AIMS CDT - Signal Processing

Michaelmas Term 2025

Xiaowen Dong

Department of Engineering Science

Logistics

- Lectures
  - Monday-Thursday 10.00-12.30 @IEB
  - slides: http://www.robots.ox.ac.uk/~xdong/teaching.html
- Lab sessions
  - Monday-Thursday 14.00-17.00 @EH
  - notes: http://www.robots.ox.ac.uk/~xdong/teaching.html
  - demonstrators
    - Yin-Cong Zhi (yin-cong.zhi@ndph.ox.ac.uk)
    - Scott le Reux (scott.leroux@wolfson.ox.ac.uk)
    - Ning Zhang (ning.zhang@some.ox.ac.uk)
    - Jacob Bamberger (jacob.bamberger@some.ox.ac.uk)
- Questions & Comments: xdong@robots.ox.ac.uk

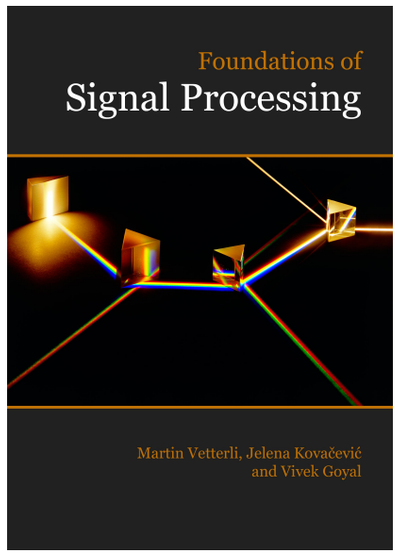
Overview

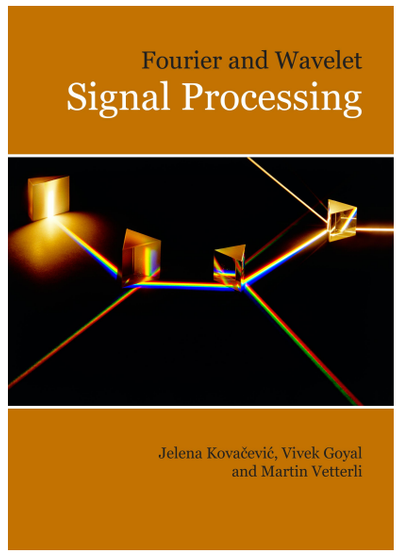
- Lecture 1: Introduction to signal processing
  - time-frequency analysis, filtering, Fourier & wavelet transforms, dictionary learning
  - Lab 1: Signal and image processing
- Lecture 2: Introduction to graph signal processing
  - graph Fourier transform, filtering & convolution, representation of graph signals
  - Lab 2: Graph signal processing
- Lecture 3: Deep learning on graphs
  - convolutional neural networks on graphs, message passing neural networks
  - Lab 3: Graph neural networks
- Lecture 4: Bayesian modelling of graph-structured data
  - Gaussian processes on graphs, Bayesian optimisation of graph-based functions
  - Lab 4: Gaussian processes on graphs

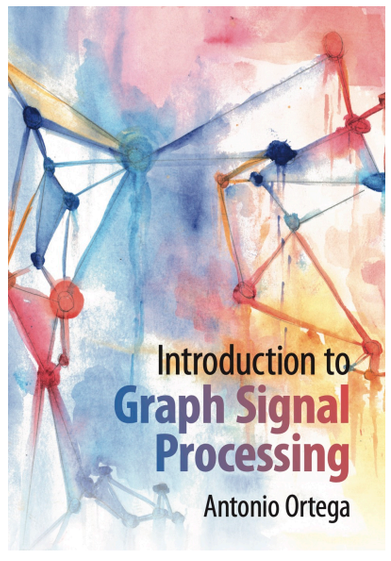
Resources

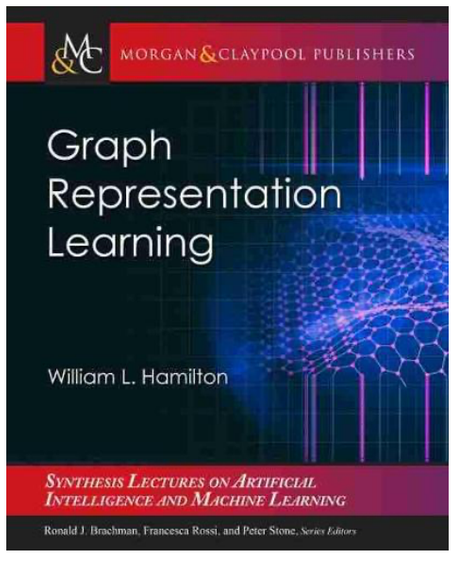
Textbooks

- Vetterli et al. Foundations of signal processing. Cambridge University Press, 2014. Available at http://www.fourierandwavelets.org
- Kovačević et al. Fourier and wavelet signal processing. Available at http://www.fourierandwavelets.org
- Ortega. Introduction to graph signal processing. Cambridge University Press, 2022.
- Hamilton. Graph representation learning. Morgan & Claypool Publishers, 2020. Available at https://www.cs.mcgill.ca/~wlh/grl_book/









Resources

- https://web.media.mit.edu/~xdong/resource.html
- https://github.com/naganandy/graph-based-deep-learning-literature

Introduction to Signal Processing

Lecture 1

- Introduction & Basic concepts and tools
- A historical overview of signal representation techniques
- Applications & Discussion

Historical notes

smoke signal tower (1570)

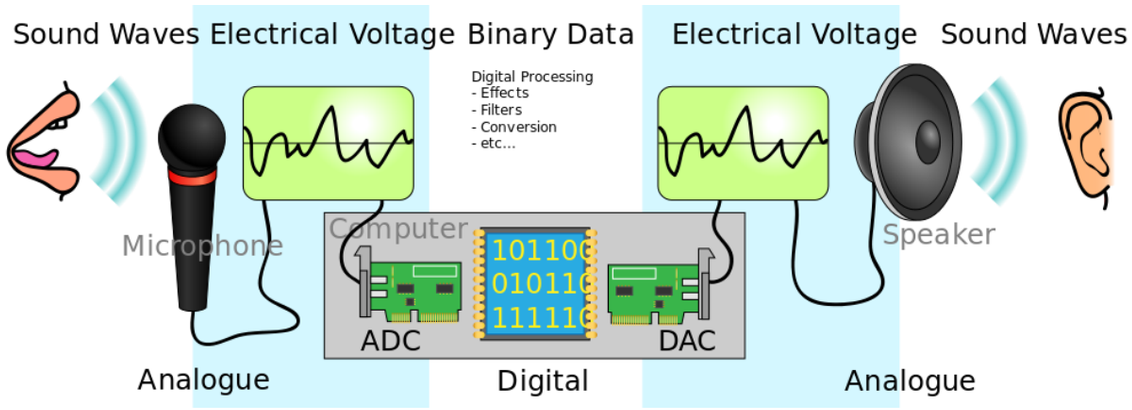
semaphore telegraph (1792)

A ●−
B −●●●
C −●−●
D −●●
E ●
F ●●−●
G −−●
H ●●●●
I ●●
J ●−−−
K −●−
L ●−●●
M −−
N −●
O −−−
P ●−−●
Q −−●−
R ●−●
S ●●●
T −
U ●●−
V ●●●−
W ●−−
X −●●−
Y −●−−
Z −−●●

Morse code (1830s)

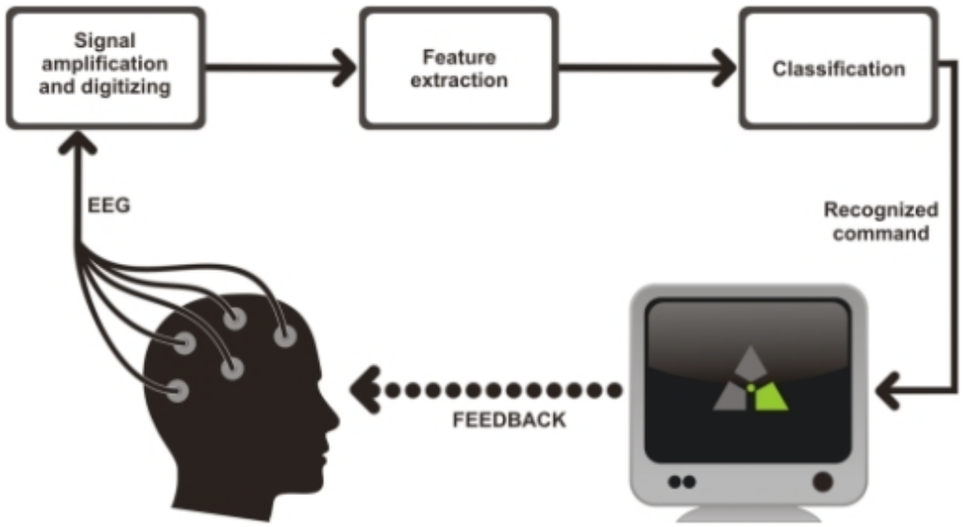
electronic communication (today)

