# Image Transforms and Compression

Jayanta Mukhopadhyay
Dept. of CSE,
IIT Kharagpur

# Image Transform

$$f(x,y) = \sum_j \sum_i \lambda_{ij} b_{ij}(x,y)$$

- Image in continuous form: $f(x,y)$: A 2-D function, where $(x,y)$ in $R^2$.

Properties of basis functions can be extended in the analysis.

- Let $B$ be a set of basis functions:

$B=\{b_i(x,y) \mid i=..,-1,0,1,2,3,\ldots.\}$, $b_i(x,y)$ in $R$ or $C$.

- Let $f(x,y)$ be expanded using $B$ as follows:

$$f(x,y) = \sum_i \lambda_i b_i(x,y)$$

Coefficients of transform

The **transform** of $f$ w.r.t. $B$ is given by $\{\lambda_i | i = \cdots. -1,0,1,2,3, \ldots\}$.

Indexing may be multidimensional say, $\lambda_{ij}$.

# Orthogonal Expansion and 1-D Transforms

$$f(x) = \sum_i \lambda_i b_i(x)$$

- Inner product: $$< f, g >= \int f(x) g^*(x) dx$$

- Orthogonal expansion: If $B$ satisfies :

$$\langle b_i, b_j \rangle = 0, \ for\ i \neq j$$
$$= c_i \ Otherwise\ (for\ i = j), where\ c_i > 0$$

- Transform coefficients in O.E.: $$\lambda_i = \frac{1}{c_i} \langle f, b_i \rangle$$
  - $c_i$=1→ orthonormal expansion. $$\lambda_i = \langle f, b_i \rangle$$

Forward transform

- Inverse transform: $$f(x) = \int_{i=-\infty}^{\infty} \lambda_i b_i(x) di$$

# Fourier transform

Complete base

$$B = \{e^{j\omega x} \mid -\infty < \omega < \infty\}$$

Unit impulse function

Orthogonality: $$\int_{-\propto}^{\propto} e^{j\omega x}\,dx = \begin{cases} 2\pi\delta(x), & for\ \omega = 0 \\ 0, & otherwise. \end{cases}$$

Fourier Transform: $$\mathcal{F}\big(f(x)\big) = \hat{f}(j\omega) = \int_{-\propto}^{\propto} f(x)e^{-j\omega x}\,dx$$

Inverse Transform: $$f(x) = \frac{1}{2\pi}\int_{-\propto}^{\propto} \hat{f}(j\omega)\,e^{j\omega x}\,d\omega$$

Full reconstruction

$$e^{-j\omega x} = \cos(\omega x) - j\sin(\omega x)$$

$$\hat{f}(j\omega) = \int_{-\infty}^{\infty} f(x)(\cos(\omega x) - j\sin(\omega x))\,dx$$

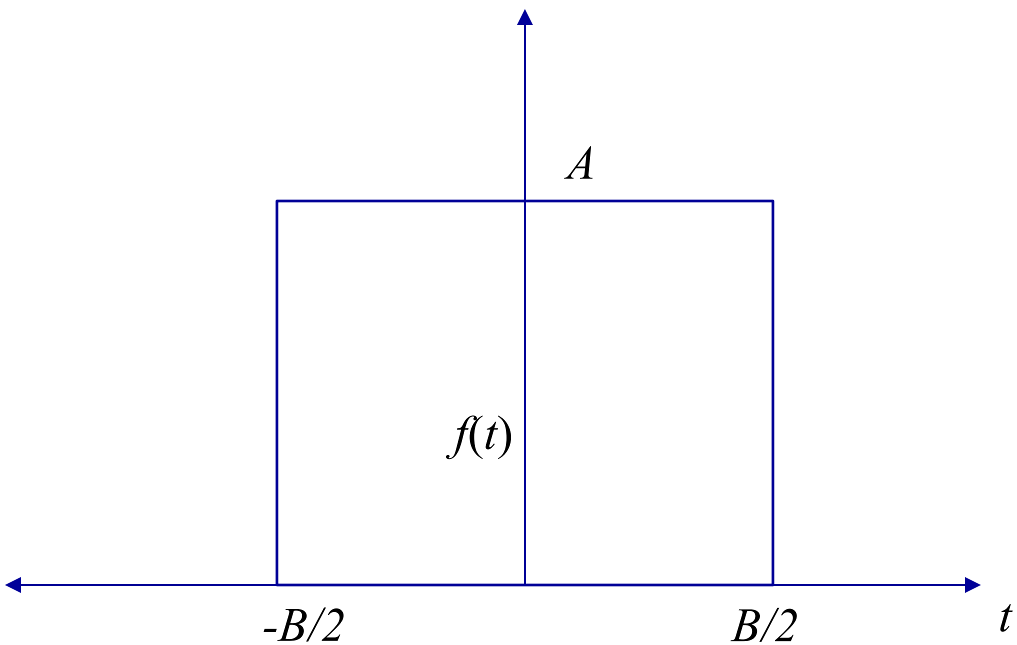
$$C = \{\cos(\omega x) \mid -\infty < \omega < \infty\} \qquad S = \{\sin(\omega x) \mid -\infty < \omega < \infty\}$$

Orthogonal

But not complete!

# Fourier transform of a square pulse

- $f(t) = A,\ -B/2 \leq t \leq B/2$




*sinc(ωB/2)*

$$F(j\omega) = AB\,\frac{\sin\left(\frac{\omega B}{2}\right)}{\frac{\omega B}{2}}$$

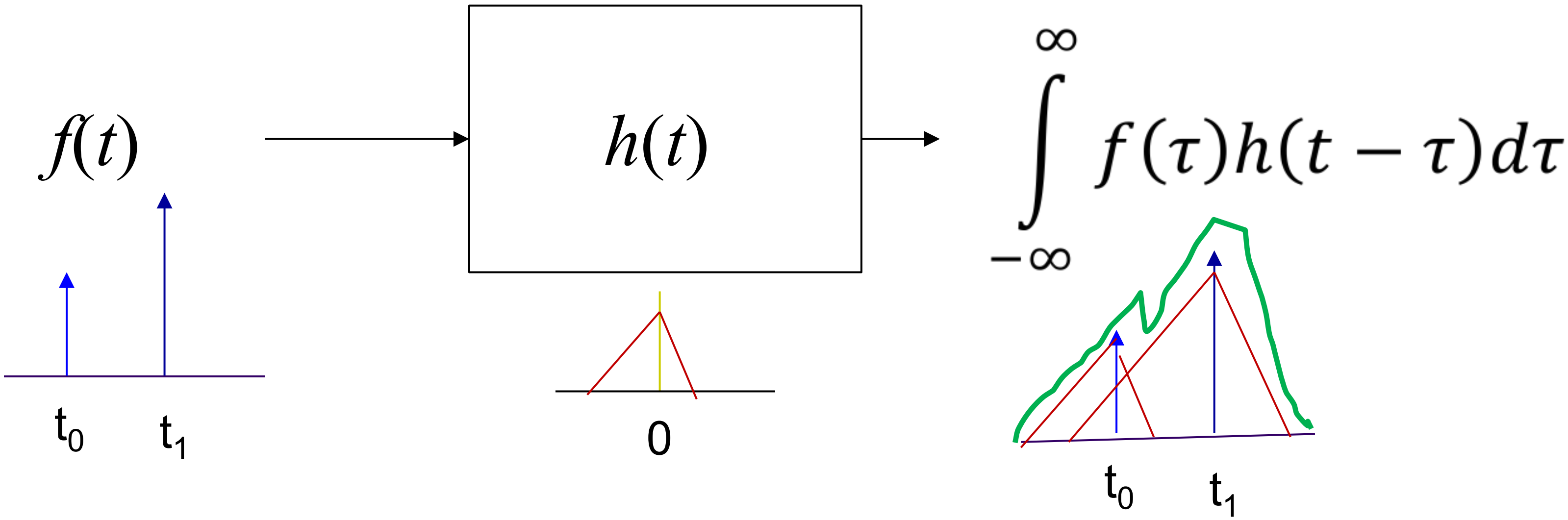
# Fourier transform of a Gaussian Pulse

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}$$

$$G(\omega) = e^{-\frac{\omega^2\sigma^2}{2}}$$

- Transform is also a Gaussian function.
- Standard deviation in the Fourier domain (angular frequency) is reciprocal of that in the time domain.

# Convolution and Fourier Transform




- Convolved output: Sum of scaled and shifted impulse responses.

$$F\big(f(t) * h(t)\big) = \int \left( \int f(\tau)\, h(t-\tau) d\tau \right) e^{-j\omega t} dt \quad = \int f(\tau) \int h(t-\tau) e^{-j\omega t} dt\, d\tau$$

$$= \int f(\tau)\, H(j\omega) e^{-j\omega\tau} d\tau$$

$$= H(j\omega) F(j\omega)$$

# Fourier transform of unit impulse

- Definition and properties of unit infinite impulse

$$\delta(t) = \infty, t = 0$$
$$= 0, \quad \text{otherwise}$$

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt = f(t_0)$$

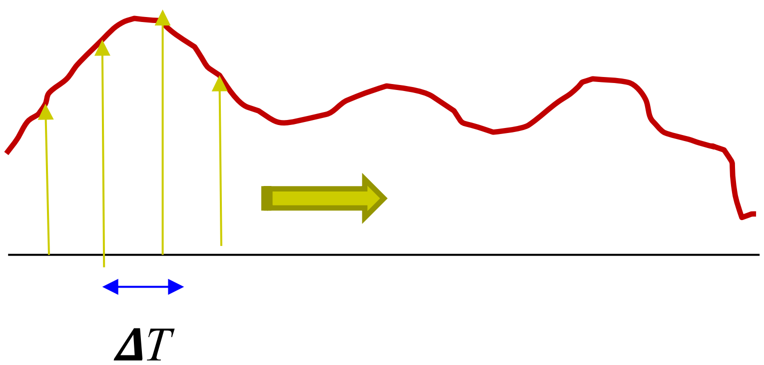
$$F\big(\delta(t)\big) = \int_{-\infty}^{\infty} \delta(t)\, e^{-j\omega t} dt \qquad = e^{-j\omega.0} = 1$$

$$F\big(\delta(t - T)\big) = e^{-j\omega T} \quad \xLeftrightarrow[\text{Duality}]{} \quad F\big(e^{j\omega_0 t}\big) = \delta(\omega - \omega_0)$$

Train of impulses: $\sum \delta(t - n\Delta T) \xrightarrow{\mathcal{F}} \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta(\omega - \frac{2\pi n}{\Delta T})$

Fourier series of a period $\Delta T$
with unit impulse in a period

# Fourier transform of a sampled function



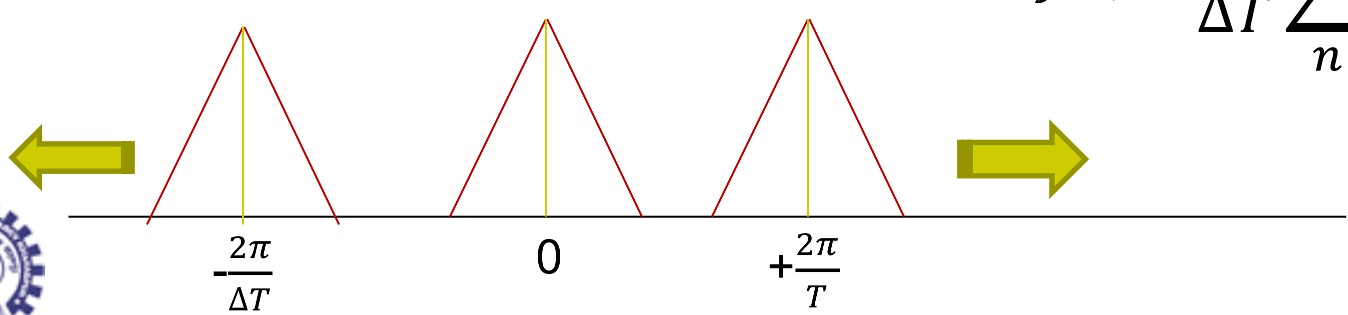

$$f_s(t) = \sum f(t)\delta(t - n\Delta T)$$

$$= f(t) \sum \delta(t - n\Delta T)$$

- $f_s(t) = f(t)$, for $t = n\,\Delta T$, n: an integer.

