University of Minnesota
Driven to Discover℠

EE 5561: Image Processing and Applications

Lecture 8

Mehmet Akçakaya

Recap of Last Lecture

- Linear algebra review
  - Vector spaces, bases
  - Norms, inner products
  - Matrix representation of linear transformations
  - Matrix transpose/Hermitian, inverse, null space, range space, rank
  - Diagonal, symmetric, unitary, circulant matrices; diagonalization
  - Least squares
- Today
  - Use linear algebra to derive a new transform (wavelets)
  - Energy compaction + another transform (DCT)
  - Random vectors review

Wavelets

- Yet another linear transform
  - Leads to good "energy compaction" for images
  - i.e. good compressibility
- We will do this using linear algebra
  - We will do Haar wavelets
  - Now how it is normally derived
  - Deeper theory: EE 5551

Wavelets

– Start with a simple averaging filter

$$\mathbf{h} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{h} * \mathbf{x} \quad \Rightarrow \quad \frac{1}{2}x_n + \frac{1}{2}x_{n-1} = y_n$$

- Write this as a matrix

$$\mathbf{H} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdots \\ & & \ddots & & \end{bmatrix}$$

- Invertible?
  - No, we only have the average

Wavelets

- But if we also have convolution with

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

$$\mathbf{z} = \mathbf{g} * \mathbf{x} \quad \Rightarrow \quad \frac{1}{2}x_n - \frac{1}{2}x_{n-1} = z_n$$

then the whole system is invertible

since we have

$$\begin{aligned} x_n &= y_n + z_n \\ x_{n-1} &= y_n - z_n \end{aligned} \qquad \text{or} \qquad x_{n-1} = y_{n-1} + z_{n-1}$$

- In matrix form

$$\mathbf{G} = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 & \cdots \\ & & \ddots & & \end{bmatrix}$$

Wavelets

- Overall system becomes

$$\begin{bmatrix} \mathbf{H} \\ \mathbf{G} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}$$

- How do we get **x** back?
  - Going with the first approach from earlier

$$x_1 = y_2 - z_2$$
$$x_2 = y_2 + z_2$$
$$x_3 = y_4 - z_4$$
$$x_4 = y_4 + z_4$$

Only need two rows
i.e. we don’t need y_1, z_1, y_3, z_3, …

Wavelets

- So we can truncate the matrix by "downsampling"

$$\tilde{\mathbf{W}}_N = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdots \\ & & \ddots & & \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots \\ 0 & 0 & -\frac{1}{2} & \frac{1}{2} & \cdots \\ & & \ddots & & \end{bmatrix} = \begin{bmatrix} \mathbf{H}_D \\ \mathbf{G}_D \end{bmatrix}$$

- Note

$$\tilde{\mathbf{W}}_N^{-1} = \begin{bmatrix} 1 & 0 & & -1 & 0 & \\ 1 & 0 & & 1 & 0 & \\ 0 & 1 & \cdots & 0 & -1 & \cdots \\ 0 & 1 & \cdots & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & & \end{bmatrix}$$

Note $\tilde{\mathbf{W}}_N^H = \frac{1}{2}\tilde{\mathbf{W}}_N^{-1}$

i.e. "almost" unitary/orthogonal

Wavelets

- Thus if we set $\mathbf{W}_N = \sqrt{2}\tilde{\mathbf{W}}_N$, it'll be unitary
- Corresponding filter is

$$\mathbf{h} = \left[\begin{array}{cc} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{array}\right]$$

$$H(w) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}e^{iw} \quad \Rightarrow \quad \begin{array}{l} H(\pi) = 0 \\ H(0) = \sqrt{2} \end{array}$$

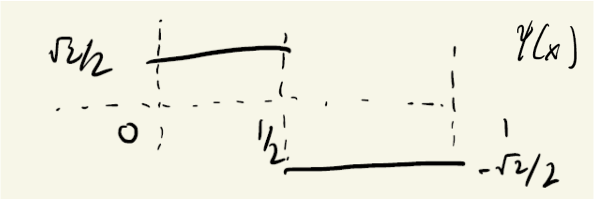
$\longleftarrow$ Low-pass as expected

Wavelets

- Wavelet ↔ "wave-like" oscillation

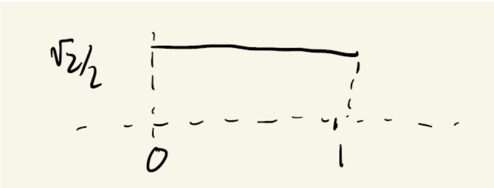
  ~integrate to zero

- Wavelets are usually derived in continuous-space first (like we did in Fourier analysis)
  - In this case the Haar wavelet in continuous space is