Modern signal processing applications

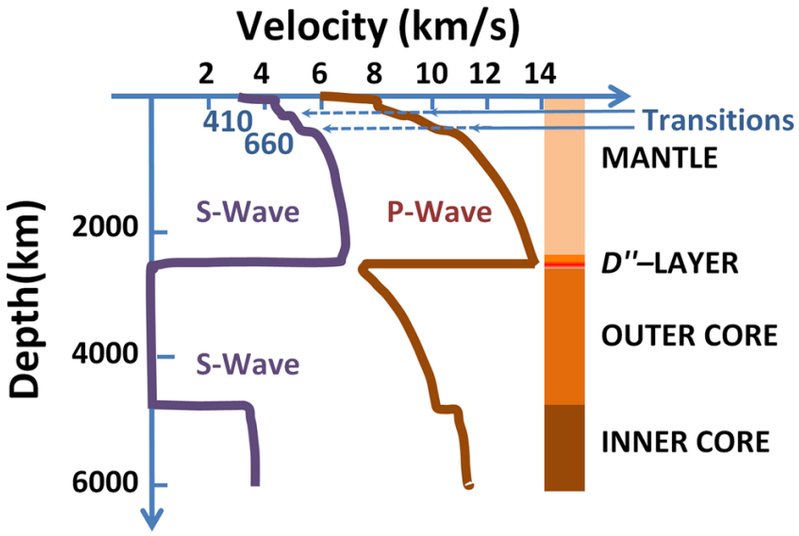


speech processing

image denoising



EEG signal classification



seismology

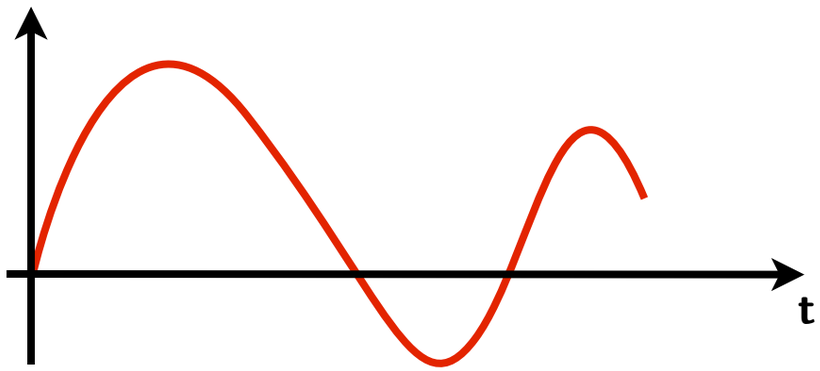
Signal types

amplitude \ time	continuous	discrete
continuous		
discrete		

Signal types

continuous **time** discrete

continuous **amplitude** discrete

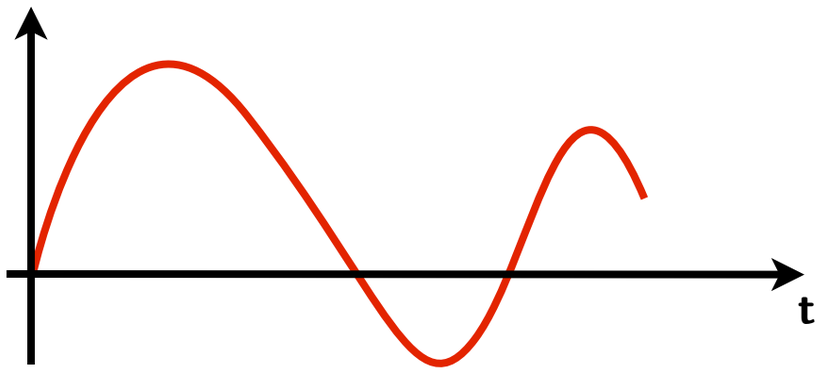


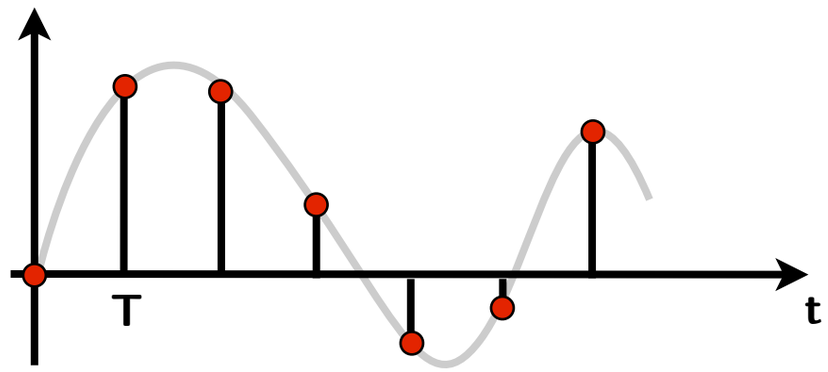
Signal types

continuous **time** discrete

continuous **amplitude**

discrete

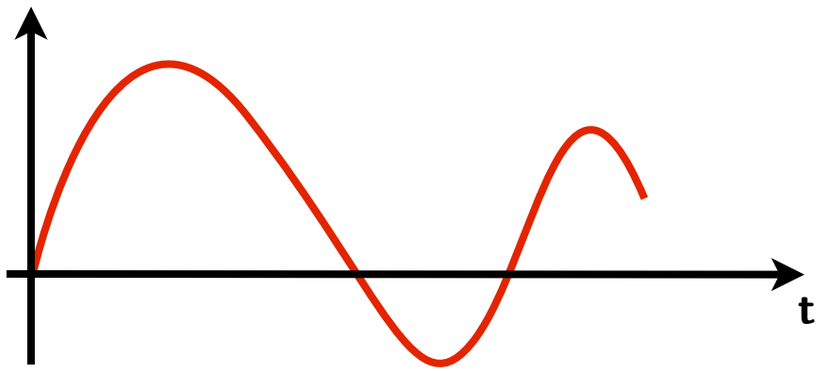


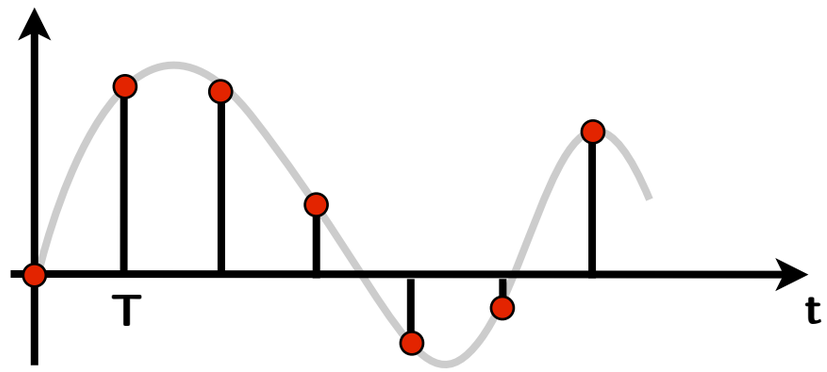


Signal types

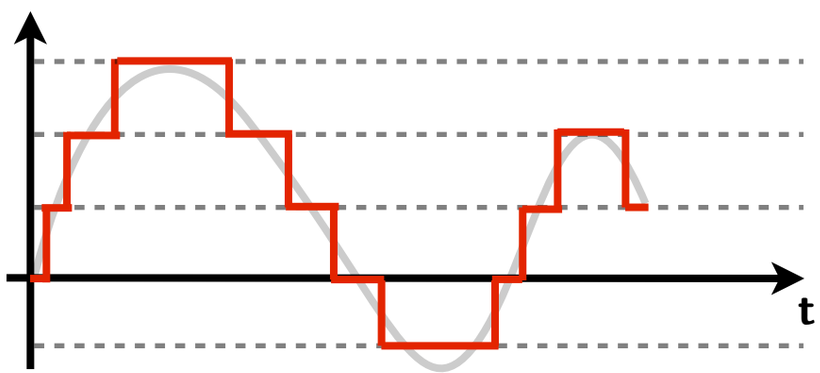
continuous **time** discrete

continuous **amplitude**





discrete

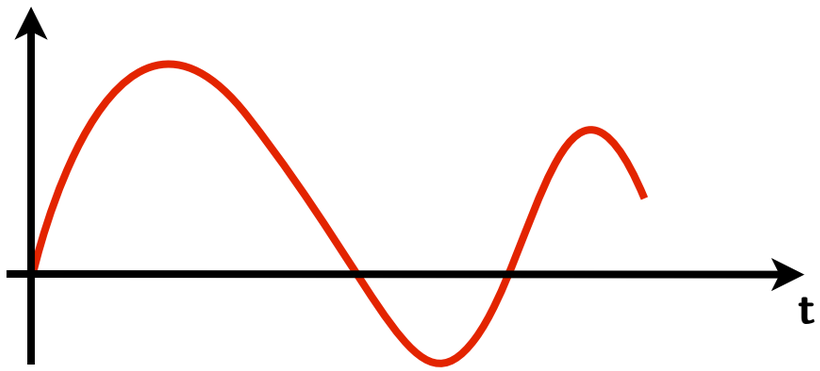


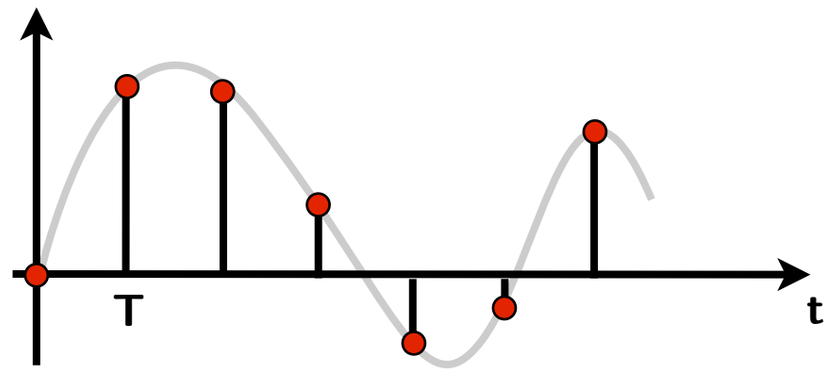
Signal types

continuous **time** discrete

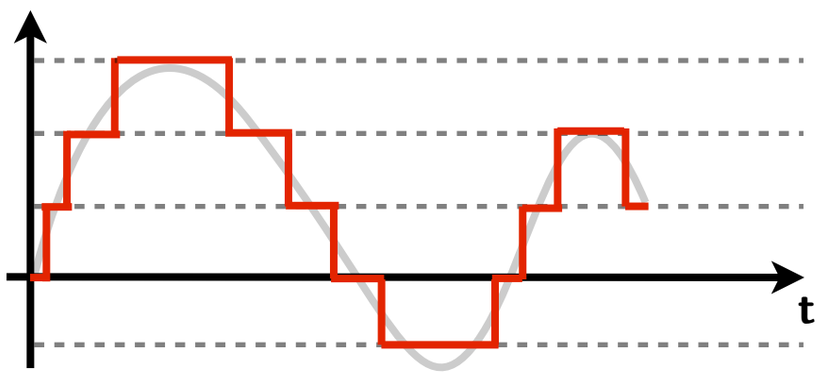
amplitude

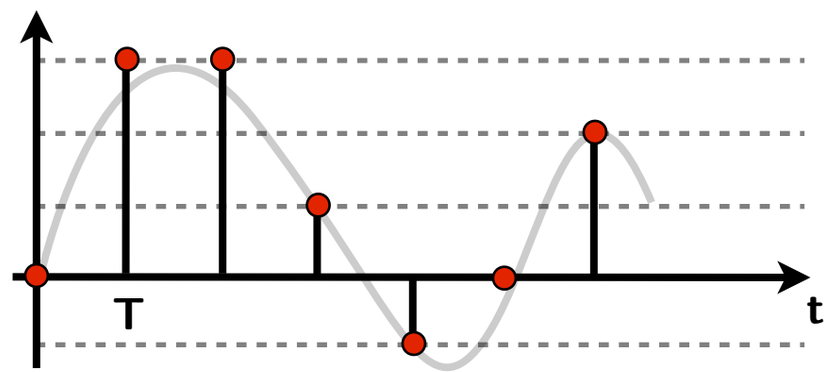
continuous





discrete

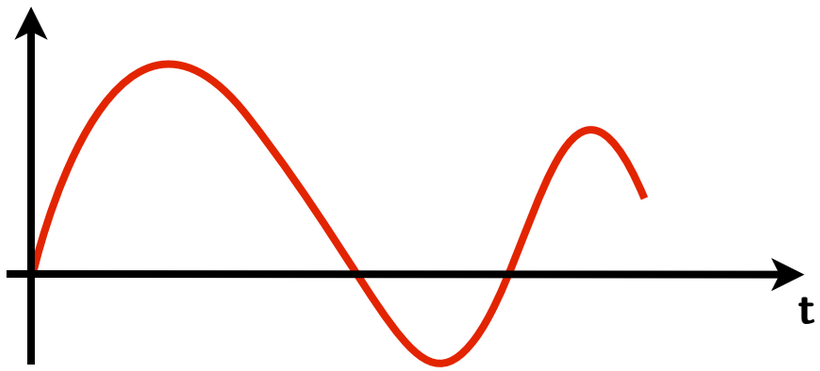


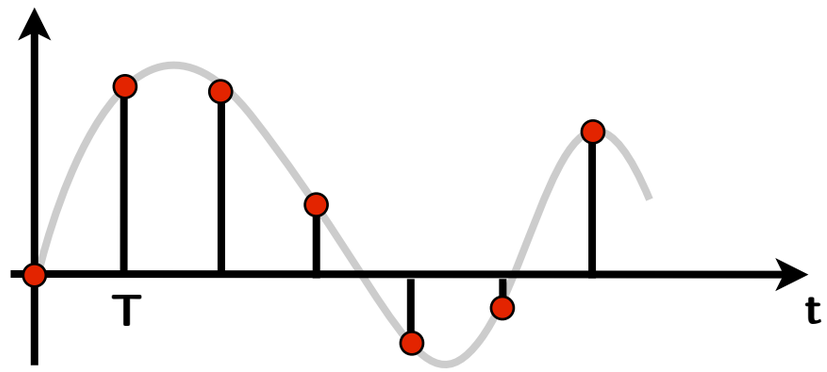


Signal types

continuous **time** discrete

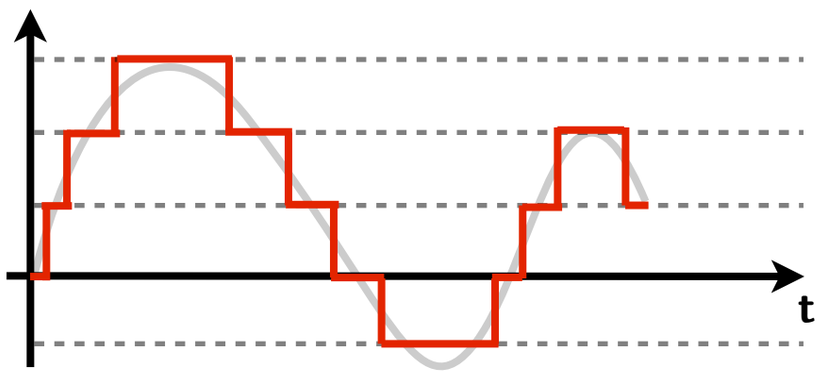
amplitude — continuous

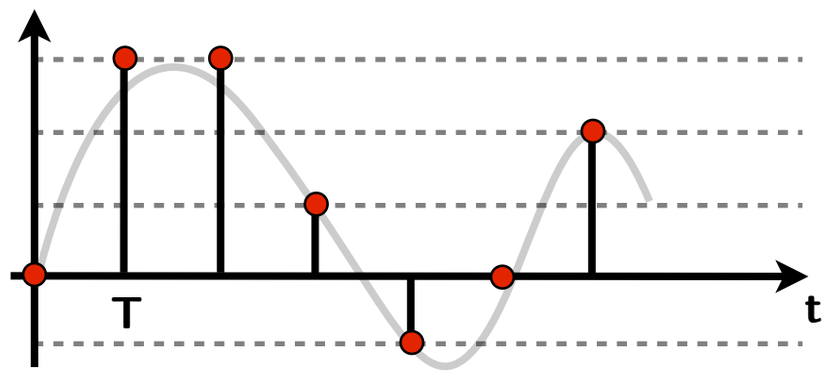




analogue signal

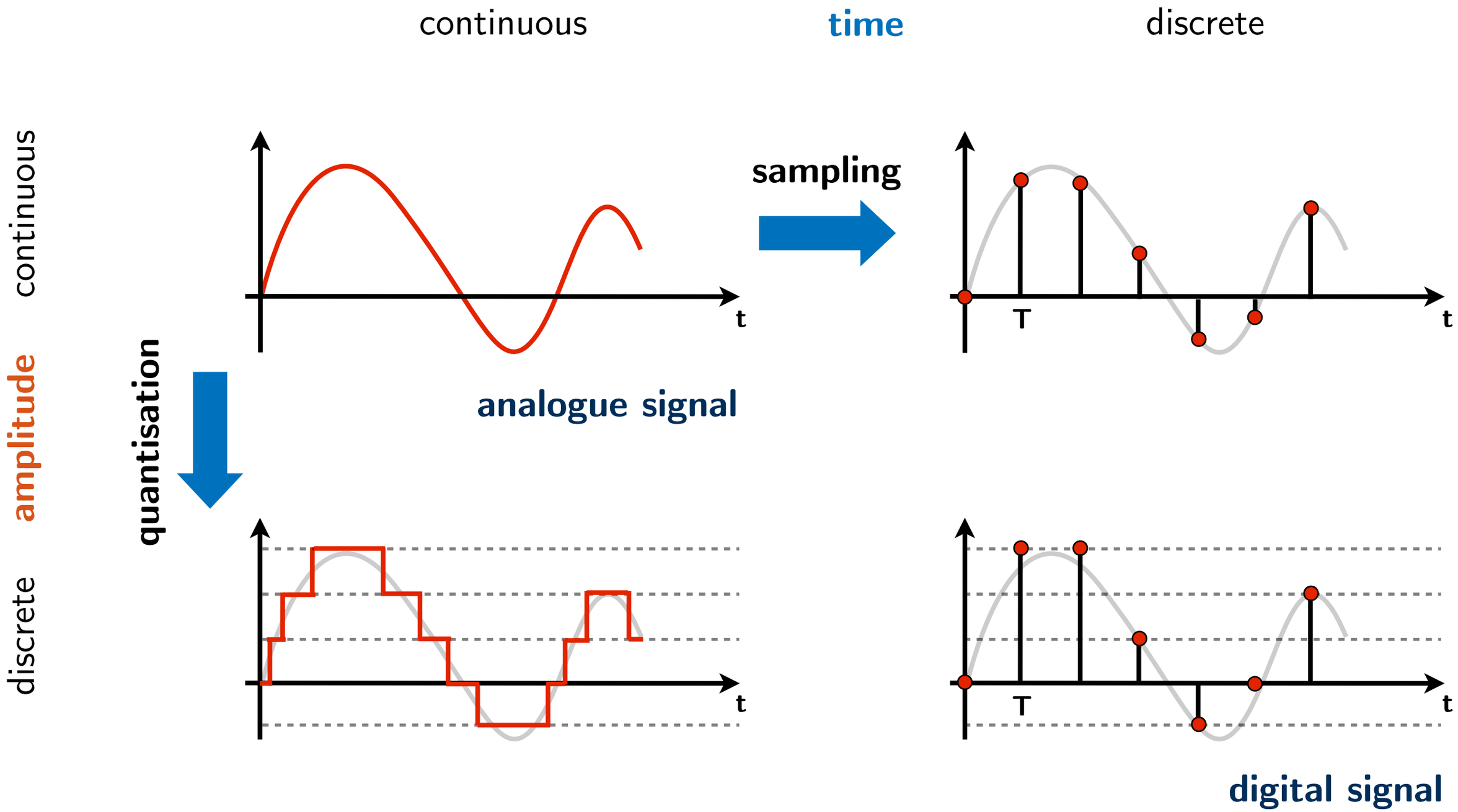
amplitude — discrete





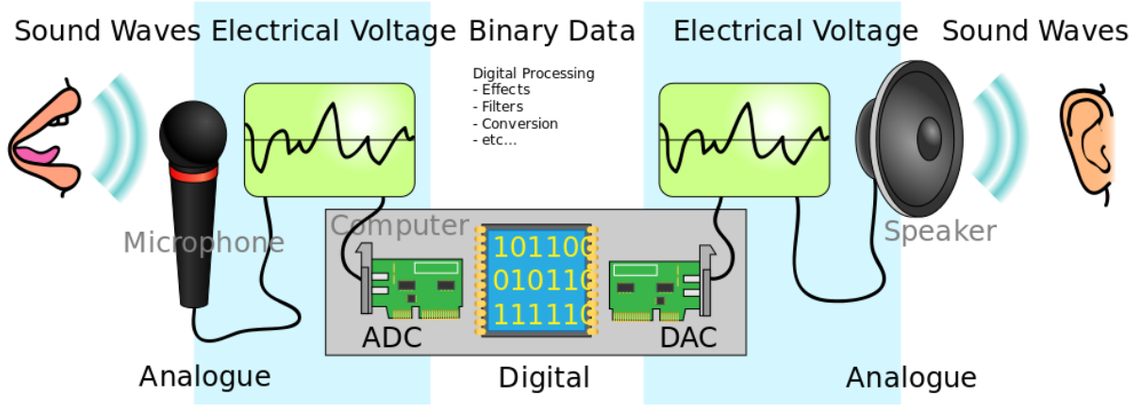
digital signal

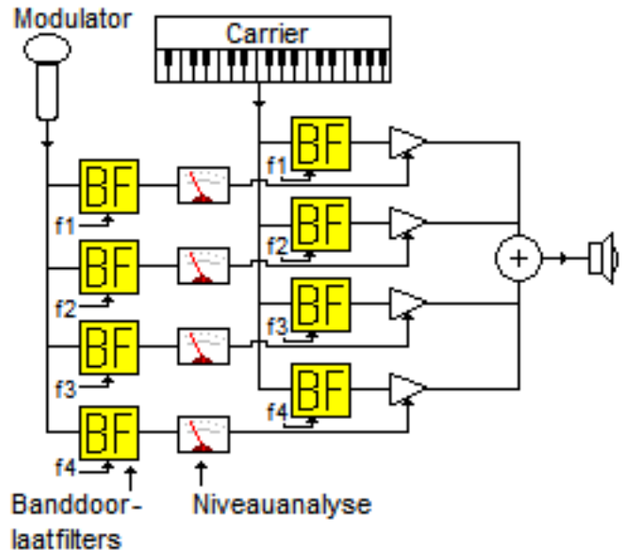
Signal types



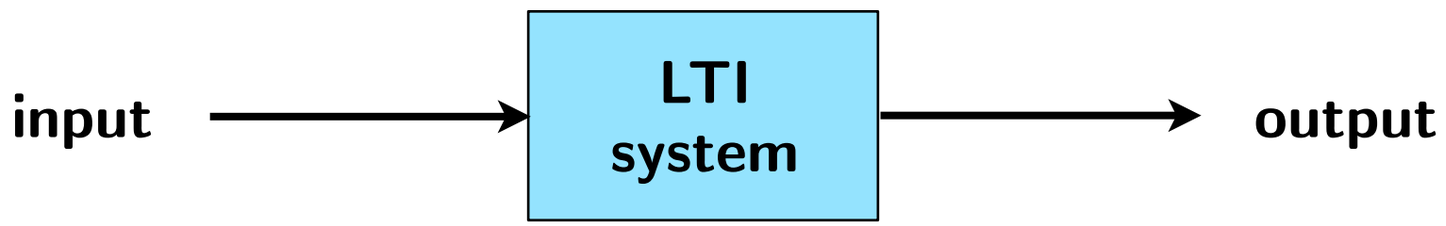
Analogue vs Digital signal processing

- Many signals of practical interest are analogue: e.g., speech, seismic, radar, and sonar signals
- Analogue signal processing systems are based on analogue equipment: e.g., channel vocoder
- Dramatic advance of digital computing moves the trend towards digital systems





Signal processing as linear processes



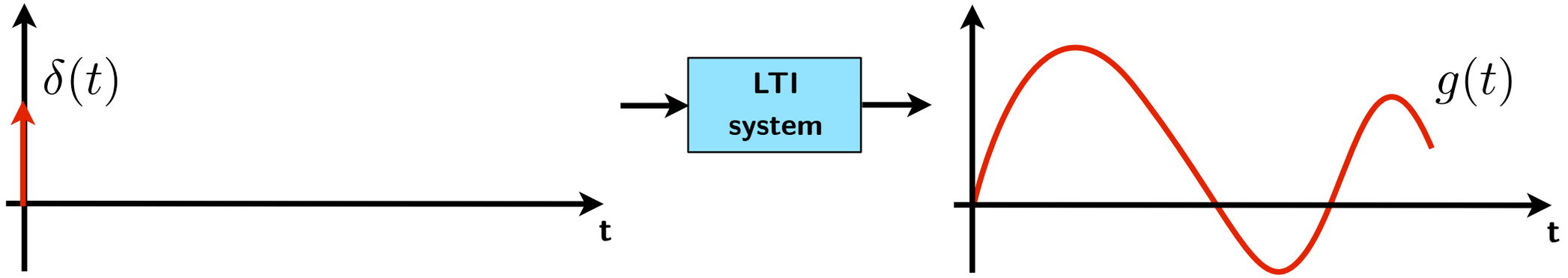
amplification/attenuation, filtering, (un-)mixing, etc.

- Linear time-invariant (LTI) system whose input-output characteristics can be defined by
  - **impulse response** in **time** domain
  - **transfer function** in **frequency** domain

- There is an **invertible** mapping between time- and frequency-domain representations

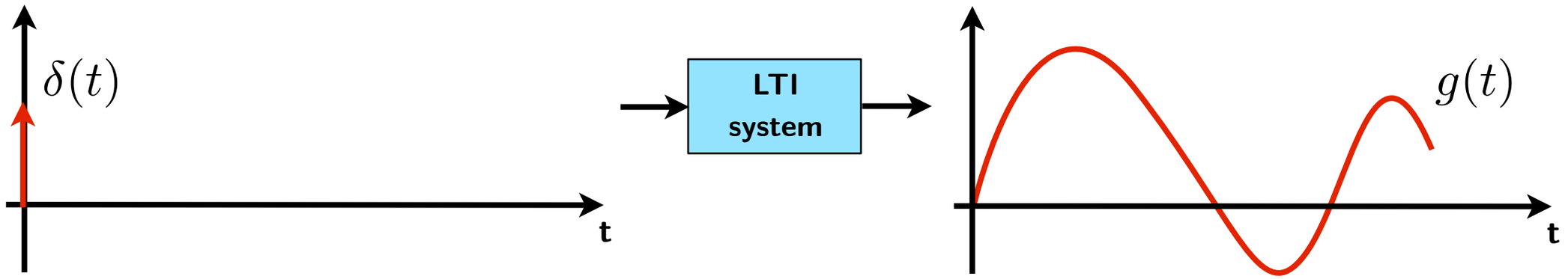
Time-domain analysis - Convolution

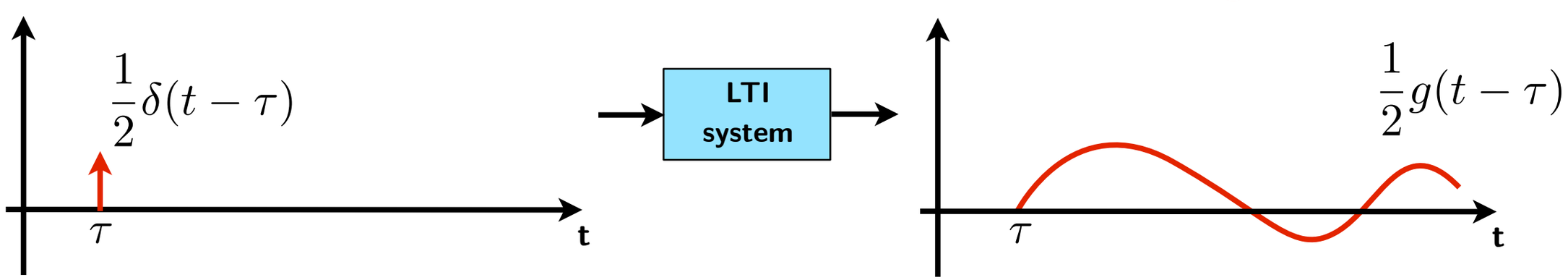
- Convolution allows the evaluation of the output signal from an linear time-invariant (LTI) system, given its impulse response and input signal



Time-domain analysis - Convolution

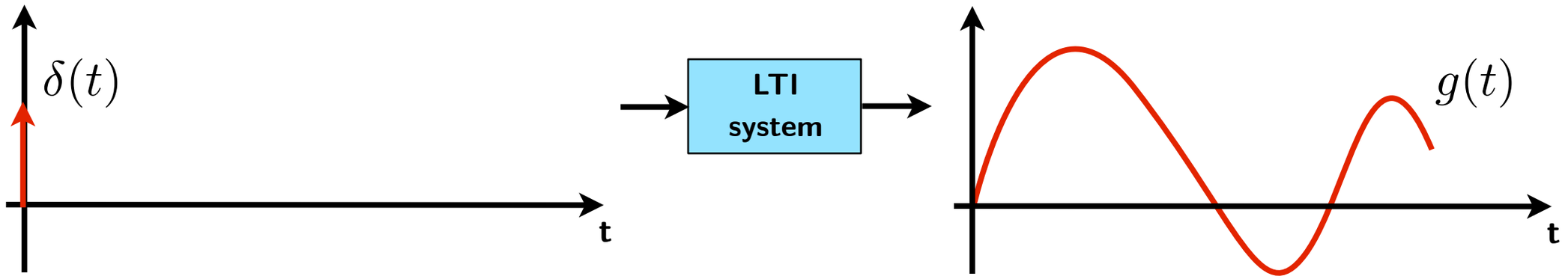
- Convolution allows the evaluation of the output signal from an linear time-invariant (LTI) system, given its impulse response and input signal

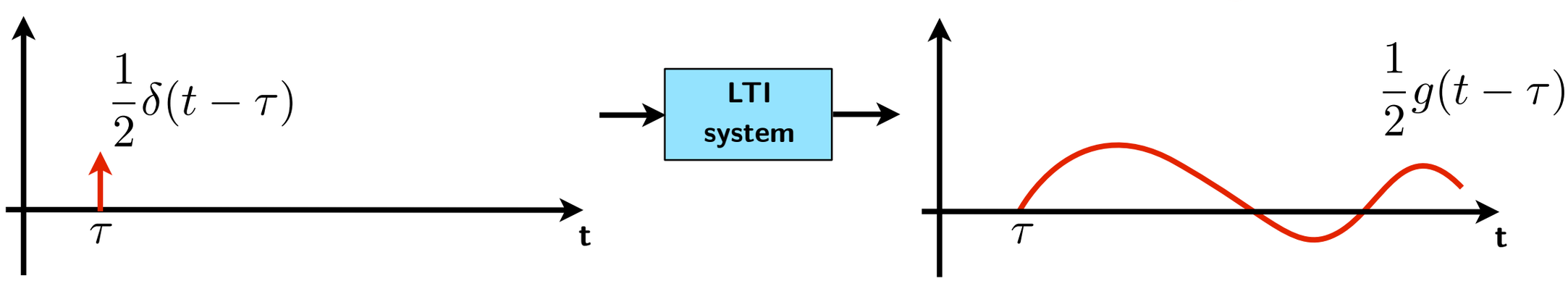




Time-domain analysis - Convolution

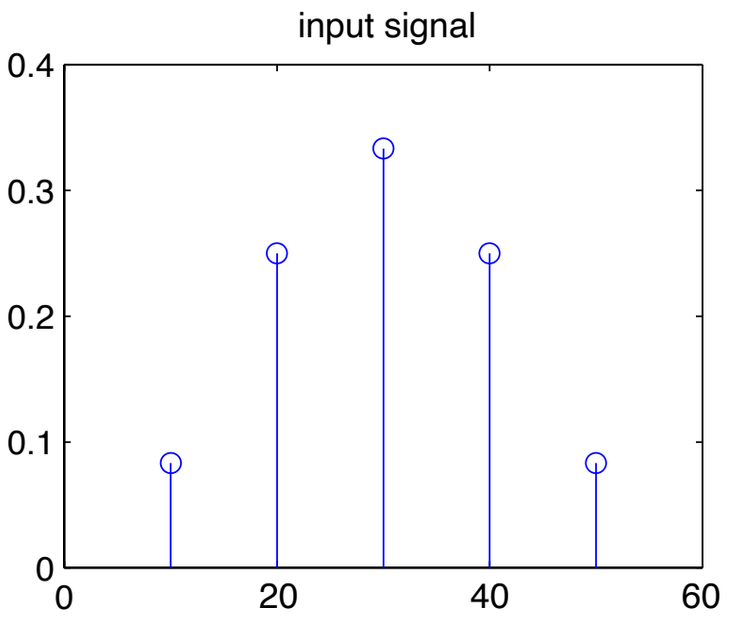
- Convolution allows the evaluation of the output signal from an linear time-invariant (LTI) system, given its impulse response and input signal

