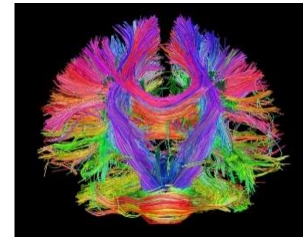



CS-570
Statistical Signal Processing

Lecture 5: Signal Representation

Spring Semester 2019

Grigorios Tsagkatakis

Today's Objectives

Topics:

- Fourier and Wavelet Transform
- Sparsity and Dictionary Learning

Disclaimer: Material used:

Applied Digital Signal Processing, Manolakis and Ingle

Linear inverse problems

- Many classic problems in computer can be posed as linear inverse problems

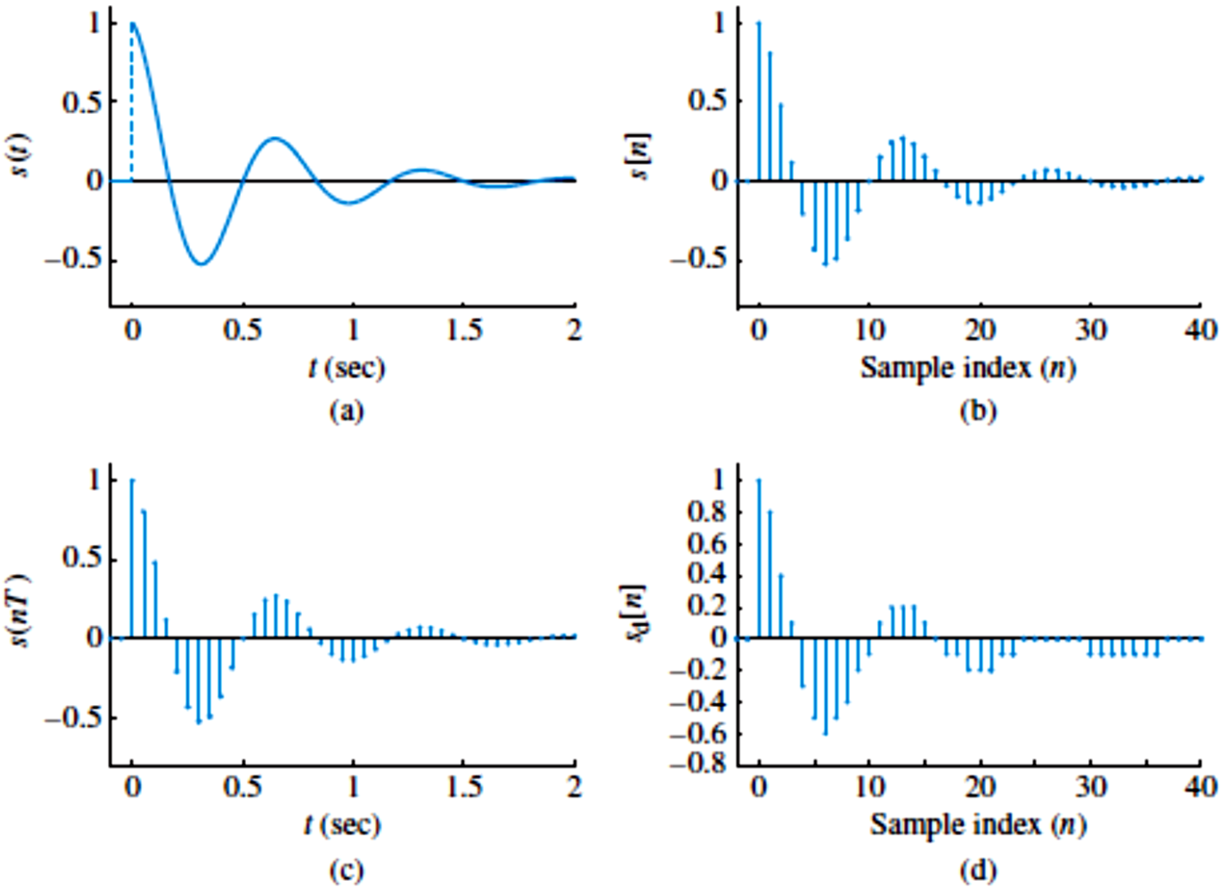
- Notation
  - **Signal** of interest $x \in \mathbb{R}^N$
  - **Observations** $y \in \mathbb{R}^M$
  - Measurement model $y = \Phi x + e$

measurement matrix (Φ)

measurement noise (e)

- Problem definition: given y, recover x

Signals



Plots illustrating the graphical representation of continuous-time signals (a), discrete-time signals (b) and (c), and digital signals (d).

Systems

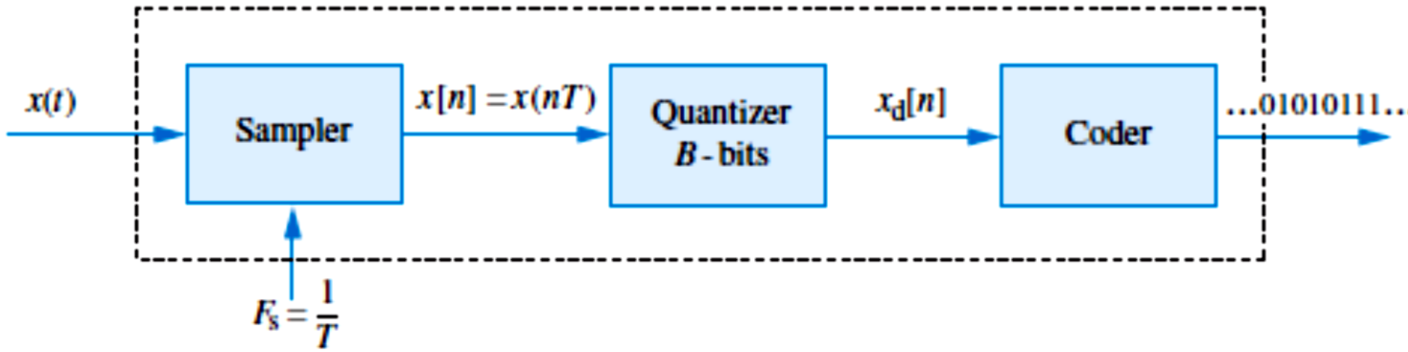
- A *continuous-time system* is a system which transforms a continuous- time input signal *x(t)* into a continuous-time output signal *y(t)*.