(this is **g** in discrete-space)

  - And the other function (Haar scaling function) is



(this is **h** in discrete-space)

Haar Wavelet

- **X**: *N*×*N* image
  - We apply $\mathbf{W}_N$ to row & columns, i.e.

$$\mathbf{W}_N \mathbf{X} \mathbf{W}_N^T$$

or

$$\mathcal{W}\{\text{vec}(\mathbf{X})\}$$

linear operator

  - Application

$$\mathbf{W}_N \mathbf{X} \mathbf{W}_N^T = \begin{bmatrix} \mathbf{H}_D \mathbf{X} \mathbf{H}_D^T & \mathbf{H}_D \mathbf{X} \mathbf{G}_D^T \\ \mathbf{G}_D \mathbf{X} \mathbf{H}_D^T & \mathbf{G}_D \mathbf{X} \mathbf{G}_D^T \end{bmatrix} \triangleq \begin{bmatrix} \underset{\text{blur}}{\mathbf{B}} & \underset{\text{vertical}}{\mathbf{V}} \\ \underset{\text{horizontal}}{\mathbf{H}} & \underset{\text{diagonal}}{\mathbf{D}} \end{bmatrix}$$

Haar Wavelet

X

$W_N X$

Haar Wavelet

X

W_N X

window-leveled
to show details

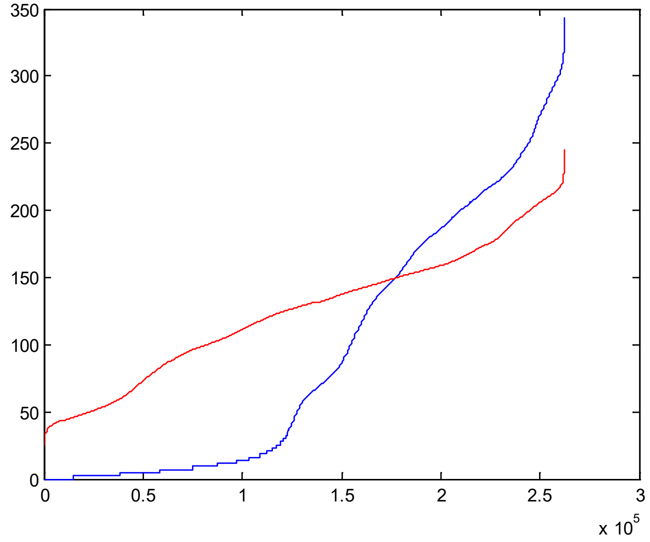
Haar Wavelet

X

$W_N \, X \, W_N^T$

Haar Wavelet

- Energy Compaction
  - Idea: Few of the wavelet coefficients capture most of the energy of the image
  - In Fourier domain, we saw that the central frequencies contain most of the energy
  - But those frequencies do not contain sharp image structures
  - Wavelets offer an alternative



Energy "compaction":
sorted (abs value) image (red) vs. wavelet (blue) coefficients across the image