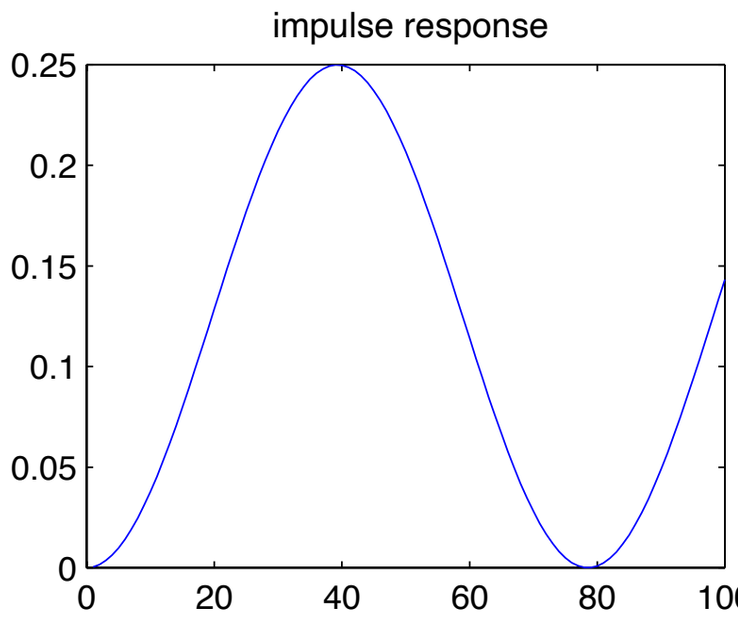


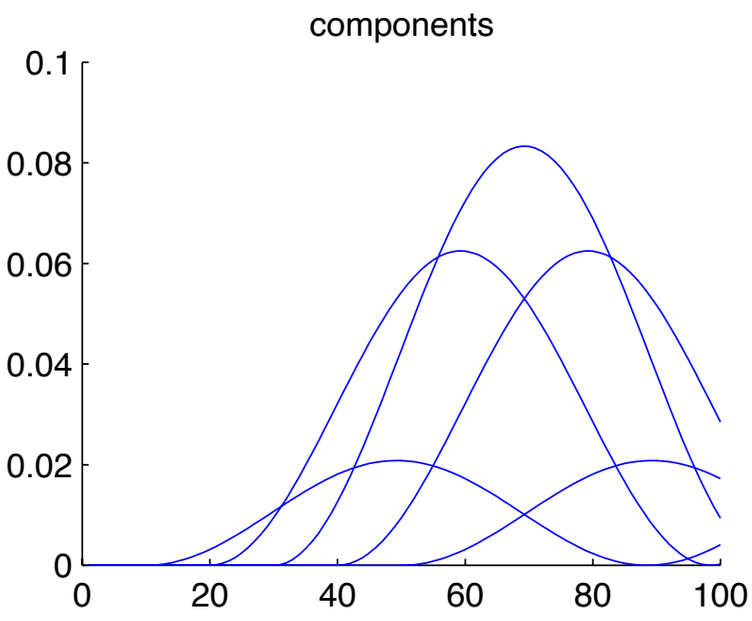


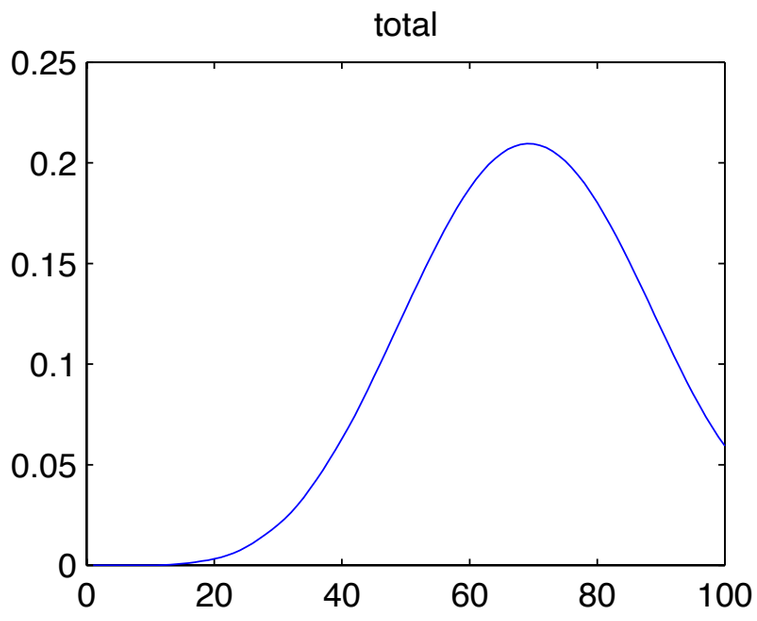
- Evaluate system output for
  - input: succession of impulse functions (which generate weighted impulse responses)
  - output: sum of the effect of each impulse function

Time-domain analysis - Convolution

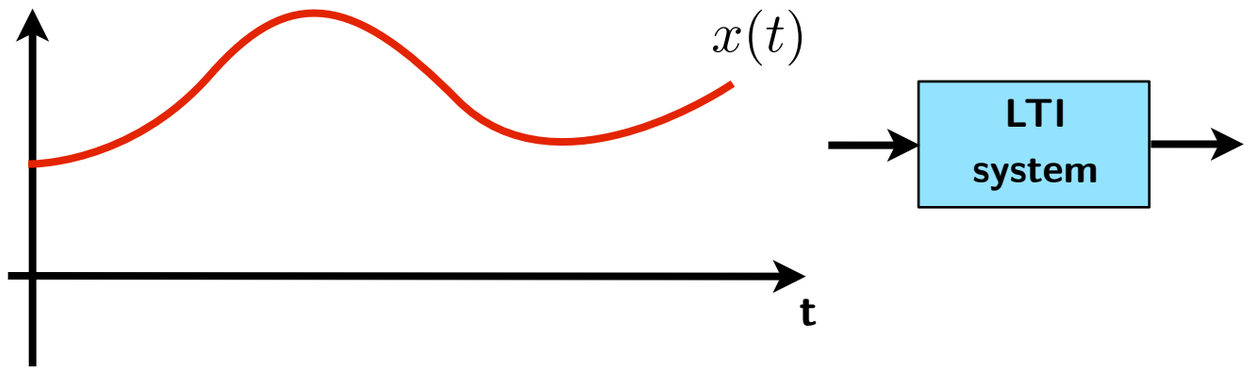




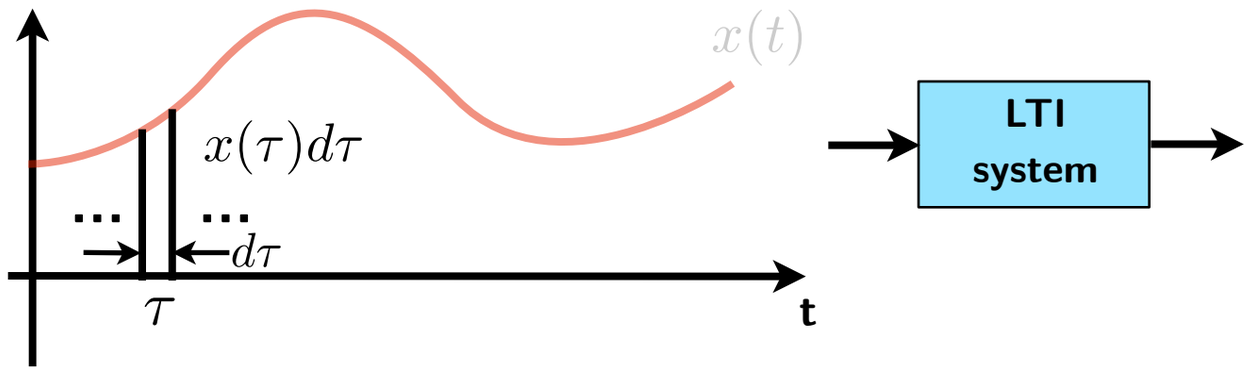




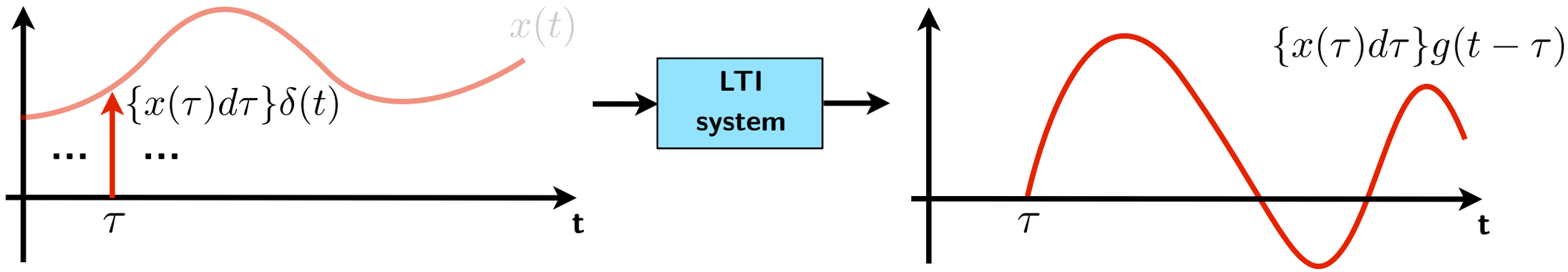
Time-domain analysis - Convolution



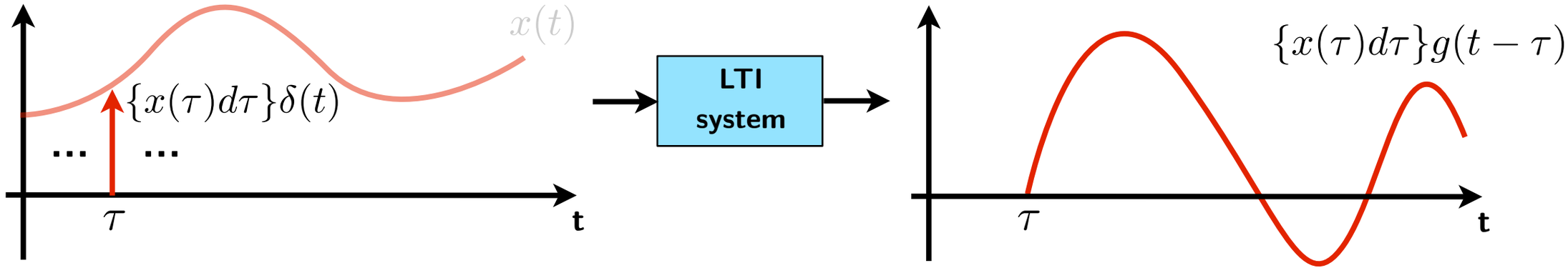
Time-domain analysis - Convolution



Time-domain analysis - Convolution



Time-domain analysis - Convolution

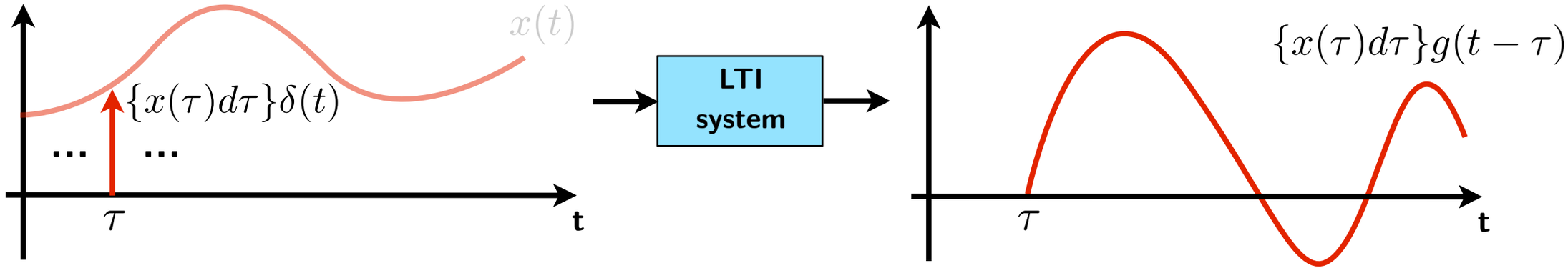


- this gives the convolution integral

$$y(t) = \sum_{\tau} \{x(\tau)d\tau\}g(t-\tau) \overset{d\tau \to 0}{\longrightarrow} \int_0^{\infty} x(\tau)g(t-\tau)d\tau$$

- system output is **convolution** of input and impulse response

Time-domain analysis - Convolution



- this gives the convolution integral

$$y(t) = \sum_{\tau} \{x(\tau)d\tau\}g(t-\tau) \overset{d\tau \to 0}{\longrightarrow} \int_0^{\infty} x(\tau)g(t-\tau)d\tau$$

- system output is **convolution** of input and impulse response
- **impulse response** characterises the system in time domain

Frequency-domain analysis

- Consider the following LTI system

$$x(t) = e^{j\omega t} \longrightarrow \boxed{g(t)} \longrightarrow y(t)$$

$$y(t) = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} g(\tau) d\tau = e^{j\omega t}\ G(j\omega)$$

  - $e^{j\omega t}$ is an eigenfunction of an LTI system with eigenvalue $G(j\omega)$, which is the Fourier transform of the impulse response $g(t)$

Frequency-domain analysis

- Consider the following LTI system

$$x(t) = e^{j\omega t} \longrightarrow \boxed{g(t)} \longrightarrow y(t)$$

$$y(t) = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} g(\tau) d\tau = e^{j\omega t}\ G(j\omega)$$

- $e^{j\omega t}$ is an eigenfunction of an LTI system with eigenvalue $G(j\omega)$, which is the Fourier transform of the impulse response $g(t)$

- $G(j\omega)$, the **frequency response**, characterises the system in frequency domain

Fourier transform

- Fourier transform (FT) and inverse FT

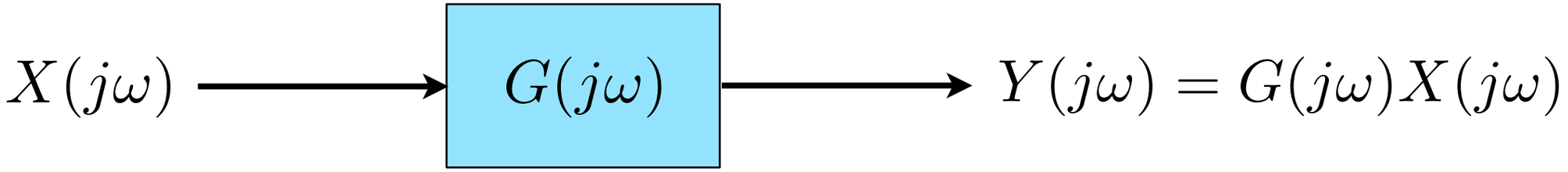
$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$$

Fourier transform

- Fourier transform (FT) and inverse FT

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$$

- FT of output is **multiplication** of FT of input and frequency response

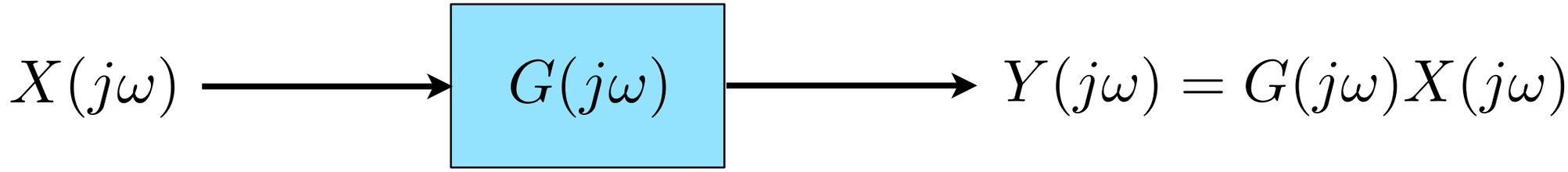


Fourier transform

- Fourier transform (FT) and inverse FT

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$$

- FT of output is **multiplication** of FT of input and frequency response

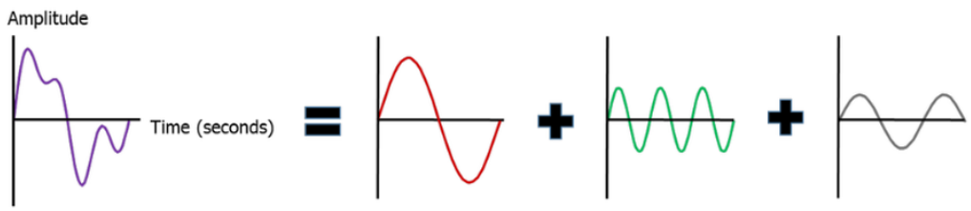


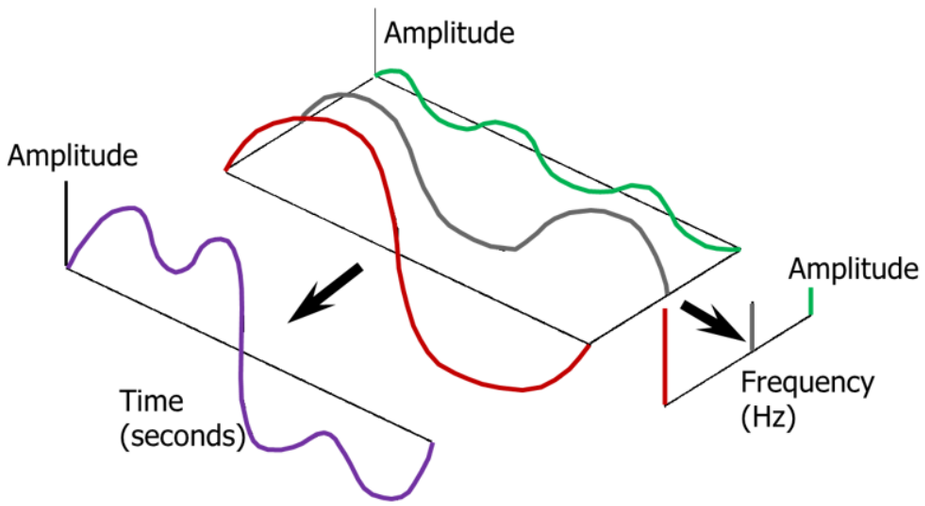
- System output via inverse FT

$$y(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} Y(j\omega)e^{j\omega t}d\omega$$

Fourier series

- **Fourier series (FS)** for **periodic signal**





fundamental frequency

$$f_0 = 1/T_0$$

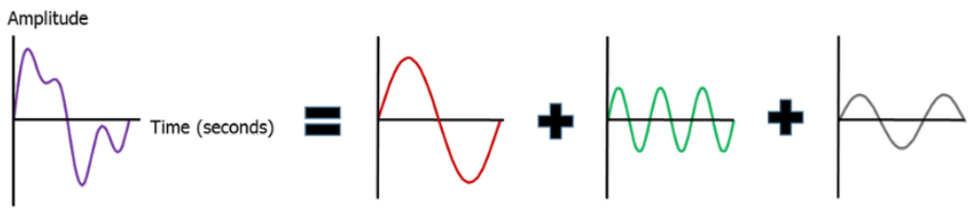
$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_0 t}$$

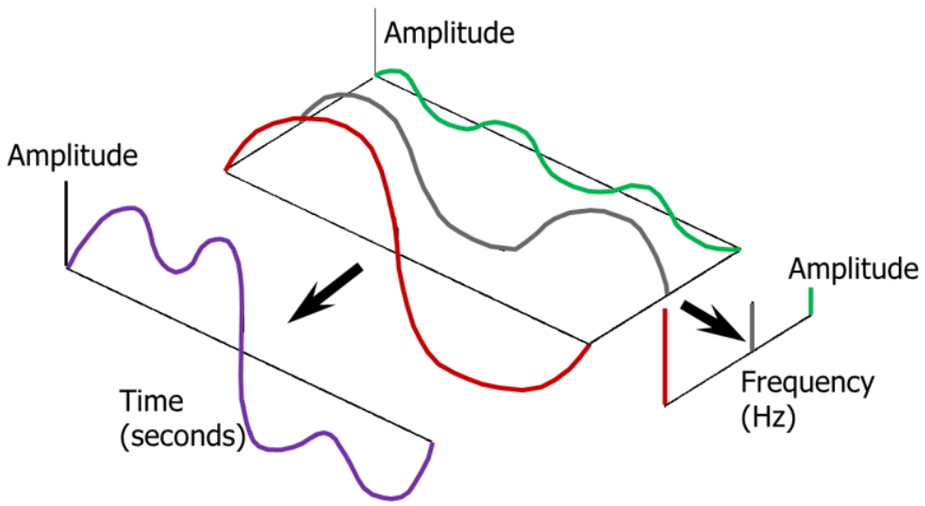
fundamental period

$$C_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-j2\pi n f_0 t} dt$$

Fourier series

- **Fourier series (FS)** for **periodic signal**





fundamental frequency

$$f_0 = 1/T_0$$

$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_0 t}$$

fundamental period

$$C_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-j2\pi n f_0 t} dt$$

- When the period approaches infinity, the spectrum becomes continuous leading to **FT** for **aperiodic signal** (previous slide)

Time domain vs Frequency domain

- **Theorem**

  Convolution in time domain is equivalent to multiplication in frequency domain, i.e.,

$$y(t) = g(t) * x(t) \equiv \mathcal{F}^{-1}\{Y(j\omega) = G(j\omega)X(j\omega)\}$$

- **Proof**

$$\begin{aligned}\mathcal{F}\{g(t) * x(t)\} &= \int_t \int_\tau g(t-\tau)x(\tau)d\tau e^{-j\omega t}dt \\ &= \int_\tau x(\tau)e^{-j\omega\tau}d\tau\mathcal{F}\{g(t)\} \\ &= \mathcal{F}\{g(t)\}\mathcal{F}\{x(t)\}\end{aligned}$$

Time domain vs Frequency domain

- **Theorem**

  Convolution in time domain is equivalent to multiplication in frequency domain, i.e.,

$$y(t) = g(t) * x(t) \equiv \mathcal{F}^{-1}\{Y(j\omega) = G(j\omega)X(j\omega)\}$$

- **Proof**

$$\begin{aligned}\mathcal{F}\{g(t) * x(t)\} &= \int_t \int_\tau g(t-\tau)x(\tau)d\tau e^{-j\omega t}dt \\ &= \int_\tau x(\tau)e^{-j\omega\tau}d\tau\mathcal{F}\{g(t)\} \\ &= \mathcal{F}\{g(t)\}\mathcal{F}\{x(t)\}\end{aligned}$$

Time domain vs Frequency domain

- **Theorem**

  Convolution in time domain is equivalent to multiplication in frequency domain, i.e.,

$$y(t) = g(t) * x(t) \equiv \mathcal{F}^{-1}\{Y(j\omega) = G(j\omega)X(j\omega)\}$$

- **Proof**

$$\begin{aligned}\mathcal{F}\{g(t) * x(t)\} &= \int_t \int_\tau g(t-\tau)x(\tau)d\tau e^{-j\omega t}dt \\ &= \int_\tau x(\tau)e^{-j\omega\tau}d\tau \mathcal{F}\{g(t)\} \\ &= \mathcal{F}\{g(t)\}\mathcal{F}\{x(t)\}\end{aligned}$$

Time domain vs Frequency domain

- **Theorem**

  Convolution in time domain is equivalent to multiplication in frequency domain, i.e.,

$$y(t) = g(t) * x(t) \equiv \mathcal{F}^{-1}\{Y(j\omega) = G(j\omega)X(j\omega)\}$$

- **Proof**

$$\begin{aligned}\mathcal{F}\{g(t) * x(t)\} &= \int_t \int_\tau g(t-\tau)x(\tau)d\tau e^{-j\omega t}dt \\ &= \int_\tau x(\tau)e^{-j\omega\tau}d\tau \mathcal{F}\{g(t)\} \\ &= \mathcal{F}\{g(t)\}\mathcal{F}\{x(t)\}\end{aligned}$$

Time domain vs Frequency domain

- **Theorem**

  Convolution in time domain is equivalent to multiplication in frequency domain, i.e.,