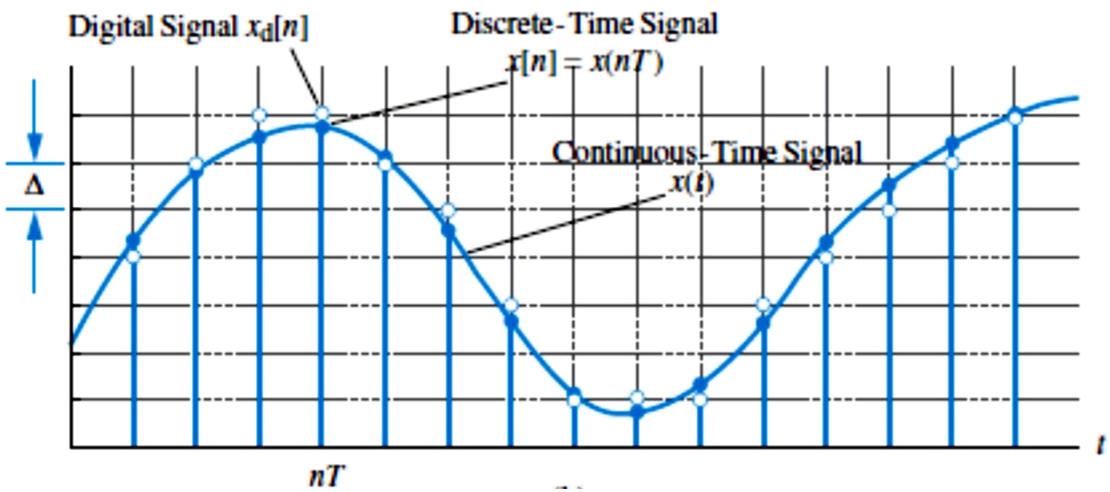
$$y(t) = \int_{-\infty}^{t} x(\tau)d\tau \qquad x(t) \xrightarrow{\mathcal{H}} y(t) \quad \text{or} \quad y(t) = \mathcal{H}\{x(t)\},$$

- A system that transforms a discrete-time input signal *x*[*n*] into a discrete-time output signal *y*[*n*], is called a *discrete-time system*.

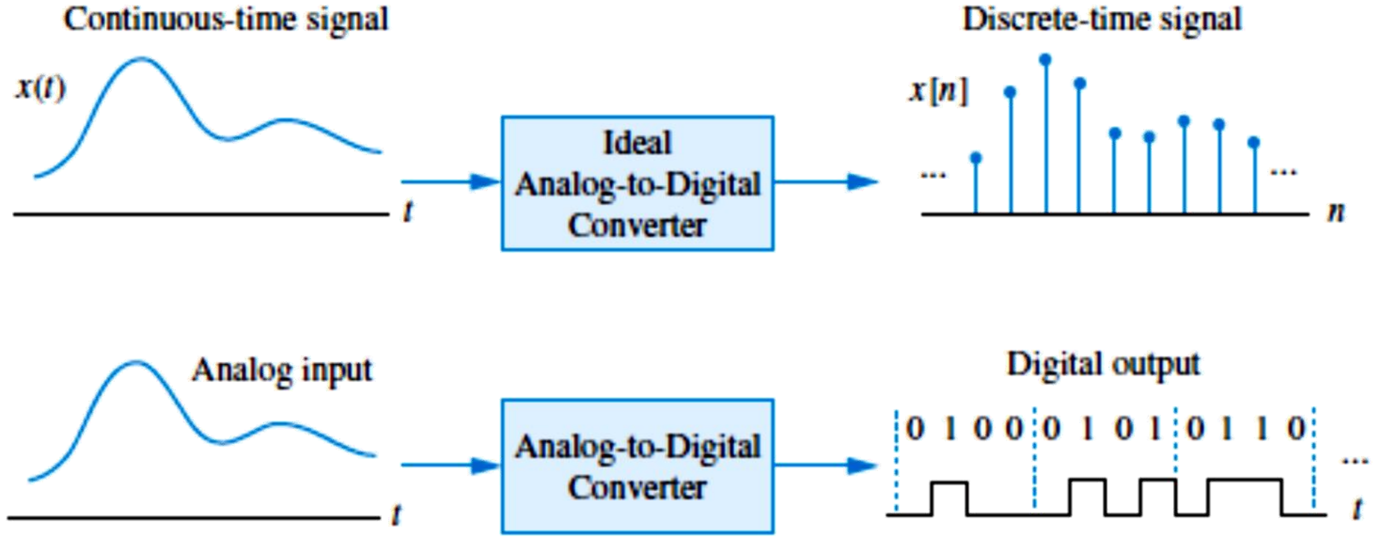
$$y[n] = \sum_{k=-\infty}^{n} x[k]. \qquad x[n] \xrightarrow{\mathcal{H}} y[n] \quad \text{or} \quad y[n] = \mathcal{H}\{x[n]\},$$

Analog-to-digital





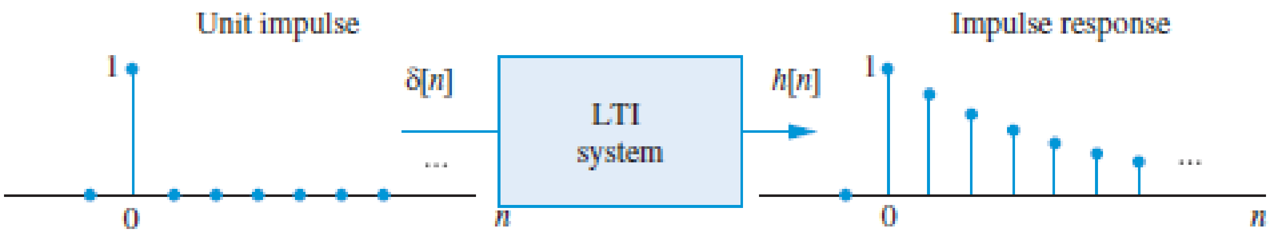
Analog-to-digital



(b)

Discrete-time system properties

Property	Input		Output
	$x[n]$	$\overset{\mathcal{H}}{\longmapsto}$	$y[n]$
	$x_k[n]$	$\overset{\mathcal{H}}{\longmapsto}$	$y_k[n]$
Linearity	$\sum_k c_k x_k[n]$	$\overset{\mathcal{H}}{\longmapsto}$	$\sum_k c_k y_k[n]$
Time-invariance	$x[n-n_0]$	$\overset{\mathcal{H}}{\longmapsto}$	$y[n-n_0]$
Stability	$\|x[n]\| \le M_x < \infty$	$\overset{\mathcal{H}}{\longmapsto}$	$\|y[n]\| \le M_y < \infty$
Causality	$x[n] = 0$ for $n \le n_0$	$\overset{\mathcal{H}}{\longmapsto}$	$y[n] = 0$ for $n \le n_0$

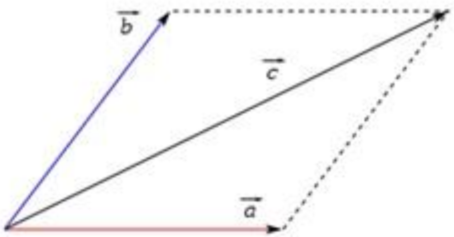


The impulse response of a linear time-invariant system

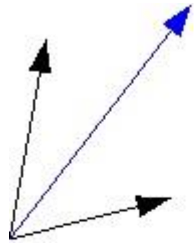
Principle of superposition

Examples:

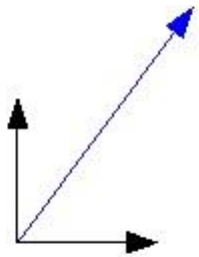
Vector addition



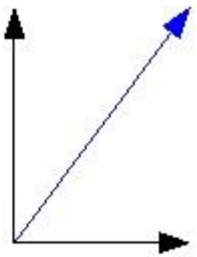
Especially useful
Combine vectors from basis vectors
(linear independent vectors)



Even more useful
orthonormal basis vectors



$$\boldsymbol{y} = \sum_{i=1}^{N} x_i e_i$$



$$\boldsymbol{y} = x_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + x_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

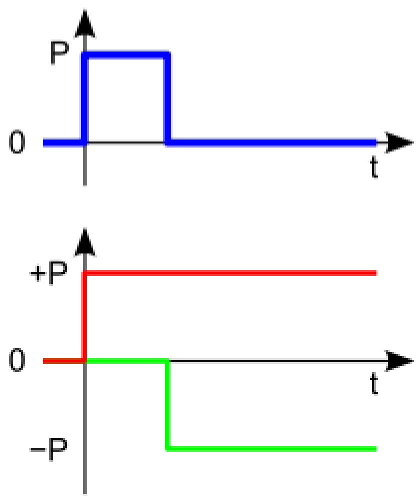
projections can easily be calculated
with the scalar product because $e_i e_j = \delta_{ij}$

Principle of superposition

Superposition = representation as linear combination:

same idea for functions:

$$x(t) = \sum_{i=1}^{n} a_i x_i(t)$$



built a rect- from step-functions

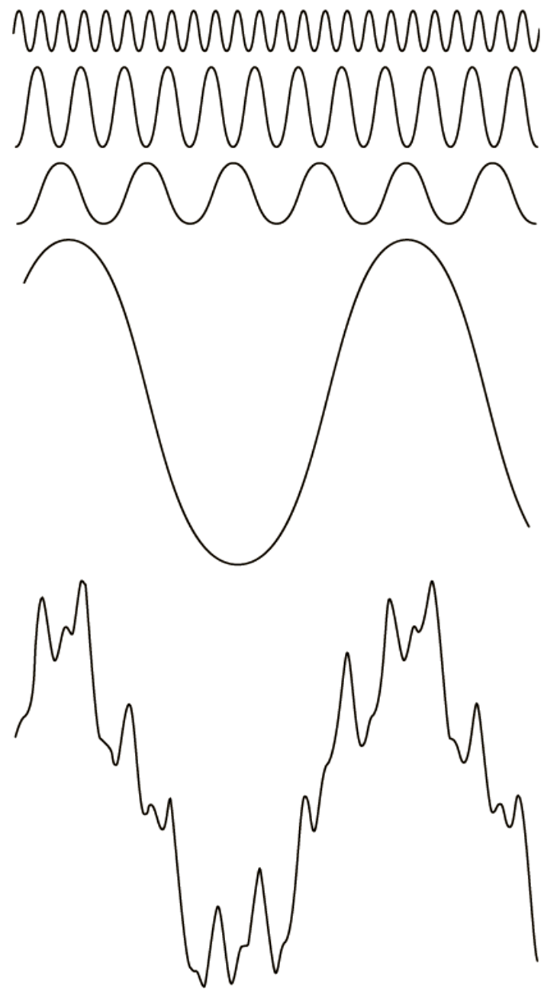
$$f(x) = \int_{-\infty}^{+\infty} f(y)\delta(x-y)\,dy$$

split a function into pointwise contributions
with the δ-function (distribution)

$$1 = \int_{-\infty}^{+\infty} \delta(y)\,dy$$

$$f(0) = \int_{-\infty}^{+\infty} f(y)\delta(y)\,dy$$

Fourier Series: Example



Fourier Series and Fourier Transform: History

- Fourier Series

  Any periodic function can be expressed as the sum of sines and /or cosines of different frequencies, each multiplied by a different coefficients

- Fourier Transform

  Any function that is not periodic can be expressed as the integral of sines and /or cosines multiplied by a weighing function