Not really "very" compact here…

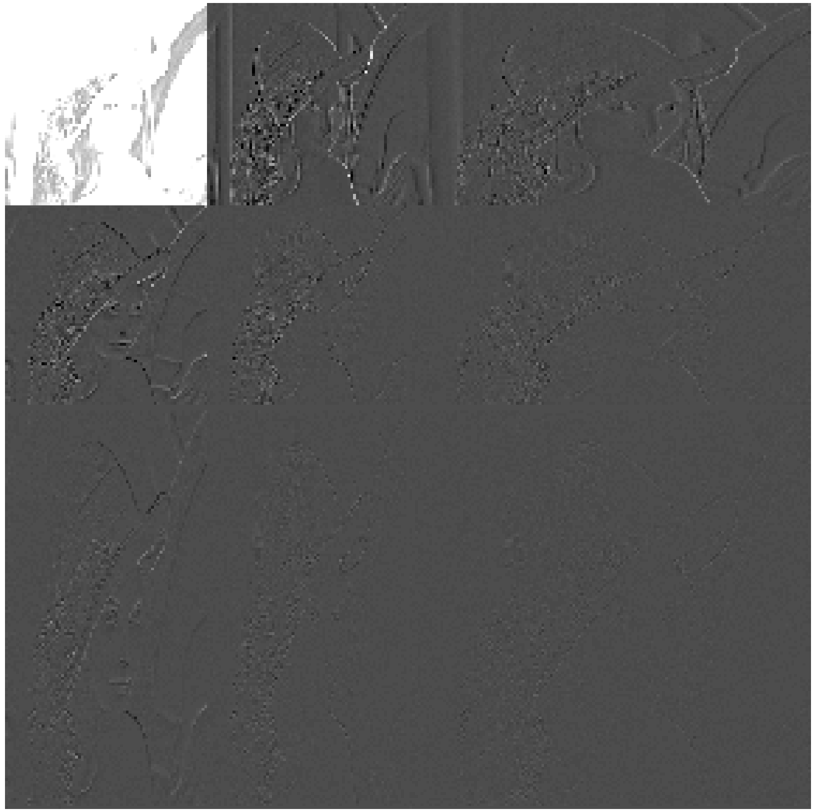
Haar Wavelet

- Energy Compaction
  - How to get more compaction?
  - Iterate! → Extract blur component (averaged image) & compute its wavelet transform

Haar Wavelet

$\mathbf{X}$

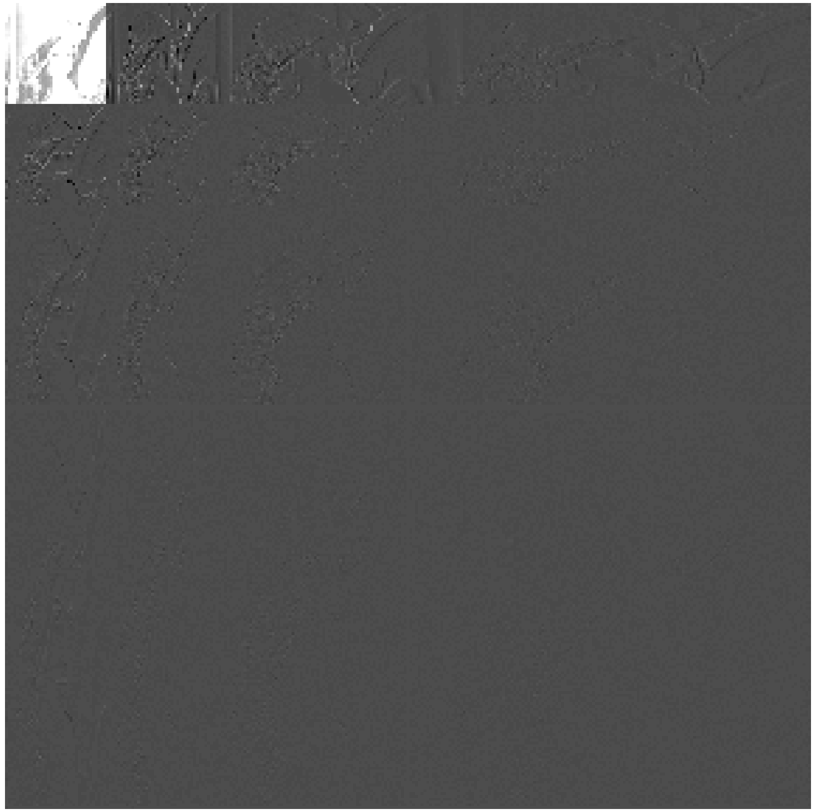
2-levels: $\mathbf{W}_N \, \mathbf{X} \, \mathbf{W}_N^T$