  $= 0$, otherwise

$$F(j\omega) * \frac{1}{\Delta T}\sum_n \delta(\omega - \frac{2\pi n}{\Delta T})$$



# Even and odd functions

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2}$$

$$\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

- Even: $f(-x)=f(x)$ for all $x$.
- Odd: $f(-x)=-f(x)$ for all $x$. $\rightarrow f(0)=0$.
- For even $f(x)$ :

$$\int_{-\infty}^{\infty} f(x)(\sin(\omega x))\,dx = 0$$

- For odd $f(x)$ :

$$\int_{-\infty}^{\infty} f(x)(\cos(\omega x))\,dx = 0$$

- Full reconstruction possible with cosines (sines) only if it is even (odd).

# Discrete representation

- Discrete representation of a function:

$$f(n) = \{f\ (nX_0) | n \epsilon \mathbb{Z}\}$$

Set of integers

Sampling interval

- Can be considered as a vector in an infinite dimensional vector space.
- In our context, it is of a finite dimensional space, e.g. $\{f(n),\ n=0,1,..N-1\}$, or
- $\boldsymbol{f}=[f(0)\ f(1)\ \dots\ f(N-1)]^T$.

# Discrete Linear Transform: A general form

- For $n$-dimensional vector $X$ any linear transform,
  - e.g. $Y_{m\text{x}1}=B_{m\text{x}n}X_{n\text{x}1}$
  - $X_{n\text{x}1}$: *A column vector of dimension n.*
  - $Y_{m\text{x}1}$: *A column vector of dimension m.*
  - $B_{m\text{x}n}$: *A matrix of dimension m*x*n.*
- Has inverse transform if $B$ is a square matrix and invertible.

# Basis vectors

- $B$ is the transformation matrix.
- Rows of $B$ are called basis vectors.

$$B = \begin{bmatrix} \boldsymbol{b}_0^{*T} \\ \boldsymbol{b}_1^{*T} \\ \vdots \\ \boldsymbol{b}_n^{*T} \end{bmatrix}$$

- $Y(i) = <\boldsymbol{b}_i^{*T}.\ X>$

dot product or inner product.

- Orthogonality condition:

$$\begin{aligned} < \boldsymbol{b}_i^{*T}.\boldsymbol{b}_j > &= 0\ if\ i \neq j \\ &= c_i, \qquad otherwise \end{aligned}$$

# Discrete Fourier Transform (DFT)

$$b_k(n) = \frac{1}{\sqrt{N}} e^{j2\pi \frac{k}{N} n}, \text{ for } 0 \leq n \leq N-1, \text{ and } 0 \leq k \leq N-1.$$

$F(k)$ ⟺ $$\hat{f}(k) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi \frac{k}{N} n} \text{ for } 0 \leq k \leq N-1.$$

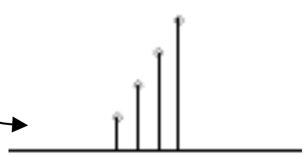
$$\hat{f}(N+k) = \hat{f}(k)$$

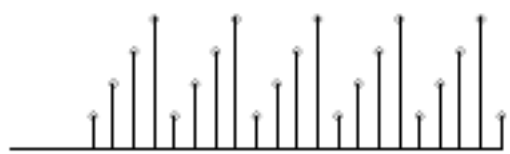
k/N: Normalized frequency

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}(k) e^{j2\pi \frac{k}{N} n} \text{ for } 0 \leq n \leq N-1.$$

Fundamental frequency: $1/(NX_0)$

A single period





$f(n+N)=f(n)$

DFT: Fourier series of a periodic function

# DFT as Fourier Series

$$f(n) = \frac{1}{N}\sum_{k=0}^{N-1} \hat{f}(k) e^{j2\pi\frac{k}{N}n} \text{ for } 0 \le n \le N-1.$$

$$\hat{f}(k) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi\frac{k}{N}n} \text{ for } 0 \le k \le N-1.$$

- For a periodic sequence of period $N$: $f(n+N)=f(n)$
- Sampling interval: $X_0$
- Period: $N \,.\, X_0$
  - Fundamental period: $1/(NX_0)$
  - Fourier series: Components of $k/(NX_0)$, $k$=0,1,2,…

$$F\left(\frac{k}{NX_0}\right) = \frac{1}{NX_0}\sum_{n=0}^{N-1} f(nX_0)\, e^{-j2\pi\frac{k}{NX_0}nX_0}\Delta x$$

$$F(k) = \frac{1}{N}\sum_{n=0}^{N-1} f(n)\, e^{-j2\pi\frac{k}{N}n}$$

In the DFT expression the normalization term (1/N) is adjusted.

For any integer $k$, $e^{j2\pi k} = 1$

# DFT properties

- Linearity: $DFT\,(a.f(n) + b(g(n)) = aF(k) + bG(k)$
- Circular time shifting $DFT(f(\langle n - n_0 \rangle_N) = e^{-j2\pi \frac{k}{N} n_0} F(k)$
- Periodicity:

$$F(k+N) = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi \frac{k+N}{N} n} = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi (\frac{k}{N}+1) n} = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi \frac{k}{N} n} = F(k)$$

- Symmetry

$$F(N-k) = F(-k) = \sum_{n=0}^{N-1} f(n)\, e^{j2\pi \frac{k}{N} n} = F^*(k) \Rightarrow F\left(\frac{N}{2} + m\right) = F^*(\frac{N}{2} - m)$$

Putting, $k=N/2+m$

- Duality
  - $DFT$ of $DFT$ of $x(n) = N.\, x(<-k>_N)$
- Energy preservation

$$\vec{x}.\vec{y}^* = \frac{1}{N} \hat{\vec{x}}.\hat{\vec{y}}^* \Rightarrow \|\vec{x}\|^2 = \vec{x}.\vec{x}^* = \frac{1}{N} \hat{\vec{x}}.\hat{\vec{x}}^* = \frac{1}{N} \left\|\hat{\vec{x}}\right\|^2$$

- Freq. Shifting $DFT(f(n) e^{j2\pi \frac{k_0}{N} n}) = F(< k - k_0 >_N)$

# Centering DFT

$$DFT(f(n)e^{j2\pi\frac{k_0}{N}n}) = F(< k - k_0 >_N)$$

- Multiplying $k_0$ th sinusoid shifts transform to $k_0$.
- Let $k_0=N/2$
  - ➔ $f(n)\ (-1)^n$
  - ➔ $F(<k - N/2>_N)$
  - Centers the Fourier transform bringing the 0 th freq. component in the center.
- A useful trick to center the transform
  - Multiply by $(-1)^n$ and then compute DFT.

# Fast Fourier Transform (FFT)

$$F(k) = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi \frac{k}{N} n}$$

Assume N even

$$F(k) = \sum_{m=0}^{\frac{N}{2}-1} f(2m)\, e^{-j2\pi \frac{k}{N}(2m)} + \sum_{m=0}^{\frac{N}{2}-1} f(2m+1)\, e^{-j2\pi \frac{k}{N}(2m+1)}$$

$$F(k) = \sum_{m=0}^{\frac{N}{2}-1} f(2m)\, e^{-j2\pi \frac{k}{N/2}(m)} + \sum_{m=0}^{\frac{N}{2}-1} f(2m+1)\, e^{-j2\pi \frac{k}{\frac{N}{2}}(m+1/2)}$$

$$F(k) = \sum_{m=0}^{\frac{N}{2}-1} f(2m)\, e^{-j2\pi \frac{k}{N/2}(m)} + e^{-j2\pi \frac{k}{\frac{N}{2}}(1/2)} \sum_{m=0}^{\frac{N}{2}-1} f(2m+1)\, e^{-j2\pi \frac{k}{\frac{N}{2}}(m)}$$

$$F(k) = \sum_{m=0}^{\frac{N}{2}-1} f(2m)\, e^{-j2\pi \frac{k}{N/2}(m)} + e^{-j2\pi \frac{k}{N}} \sum_{m=0}^{\frac{N}{2}-1} f(2m+1)\, e^{-j2\pi \frac{k}{\frac{N}{2}}(m)}$$

Cooley, I.W., and Tukey, I.W., "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Comp., vol. 19, pp. 297-301, April 1965.

# Fast Fourier Transform (FFT) (Cooley and Tukey (1965))

$$F(k) = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi \frac{k}{N} n}$$

Divide and conquer strategy

Exploiting other properties of DFT can be reduced to O(N log(N))

Assume N even

$$F(k) = \sum_{m=0}^{\frac{N}{2}-1} f(2m)\, e^{-j2\pi \frac{k}{N/2}(m)} + e^{-j2\pi \frac{k}{N}} \sum_{m=0}^{\frac{N}{2}-1} f(2m+1)\, e^{-j2\pi \frac{k}{\frac{N}{2}}(m)}$$

DFT of order N/2 of Even terms

DFT of order N/2 of Odd terms

Danielson-Lanczos lemma

Danielson, G.C. and Lanczos, C., "Some Improvements In Practical Fourier Analysis and Their Application to X-Ray Scattering from Liquids," J. Franklin Institute, vol. 233, pp. 365-380 and 435-452, 1942.

# Convolution Multiplication Property (CMP)

$$f(t) \longrightarrow \boxed{h(t)} \longrightarrow \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau$$

$$F(j\omega) \longrightarrow \boxed{H(j\omega)} \longrightarrow H(j\omega)F(j\omega)$$

CMP for Fourier Transform

$$\widehat{f \otimes h}(k) = \hat{f}(k)\hat{h}(k)$$

# CMP for DFT

CMP for DFT holds for circular convolution.

**Linear convolution**

$$f(n) \longrightarrow \boxed{h(n)} \longrightarrow \sum_{m=-\infty}^{\infty} f(m)h(n-m)$$

- Periodic convolution: Convolution between two finite sequences with periodic extension.
- It is defined if both have the same period, providing a periodic sequence with the same period.