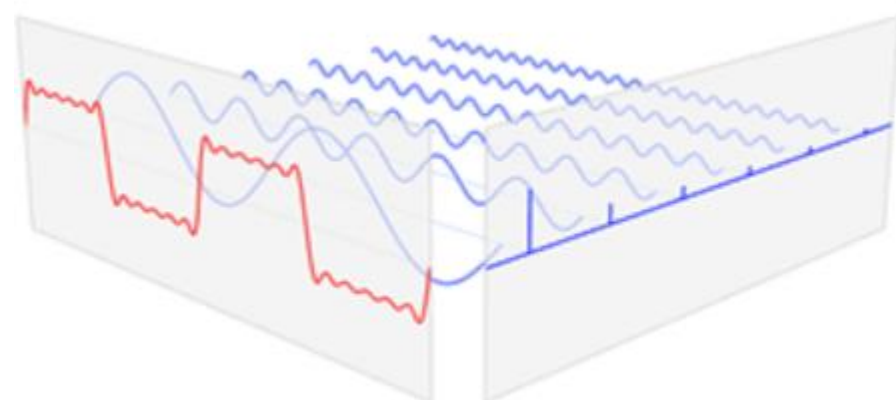
Fourier series

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right)$$

$$a_0 = \frac{1}{L}\int_0^{2L} f(x)\, dx$$

$$a_n = \frac{1}{L}\int_0^{2L} f(x) \cos\left(\frac{n\pi x}{L}\right) dx$$

$$b_n = \frac{1}{L}\int_0^{2L} f(x) \sin\left(\frac{n\pi x}{L}\right) dx$$



Discrete Fourier Transform

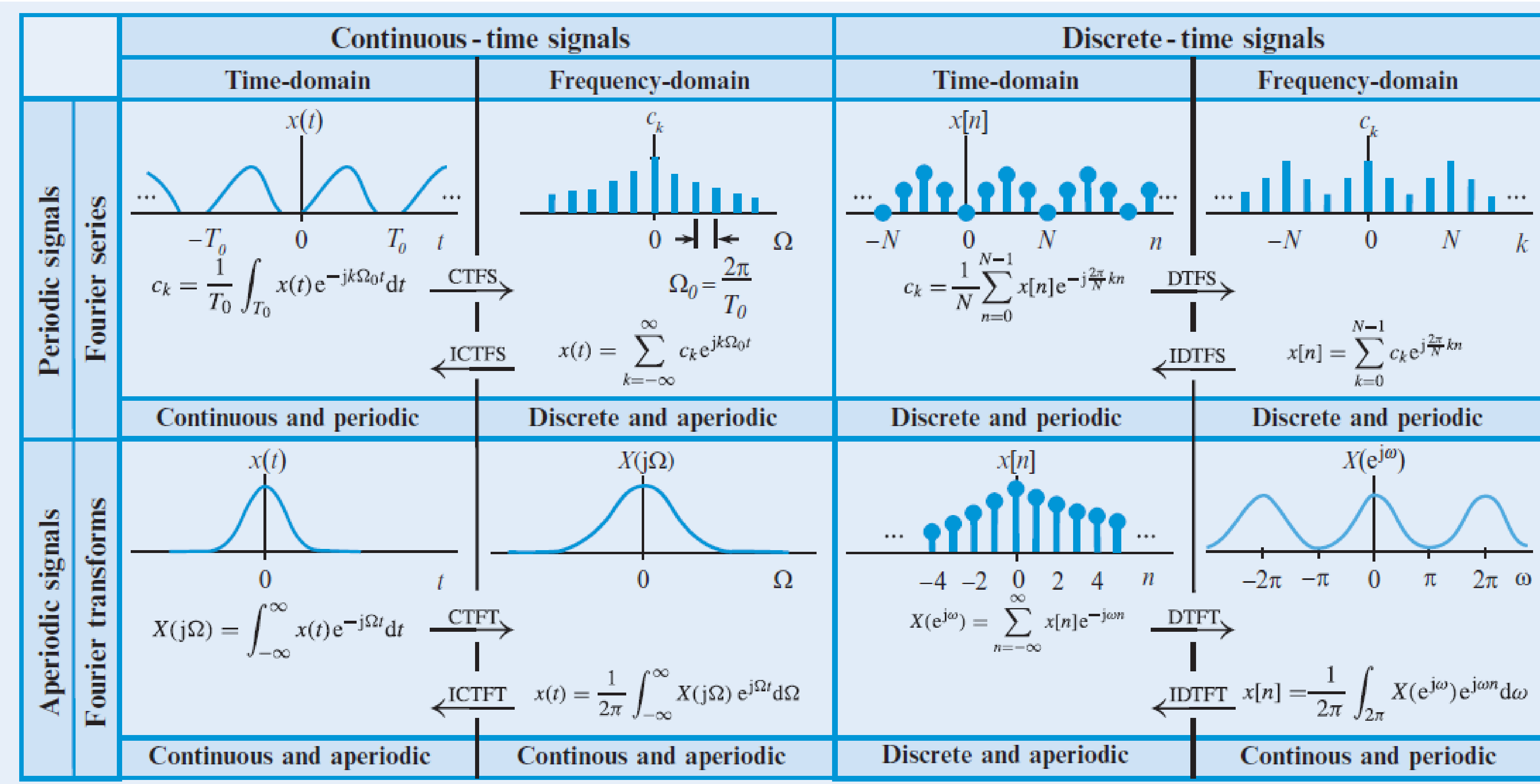
- A discrete signal x[n] of length N can be represented as

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-jk\omega_0 n}$$

- Given a representation X[k], the original signal can be recovered by

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{jk\omega_0 n}$$

Fourier representation of signals



Why?

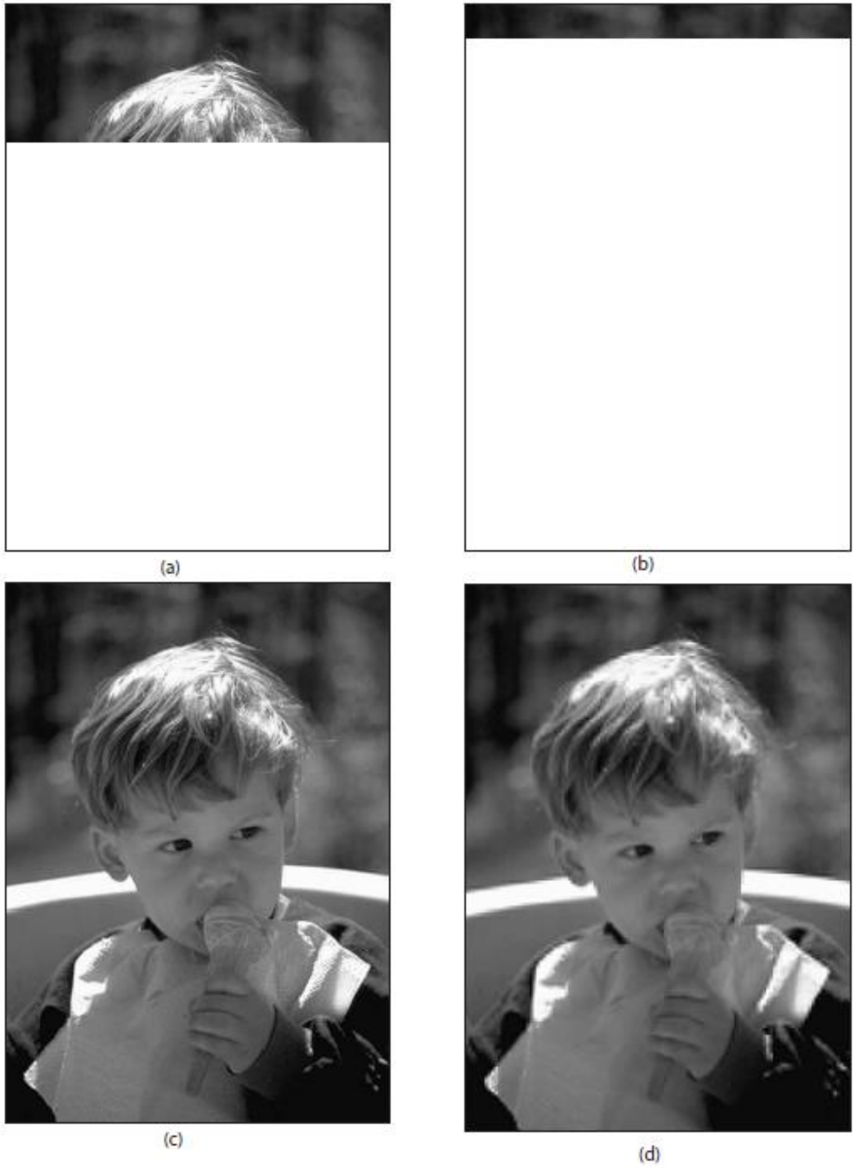
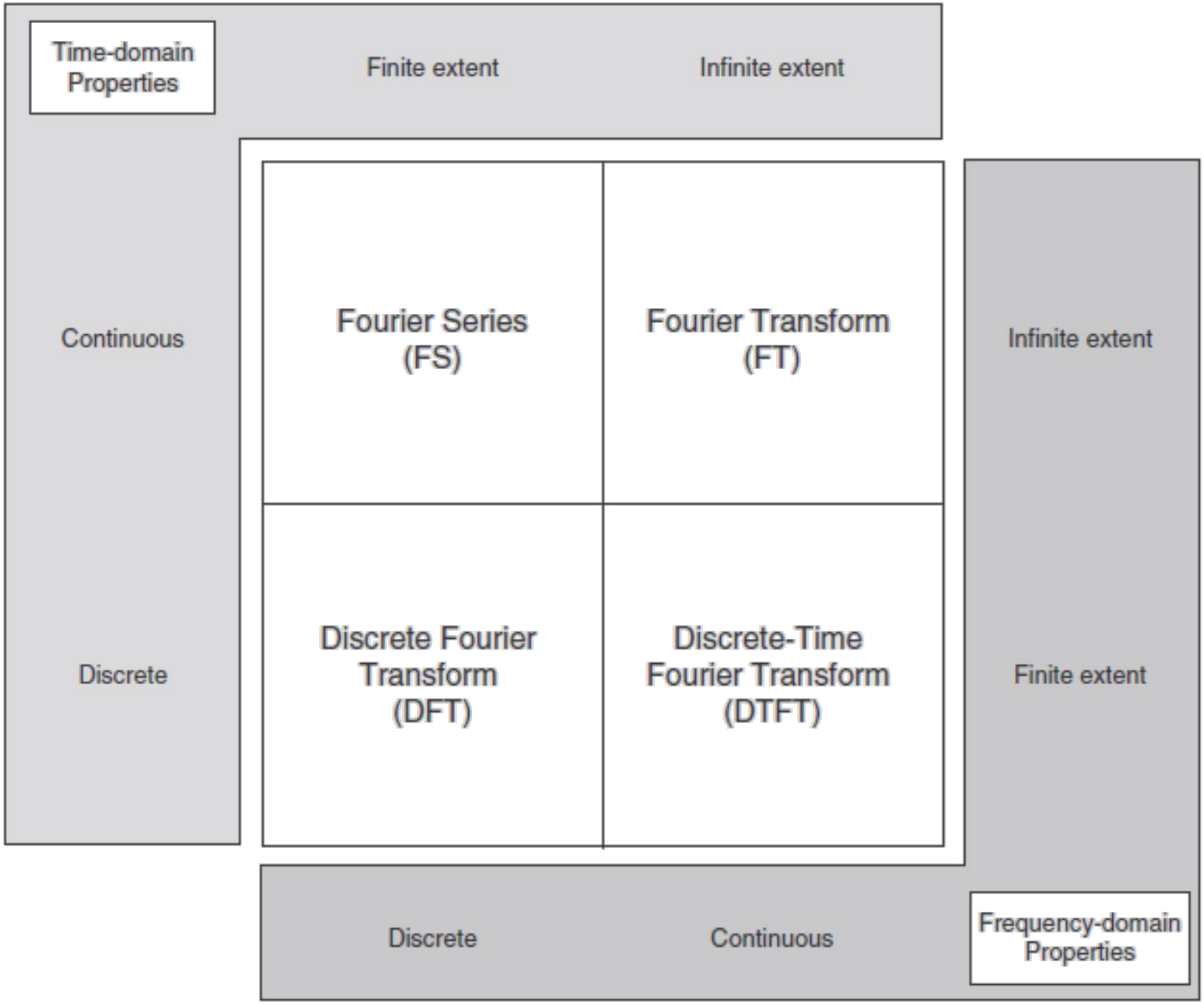