Haar Wavelet

X

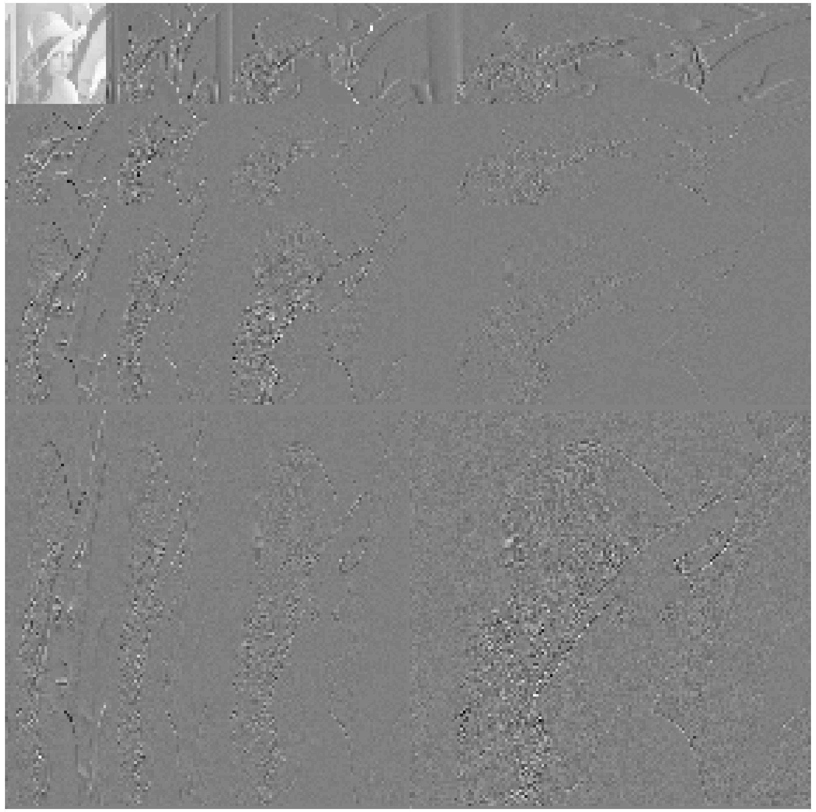
3-levels: $\mathbf{W}_N \, \mathbf{X} \, \mathbf{W}_N^T$



Haar Wavelet

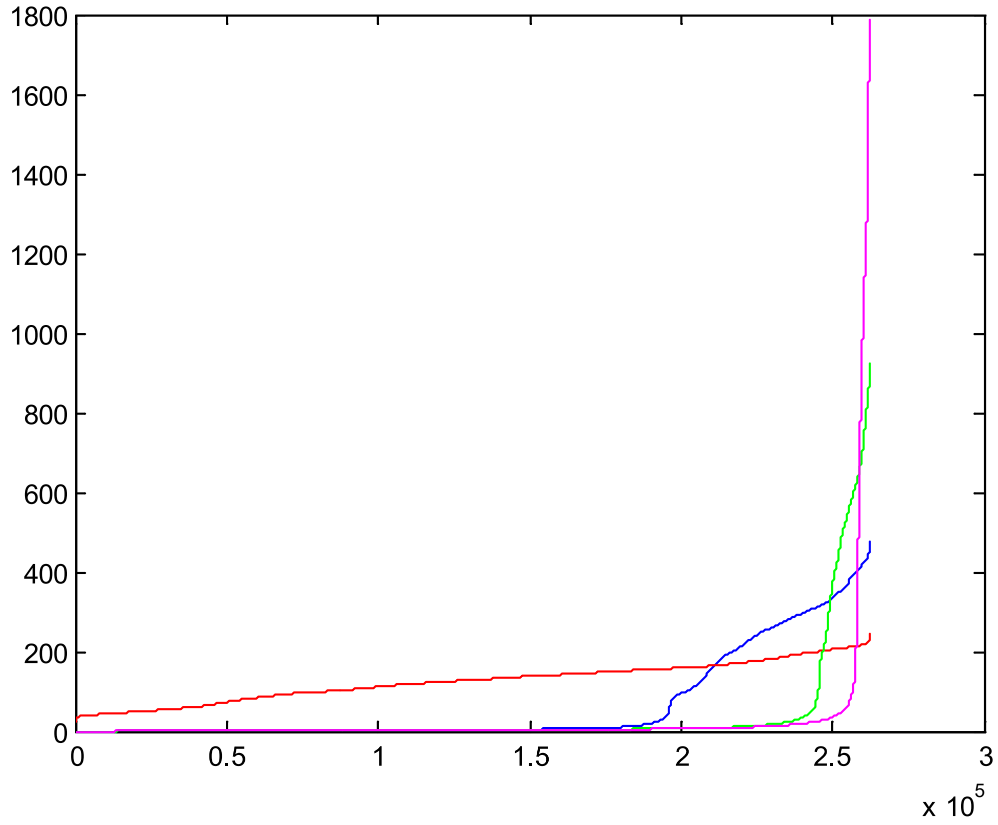
X

3-levels: $\mathbf{W}_N \, \mathbf{X} \, \mathbf{W}_N^T$



Haar Wavelet

Energy "compaction":
sorted (abs value) image (red) vs. wavelet
1-level (blue), 2-level (green), 3-level(magenta) coefficients across the image