**Circular Convolution**

$$\begin{aligned} f \circledast h(n) &= \sum_{m=0}^{N-1} f(m)h(n-m), \\ &= \sum_{m=0}^{n} f(m)h(n-m) + \sum_{m=n+1}^{N-1} f(m)h(n-m+N). \end{aligned}$$

# Circular Cross Correlation

- Cross correlation with periodic extensions of both the functions.

$$f \odot h(n) = \sum_{m=0}^{N-1} f(m)\, h(m+n)$$

$$f \odot h(n) = \sum_{m=0}^{N-n-1} f(m)\, h(m+n) + \sum_{m=N-n}^{N-1} f(m)\, h(m+n-N)$$

$$\widehat{f \odot h}(k) = \hat{f}(k).\hat{h}(k)^{*}$$

# Filtering in the transform domain

$$F(k) \longrightarrow \boxed{H(k)} \longrightarrow G(k)=H(k)\,F(k)$$

- Use sufficient 0 padding at the both end to make circular convolution equivalent to linear convolution
  - To take care of boundary effect.
  - The length of $f(n)$ and $h(n)$ should be the same.
- $H(k)$ usually provided as symmetric about the center.
  - $0^{th}$ freq. at the N/2 th element.
- Center $F(k)$ as $F_c(k)$ by multiplying f(n) with $(-1)^n$
- Obtain $G(k)= H(k) \,.\, F_c(k)$
- Multiply $G(k)$ by $(-1)^k$ and perform IDFT to get $g(n)$.

# DFT: A linear transform

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi\frac{kn}{N}} \quad for\ 0 \le k \le N-1$$

$$\begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi\frac{1}{N}} & \cdots & e^{-j2\pi\frac{N-1}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi\frac{N-1}{N}} & \cdots & e^{-j2\pi\frac{(N-1)^2}{N}} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix}$$

$$\mathcal{F}_N = \left[e^{-j2\pi\frac{k}{N}n}\right]_{0 \le (k,n) \le N-1}$$

$$\boldsymbol{F} = \mathcal{F}_N \boldsymbol{f}$$

$$\boldsymbol{f} = \mathcal{F}_N^{-1} \boldsymbol{F}$$

Hermitian transpose

$$\mathcal{F}_N^{-1} = \frac{1}{N} \mathcal{F}_N^H$$

# Generalized Discrete Fourier Transform (GDFT)

$$\mathbf{F}_{\alpha,\beta} = \left[e^{-j2\pi\frac{k+\alpha}{N}(n+\beta)}\right]_{0\leq(k,n)\leq N-1}.$$

$$\begin{aligned}
\mathbf{F}_{0,0}^{-1} &= \tfrac{1}{N}\mathbf{F}_{0,0}^{H} &= \tfrac{1}{N}\mathbf{F}_{0,0}^{*},\\
\mathbf{F}_{\frac{1}{2},0}^{-1} &= \tfrac{1}{N}\mathbf{F}_{\frac{1}{2},0}^{H} &= \tfrac{1}{N}\mathbf{F}_{0,\frac{1}{2}}^{*},\\
\mathbf{F}_{0,\frac{1}{2}}^{-1} &= \tfrac{1}{N}\mathbf{F}_{0,\frac{1}{2}}^{H} &= \tfrac{1}{N}\mathbf{F}_{\frac{1}{2},0}^{*}, \text{ and}\\
\mathbf{F}_{\frac{1}{2},\frac{1}{2}}^{-1} &= \tfrac{1}{N}\mathbf{F}_{\frac{1}{2},\frac{1}{2}}^{H} &= \tfrac{1}{N}\mathbf{F}_{\frac{1}{2},\frac{1}{2}}^{*}.
\end{aligned}$$

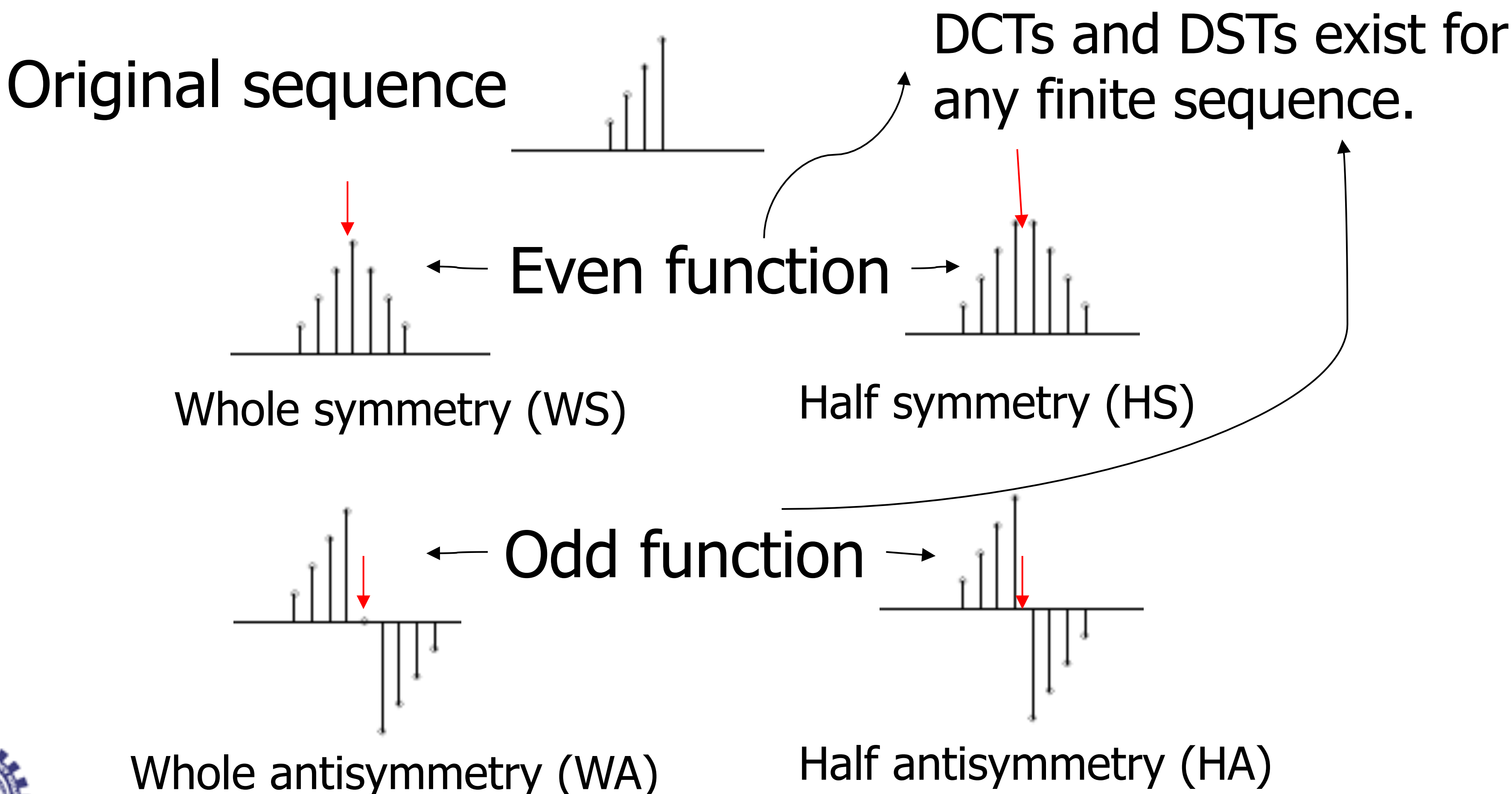
$$b_k^{(\alpha,\beta)}(n) = \frac{1}{\sqrt{N}}e^{j2\pi\frac{k+\alpha}{N}(n+\beta)}, \text{ for } 0\leq n\leq N-1, \text{ and } 0\leq k\leq N-1$$

$$\hat{f}_{\alpha,\beta}(k) = \sum_{n=0}^{N-1} f(n)e^{-j2\pi\frac{k+\alpha}{N}(n+\beta)}, \text{ for } 0\leq k\leq N-1$$

$$f(n) = \frac{1}{N}\sum_{k=0}^{N-1} \hat{f}_{\alpha,\beta}(k)e^{j2\pi\frac{k+\alpha}{N}(n+\beta)}, \text{ for } 0\leq n\leq N-1$$

| $\alpha$ | $\beta$ | Transform name | Notation |
|---|---|---|---|
| 0 | 0 | Discrete Fourier Transform ($DFT$) | $\hat{f}(k)$ |
| 0 | $\frac{1}{2}$ | Odd Time Discrete Fourier Transform ($OTDFT$) | $\hat{f}_{0,\frac{1}{2}}(k)$ |
| $\frac{1}{2}$ | 0 | Odd Frequency Discrete Fourier Transform ($OFDFT$) | $\hat{f}_{\frac{1}{2},0}(k)$ |
| $\frac{1}{2}$ | $\frac{1}{2}$ | Odd Frequency Odd Time Discrete Fourier Transform ($O^2DFT$) | $\hat{f}_{\frac{1}{2},\frac{1}{2}}(k)$ |

# Symmetric / Antisymmetric extension of a finite sequence

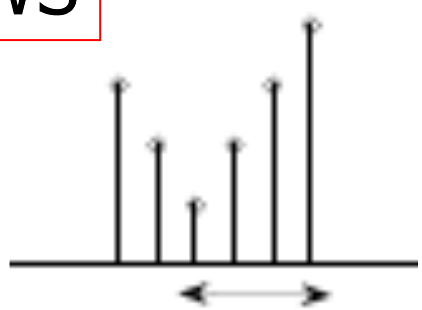


# Discrete Cosine / Sine Transforms

$$\alpha(p) = \begin{cases} \sqrt{\frac{1}{2}}, & \text{for } p = 0 \text{ or } N, \\ 1, & \text{otherwise.} \end{cases}$$

- Types of symmetric / antisymmetric extensions at the two ends of a sequence and a type of GDFT→ DCTs / DSTs

WSWS



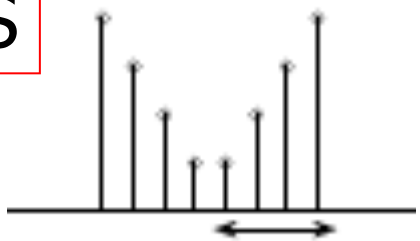
$\xrightarrow{F_{0,0}}$ Type-I Even DCT

$$C_{1e}(x(n)) = X_{Ie}(k) = \sqrt{\frac{2}{N}}\alpha^2(k)\sum_{n=0}^{N} x(n)\cos\left(\frac{2\pi nk}{2N}\right),\ 0 \le k \le N,$$

# Discrete Cosine / Sine Transforms

$$\alpha(p) = \begin{cases} \sqrt{\frac{1}{2}}, & \text{for } p = 0 \text{ or } N, \\ 1, & \text{otherwise.} \end{cases}$$

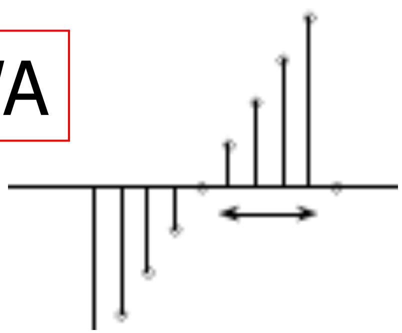
HSHS



$\xrightarrow{F_{0,1/2}}$ Type-2 Even DCT

$$C_{2e}(x(n)) = X_{IIe}(k) = \sqrt{\frac{2}{N}}\alpha(k)\sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi k(n+\frac{1}{2})}{2N}\right),\ 0 \le k \le N-1$$

WAWA



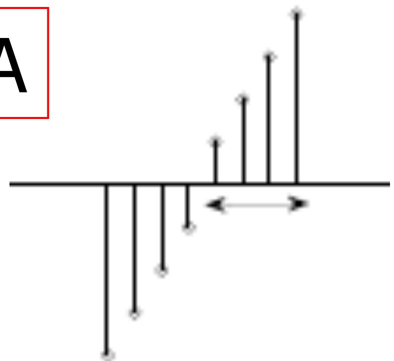
$\xrightarrow{jF_{0,0}}$ Type-1 Even DST

$$S_{1e}(x(n)) = X_{sIe}(k) = \sqrt{\frac{2}{N}}\sum_{n=1}^{N-1} x(n)\sin\left(\frac{2\pi kn}{2N}\right),\ 1 \le k \le N-1$$

# Discrete Cosine / Sine Transforms

$$\alpha(p) = \begin{cases} \sqrt{\frac{1}{2}}, & \text{for } p = 0 \text{ or } N, \\ 1, & \text{otherwise.} \end{cases}$$

HAHA



$jF_{0,1/2}$

Type-2 Even DST

$$S_{2e}(x(n)) = X_{sIIe}(k) = \sqrt{\frac{2}{N}}\alpha(k)\sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi k(n+\frac{1}{2})}{2N}\right), \ 1 \le k \le N-1$$

There exist 16 different types of DCTs and DSTs.