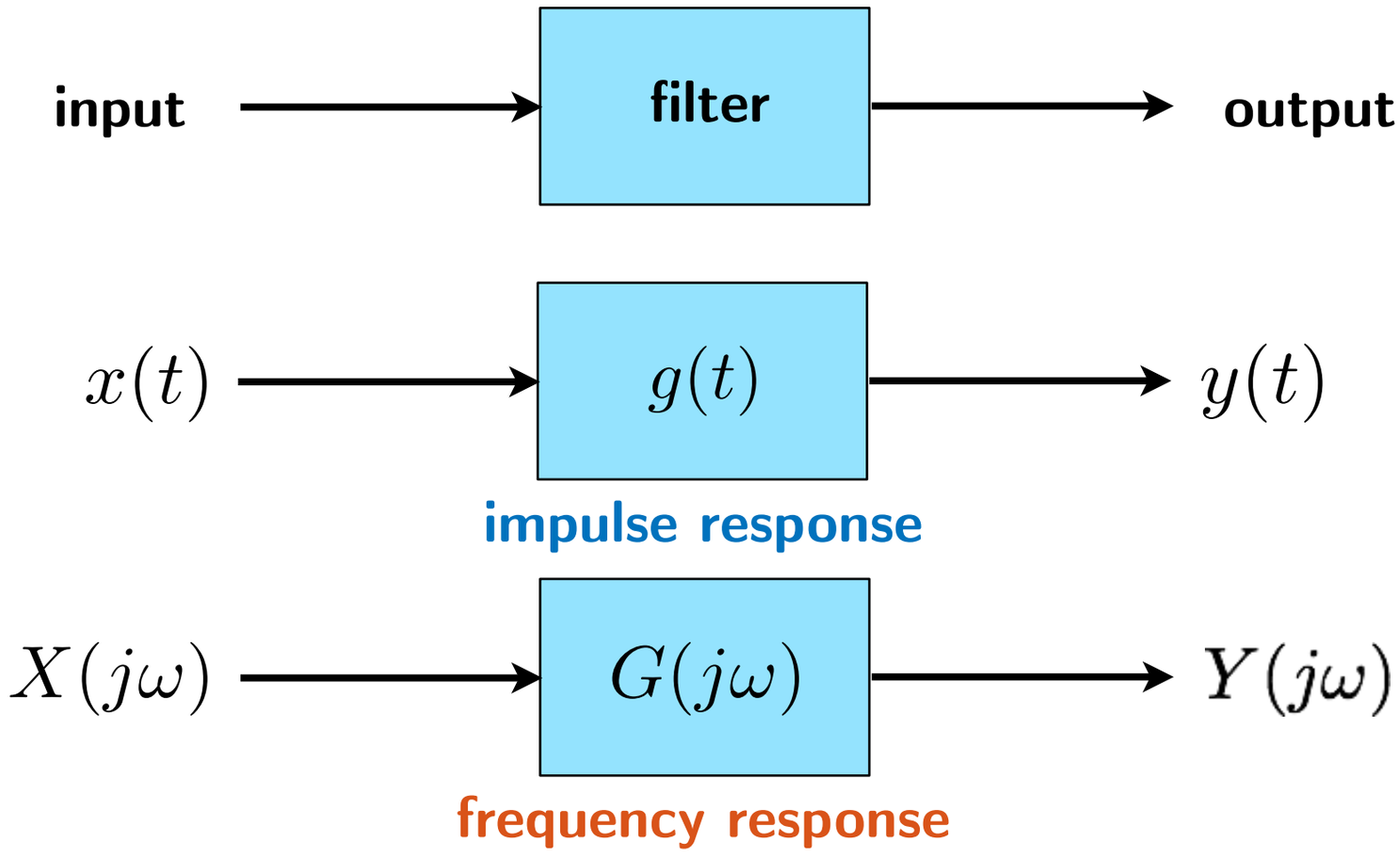
$$y(t) = g(t) * x(t) \equiv \mathcal{F}^{-1}\{Y(j\omega) = G(j\omega)X(j\omega)\}$$

- **Proof**
$$\begin{aligned}\mathcal{F}\{g(t) * x(t)\} &= \int_t \int_\tau g(t-\tau)x(\tau)d\tau e^{-j\omega t}dt \\ &= \int_\tau x(\tau)e^{-j\omega\tau}d\tau\mathcal{F}\{g(t)\} \\ &= \mathcal{F}\{g(t)\}\mathcal{F}\{x(t)\}\end{aligned}$$

  This allows us to move losslessly between **time** and **frequency** domains, choosing whichever is the easier to work with

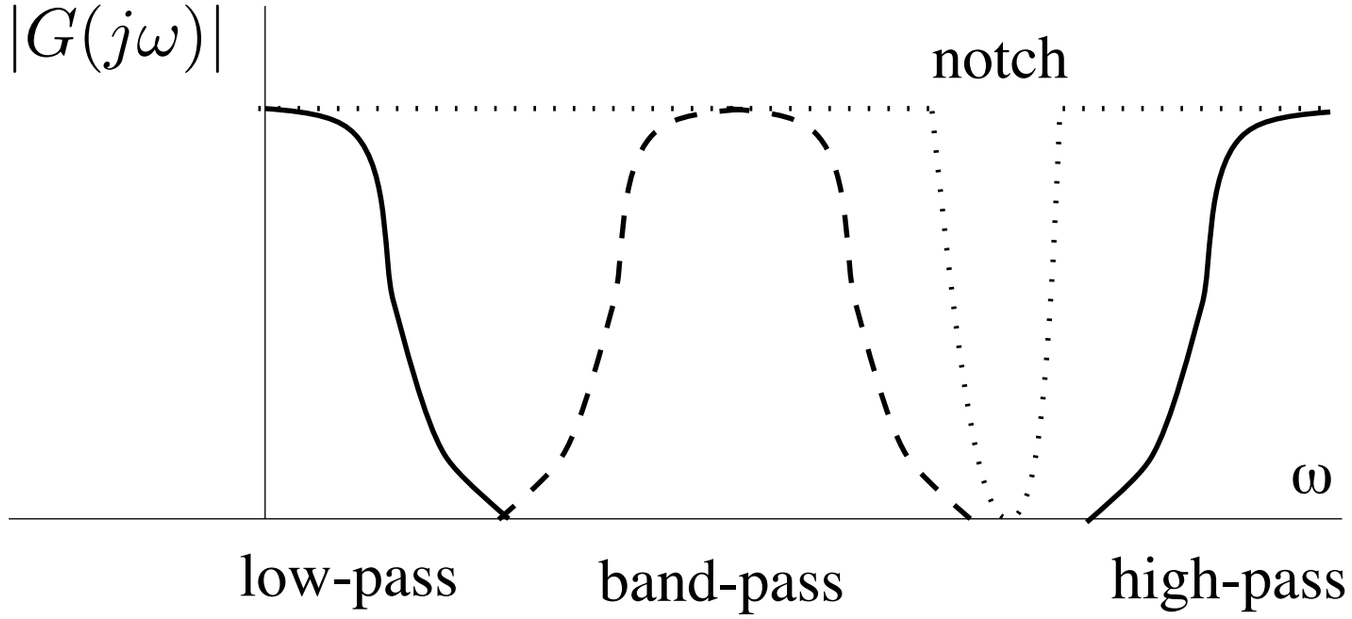
Frequency filtering

- Filtering as input-output relationship



Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component



Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component

original

Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component

original **low-pass filtered**

Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component

original **low-pass filtered** **high-pass filtered**

Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component

original **low-pass filtered** **high-pass filtered**



Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component

original **low-pass filtered** **high-pass filtered**




Frequency filtering

- Filters are **frequency-selective** linear systems
  - **low-pass:** extract average or eliminate high-frequency fluctuations
  - **high-pass:** follow small-amplitude high-frequency perturbations in presence of much larger slowly-varying component

original **low-pass filtered** **high-pass filtered**



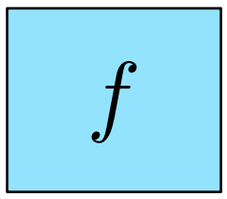


Frequency filtering

Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

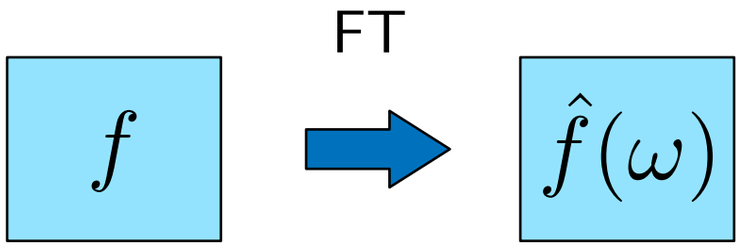
Frequency filtering

Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$



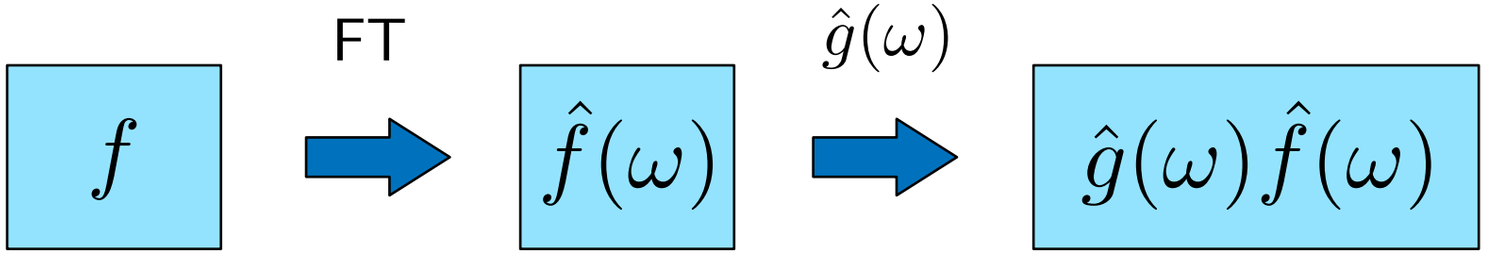
Frequency filtering

Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$



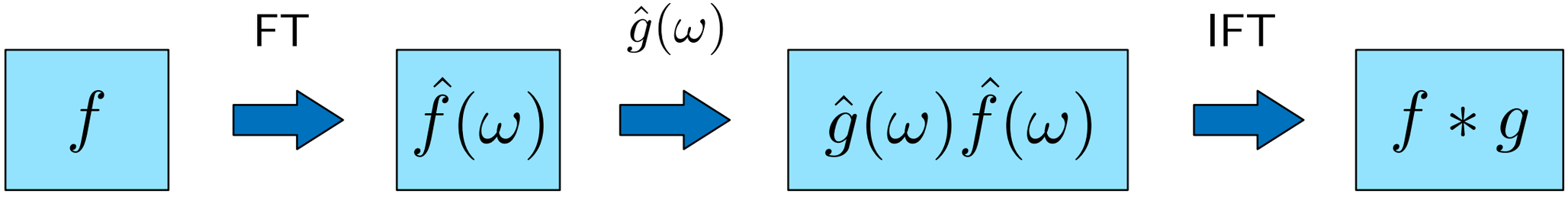
Frequency filtering

Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$



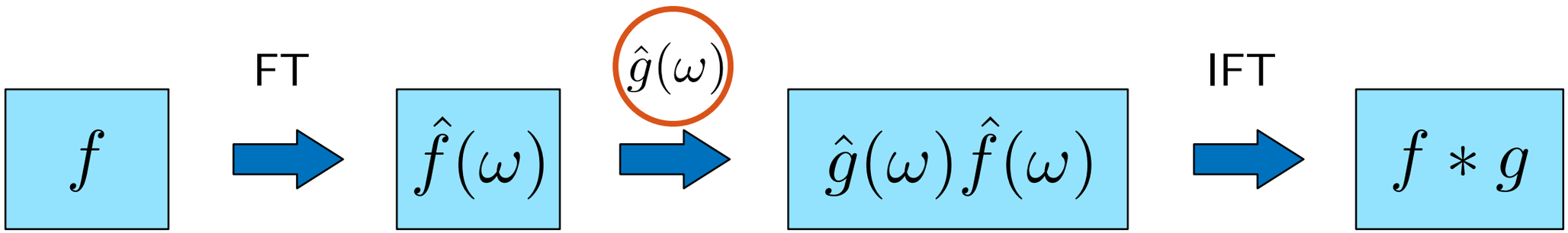
Frequency filtering

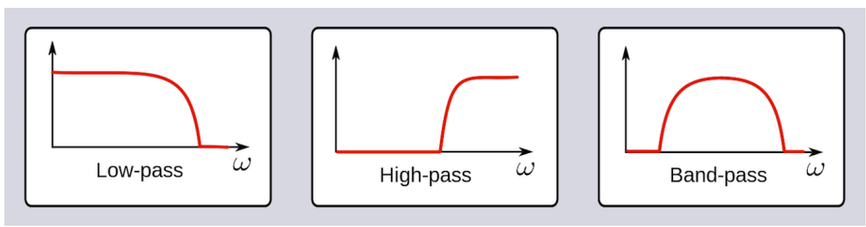
Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$



Frequency filtering

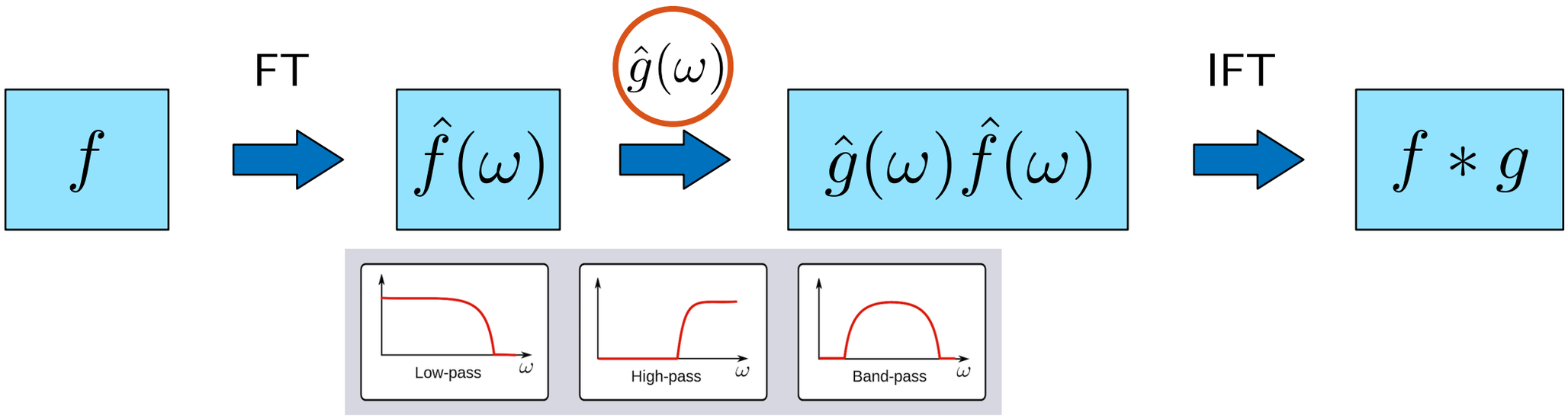
Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$





Frequency filtering

Fourier transform: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$



Takeaway: signal processing (filtering) requires two considerations

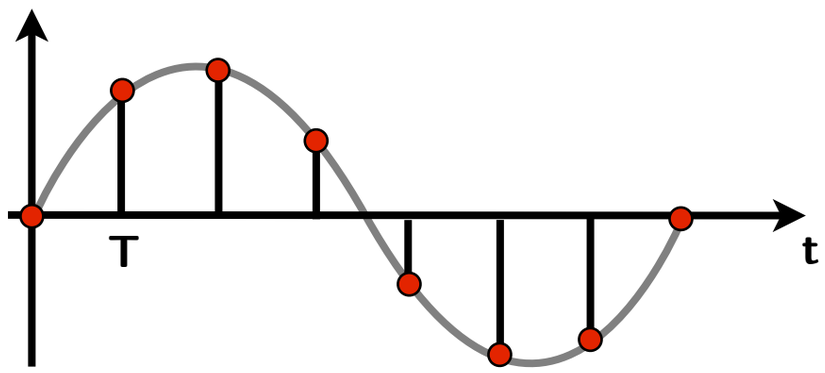
- find a **good representation** (e.g., via Fourier transform) of the signal
- design (or learn) **appropriate filters** (note that filtering = convolution!)

Lecture 1

- Introduction & Basic concepts and tools
- A historical overview of signal representation techniques
- Applications & Discussion

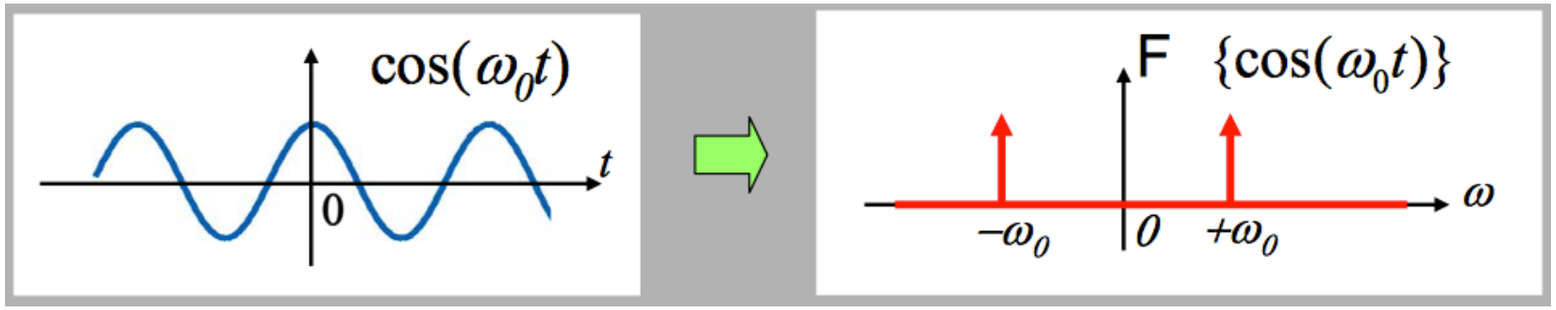
What is good representation of a signal?

- Sum of delta functions in time or space (sampling domain)
  - good for display or playback
  - not good for analysis (e.g., denoising, compression)



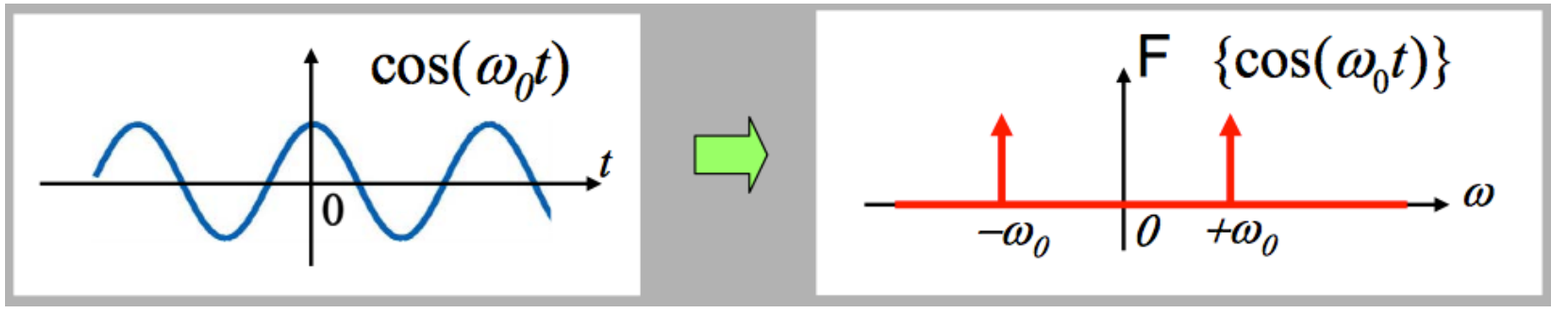
What is good representation of a signal?

- Representation often involves transformation of the signal into a new domain where signal characteristics are revealed
  - example: **Fourier coefficients** reveal **rate of change** of the signal



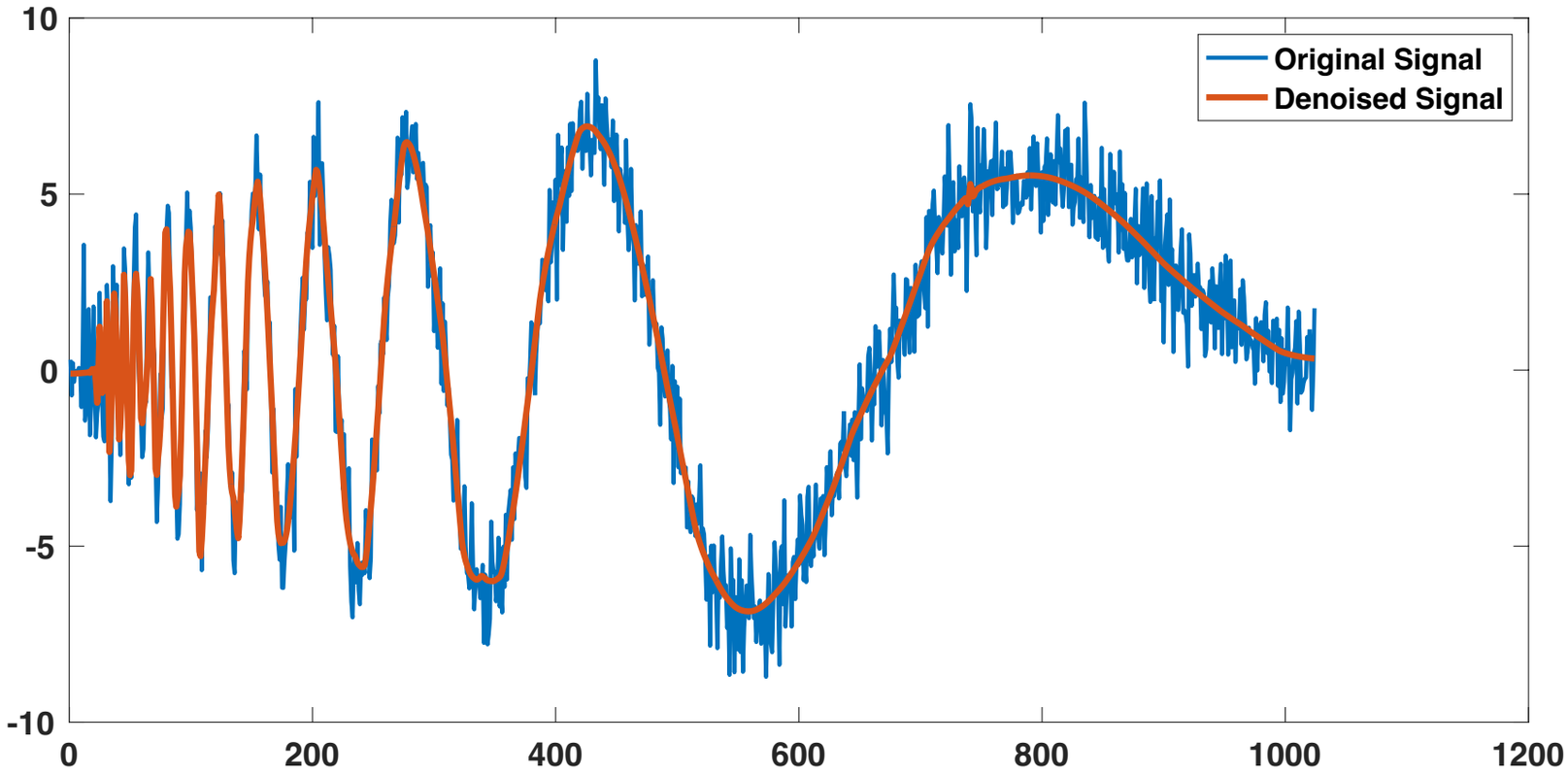
What is good representation of a signal?