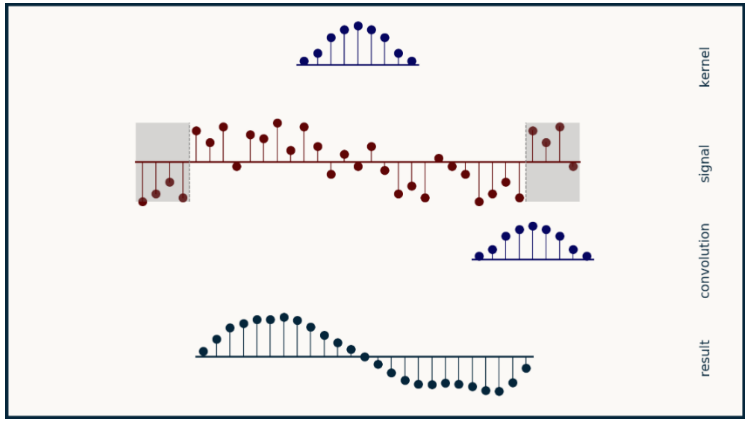
Wavelets

- Energy compaction
  - Wavelets are good
  - Wavelets are used in the JPEG2000 standard for compression
  - But a more complicated wavelet than Haar…

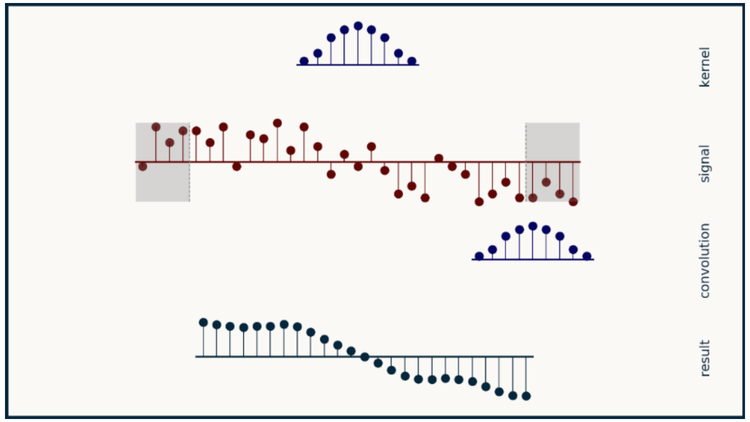
- How to design more complicated wavelets?
  - Vanish more moments
  - i.e. $\int \psi(x)dx = 0$ and $\int x\psi(x)dx = 0$ etc.
  - Fourier analysis can help us:
    - n^{th} moment = 0 → n^{th} derivative of FT at ω = 0 is 0
    - Smoother decay from mid-frequencies to 0 in frequency domain
  - Beyond our scope
  - We won’t derive them, but we will use them

More Energy Compaction

- We saw energy compaction is important
- Other transforms?
- Recall when we talked about boundaries in convolutions
  - Circular extension
  - Corresponds to what was done in DFT
  - We noted that this creates a sharp transition



  - Mirroring extension was another option
  - Better continuity at the boundary
  - This will be the idea behind our new transform



Discrete Cosine Transform

- Useful for compression
  - Images are real, but DFT is complex → not good for storage
  - In image processing, we do a lot of block processing
  - DFT works on periodic extension → discontinuities at the border
  - “Slow” decay of coefficients → poor energy compaction

  - DCT addresses these concerns
  - Real-valued
  - Eliminates discontinuities at block borders

Discrete Cosine Transform

– Mirror extension example

$$N = 4 \qquad x[n] = \{2, 4, 6, 8\}$$

periodic extension would be $\tilde{x}[n] = \{\ldots, 2, 4, 6, 8, 2, 4, 6, 8, \ldots\}$