Type-II Even DCT is used in signal, image, and video compression.

# Matrix form of Type-II DCT

$$\alpha(p) = \begin{cases} \sqrt{\frac{1}{2}}, & \text{for } p = 0 \text{ or } N, \\ 1, & \text{otherwise.} \end{cases}$$

- Matrix form:

N-point DCT

$$C_N = \left[ \sqrt{\frac{2}{N}}.\alpha(k)cos(\frac{\pi k(2n+1)}{2N}) \right]_{0 \leq (k,n) \leq N-1}.$$

- Each row is either symmetric (even row) or antisymmetric (odd row).

$$C_N(k, N-1-n) = \begin{cases} C_N(k,n) & for\ k\ even \\ -C_N(k,n) & for\ k\ odd \end{cases}$$

$$X = C_N.\mathbf{x} \qquad C_N^{-1} = C_N^T$$

# Antiperiodic extension and skew-circular convolution

- Antiperiodic function with an antiperiod $N$, if $f(x+N)=-f(x)$.
- An antiperiodic function of antiperiod $N$ → a periodic function of period $2N$.
- Skew-circular convolution: convolution between two antiperiodic extended sequences of the same antiperiod.

$$\begin{aligned} f \circledS h(n) &= \sum_{m=0}^{N-1} f(m)h(n-m), \\ &= \sum_{m=0}^{n} f(m)h(n-m) - \sum_{m=n+1}^{N-1} f(m)h(n-m+N) \end{aligned}$$

# CMPs for DCTs

$$u(n) = x(n) \circledast y(n)$$

$$w(n) = x(n) \circledS y(n)$$

$$C_{1e}\left(u(n)\right) \quad = \quad \sqrt{2N}\, C_{1e}\left(x(l)\right) C_{1e}\left(y(m)\right)$$

$$C_{2e}\left(u(n)\right) \quad = \quad \sqrt{2N}\, C_{2e}\left(x(l)\right) C_{1e}\left(y(m)\right)$$

$$C_{3e}\left(w(n)\right) \quad = \quad \sqrt{2N}\, C_{3e}\left(x(l)\right) C_{3e}\left(y(m)\right)$$

$$f(x,y) = \sum_j \sum_i \lambda_{ij} b_{ij}(x,y)$$

# 2-D Transforms

- Easily extendable if basis functions are separable, i.e. $B=\{ b_{ij}(x,y)=g_i(x).g_j(y)\}$.

They could be from two different sets, say $b(x,y)=g(x).h(y)$.

1-D basis function

- $B$: Orthogonal if $G=\{g_i(x), i=1,2,..\}$ is orthogonal.
- $B$: Orthogonal and complete if G is so.
- Reuse of 1-D transform computation.

$$\lambda_{ij} = \sum_j g_j^*(y) \left( \sum_i f(x,y) g_i^*(x) \right)$$

# 2D Discrete Transform

$$Y_{m\text{x}n} = B_{m\text{x}m} X_{m\text{x}n} B_{n\text{x}n}{}^{\mathrm{T}}$$

- Use of separability:
  - Transform columns.
  - Transform rows.
- Input: $X_{m\text{x}n}$ 1-D Transform Matrix: *B*
- Transform columns: $[Y_1]_{m\text{x}n} = B_{m\text{x}m} X_{m\text{x}n}$
- Transform rows: $Y_{m\text{x}n} = [B_{n\text{x}n} Y_1{}^{\mathrm{T}}]^{\mathrm{T}}$

$$= Y_1 B_{n\text{x}n}{}^{\mathrm{T}}$$

$$= B_{m\text{x}m} X_{m\text{x}n} B_{n\text{x}n}{}^{\mathrm{T}}$$

# Image Transform: DFT

Image: $f(m,n)$, of size $M$ x $N$

$$F(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi\frac{km}{M}} e^{-j2\pi\frac{ln}{N}}$$

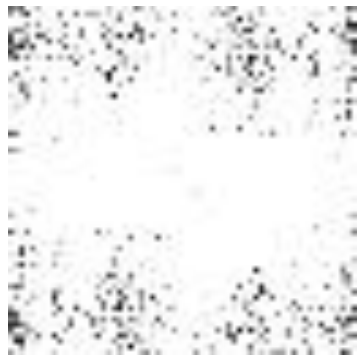
$$f(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k,l) e^{j2\pi\frac{km}{M}} e^{j2\pi\frac{ln}{N}}$$

Property of separability

$$\boldsymbol{F} = \mathcal{F}_m \boldsymbol{f} \mathcal{F}_N^T$$

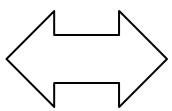
$$F(k,l) = \sum_{m=0}^{M-1} e^{-j2\pi\frac{km}{M}} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi\frac{ln}{N}}$$

# DFT Examples:



Magnitude

Phase



Magnitudes and phases are shown by bringing them into displayable range, and shifting the origin at the center of image.

# 2D DCT

$$\alpha(p) = \begin{cases} \sqrt{\frac{1}{2}}, & \text{for } p = 0 \text{ or } N, \\ 1, & \text{otherwise.} \end{cases}$$

- Type-I:

$$X_I(k,l) \;=\; \frac{2}{N}.\alpha\;(k).\alpha\;(l).\sum_{m=0}^{M}\sum_{n=0}^{N}(x(m,n)\cos(\frac{m\pi k}{M})\cos(\frac{n\pi l}{N})), \quad 0 \le k \le M, 0 \le l \le N.$$

- Matrix Representation:

$$X_{II}(k,l) \;=\; \frac{2}{N}.\alpha(k).\alpha(l).\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(x(m,n)\cos(\frac{(2m+1)\pi k}{2M})\cos(\frac{(2n+1)\pi l}{2N})), \quad 0 \le k \le M-1, 0 \le l \le N-1.$$

$$X = DCT(x) = C_M.\mathrm{x}.C_N^T$$

# An example:

Input image

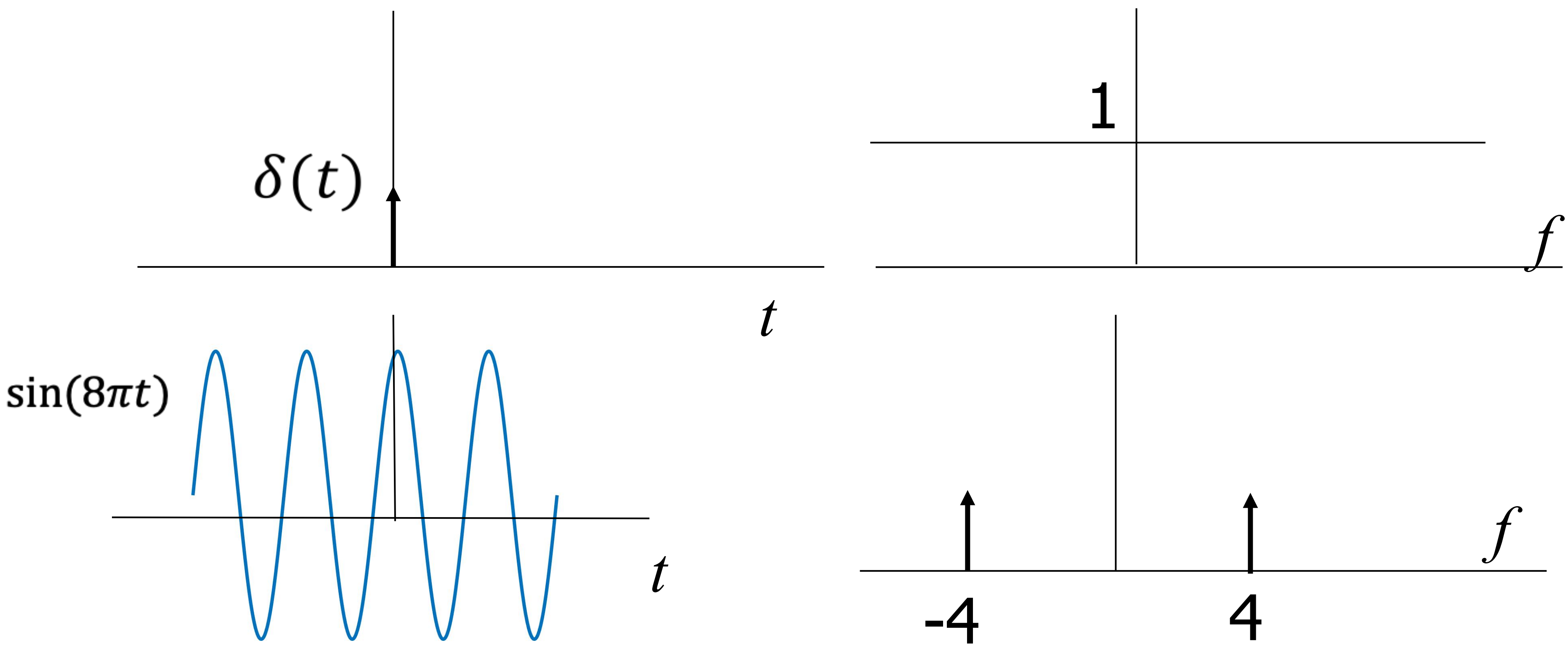
Discrete Cosine Transform

There are 16 different types of DCTs and DSTs.

# Wavelets

- Functions to have *ideally* finite support in both its original domain (say, time or space) and also in the transform domain (i.e., the frequency domain).
  - No such function exists truly satisfying it.
  - Attempts to match these properties as far as possible.
- Acts as basis functions.
- Good localization property in both domain.

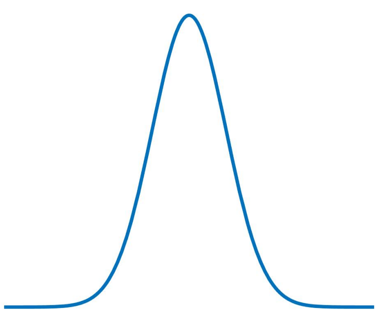
# A few examples



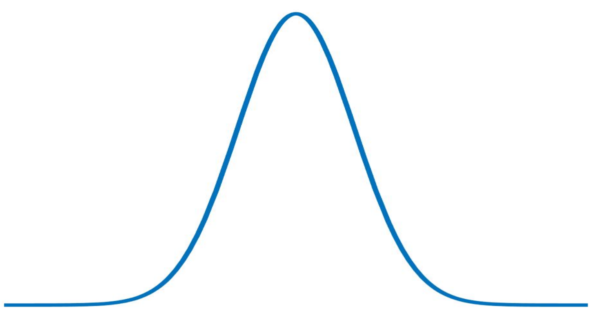
# An interesting function

- Same form in time and frequency domain
  - Gaussian
- Analogy from Heisenberg's uncertainty principle

$$\sigma_t^2 \sigma_f^2 \geq \frac{1}{4}$$



Variance of $t$ weighted by $g^2(t)$. Similarly for $f$. For any function it holds !!