Figure 4.3: Sending image data in usual format vs. sending only low frequency data. a) 25% of data in usual format. b) 6.25% of data in usual format. c) 25% of lowest frequency data. d) 6.25% of lowest frequency data.


Time-domain Properties
Finite extent
Infinite extent
Continuous
Discrete
Fourier Series (FS)
Fourier Transform (FT)
Discrete Fourier Transform (DFT)
Discrete-Time Fourier Transform (DTFT)
Infinite extent
Finite extent
Discrete
Continuous
Frequency-domain Properties

The Discrete Fourier Transform (DFT) matrix

- The $n \times n$ Fourier matrix is defined as:

$$F_n = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & \dots & w^{(n-1)} \\ 1 & w^2 & w^4 & \dots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{(n-1)} & w^{2(n-1)} & \dots & w^{(n-1)(n-1)} \end{bmatrix}$$

- Number the first row and column with 0.
- We define $w = e^{-\mathbf{i}\frac{2\pi}{n}}$. For w is preferable to use polar representation. $F_n(i,j) = w^{ij}$.

The DFT matrix for n=4

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -\mathbf{i} & (-\mathbf{i})^2 & (-\mathbf{i})^3 \\ 1 & (-\mathbf{i})^2 & (-\mathbf{i})^4 & (-\mathbf{i})^6 \\ 1 & (-\mathbf{i})^3 & (-\mathbf{i})^6 & (-\mathbf{i})^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -\mathbf{i} & (-\mathbf{i})^2 & (-\mathbf{i})^3 \\ 1 & (-\mathbf{i})^2 & (-\mathbf{i})^0 & (-\mathbf{i})^2 \\ 1 & (-\mathbf{i})^3 & (-\mathbf{i})^2 & (-\mathbf{i})^1 \end{bmatrix}$$

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

The columns of this matrix are orthogonal. $F_4=0.5*F_4$ to make them orthonormal

DFT in matrix form

- Fourier-Matrix $$(W_N)_{jk} = \frac{1}{N} e^{-2\pi i \frac{jk}{N}}$$
- DFT of x: $$\hat{x} = (\hat{x}_0, ..., \hat{x}_{N-1})$$