Instead consider

$$y[n] = \begin{cases} x[n] & 0 \leq n \leq N-1 \\ x[2N-1-n] & N \leq n \leq 2N-1 \\ 0 & \text{otherwise} \end{cases}$$

For the example above $y[n] = \{2, 4, 6, 8, 8, 6, 4, 2\}$

Discrete Cosine Transform

- Take y[n] above and compute its DFT

$$y[n] = \begin{cases} x[n] & 0 \le n \le N-1 \\ x[2N-1-n] & N \le n \le 2N-1 \\ 0 & \text{otherwise} \end{cases}$$

$$Y[k] = \sum_{n=0}^{2N-1} y[n] e^{-i\frac{2\pi}{2N}kn} = \sum_{n=0}^{2N-1} y[n] W_{2N}^{kn} \qquad \text{where} \qquad W_{2N} = e^{-i\frac{2\pi}{2N}}$$

$$= \sum_{n=0}^{N-1} x[n] W_{2N}^{kn} + \sum_{n=N}^{2N-1} x[2N-1-n] W_{2N}^{kn}$$

$$= \sum_{n=0}^{N-1} x[n] W_{2N}^{kn} + \sum_{n=0}^{N-1} x[n] W_{2N}^{k(2N-1-n)}$$

$$= \sum_{n=0}^{N-1} x[n] \left[W_{2N}^{kn} + W_{2N}^{-kn-k} \right]$$

$$= W_{2N}^{-k/2} \sum_{n=0}^{N-1} x[n] \left(W_{2N}^{k(n+\frac{1}{2})} + W_{2N}^{-k(n+\frac{1}{2})} \right) = W_{2N}^{-k/2} \sum_{n=0}^{N-1} x[n] 2\cos\left(\frac{2\pi k(2n+1)}{4} \right)$$

Discrete Cosine Transform

- Take y[n] above and compute its DFT

$$y[n] = \begin{cases} x[n] & 0 \leq n \leq N-1 \\ x[2N-1-n] & N \leq n \leq 2N-1 \\ 0 & \text{otherwise} \end{cases}$$

$$Y[k] = 2W_{2N}^{-k/2} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi k(2n+1)}{4}\right)$$

- *N* point x[*n*] → 2*N* complex Y[0], … Y[2*N*-1]
- How is this compression?
  - Note $Y[N] = \sum_{n=0}^{2N-1} y[n] e^{-i\frac{2\pi}{2N}Nn} = \sum_{n=0}^{2N-1} y[n](-1)^n = 0$ since y[n] is symmetric
  - x[n] real → $W_{2N}^{k/2} Y[k]$ is real
  - $W_{2N}^{(2N-k)/2} Y[2N-k] = W_{2N}^{k/2} Y[k]$ → Only need to save half of Y[*k*]

Discrete Cosine Transform

– Take y[n] above and compute its DFT

$$y[n] = \begin{cases} x[n] & 0 \le n \le N-1 \\ x[2N-1-n] & N \le n \le 2N-1 \\ 0 & \text{otherwise} \end{cases}$$

$$Y[k] = 2W_{2N}^{-k/2} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi k(2n+1)}{4}\right)$$

