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}f(x,y)\prod_{i=0}^{n-1}(-1)^{\{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)\}}}_{2-D\ \text{Forward Transform}}$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

$$\underbrace{h(x,y,u,v)=\frac{1}{N}\prod_{i=0}^{n-1}(-1)^{\{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)\}}}_{2-D\ \text{Inverse Kernel}}$$

$$\underbrace{f(x,y)=\frac{1}{N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}W(u,v)\prod_{i=0}^{n-1}(-1)^{\{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)\}}}_{2-D\ \text{Inverse Transform}}$$

Property

- **Separable**

$$W(u) = \frac{1}{2}[W_{even}(u) + W_{odd}(u)]$$

$$W(u+M) = \frac{1}{2}[W_{even}(u) - W_{odd}(u)]$$

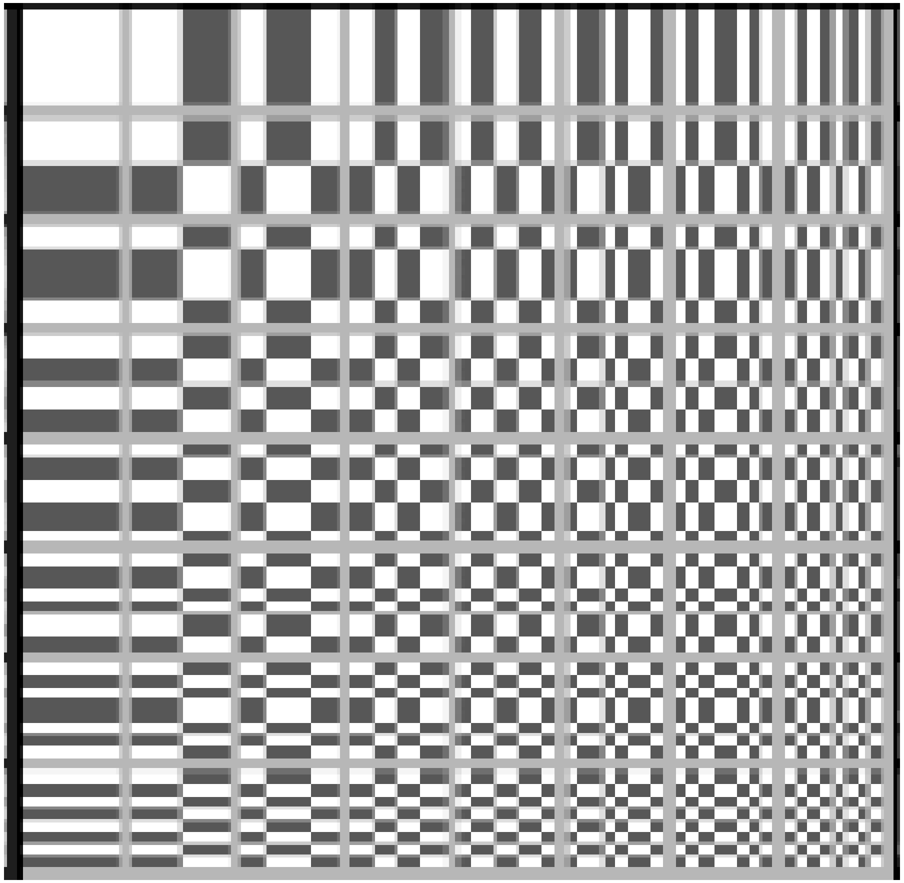
$$where,$$

$$u = 0,1,\ldots,\frac{N}{2} - 1$$

$$M = \frac{N}{2}$$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Basis images of Walsh Transform



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Hadamard Transfrom

$$\underbrace{g(x,y,u,v)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1}\{b_i(x)b_i(u)+b_i(y)b_i(v)\}}}_{2D\ \text{Forward Kernel}}$$

$$\underbrace{h(x,y,u,v)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1}\{b_i(x)b_i(u)+b_i(y)b_i(v)\}}}_{2D\ \text{Inverse Kernel}}$$

Hadamard Matrix

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

$u \longrightarrow$

$x \downarrow$	0(0)	1(7)	2(3)	3(4)	4(1)	5(6)	6(2)	7(5)
0	+	+	+	+	+	+	+	+
1	+	-	+	-	+	-	+	-
2	+	+	-	-	+	+	-	-
3	+	-	-	+	+	-	-	+
4	+	+	+	+	-	-	-	-
5	+	-	+	-	-	+	-	+
6	+	+	-	-	-	-	+	+
7	+	-	-	+	-	+	+	-

sign changes are not ordered (shown in backet)

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Hadamard transform

- Transform matrices can be recursively generated

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

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

$$H_3 = H_1 \otimes H_2, \;\; H_2 = H_1 \otimes H_1$$

Note: Hadamard Coefficients need reordering to concentrate energy

Contd..