- Representation often involves transformation of the signal into a new domain where signal characteristics are revealed
  - example: **Fourier coefficients** reveal **rate of change** of the signal



- Usefulness of the representation depends on the analysis goal
  - which may vary but all shares the core desire for **simplification**

Example: Denoising



goal: recover signal from noisy observation

Example: Compression

original

JPEG 2000 (10% in size)

JPEG 2000 (1% in size)

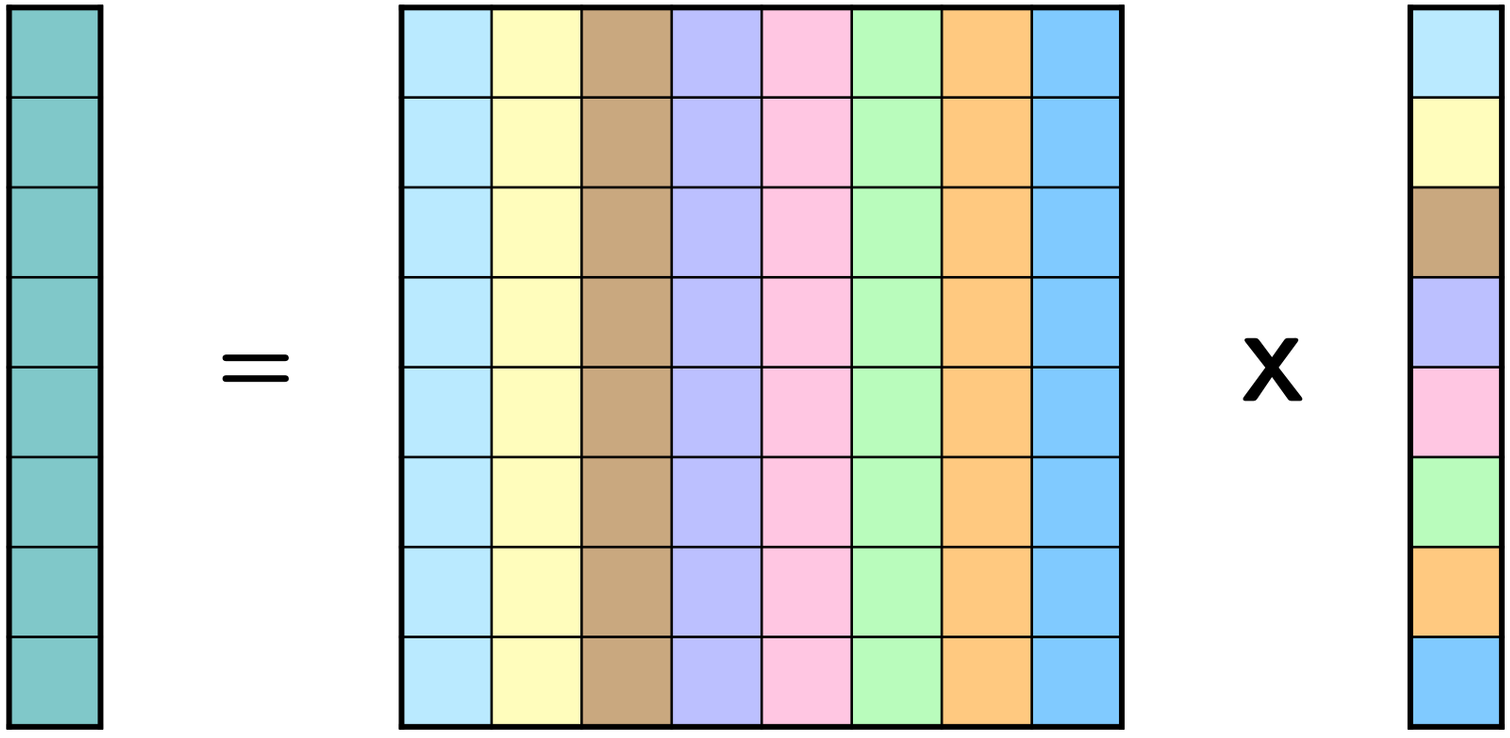
goal: compress signal without sacrificing quality

Example: Recognition

goal: capture true “causes” of signal

Signal representation via dictionaries

signal **dictionary** **coefficients**

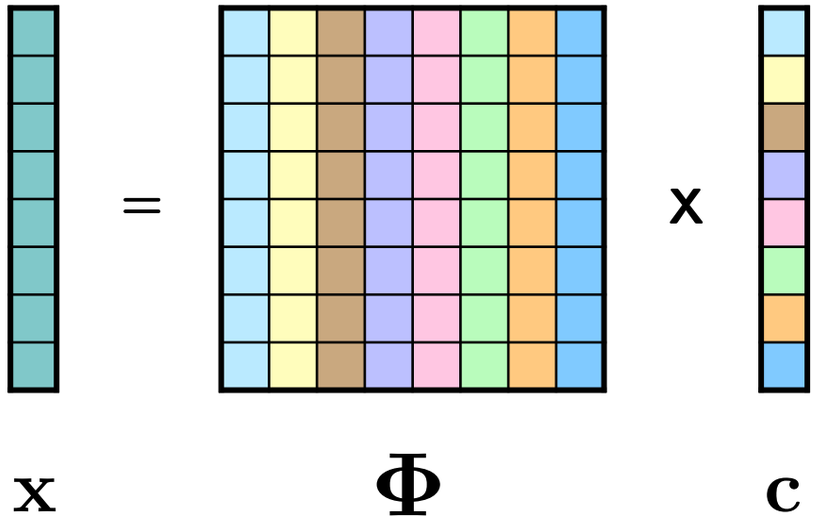


$\mathbf{x}$ $\quad$ $\mathbf{\Phi} = [\boldsymbol{\phi_0}\ \boldsymbol{\phi_1}\ \cdots\ \boldsymbol{\phi_{N-1}}]$ $\quad$ $\mathbf{c}$

“atoms”

Signal representation via dictionaries

- Complete dictionaries

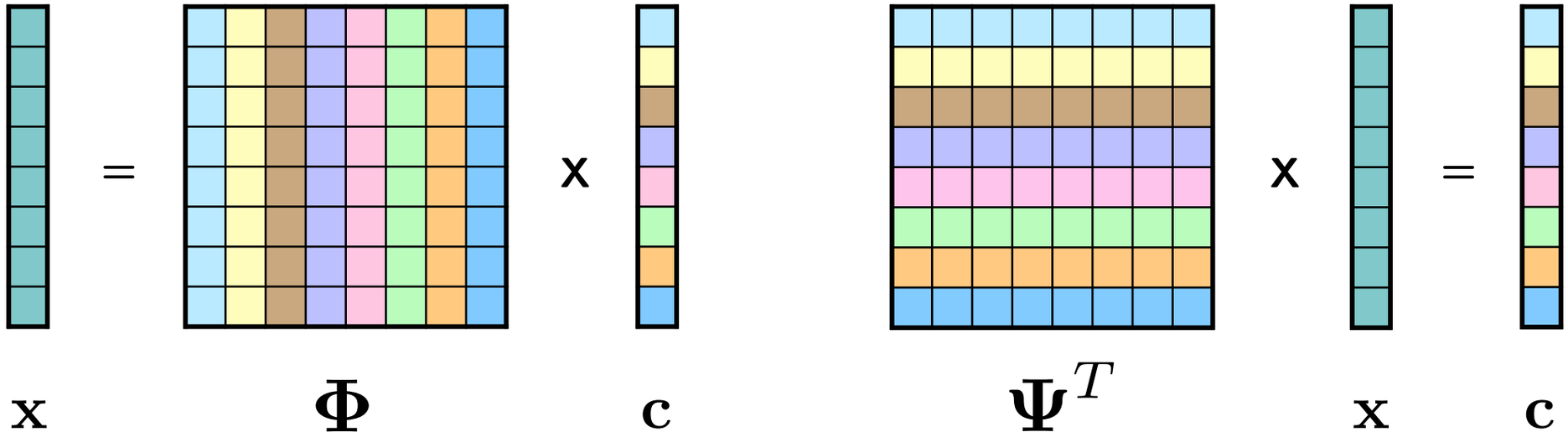


$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \boldsymbol{\phi}_n$$

Signal representation via dictionaries

- Complete dictionaries

synthesis **analysis**

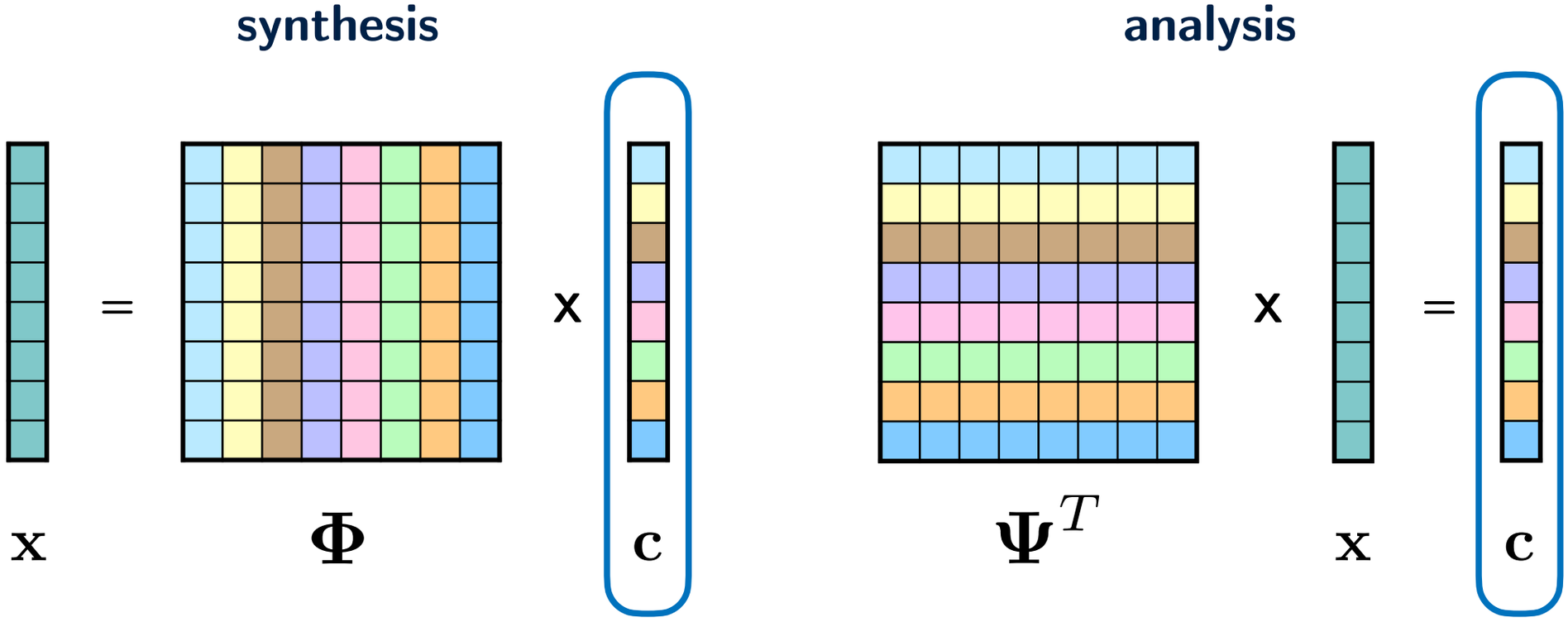


$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \boldsymbol{\phi}_n$$

$$\mathbf{c}_n = \boldsymbol{\psi}_n^T \mathbf{x}$$

Signal representation via dictionaries

- Complete dictionaries



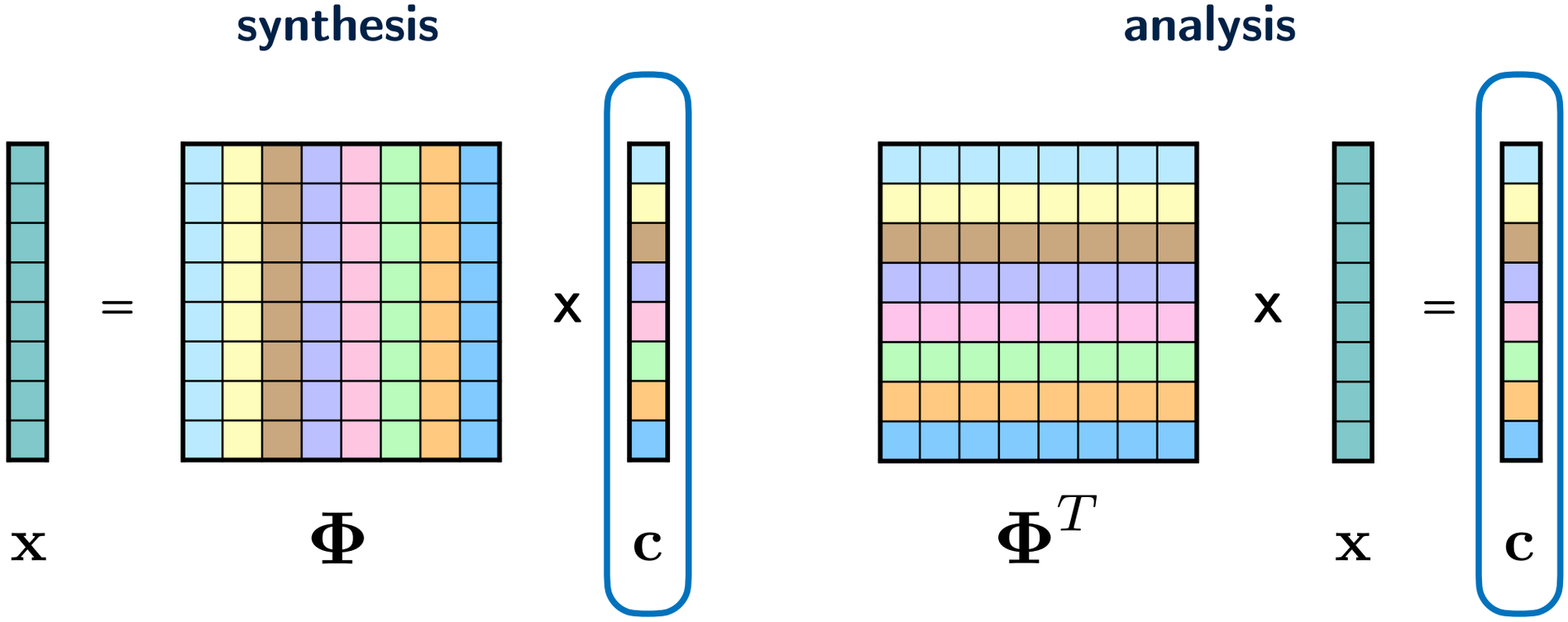
$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \boldsymbol{\phi}_n$$

$$\mathbf{c}_n = \boldsymbol{\psi}_n^T \mathbf{x}$$

equivalent for complete and biorthogonal dictionaries

Signal representation via dictionaries

- Complete dictionaries



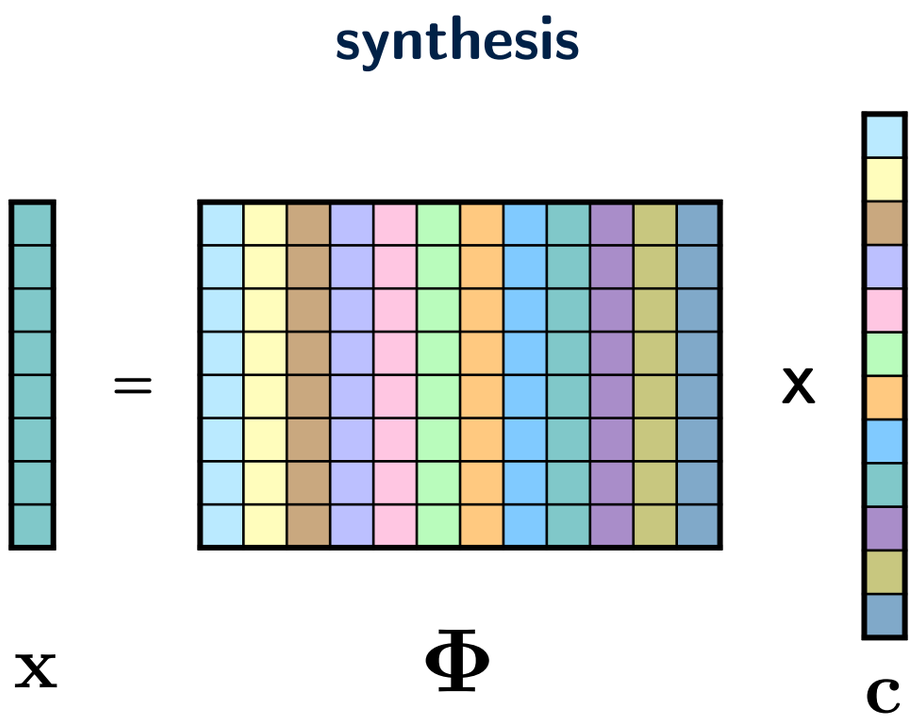
$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \phi_n$$

$$\mathbf{c}_n = \phi_n^T \mathbf{x}$$

equivalent for complete and biorthogonal dictionaries

Signal representation via dictionaries

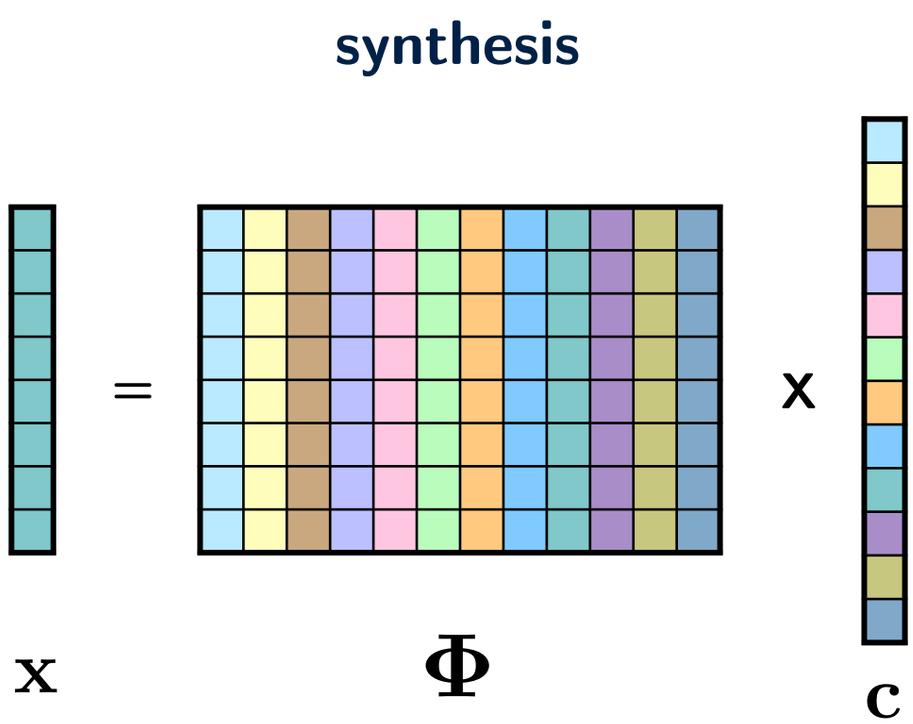
- Overcomplete (redundant) dictionaries



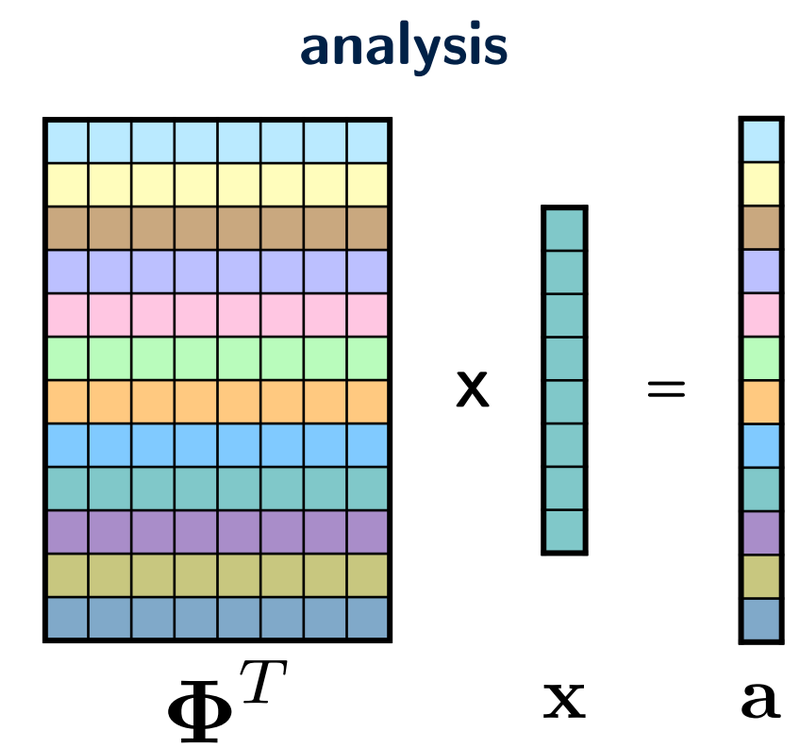
$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \boldsymbol{\phi}_n$$