$$\hat{x} = W_N x$$

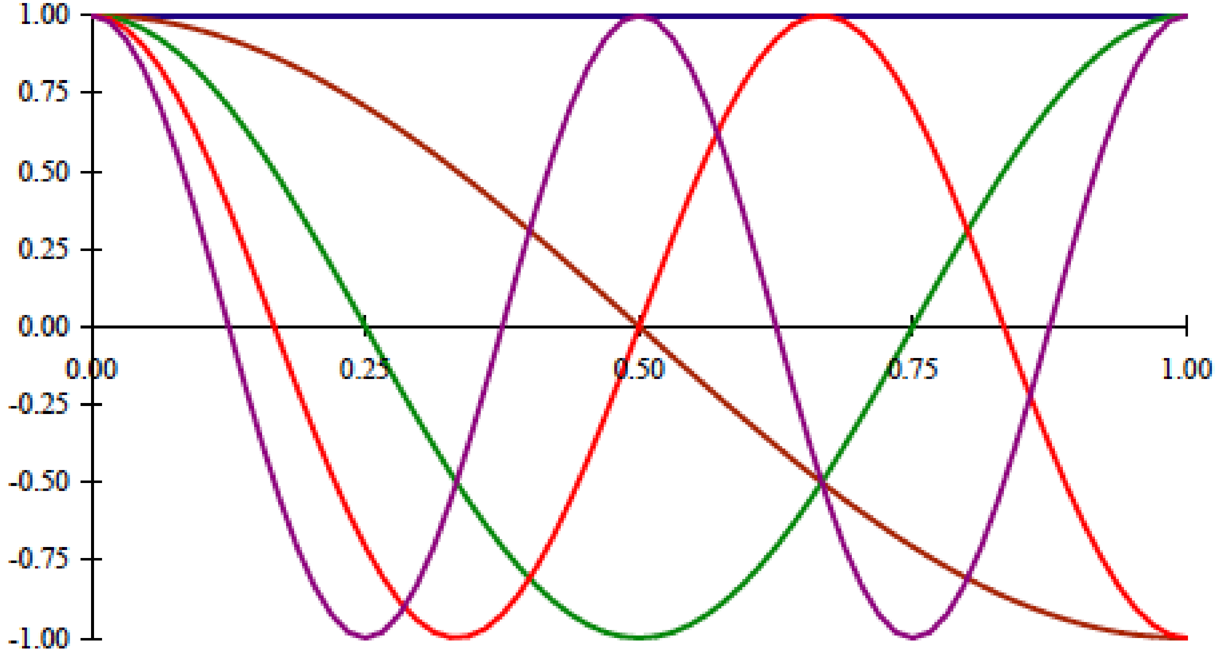
- Matrix-Vector-Product:
  - N^2 Multiplications
  - N(N-1) Additions
  - Arithmetic Complexity: O(N^2)

Discrete Cosine Transform (DCT)

- A variant of the Discrete Fourier Transform – using only real numbers
- Periodic and symmetric
- The energy of a DCT transformed data (if the original data is correlated) is concentrated in a few coefficients – well suited for compression.

1-D DCT

$$F(k) = \sum_{i=0}^{N-1} \cos\left(\frac{\pi}{N}(i+\frac{1}{2})k\right) f(i)$$



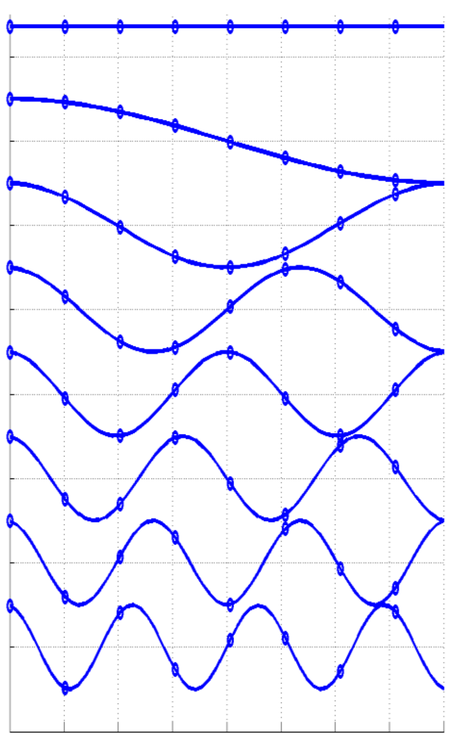
1-D DCT

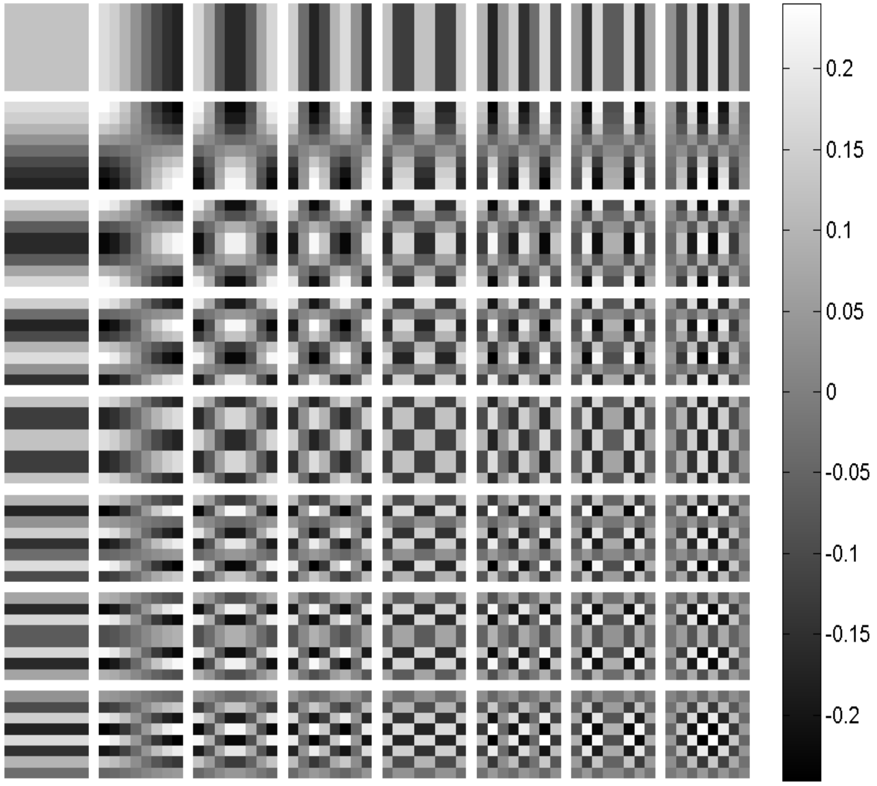
- The one-dimensional **DCT** of order n is defined by an $N \times N$ matrix C whose entries are

$$c(k,n) = \begin{cases} \dfrac{1}{\sqrt{N}}, & k = 0, 0 \le n < N-1 \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{\pi(2n+1)k}{2N}, & 1 \le k \le N-1, 0 \le n < N-1 \end{cases}$$

$$C = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{2n} & \cos\frac{3\pi}{2n} & \cdots & \cos\frac{(2n-1)\pi}{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \cos\frac{(n-1)\pi}{2n} & \cos\frac{(n-1)3\pi}{2n} & \cdots & \cos\frac{(n-1)(2n-1)\pi}{2n} \end{bmatrix}$$

From 1D-DCT to 2D-DCT





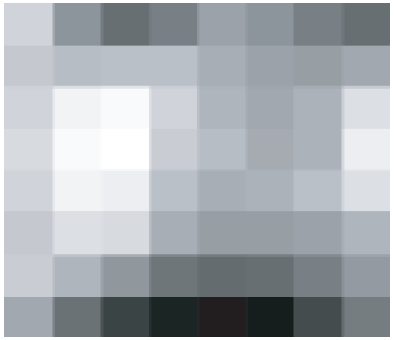
2D DCT

$$F(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \cos\left(\frac{\pi}{N}(i+\frac{1}{2})k\right)\cos\left(\frac{\pi}{N}(j+\frac{1}{2})l\right)f(i,j)$$

Like the 2D Fast Fourier Transform, the 2D DCT can be implemented in two stages, i.e., first
computing the DCT of each line in the block and then computing the DCT of each resulting column.