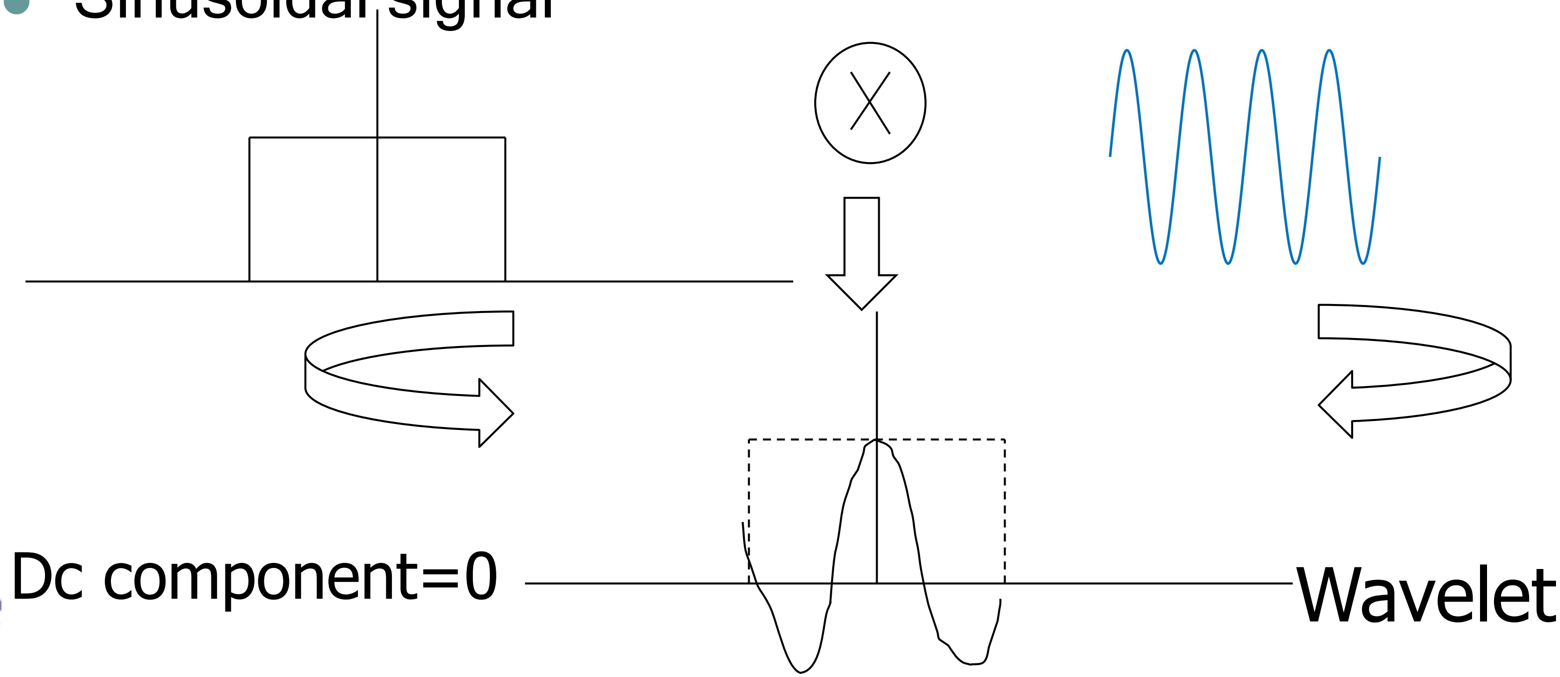
$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}$$

$$G(\omega) = e^{-\frac{\omega^2\sigma^2}{2}}$$

# Designing wavelet: An intuitive approach

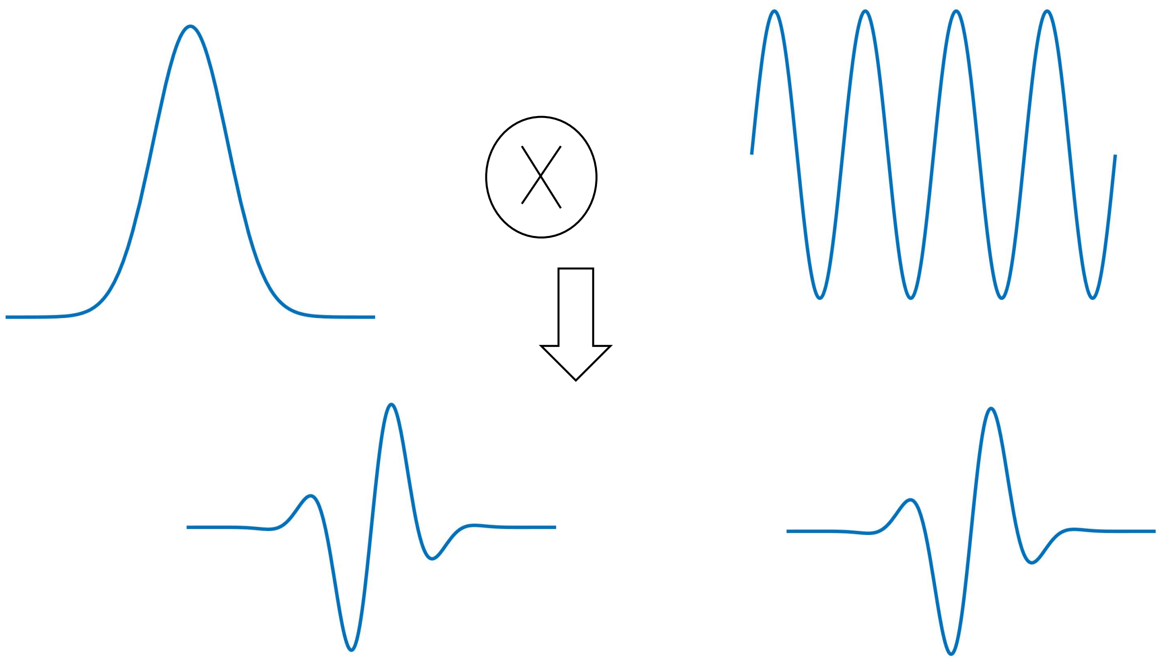
- Time limited signal:
  - Square pulse
- Band limited signal:
  - Sinusoidal signal

- Wavelet to satisfy both?
  - Multiply them!!



# Gabor wavelet (1-D)

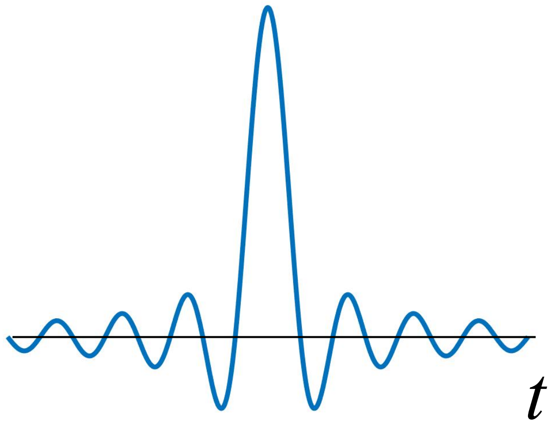
$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} \qquad e^{j2\pi f t}$$

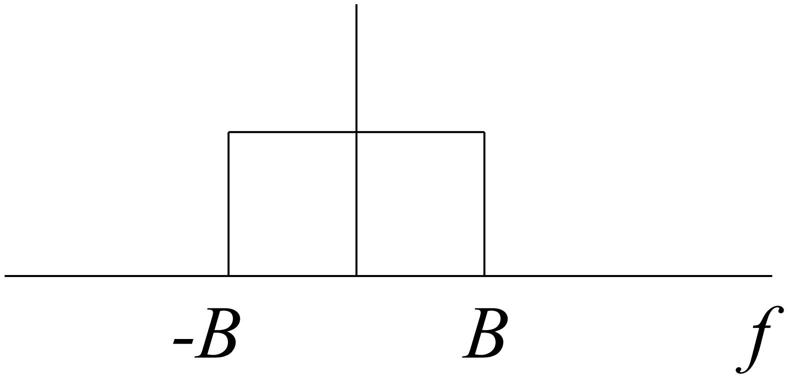


Real part  Imaginary part

# Shannon wavelet

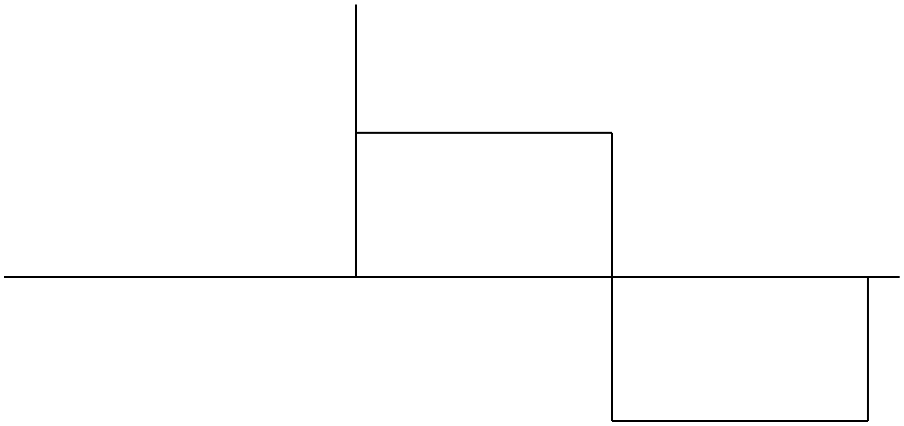
$$h(t) = 2B\frac{sin(2\pi Bt)}{2\pi Bt} = 2Bsinc(2Bt)$$



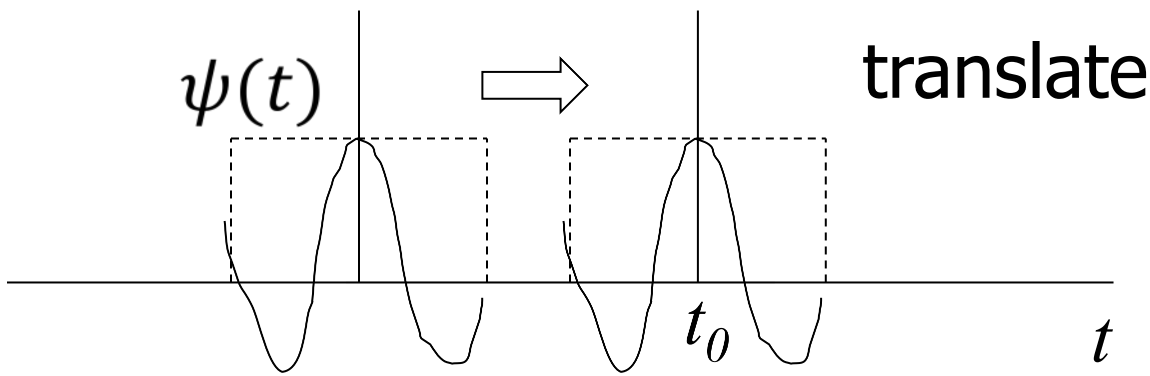
# Haar Wavelet

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} < t \leq 1 \\ 0 & Otherwise \end{cases}$$