Signal representation via dictionaries

- Overcomplete (redundant) dictionaries



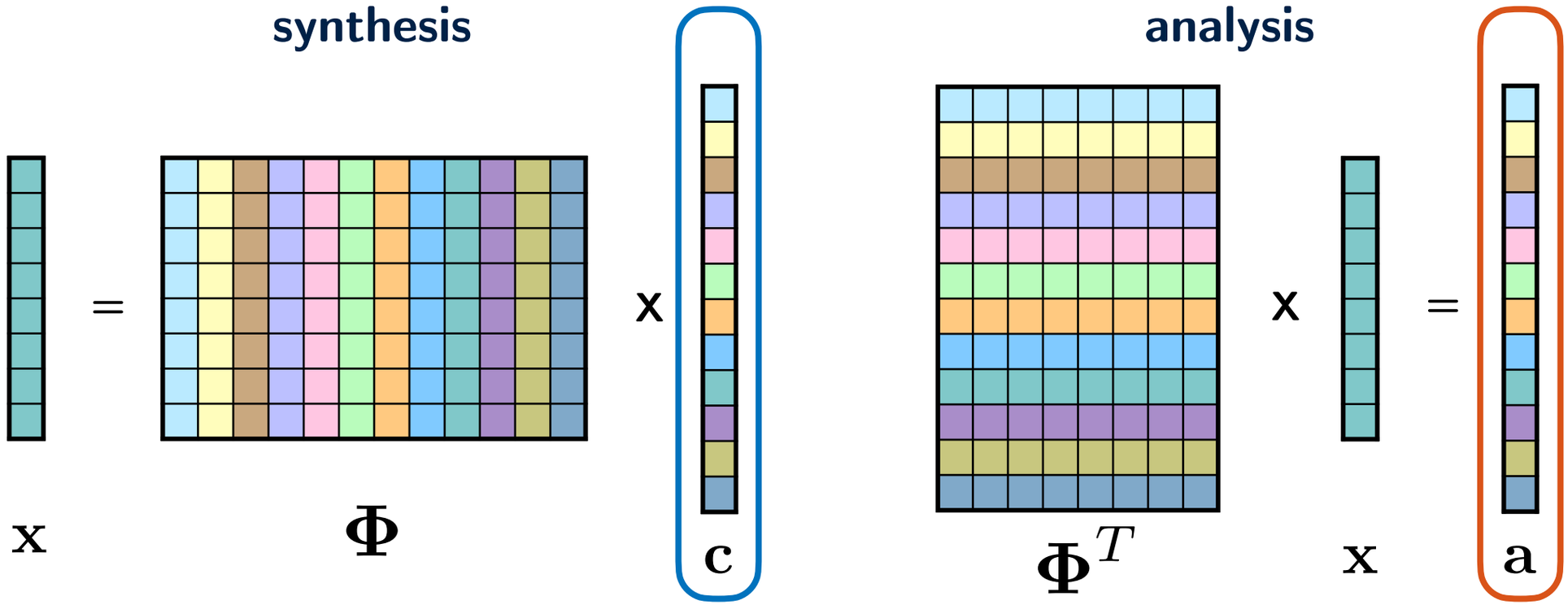
$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \phi_n$$



$$\mathbf{a}_n = \phi_n^T \mathbf{x}$$

Signal representation via dictionaries

- Overcomplete (redundant) dictionaries

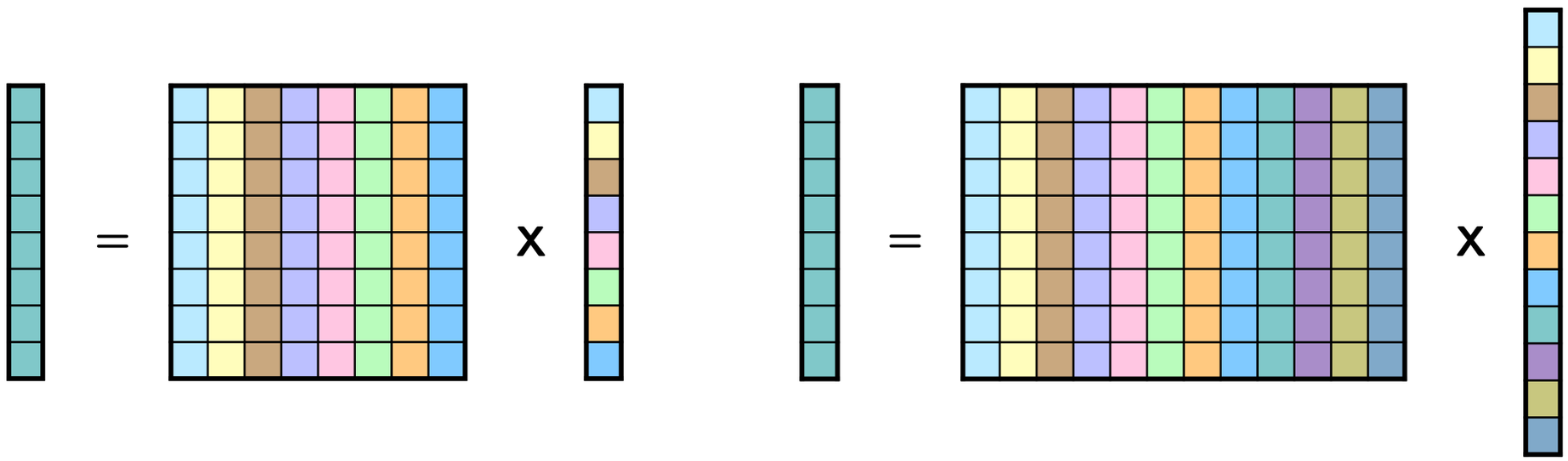


$$\mathbf{x} = \mathbf{\Phi c} = \sum_n \mathbf{c}_n \phi_n$$

$$\mathbf{a}_n = \phi_n^T \mathbf{x}$$

not equivalent for overcomplete dictionaries

Signal representation via dictionaries

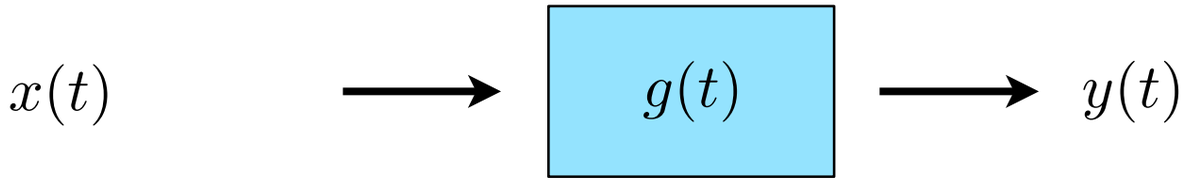


Two sources of dictionary design

- mathematical modelling of data (transforms/analytic dictionaries)
- a set of realisations of data (learned dictionaries)

1960s: Fourier basis and DFT

- recall the LTI system



1960s: Fourier basis and DFT

- recall the LTI system

$$x(t) = e^{j\omega t} \longrightarrow \boxed{g(t)} \longrightarrow y(t)$$

$$y(t) = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} g(\tau) d\tau = e^{j\omega t} G(\omega)$$

1960s: Fourier basis and DFT

discrete Fourier transform

- recall the LTI system

$$x(t) = e^{j\omega t} \longrightarrow \boxed{g(t)} \longrightarrow y(t)$$

$$y(t) = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} g(\tau) d\tau = e^{j\omega t} G(\omega)$$

$$X(\omega) \longrightarrow \boxed{G(\omega)} \longrightarrow Y(\omega) = X(\omega)G(\omega)$$

Fourier basis functions (real part)

Fourier basis diagonalises convolution operator

1960s: Fourier basis and DFT

- Fourier basis describes a signal in terms of its **global** frequency content and hence is good at representing **uniformly smooth** signals

1960s: Fourier basis and DFT

- Fourier basis describes a signal in terms of its **global** frequency content and hence is good at representing **uniformly smooth** signals
- **discrete Fourier transform (DFT)** provides an orthogonal dictionary: $\phi_n(k) = e^{j\frac{2\pi}{N}nk}$

$$\begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{pmatrix} = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \dots & W^{N-3} \\ \vdots & & & & & \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \dots & W \end{pmatrix} \begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{pmatrix} \quad \text{with} \quad W = e^{j\frac{2\pi}{N}}$$

- fast Fourier transform (FFT) reduces complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$

1960s: Fourier basis and DFT

- Fourier basis describes a signal in terms of its **global** frequency content and hence is good at representing **uniformly smooth** signals
- **discrete Fourier transform (DFT)** provides an orthogonal dictionary: $\phi_n(k) = e^{j\frac{2\pi}{N}nk}$

$$\begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{pmatrix} = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \dots & W^{N-3} \\ \vdots & & & & & \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \dots & W \end{pmatrix} \begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{pmatrix} \quad \text{with} \quad W = e^{j\frac{2\pi}{N}}$$

- fast Fourier transform (FFT) reduces complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$
- can be made into a real transform called **discrete cosine transform (DCT)**

The need for sparsity

- projection onto a fixed subset of DFT/DCT atoms leads to **compaction**

$$\mathbf{x} \approx \sum_{n \in \mathcal{S}_k} (\mathbf{\Psi}_n^T \mathbf{x}) \mathbf{\Phi}_n$$

The need for sparsity

- projection onto a fixed subset of DFT/DCT atoms leads to **compaction**

$$\mathbf{x} \approx \sum_{n \in \mathcal{S}_k} (\mathbf{\Psi}_n^T \mathbf{x}) \mathbf{\Phi}_n$$

- from compaction (simplicity) to **sparsity**: signal as linear combination of a few atoms
- sparsity requires shift from linear to **nonlinear** approximation

$$\mathbf{x} \approx \sum_{k \in \mathcal{K}} c_k \phi_k$$

subset of atoms (different for each x)

The need for sparsity

- projection onto a fixed subset of DFT/DCT atoms leads to **compaction**

$$\mathbf{x} \approx \sum_{n \in \mathcal{S}_k} (\mathbf{\Psi}_n^T \mathbf{x}) \mathbf{\Phi}_n$$

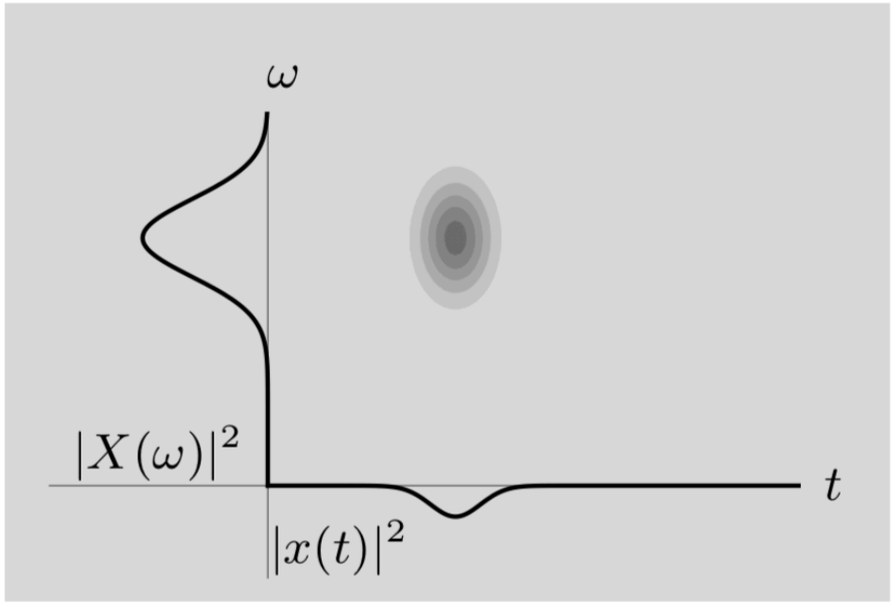
- from compaction (simplicity) to **sparsity**: signal as linear combination of a few atoms
- sparsity requires shift from linear to **nonlinear** approximation

$$\mathbf{x} \approx \sum_{k \in \mathcal{K}} c_k \phi_k$$

subset of atoms (different for each x)

- sparsity requires **localisation:** atoms with concentrated support
  - allow more flexible representations based on local characteristics
  - limit effects of irregularities (a main source of large coefficients)

Time-frequency representation

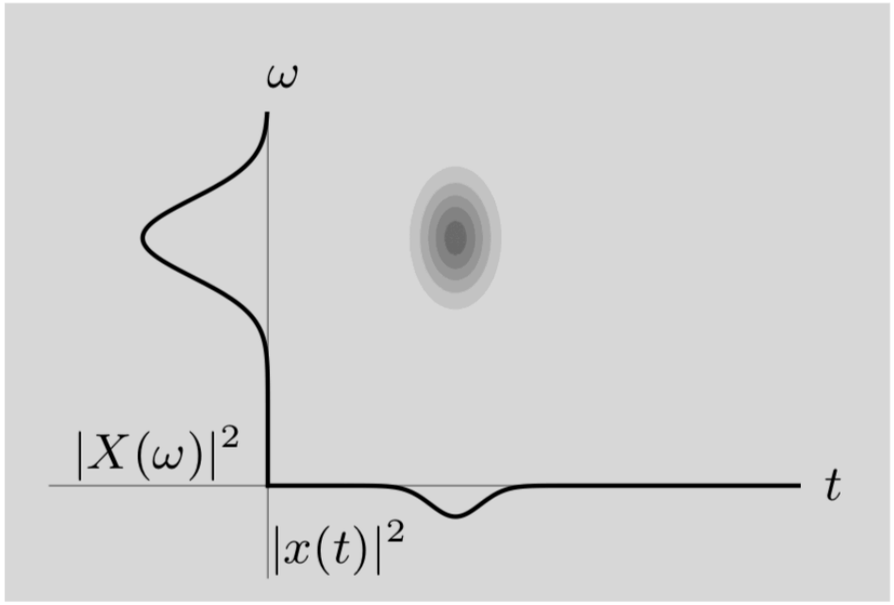


time localisation

$$\mu_t = \frac{1}{||x||^2} \int_{-\infty}^{\infty} t|x(t)|^2 dt$$

$$\Delta_t = \left(\frac{1}{||x||^2} \int_{-\infty}^{\infty} (t-\mu_t)^2 |x(t)|^2 dt \right)^{\frac{1}{2}}$$

Time-frequency representation



time localisation

$$\mu_t = \frac{1}{||x||^2} \int_{-\infty}^{\infty} t|x(t)|^2 dt$$

$$\Delta_t = \left(\frac{1}{||x||^2} \int_{-\infty}^{\infty} (t - \mu_t)^2 |x(t)|^2 dt \right)^{\frac{1}{2}}$$

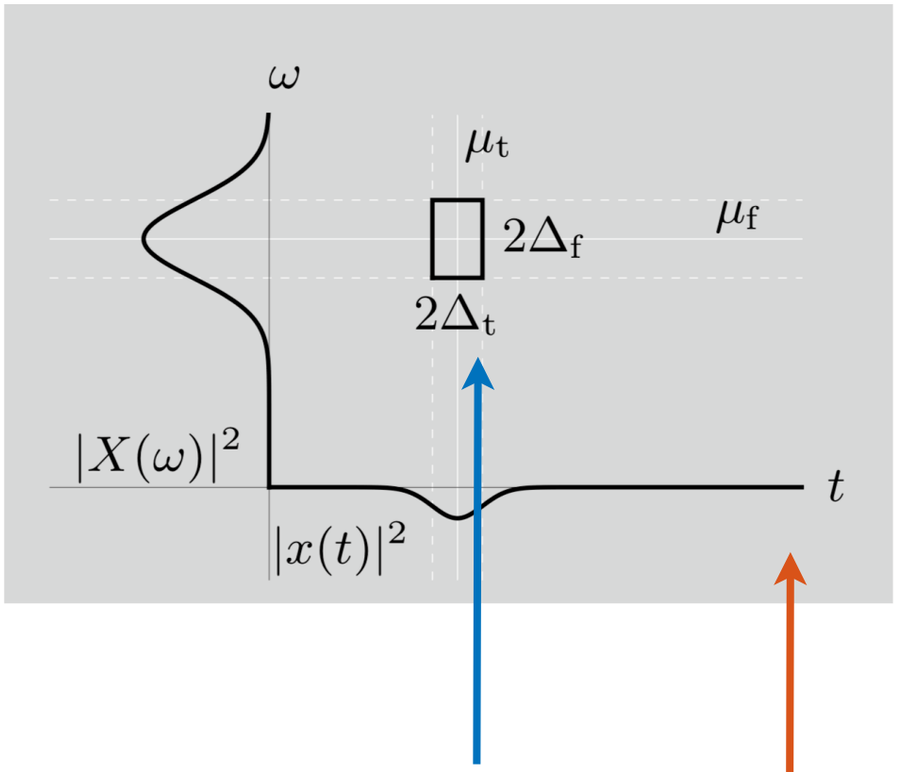
frequency localisation

$$\mu_f = \frac{1}{2\pi||x||^2} \int_{-\infty}^{\infty} \omega |X(\omega)|^2 d\omega$$

$$\Delta_f = \left(\frac{1}{2\pi||x||^2} \int_{-\infty}^{\infty} (\omega - \mu_f)^2 |X(\omega)|^2 d\omega \right)^{\frac{1}{2}}$$

Vetterli et al., "Foundations of signal processing," Cambridge University Press, 2014.

Time-frequency representation



time-frequency tile (Heisenberg box)

time-frequency plane

time localisation

$$\mu_t = \frac{1}{||x||^2} \int_{-\infty}^{\infty} t|x(t)|^2 dt$$

$$\Delta_t = \left(\frac{1}{||x||^2} \int_{-\infty}^{\infty} (t-\mu_t)^2 |x(t)|^2 dt \right)^{\frac{1}{2}}$$

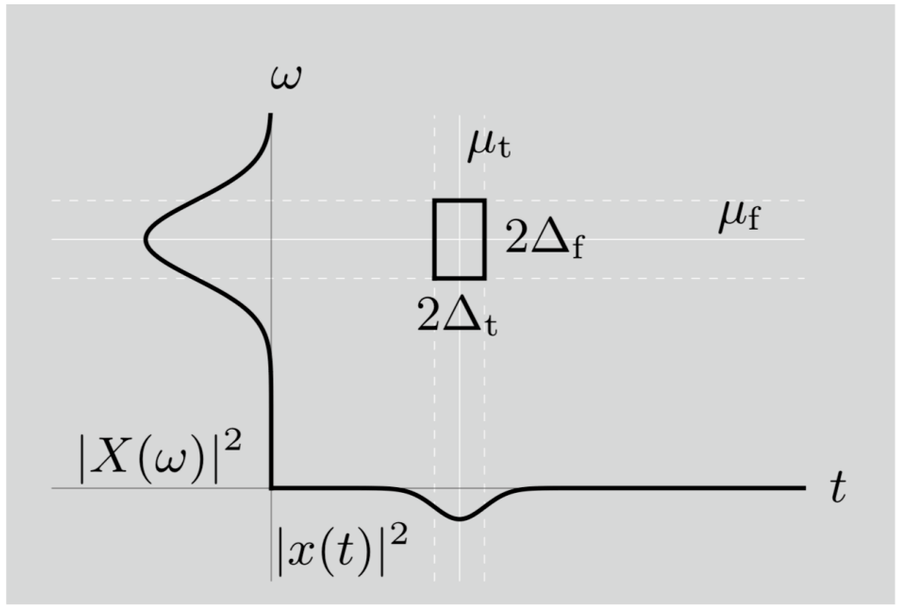
frequency localisation

$$\mu_f = \frac{1}{2\pi||x||^2} \int_{-\infty}^{\infty} \omega |X(\omega)|^2 d\omega$$

$$\Delta_f = \left(\frac{1}{2\pi||x||^2} \int_{-\infty}^{\infty} (\omega-\mu_f)^2 |X(\omega)|^2 d\omega \right)^{\frac{1}{2}}$$

Vetterli et al., "Foundations of signal processing," Cambridge University Press, 2014.