Like the FFT, each of the DCTs can also be computed in O(N logN) time.

DCT



PICTURE MATRIX

40	24	15	19	28	24	19	15
38	34	35	35	31	28	27	29
40	47	49	40	33	29	32	43
42	49	50	39	34	30	32	46
40	47	46	35	31	32	35	43
38	43	42	31	27	27	28	33
39	33	25	17	14	15	19	26
29	16	6	1	-4	0	7	18

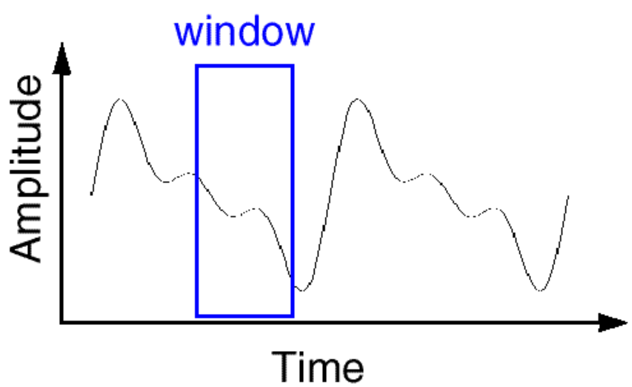
Transformation

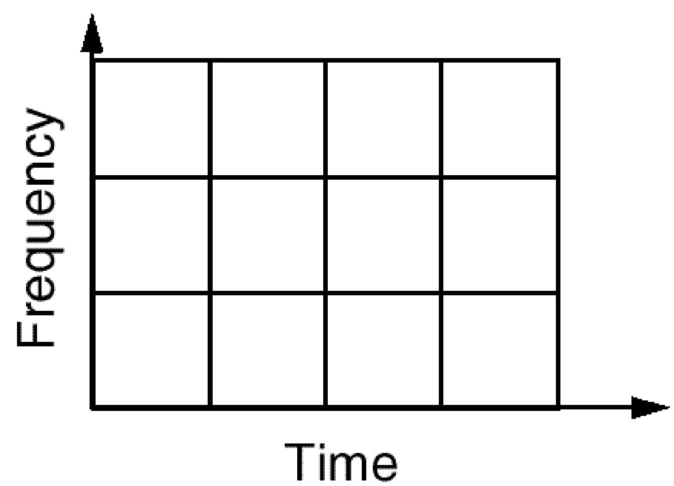
DCT COEFFICENTS

239	32	27	-12	3	-5	3	1
34	-3	-19	6	3	0	-1	1
-70	2	8	23	9	6	-1	-1
5	0	-6	11	-2	0	-1	1
-17	-3	6	6	3	-1	0	0
2	4	2	2	1	-2	0	1
-3	0	0	-1	-1	-1	0	0
1	-1	3	1	0	0	0	0

Short Time Fourier Analysis

In order to analyze small section of a signal, Denis Gabor (1946), developed a technique, based on the FT and using *windowing* : STFT





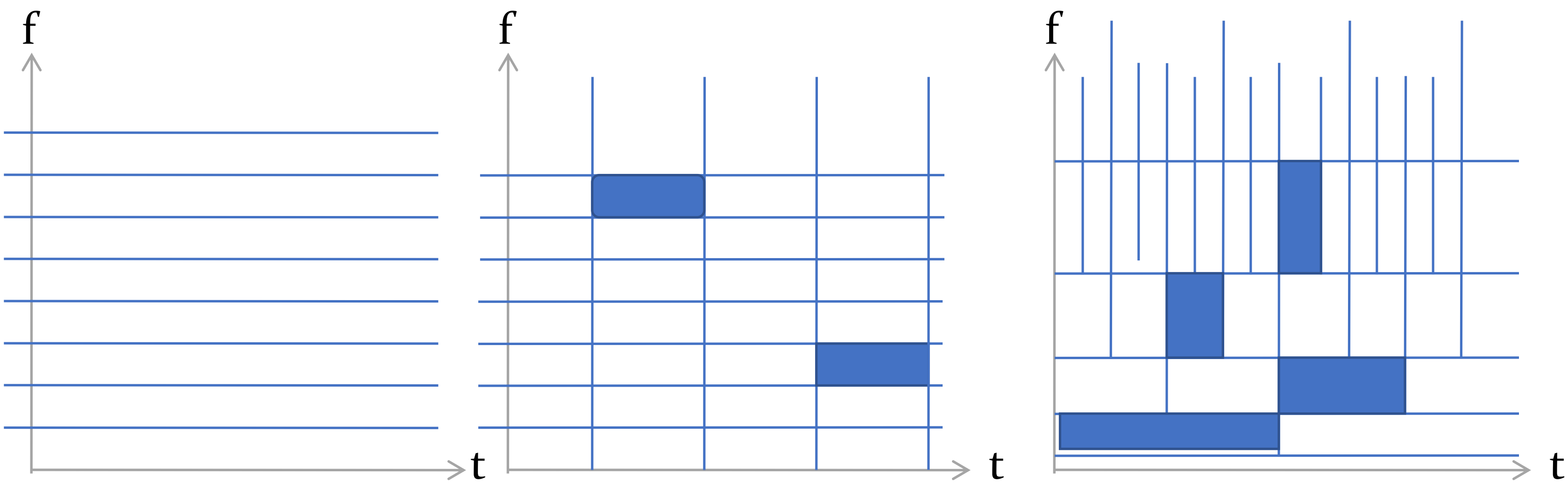
STFT (or: Gabor Transform)

- A compromise between time-based and frequency-based views of a signal.
- both time and frequency are represented in limited precision.
- The precision is determined by the size of the window.
- Once you choose a particular size for the time window - it will be the same for all frequencies.

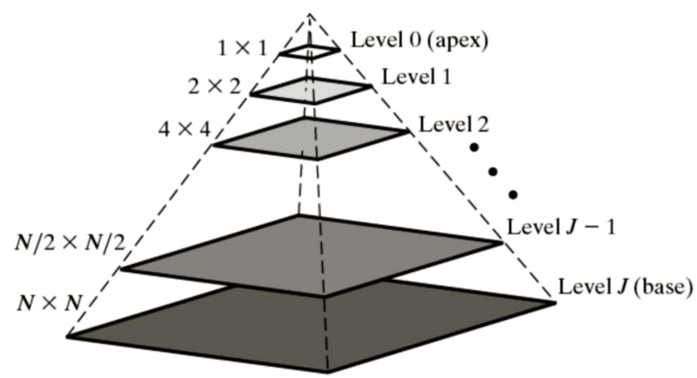
- Limitation

Many signals require a more flexible approach - so we can <u>vary the window size</u> to determine more accurately either time or frequency.