- N=2

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

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

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Ordered Hadamard Transformation

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

$$where,$$

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

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

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

.

.

.

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

$b_i(u)$* is changed to *$p_i(u)$

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Contd..

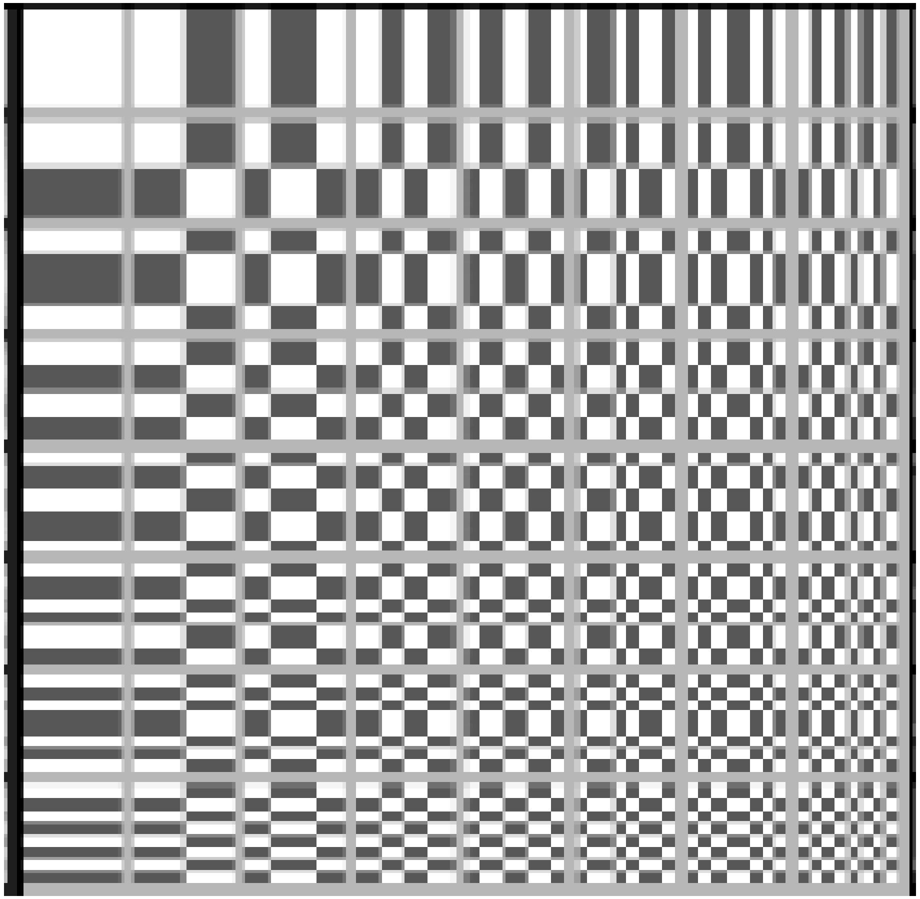
N = 8

$u \rightarrow$; $x \downarrow$

	0	1	2	3	4	5	6	7
0	+	+	+	+	+	+	+	+
1	+	+	+	+	-	-	-	-
2	+	+	-	-	-	-	+	+
3	+	+	-	-	+	+	-	-
4	+	-	-	+	+	-	-	+
5	+	-	-	+	-	+	+	-
6	+	-	+	-	-	+	-	+
7	+	-	+	-	+	-	+	-

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Basis images of Hadamard Transform



Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Properties of Hadamard Transform

- Unlike Other Transform, the elements of the basis vectors of the Hadamard transform take only the binary values +1 , -1, therefore well suited for DSP.
- Hadamard transform H is real, symmetric, and orthogonal
- Hadamard transform is a fast transform. The one dimensional transform can be implemented in $O(N \log_2 N)$ additions and subtractions

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Similarity between Walsh and Hadamard Transform

- Basis images of Walsh and Hadamard transforms are same.
- So Hadamard transform is also known as Walsh transform or Hadamard-Walsh transform.

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Image Reconstruction

Original image

Reconstructed image

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela

Image Reconstruction

Original image

Reconstructed image

Ratnakar Dash,
Assistant Professor, CSE dept. of NIT Rourkela


Thank You!