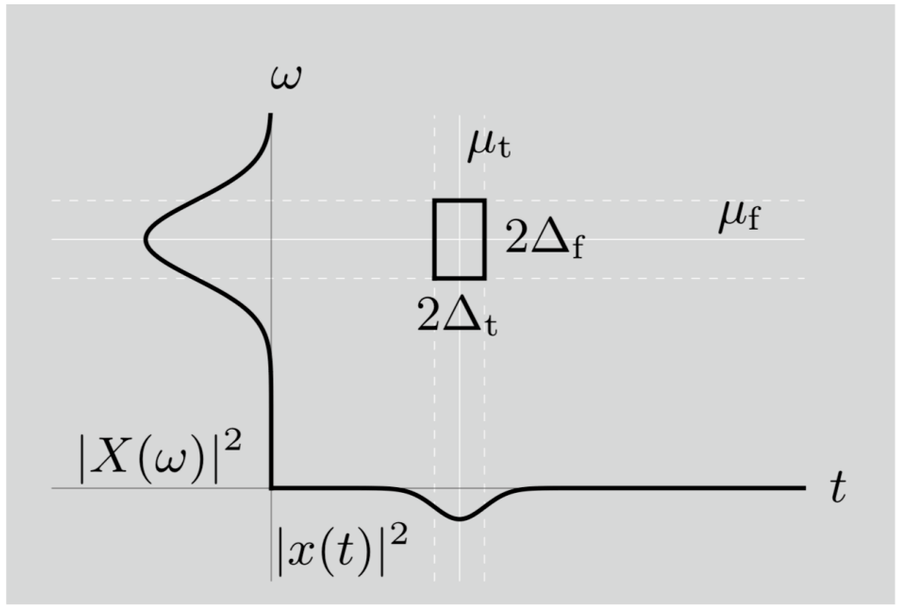
Time-frequency representation



Heisenberg's uncertainty principle

$$\text{Let } x \in \mathcal{L}^2(\mathbb{R}), \text{ then } \Delta_t \Delta_f \geq \frac{1}{2}$$

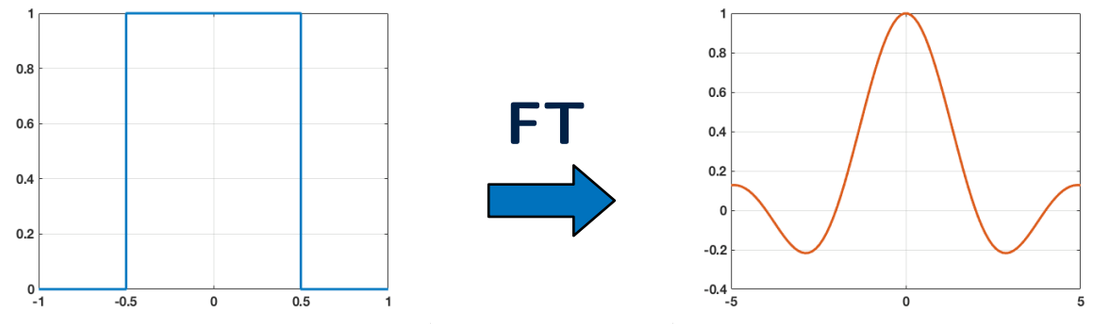
Time-frequency representation

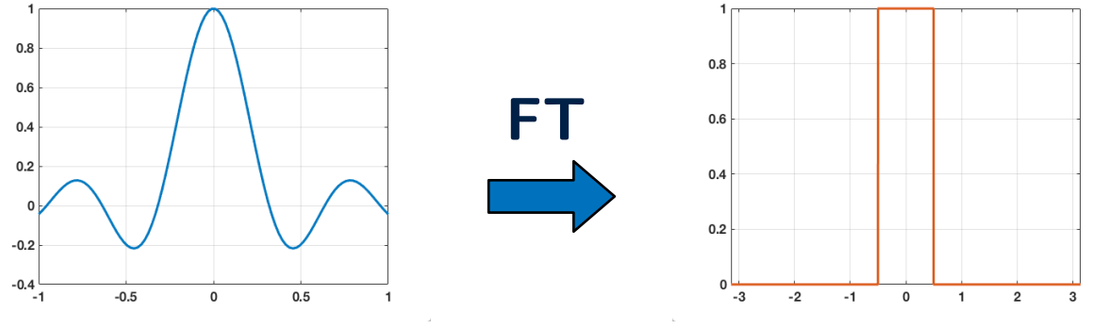


Heisenberg's uncertainty principle

$$\text{Let } x \in \mathcal{L}^2(\mathbb{R}), \text{ then } \Delta_t \Delta_f \geq \frac{1}{2}$$

examples



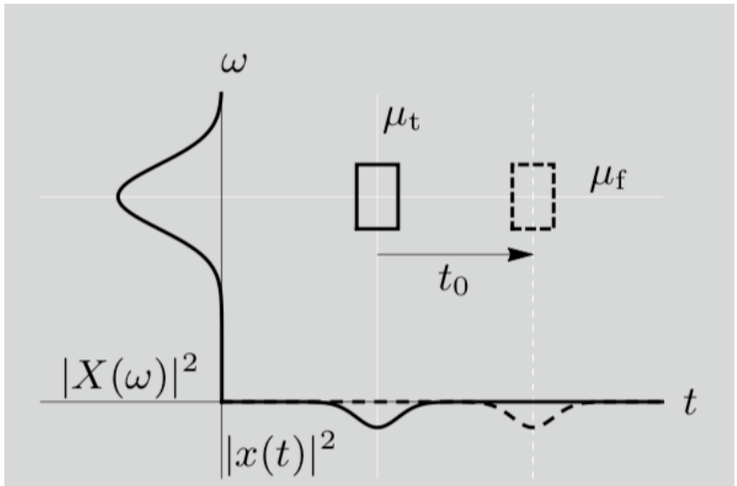


Vetterli et al., "Foundations of signal processing," Cambridge University Press, 2014.

Time-frequency representation

- Consider three basic operations

shift in time



$$y(t) = x(t - t_0)$$

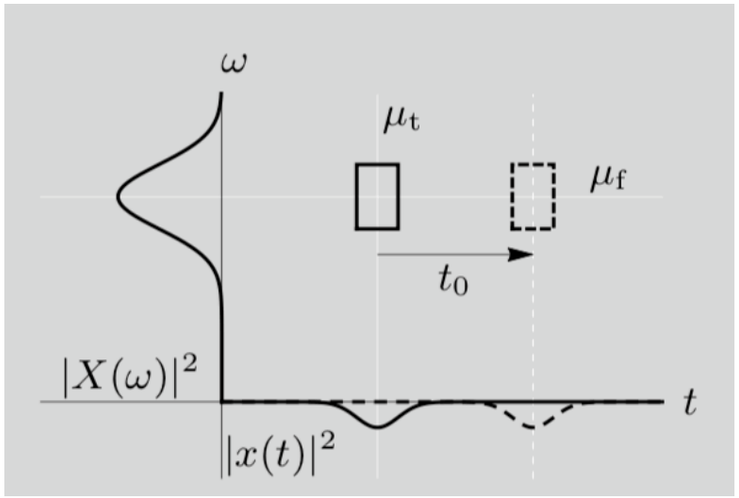
FT ⬇

$$Y(\omega) = e^{-j\omega t_0} X(\omega)$$

Time-frequency representation

- Consider three basic operations

shift in time

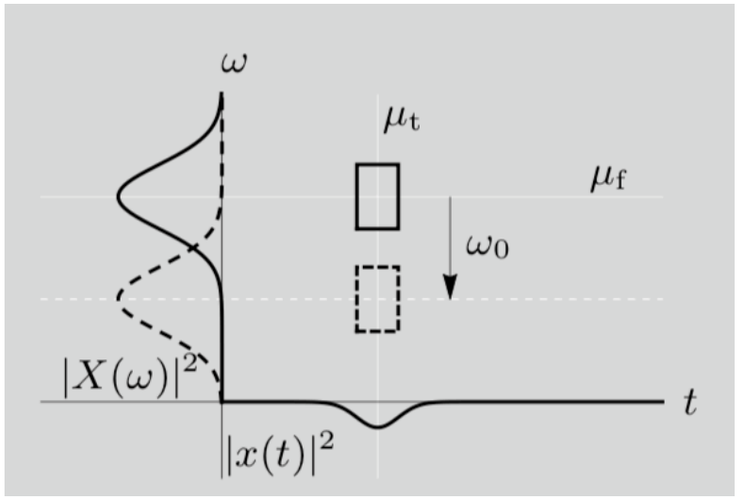


$$y(t) = x(t - t_0)$$

FT

$$Y(\omega) = e^{-j\omega t_0} X(\omega)$$

shift in frequency



$$y(t) = e^{j\omega_0 t} x(t)$$

FT

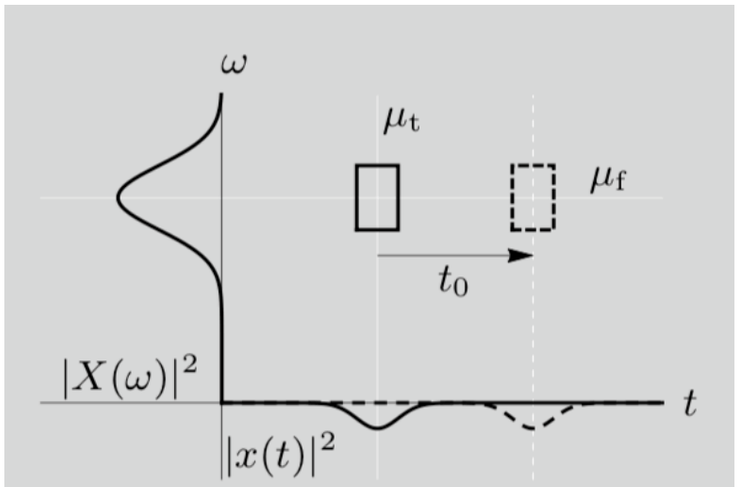
$$Y(\omega) = X(\omega - \omega_0)$$

Vetterli et al., "Foundations of signal processing," Cambridge University Press, 2014.

Time-frequency representation

- Consider three basic operations

shift in time

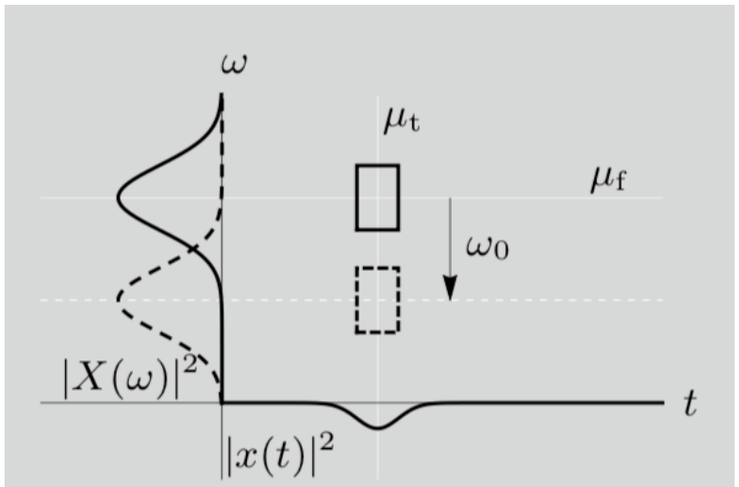


$$y(t) = x(t - t_0)$$

FT

$$Y(\omega) = e^{-j\omega t_0} X(\omega)$$

shift in frequency

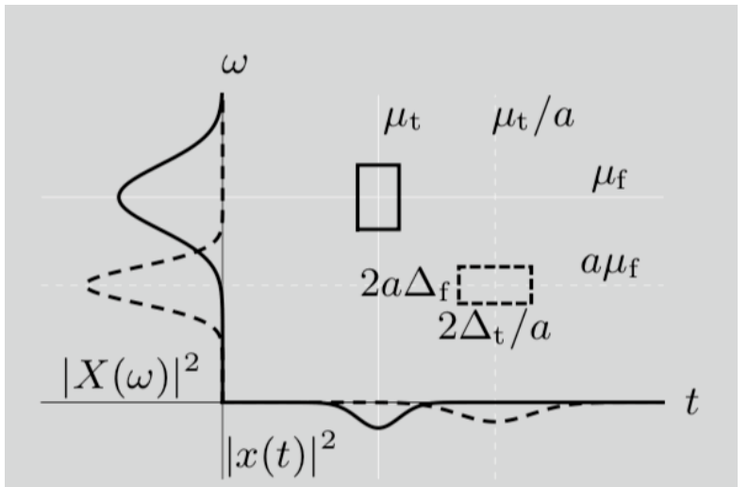


$$y(t) = e^{j\omega_0 t} x(t)$$

FT

$$Y(\omega) = X(\omega - \omega_0)$$

scaling in time



$$y(t) = \sqrt{a} x(at)$$

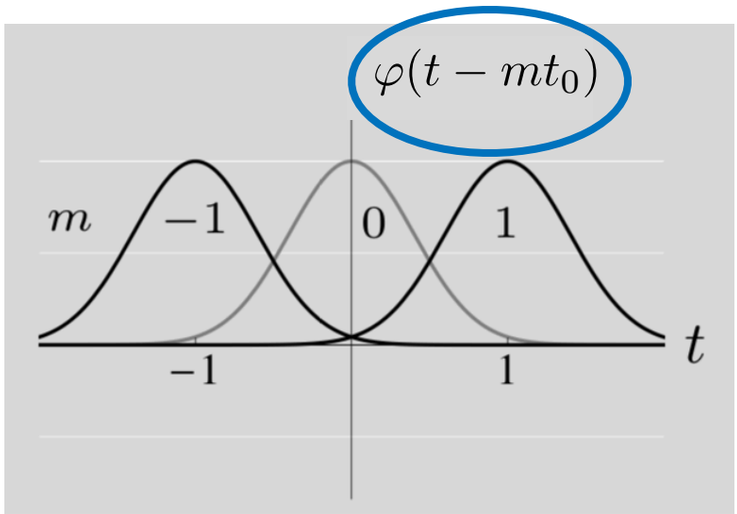
FT

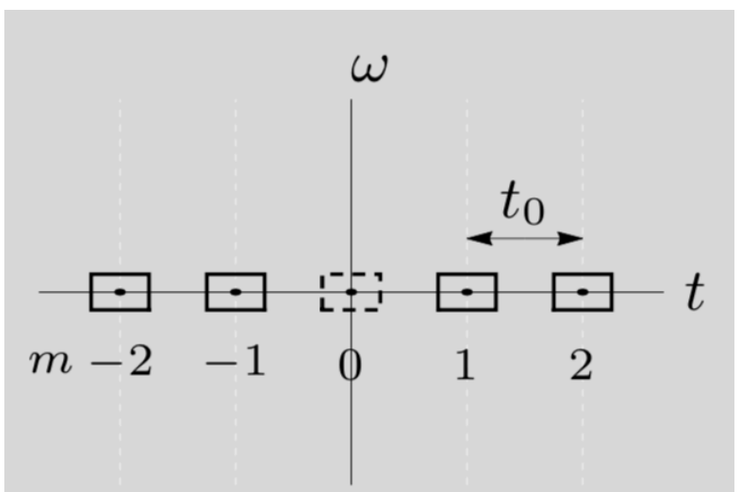
$$Y(\omega) = \frac{1}{\sqrt{a}} X(\frac{\omega}{a})$$

Time-frequency representation

- Consider three structured sets of functions

shift in time

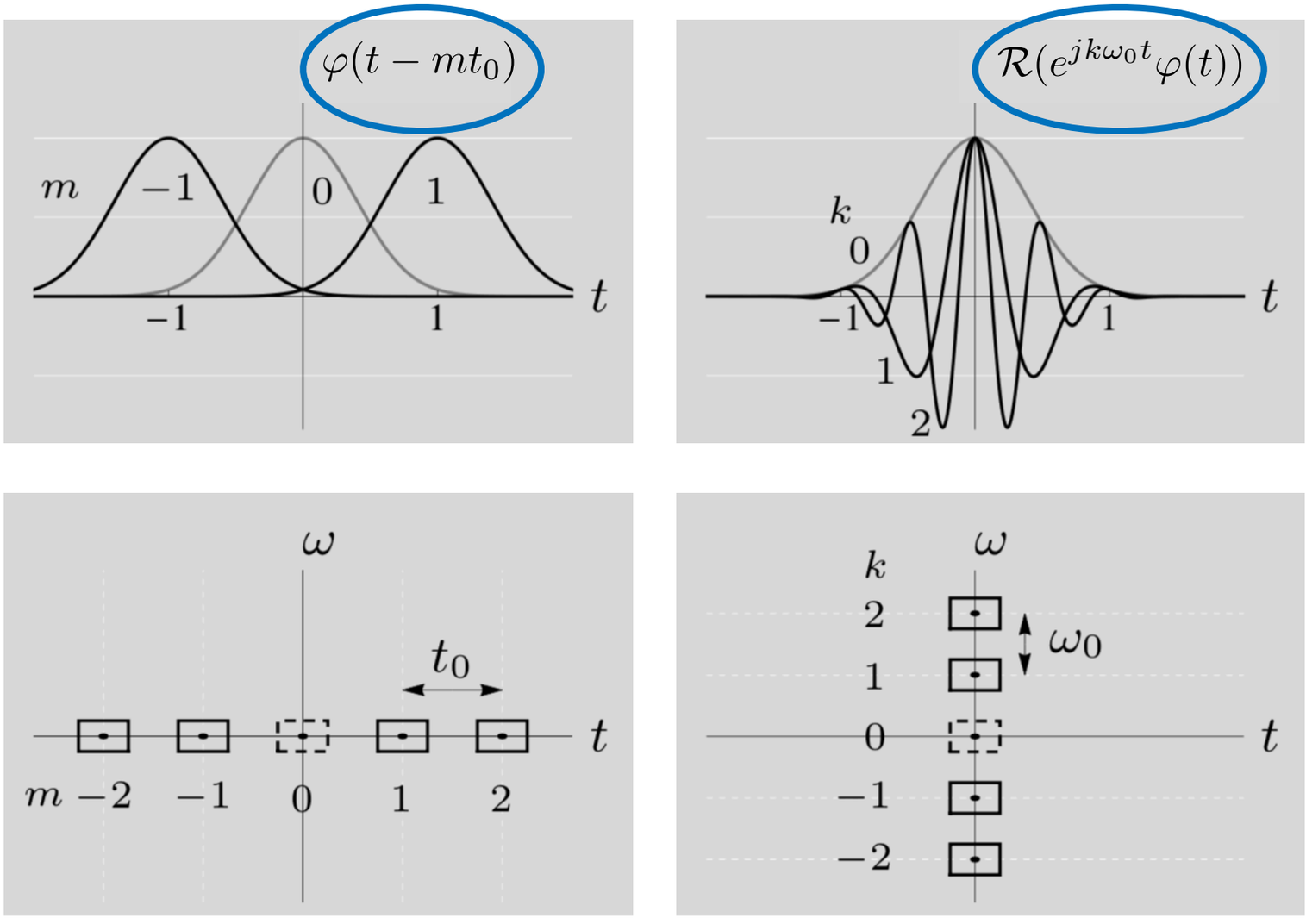




Time-frequency representation

- Consider three structured sets of functions

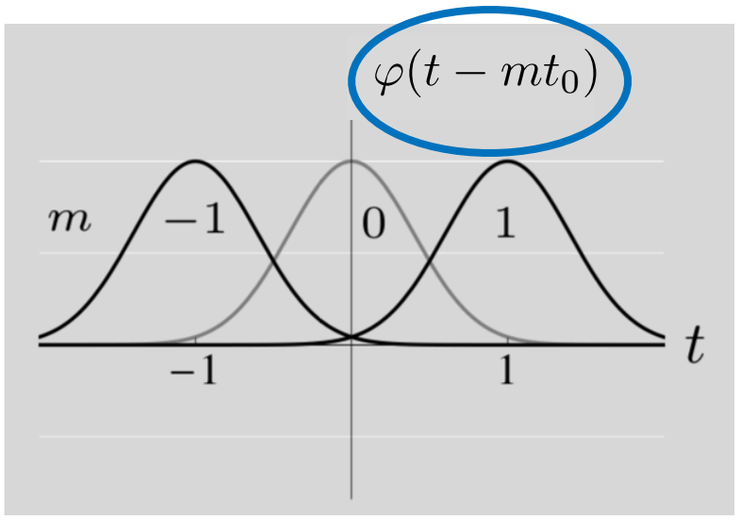
shift in time | **shift in frequency**

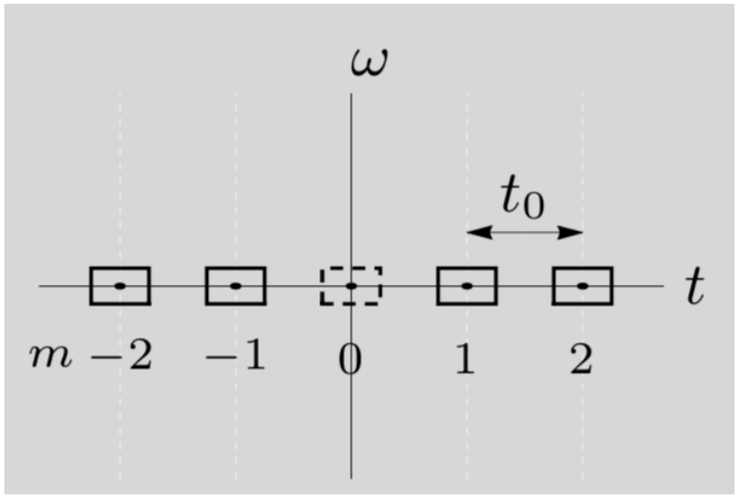


Time-frequency representation

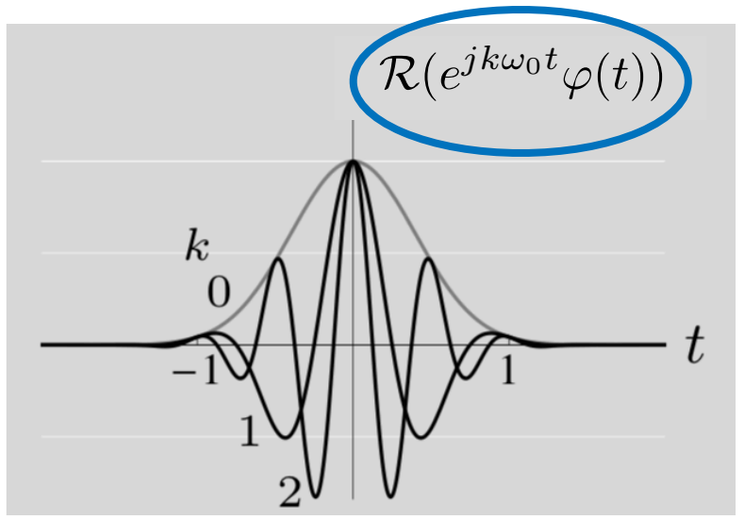
- Consider three structured sets of functions