MultiResolution Analysis

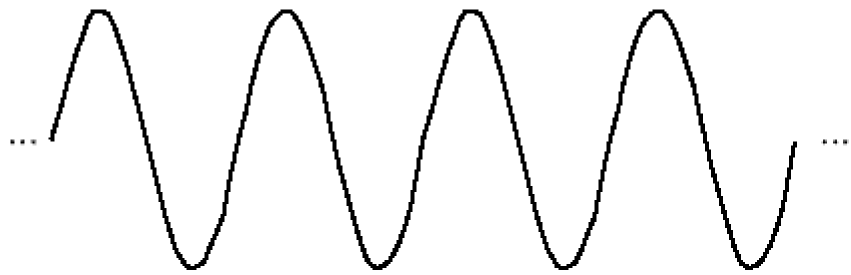


DFT STFT DWT

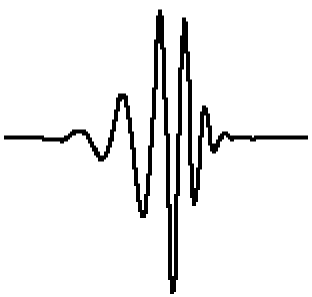


What is Wavelet Analysis ?

- And…what is a wavelet…?



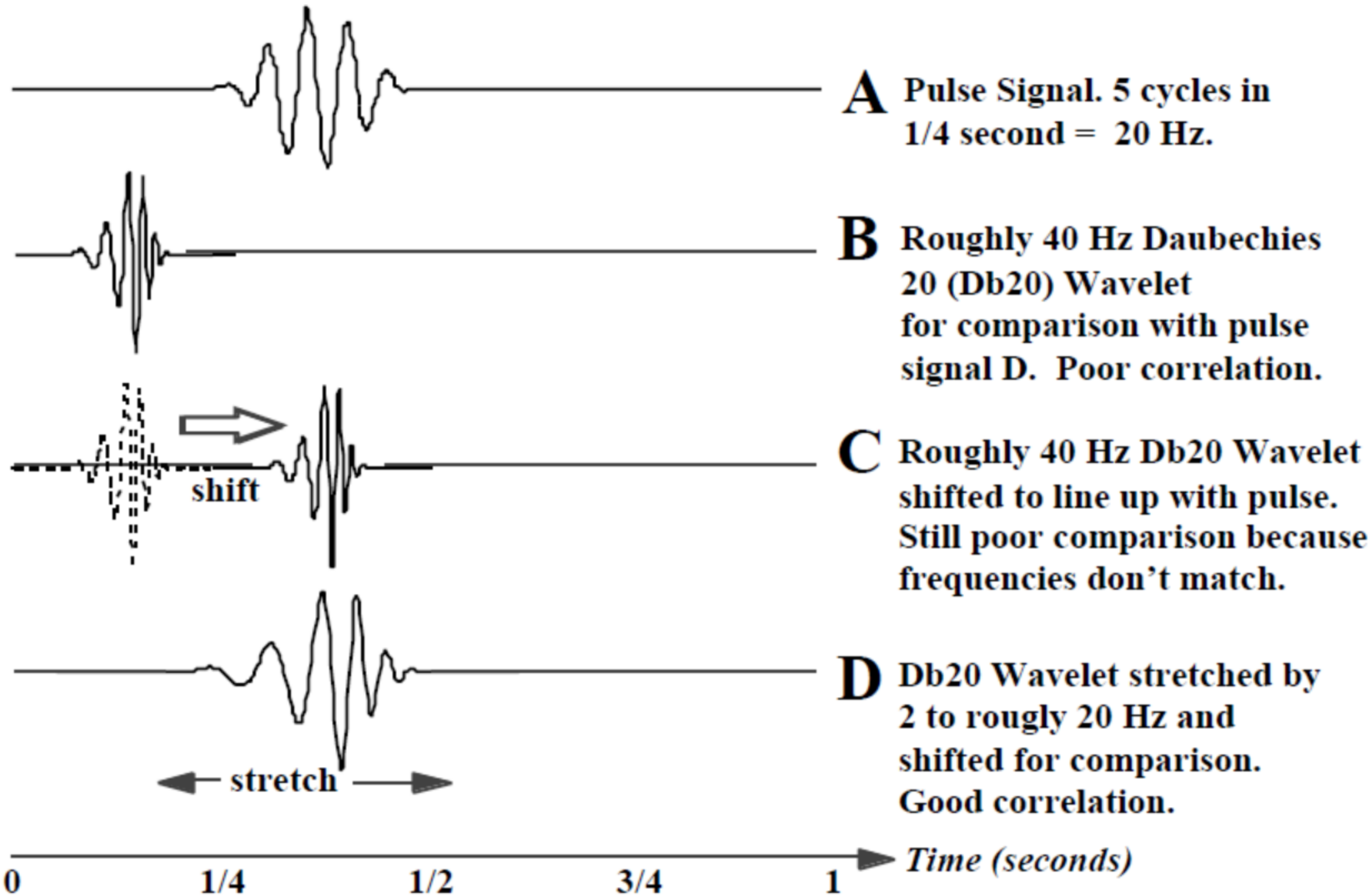
Sine Wave



Wavelet (db10)

- A wavelet is a waveform of effectively limited duration that has an average value of zero.
- Properties
  - Short time localized waves with zero integral value.
  - Possibility of time shifting.
  - Flexibility.

Shift & Scale


A
Pulse Signal. 5 cycles in
1/4 second = 20 Hz.
B
Roughly 40 Hz Daubechies
20 (Db20) Wavelet
for comparison with pulse
signal D. Poor correlation.
shift
C
Roughly 40 Hz Db20 Wavelet
shifted to line up with pulse.
Still poor comparison because
frequencies don't match.
stretch
D
Db20 Wavelet stretched by
2 to rougly 20 Hz and
shifted for comparison.
Good correlation.
0
1/4
1/2
3/4
1
Time (seconds)

Wavelets

- Wavelet transform decomposes a signal into a set of basis functions.
- These basis functions are called *wavelets*
- Wavelets are obtained from a single prototype wavelet *ψ(t)* called mother *wavelet* by *dilations* and *shifting:*

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$$

where *a* is the scaling parameter and *b* is the shifting parameter

Discrete Wavelet Transform (DWT)

- Forward $$a_{jk} = \sum_{t} f(t)\psi^{*}_{jk}(t)$$

- Inverse $$f(t) = \sum_{k}\sum_{j} a_{jk}\psi_{jk}(t)$$

Where $$\psi_{jk}(t) = 2^{j/2}\psi\left(2^{j}t - k\right)$$

DWT in images