– 1D DCT is defined as

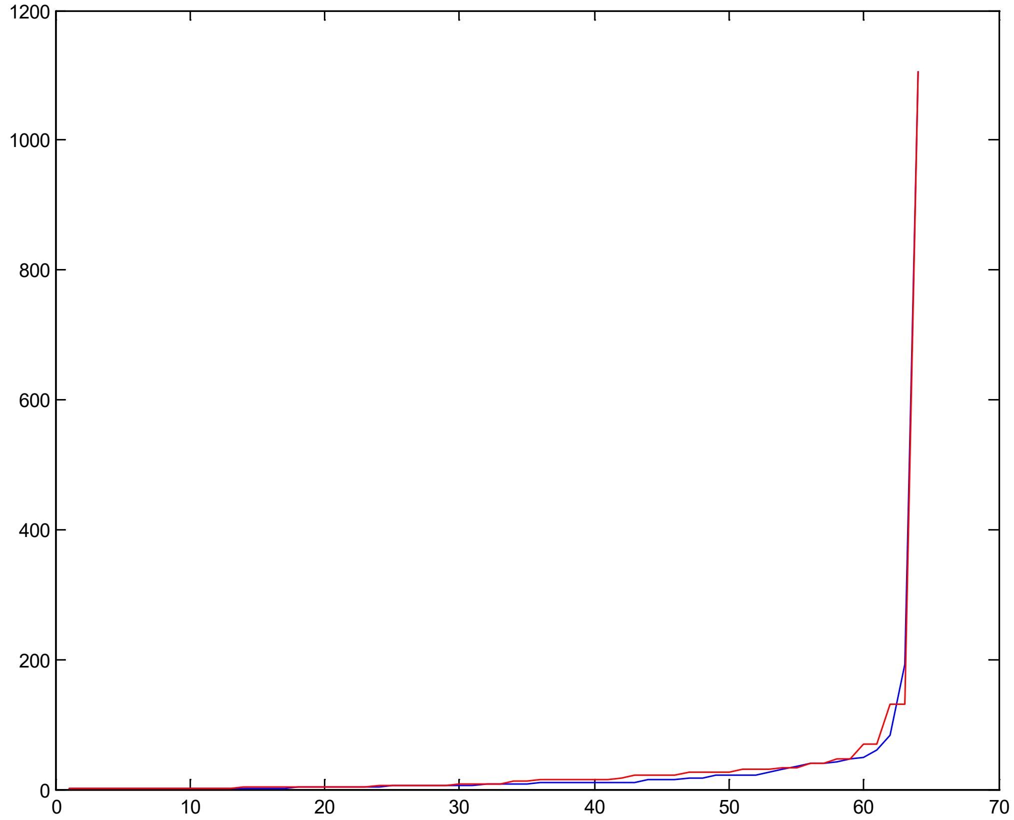
$$C_x[k] = \begin{cases} W_{2N}^{k/2} Y[k] & 0 \le k \le N-1 \\ 0 & \text{otherwise} \end{cases} = \sum_{n=0}^{N-1} 2x[n] \cos\left(\frac{\pi k(2n+1)}{2N}\right) \qquad k = 0, \ldots, N-1$$

- N-point → N-point
- DCT is real if x[n] is real
- DC component $C_x[0] = 2Y[0]$

Discrete Cosine Transform

− 2D DCT is defined analogously with 2D mirroring/symmetrizing extension

Energy “compaction”:
sorted (abs value) DCT (red) vs. DFT (blue) coefficients in an example 8×8 block



Compression: DCT vs. DFT

Original

Compression with DFT

Compression with DCT

6-fold compression

Compression: DCT vs. Wavelet

Original

Compression with DCT

Used in JPEG standard

Compression with Wavelet

Used in JPEG2000 standard

12-fold compression

Random Variables

- Last bit of math to review before starting image restoration

- Probability space

  1. Sample space, Ω
  2. Set of events, $\mathcal{F}$ (collection of subsets of Ω)
  3. Probability measure, $\mathbb{P} : \mathcal{F} \to [0, 1]$

- Random variable is a mapping from sample space (of a probability space) to real (or complex) numbers (in this course)

$$X : \Omega \to E$$

Ω has an underlying probability space (with associated $\mathbb{P}$ and $\mathcal{F}$)
E is a measureable space (for us, $\mathbb{R}$ or $\mathbb{C}$)

Random Variables

- Technically, we will consider $\omega \in \Omega$ and evaluate $X(\omega)$
- Thus when we say $\mathbb{P}(X \in B)$ for $B \subseteq \mathbb{R}$, we mean

$$\mathbb{P}\Big(\{\omega \in \Omega : X(\omega) \in B\}\Big)$$

- Hence $\mu(B) = \mathbb{P}(X \in B)$ is a probability measure & called the distribution of X
- Often times the dependence on $\omega \in \Omega$ will be implicit, and dropped in notation

- The distribution can be described by the cumulative distribution function (CDF) for real-valued X

$$F_X(x) = \mathbb{P}(X \leq x)$$

Random Variables

- As we already saw, we often talk about the probability density function (PDF) for continuous random variables (e.g. Gaussian, uniform)

$$f_X(x) = \frac{dF_X(x)}{dx}$$

and probability mass function (PMF) for discrete random variables (e.g. Bernoulli, binomial, Poisson)

$$p_X(k) = \mathbb{P}(X = k)$$

- Examples
  - Gaussian PDF $$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
  - Poisson PMF $$p_X(k) = e^{-\lambda}\frac{\lambda^k}{k!}$$