# Family of wavelets

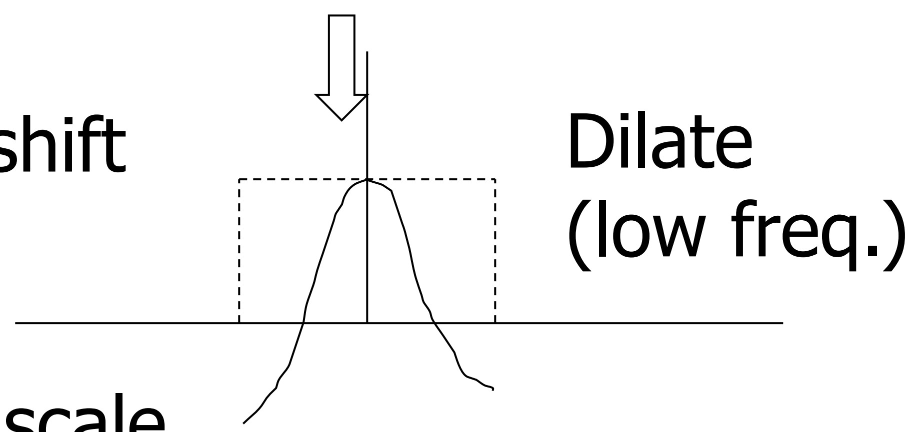
- Translate and dilate a mother wavelet




$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

shift

scale



# Continuous wavelet transform

- Forward transform

From 1-D representation to 2-D representation.

$$W(s,\tau) = \int f(t) \frac{1}{\sqrt{s}} \psi^* \left( \frac{t-\tau}{d} \right) dt$$

How correlated at that instance with the wavelet fn.

Reveals structure of function at multiple resolution.

- Inverse transform:

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^{\infty} W(s,\tau) \frac{\psi(t)}{s^2} ds d\tau$$

Fourier transform

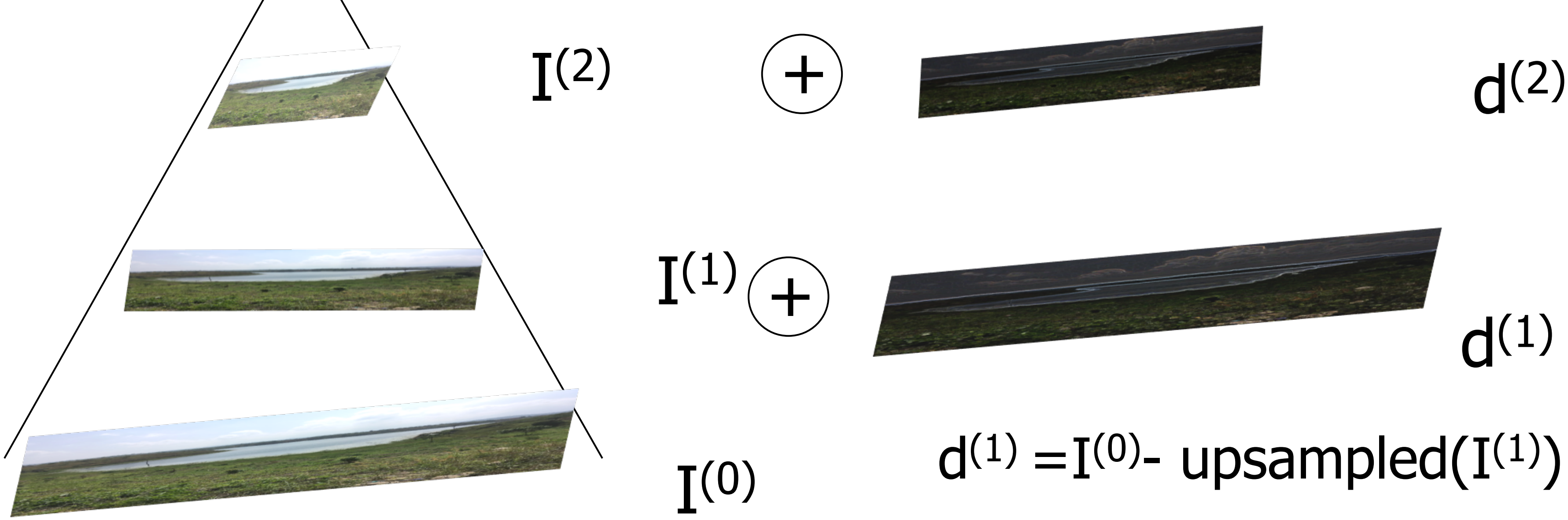
where

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|}{|\omega|} d\omega$$

# Multiresolution representation

- Gaussian Pyramid

$d^{(2)} = I^{(1)} - \text{upsampled}(I^{(2)})$




$d^{(1)} = I^{(0)} - \text{upsampled}(I^{(1)})$

$I(0) = us(us\,(I^{(2)}) + d^{(2)}) + d^{(1)}$

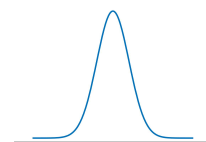
Approximation Details

$d^{(i)} = I^{(i-1)} - \text{upsampled}(I^{(i)})$

# Gaussian Pyramid: Wavelet analysis

$I^{(2)}$

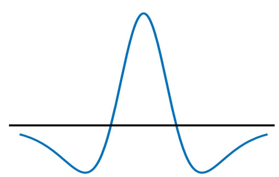
Obtained by convolution with G(x,y) and downsampling at successive stages.



Scaling function

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{\frac{-((x-x_c)^2+(y-y_c)^2)}{2\sigma^2}}$$

$d^{(2)}$



Obtained by convolution with DOG(x,y) and downsampling at successive stages.

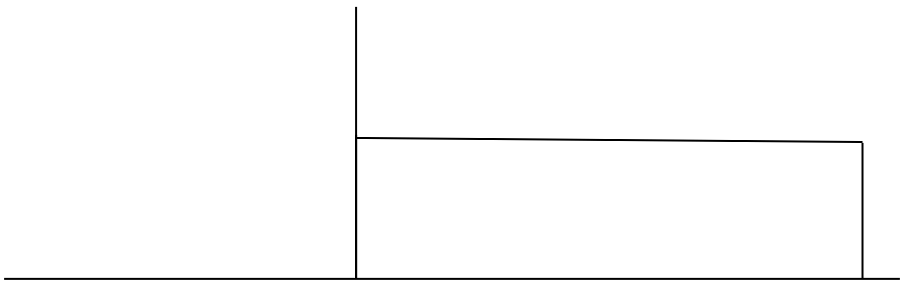
Wavelet function

$d^{(1)}$

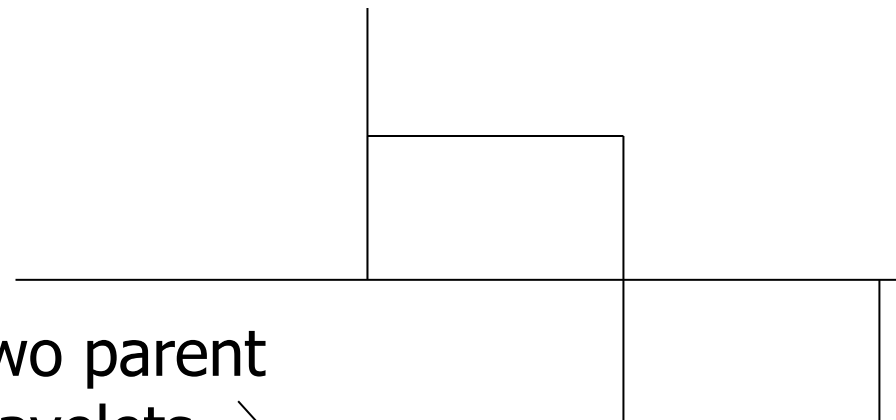
Filtering and transformation equivalent!!

# Haar Wavelet transform

- Scaling function



- Wavelet function



Two parent wavelets

$$\varphi(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0 & Otherwise \end{cases}$$

$$\psi(t) = \begin{cases} 1, & 0 \le t < \frac{1}{2} \\ -1 & \frac{1}{2} < t \le 1 \\ 0 & Otherwise \end{cases}$$

Family of translated and dilated functions from the both forms the basis.

# Discrete wavelet transform (DWT)

- Translated only at discrete grid points.
  - $k$=0, ± 1, ± 2, ......
  - Finite sequence: A finite number of basis functions.
- Scaled by powers of 2: $2^j$, $j$=0,1,...
  - Downsampling takes care of dilation of wavelets and allows to use the same function at that level.
- Family of scaling and wavelet functions:

$$\varphi_{j,k}(n) = 2^{-\frac{j}{2}}\varphi\left(2^{-j}n - k\right), j = 0,1, \ldots k = 0,1, \ldots, M$$

$$\psi_{j,k}(n) = 2^{\frac{-j}{2}}\psi\left(2^{-j}n - k\right), j = 0,1, \ldots k = 0,1, \ldots, M$$

M≤N (length of sequence)

# Haar wavelets in discrete grid

- N=8

$$\varphi(n) = \frac{1}{\sqrt{2}}(1,1,0,0,0,0,0,0)$$

$$\psi(n) = \frac{1}{\sqrt{2}}(1,-1,0,0,0,0,0,0)$$

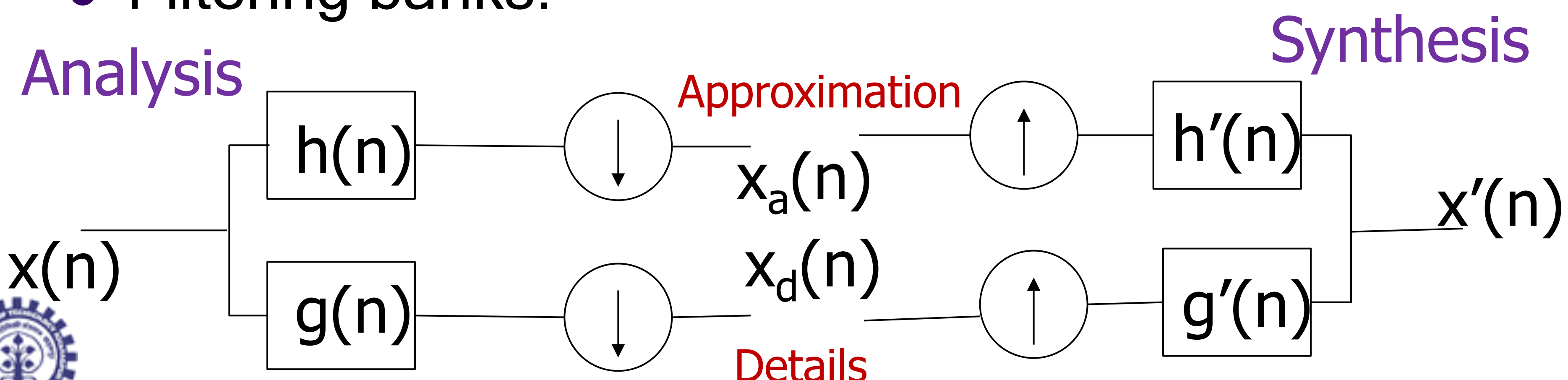
Transformation matrix:

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

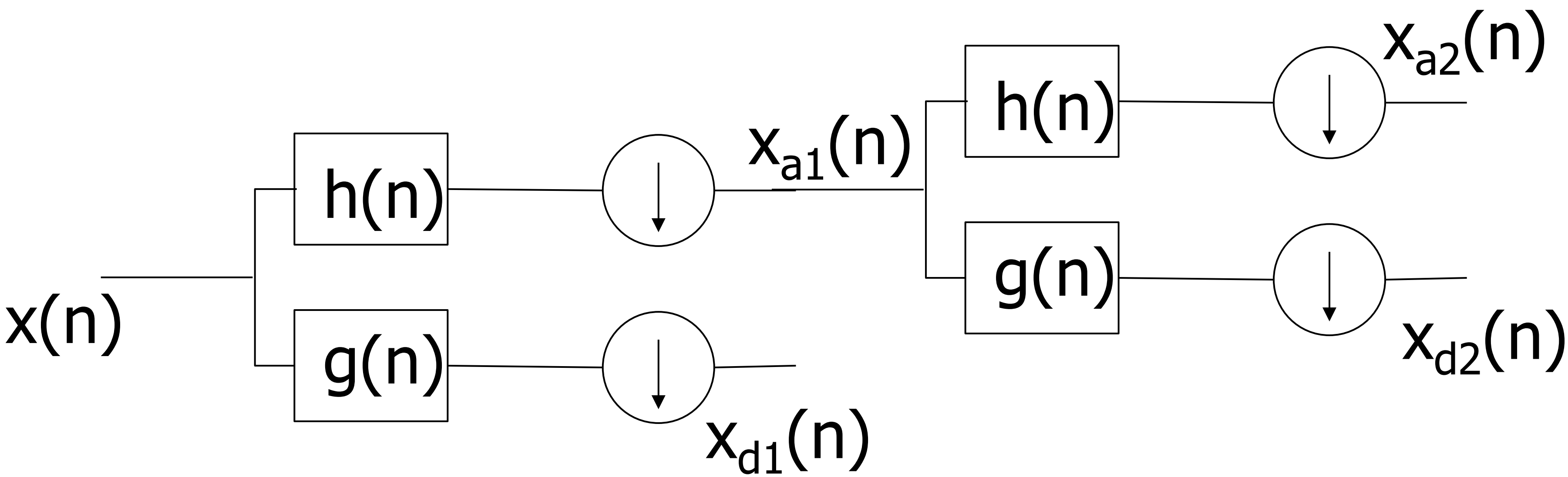
Courtesy: "Image and video processing in the compressed domain", J. Mukhopadhyay, CRC Press, 2011.

# DWT

- Translated only at discrete grid points.
- Scaled by powers of 2: $2^j$, $j=0,1,\ldots$
  - Downsampling takes care of dilation of wavelets and allows to use the same function at that level.
  - Filtering by the filter of same impulse response.
- Filtering banks.



# Dyadic decomposition




- At each level sample size is halved
  - Equivalent of scaling by 2.
- Total number of samples remain the same.

# Typical wavelet filters

- Daubechies 9/7 filters

| $n$ | Analysis filter bank | | Synthesis filter bank | |
|---|---|---|---|---|
| | $h(n)$ | $g(n-1)$ | $h'(n)$ | $g'(n+1)$ |
| 0 | 0.603 | 1.115 | 1.115 | 0.603 |
| $\pm$ 1 | 0.267 | -0.591 | 0.591 | -0.267 |
| $\pm$ 2 | -0.078 | -0.058 | -0.058 | -0.078 |
| $\pm$ 3 | -0.017 | 0.091 | -0.091 | 0.017 |
| $\pm$ 4 | 0.027 | | | 0.027 |

- Le Gall 5/3 filters

| $n$ | Analysis filter bank | | Synthesis filter bank | |
|---|---|---|---|---|
| | $h(n)$ | $g(n-1)$ | $h'(n)$ | $g'(n+1)$ |
| 0 | $\frac{6}{8}$ | 1 | 1 | $\frac{6}{8}$ |
| $\pm$ 1 | $\frac{2}{8}$ | $-\frac{1}{2}$ | $\frac{1}{2}$ | $-\frac{2}{8}$ |
| $\pm$ 2 | $-\frac{1}{8}$ | | | $-\frac{1}{8}$ |

Courtesy: "Image and video processing in the compressed domain", J. Mukhopadhyay, CRC Press, 2011.

# 2-D DWT

- Separable filters.
- Transform rows, then transform columns.

By 5/3 Analysis filters

**Applications:**

- Compression
- Denoising
- Feature representation
- Image fusion

…

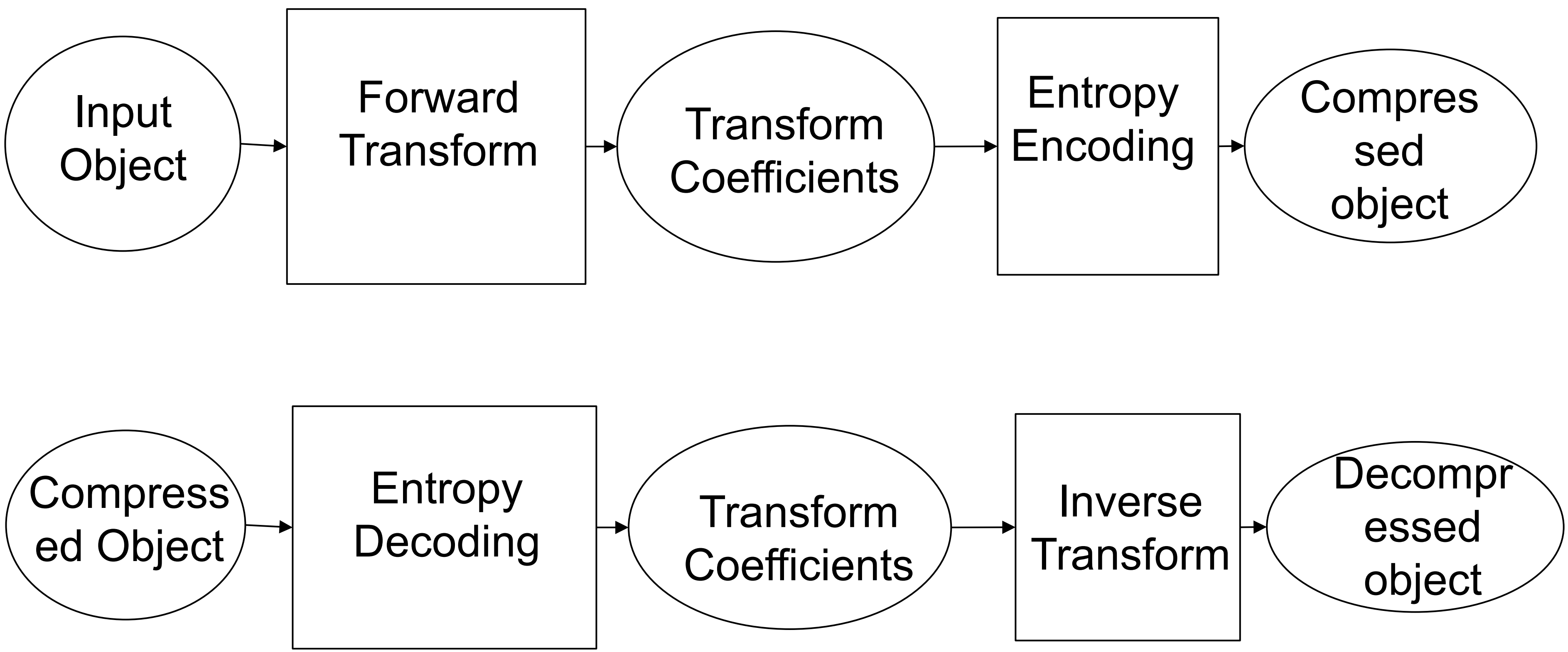
# Image compression

- An alternative representation requiring less storage compared to in original original space.
  - An analogy with representation of a circle:
    - A set of all points in its periphery.
    - Only three (non-collinear) points.
    - Center and radius
- Decompression: Reconstruction from  a compressed image in the original space.
- Lossy compression: Approximate reconstruction.
- Lossless compression: Exact reconstruction

# Desirable features

- Good reconstructibility
  - Visual quality of decompressed image should be high.
- Low redundancy
  - Spatial correlation, Channel (color) correlation, Symbol representation,
- Factorization in substructures
  - Frequency components, Space-frequency decomposition,
    - DCT, DWT

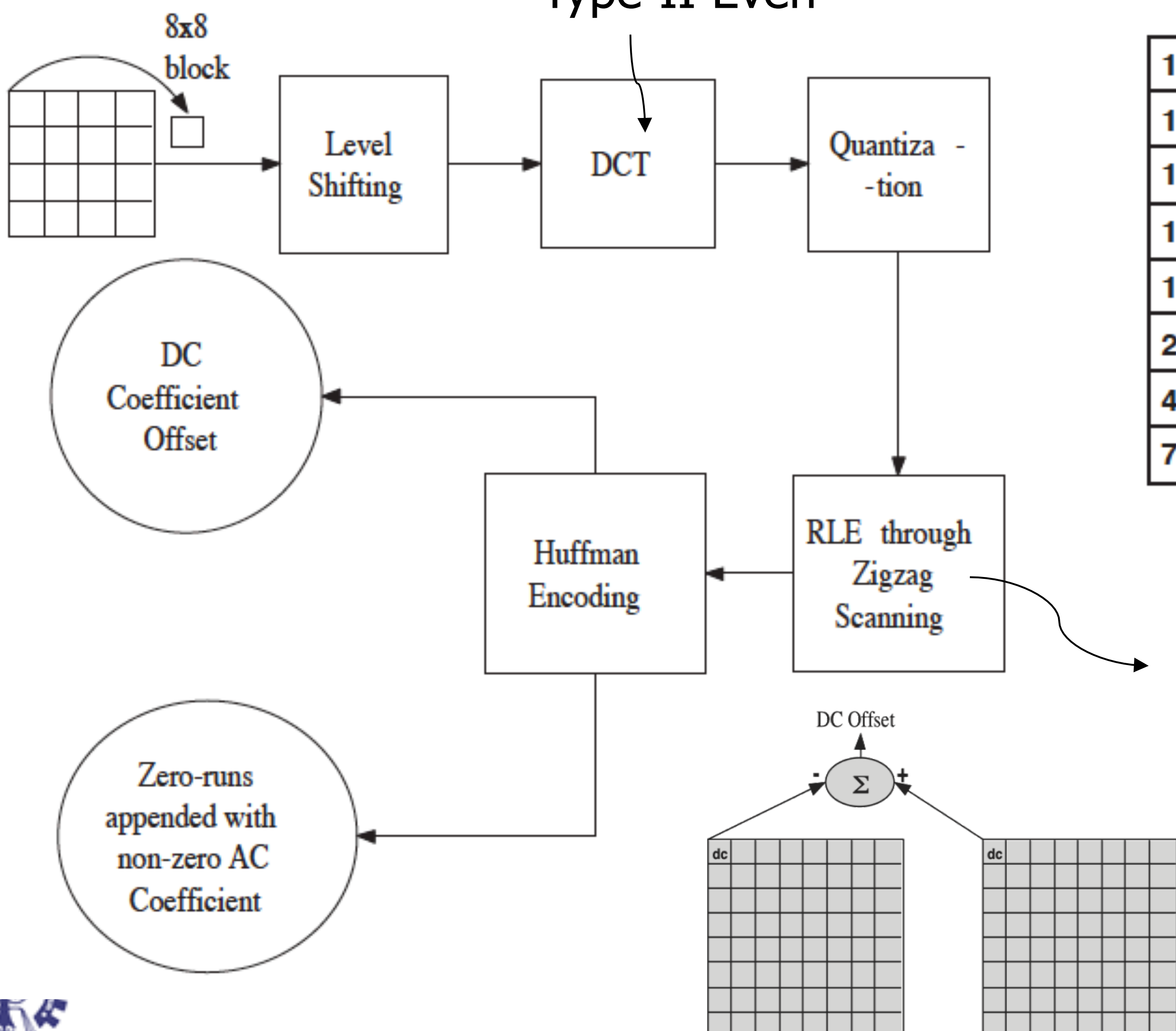
# Generic pipeline of compression and decompression



Courtesy: J. Mukhopadhyay, Image and video processing in the compressed domain, CRC Press, 2011.