Random Variables

– 2^{nd} order properties of random variables

- Mean $\mathbb{E}[X] = \int x f_X(x) dx$
- Expectation of a function of a random variable

$$\mathbb{E}[g(X)] = \int g(x) f_X(x) dx$$

- Variance $\mathrm{Var}\{X\} = \mathbb{E}\Big[X - \mathbb{E}[X]\Big]^2 = \mathbb{E}[X^2] - \mathbb{E}[X]^2$
- Correlation $\mathbb{E}[XY] = \int\int xy f_{XY}(x,y) dx\, dy$

joint PDF of X and Y

- Covariance $\mathrm{Cov}\{X, Y\} = \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)\big(Y - \mathbb{E}[Y]\big)\Big]$

Random Vectors

- A finite collection of random variables over a common probability space

$$\underline{\mathbf{X}} = (X_1, \ldots, X_n) \qquad \text{where each } X_i \text{ is a random variable}$$

- We can define the CDF

$$F_{\underline{\mathbf{X}}}(x_1, \ldots, x_n) = \mathbb{P}(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n)$$

- Corresponding PDF is

$$f_{\underline{\mathbf{X}}}(\mathbf{x}) = \frac{\partial^n}{\partial x_1 \ldots \partial x_n} F_{\underline{\mathbf{X}}}(x_1, \ldots, x_n)$$

- Similarly

$$\mathbb{P}(\underline{\mathbf{X}} \in B) = \int_B f_{\underline{\mathbf{X}}}(\mathbf{x}) d\mathbf{x} \qquad B \subseteq \mathbb{R}^n$$

Random Vectors

– 2nd order properties

- Mean

$$\mathbb{E}[\underline{\mathbf{X}}] = \int_{\mathbb{R}^n} \mathbf{x} f_{\underline{\mathbf{X}}}(\mathbf{x}) d\mathbf{x} = \begin{bmatrix} \mathbb{E}[X_1] \\ \mathbb{E}[X_2] \\ \vdots \\ \mathbb{E}[X_n] \end{bmatrix}$$

- Expectation of a function of a random vector

$$\mathbb{E}[g(\underline{\mathbf{X}})] = \int_{\mathbb{R}^n} g(\mathbf{x}) f_{\underline{\mathbf{X}}}(\mathbf{x}) d\mathbf{x}$$

- *n*×*n* correlation matrix

$$\mathbf{R}_{\underline{\mathbf{X}}} = \mathbb{E}\left[\underline{\mathbf{X}}\,\underline{\mathbf{X}}^H\right]$$

$$\text{i.e. } (\mathbf{R}_{\underline{\mathbf{X}}})_{ij} = \mathbb{E}[X_i X_j] = \int_{\mathbb{R}^n} x_i x_j f_{\underline{\mathbf{X}}}(x_1, \ldots, x_n) d\mathbf{x}$$

Random Vectors

- 2nd order properties
  - $n \times n$ covariance matrix

$$\mathbf{K}_{\underline{\mathbf{X}}} = \mathrm{Cov}\{\underline{\mathbf{X}}\} = \mathbb{E}\Big[\big(\underline{\mathbf{X}} - \mathbb{E}[\underline{\mathbf{X}}]\big)\big(\underline{\mathbf{X}} - \mathbb{E}[\underline{\mathbf{X}}]\big)^H\Big]$$

$$= \mathbf{R}_{\underline{\mathbf{X}}} - \mathbb{E}[\underline{\mathbf{X}}]\mathbb{E}[\underline{\mathbf{X}}]^H$$

$$\text{i.e. } (\mathbf{K}_{\underline{\mathbf{X}}})_{ij} = \mathrm{Cov}\{X_i, X_j\}$$

Random Vectors

− Example: Gaussian random vector

$$\underline{\mathbf{X}} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$$

n×1 mean vector

n×*n* covariance matrix

$$\boldsymbol{\mu} = \mathbb{E}[\underline{\mathbf{X}}]$$
$$\mathbf{K} = \mathrm{Cov}\{\underline{\mathbf{X}}\}$$

PDF is given as

$$f_{\underline{\mathbf{X}}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\det(\mathbf{K})|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{K}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$