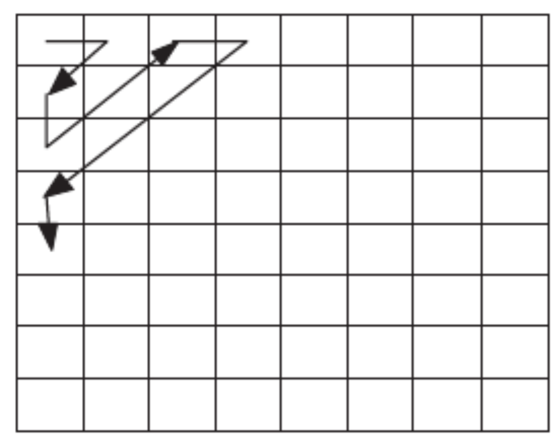
# JPEG: Baseline scheme

Type-II Even




Typical quantization mask.

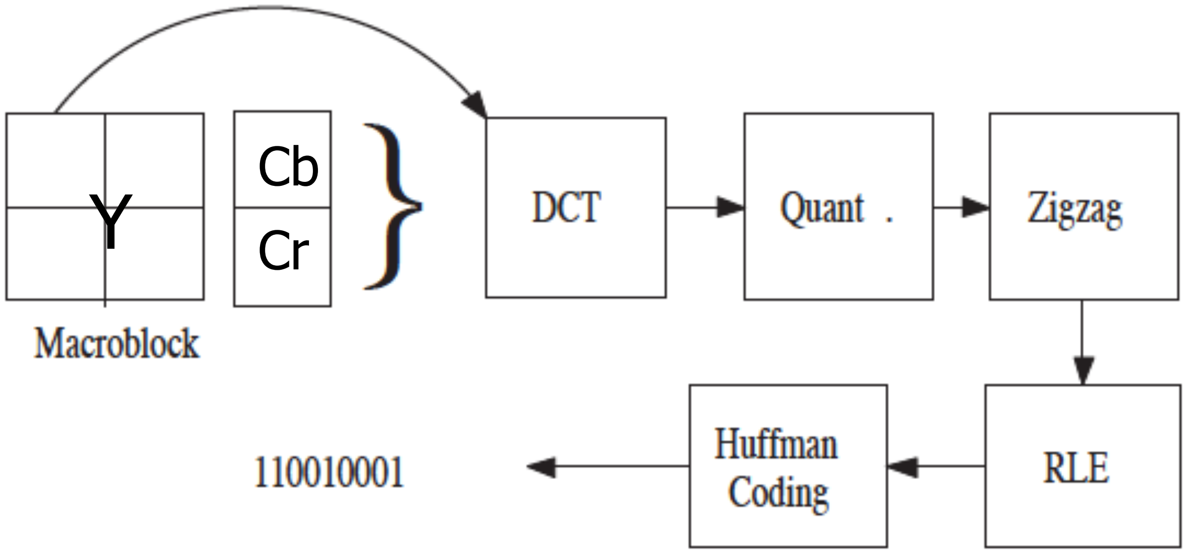
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |



# Color encoding in JPEG

- Y-Cb-Cr color space:

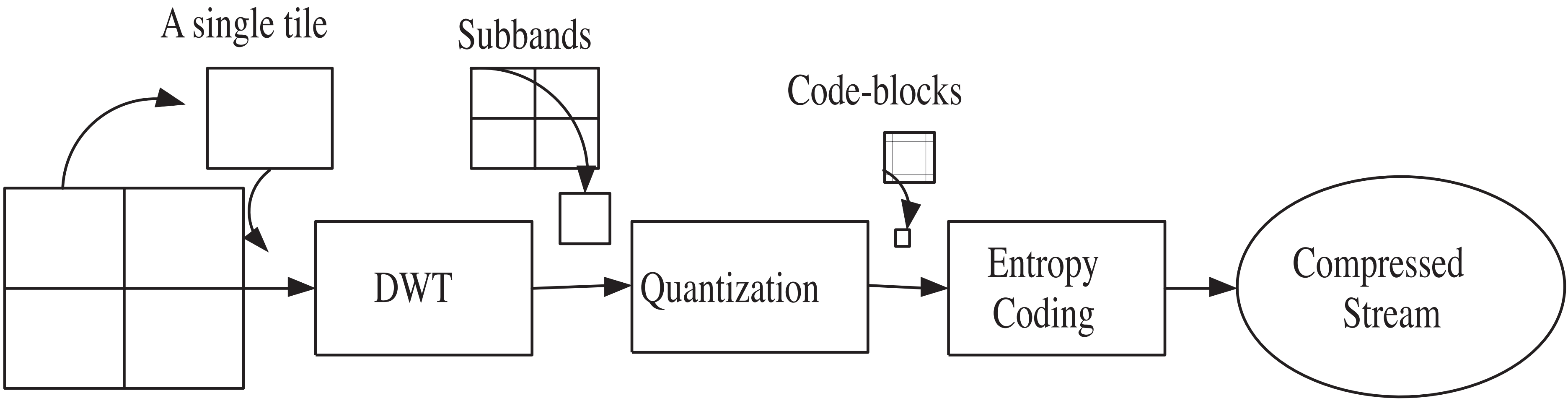
$$
\begin{aligned}
Y \; &= 0.520G + 0.098\,B + 0.256R \\
Cb &= -0.290G + 0.438\,B - 0.148R + 128 \\
Cr &= -0.366G - 0.071\,B + 0.438R + 128
\end{aligned}
$$




Courtesy: J. Mukhopadhyay, Image and video processing in the compressed domain, CRC Press, 2011.

Courtesy: J. Mukhopadhyay, Image and video processing in the compressed domain, CRC Press, 2011.

# JPEG 2000




- Lossy: Daubechies 9/7 filters
- Lossless: Le Gall 5/3 filters
- Color Transformation:
  - Lossy: Y-Cb-Cr (w/o downsampling)
  - Lossless: $Y = \left\lfloor \frac{R + 2G + B}{4} \right\rfloor$ $U = R - G$ $V = B - G$

$$G = Y - \left\lfloor \frac{U + V}{4} \right\rfloor$$

$$R = U + G$$

$$B = V + G$$

# JPEG 2000: Quantization

- Each sub-band independently quantized with a uniform quantization threshold.

$$\Delta = 2^{n-\epsilon}(1 + \frac{\mu}{2^{11}})$$

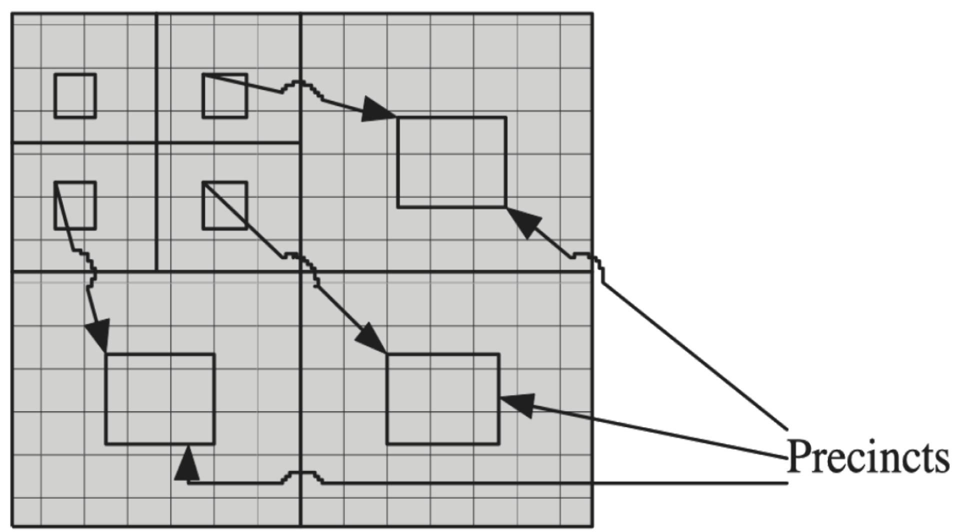
  - n: Nominal dynamic range of the sub-band, e.g. 10 for $HH_1$
  - $\epsilon$ , $\mu$ : the number of bits allotted to the exponent and mantissa respectively, of its coefficients.
- Quantized coefficient (of $X(u,v)$ )

$$X_q(u,v) = sign(X(u,v)) \left\lfloor \frac{|X(u,v)|}{\Delta} \right\rfloor$$

- For lossless compression: $\boldsymbol{\Delta}=1$
- Implicit quantization: Lowest level (LL): $\epsilon_0$ , $\mu_0$
- For the $i$ th sub-band at level $k$: $\mu_i = \mu_0$ and $\epsilon_i = \epsilon_0 + i - k$

# JPEG2000: Code Structure

- Every sub-band partitioned into a set of non-overlapping codeblocks.
- Each codeblock independently coded by a schema called Embedded Block Coding on Truncation (EBCOT).
- each bit-plane of wavelet coefficients is processed by three passes, namely, significant propagation, magnitude refinement, and clean-up.
- The resulting bit-stream encoded using Arithmetic Encoding.
- A layer formed with the output of similar passes from a group of code blocks.
- In a layer, packets formed by grouping corresponding code blocks of subbands at the same level of decomposition.
  - also known as precincts




Courtesy: J. Mukhopadhyay, Image and video processing in the compressed domain, CRC Press, 2011.

# Summary

- Image transforms involve representation of images as a linear combination of a given set of basis functions.
- For a finite discrete sequence, this is treated as a linear combination of a given set of basis vectors.
- Orthogonal set of basis functions (vectors) simplifies computation of forward and inverse transforms
  - Inner product of the function with the basis function.
  - Examples: Fourier Transform, Wavelet Transforms (may be also non-orthogonal)
- A set of basis functions may be orthogonal but not complete for exactly representing any arbitrary function.
  - Cosine and Sine Transforms in continuous domain.
  - For finite discrete sequences several orthogonal and complete transforms available: DFT, GDFTs, DCTs, DSTs, etc.

# Summary

- Alternative representation provides other insights of structure of images.
  - low frequency and high frequency components.
- May become useful for providing more compact representation.
  - A few transform coefficients.
  - Selective quantization of components, considering their effect on our perception.
    - Image compression: JPEG
- Sometimes convenient for processing.
  - Filtering, enhancement, ….

# Summary

- Wavelets represent the scale of features in an image, as well as their positions.
  - Time-scale, Space-Scale representation
- Fast computation of forward and inverse transform
- Provides multiresolution representation.
  - Enables progressive and scalable processing
- Lossy and lossless  reconstruction possible.
- useful for a number of applications including image compression.
  - JPEG2000

# Thank You