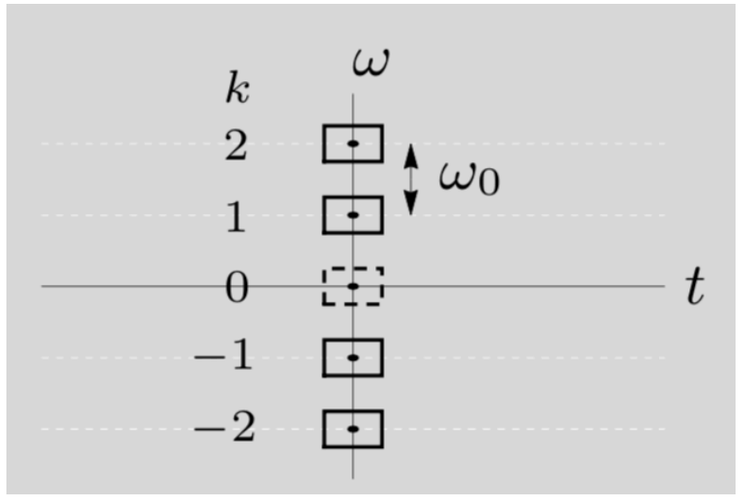
shift in time



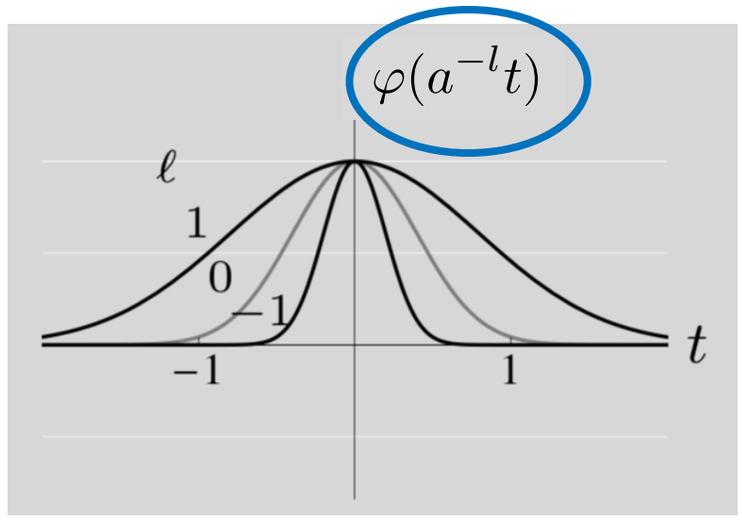


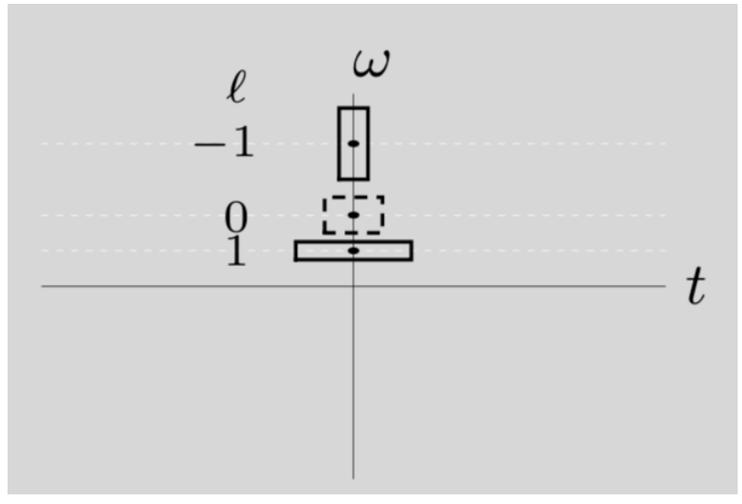
shift in frequency





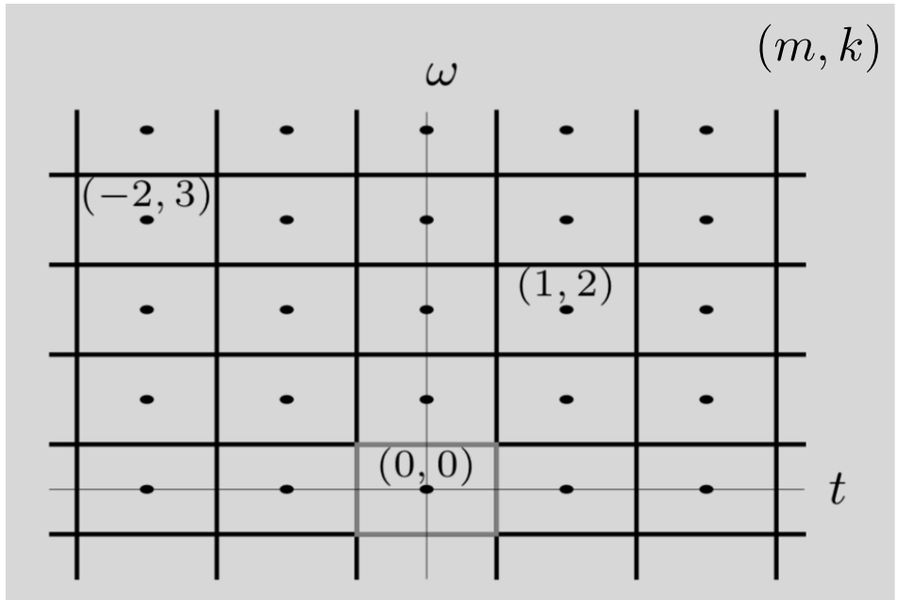
scaling in time





Time-frequency representation

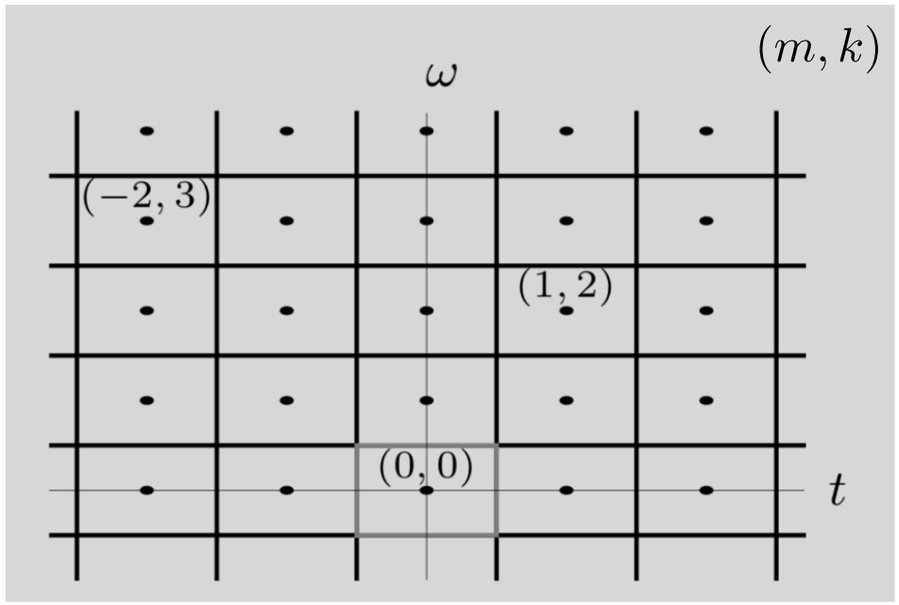
time shift and modulation



$$\varphi_{k,m}(t) = e^{jk\omega_0 t}\varphi(t - mt_0) \quad k,m \in \mathbb{Z}$$

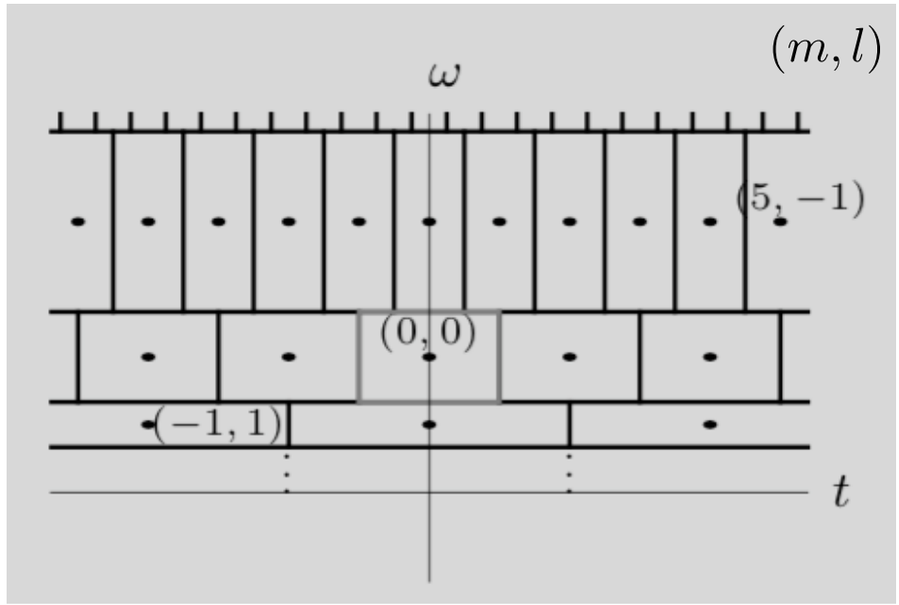
Time-frequency representation

time shift and modulation



$$\varphi_{k,m}(t) = e^{jk\omega_0 t}\varphi(t - mt_0) \quad k,m \in \mathbb{Z}$$

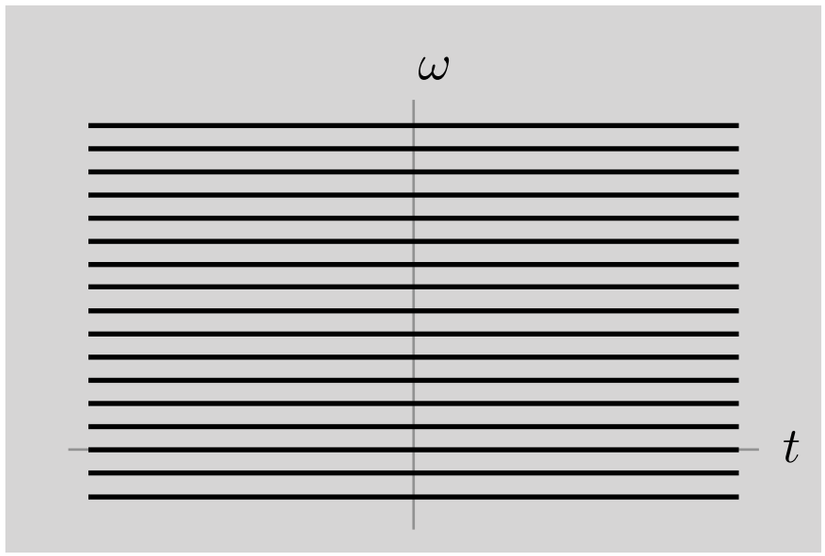
time shift and scaling



$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) \quad l,m \in \mathbb{Z}$$
$$= \varphi(a^{-l}(t - ma^l t_0))$$

1970s-80s: STFT

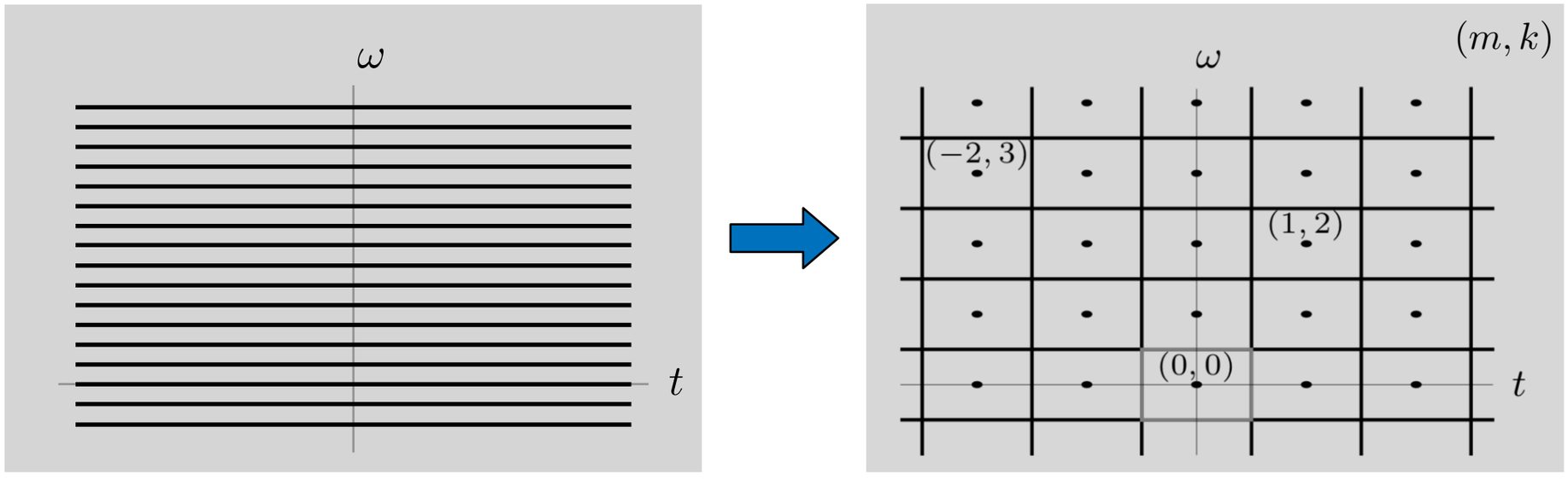
- Fourier basis functions (complex exponentials) are not localised in time



1970s-80s: STFT

- Fourier basis functions (complex exponentials) are not localised in time
- consider a set of shifted and modulated versions of a low-pass function

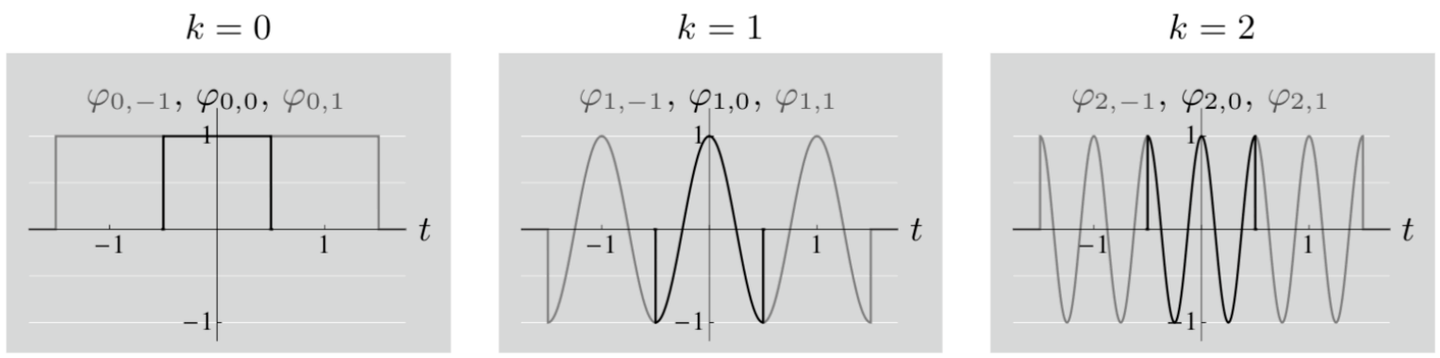
$$\varphi_{k,m}(t) = e^{jk\omega_0 t}\varphi(t - mt_0) \quad k, m \in \mathbb{Z}$$



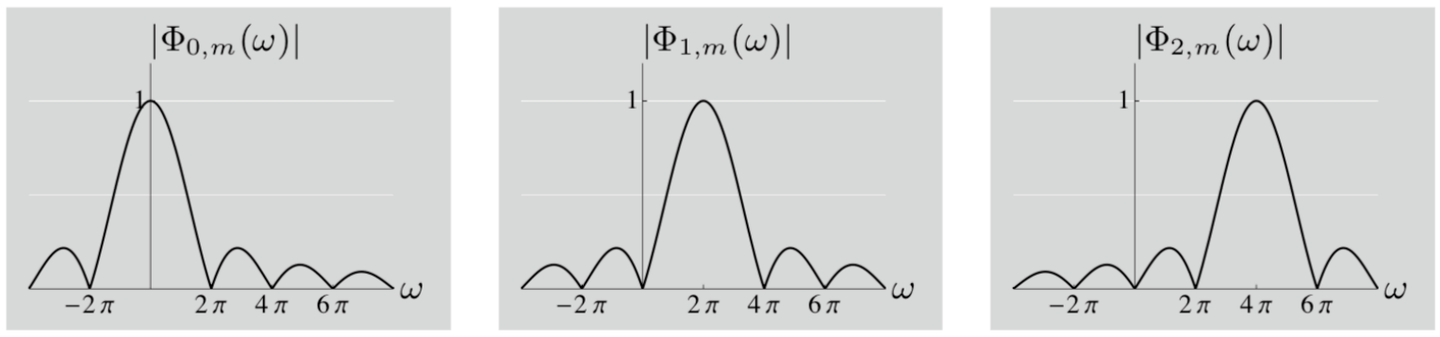
1970s-80s: STFT

- example: consider a box function and $t_0 = 1$, $\omega_0 = 2\pi$

$$\varphi_{k,m}(t) = e^{jk2\pi t}\varphi(t-m), \quad \varphi(t) = \begin{cases} 1, & \text{for } |t| \leq \frac{1}{2}; \\ 0, & \text{otherwise.} \end{cases}$$



Basis functions (real parts only).



Magnitudes of the Fourier transform.

Vetterli et al., "Foundations of signal processing," Cambridge University Press, 2014.

1970s-80s: STFT

- we can define the following transform

$$X_{k,m} = \int_{-\infty}^{\infty} x(t)\varphi_{k,m}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t - mt_0)e^{-jk\omega_0 t}dt$$

1970s-80s: STFT

- we can define the following transform

$$X_{k,m} = \int_{-\infty}^{\infty} x(t)\varphi_{k,m}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t - mt_0)e^{-jk\omega_0 t}dt$$

$$\Rightarrow \quad X(\omega,\tau) = \int_{-\infty}^{\infty} x(t)\varphi_{\omega,\tau}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t - \tau)e^{-j\omega t}dt$$

1970s-80s: STFT

- we can define the following transform

$$X_{k,m} = \int_{-\infty}^{\infty} x(t)\varphi_{k,m}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t - mt_0)e^{-jk\omega_0 t}dt$$

$$\Rightarrow \quad X(\omega,\tau) = \int_{-\infty}^{\infty} x(t)\varphi_{\omega,\tau}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t-\tau)e^{-j\omega t}dt$$

- applying **time-localised** window to the signal before taking Fourier transform: windowed or **short-time Fourier transform (STFT)**

1970s-80s: STFT

- we can define the following transform

$$X_{k,m} = \int_{-\infty}^{\infty} x(t)\varphi_{k,m}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t - mt_0)e^{-jk\omega_0 t}dt$$

$$\Rightarrow \quad X(\omega,\tau) = \int_{-\infty}^{\infty} x(t)\varphi_{\omega,\tau}^{*}(t)dt = \int_{-\infty}^{\infty} x(t)\varphi(t-\tau)e^{-j\omega t}dt$$

- applying **time-localised** window to the signal before taking Fourier transform: windowed or **short-time Fourier transform (STFT)**
- Gaussian window achieves localisation in frequency: Gabor transform
- STFT maps a 1-D function into a 2-D function (**overcomplete**)

DCT vs STFT

- discrete STFT generally provides an overcomplete dictionary

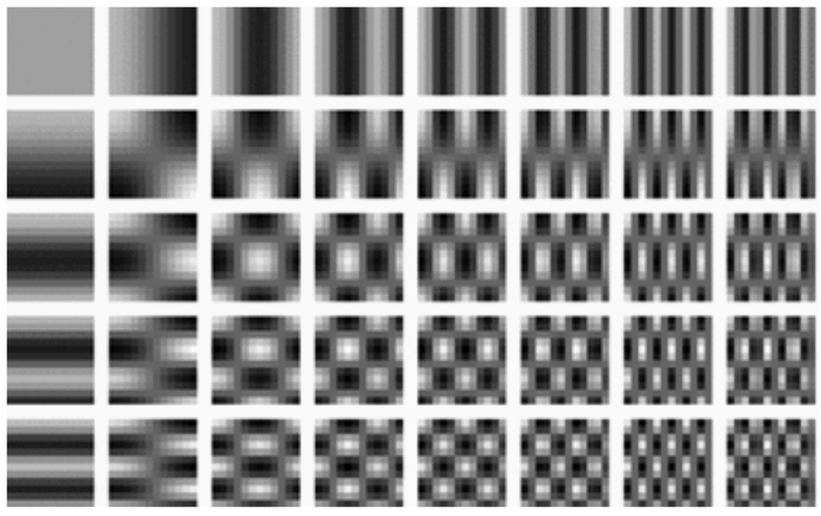
$$\phi_{k,m}(n) = e^{j\frac{2\pi}{N}nk}\varphi(n - mN)$$

DCT vs STFT

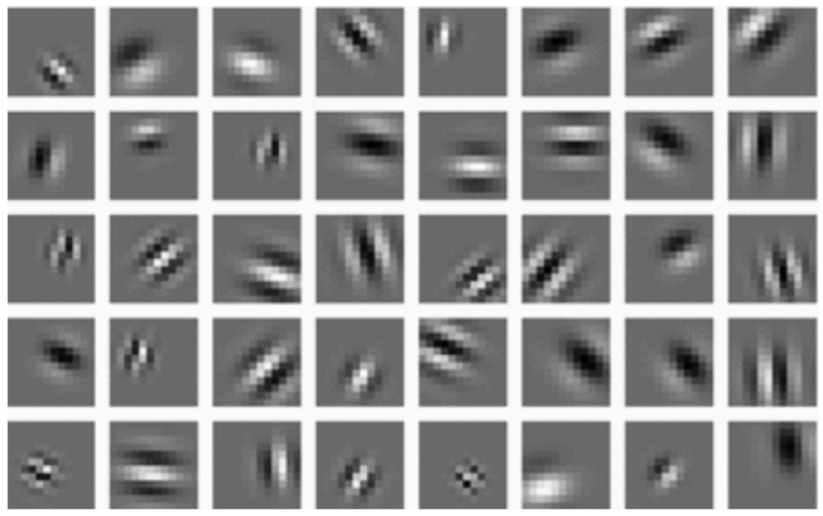
- discrete STFT generally provides an overcomplete dictionary

$$\phi_{k,m}(n) = e^{j\frac{2\pi}{N}nk}\varphi(n - mN)$$

DCT



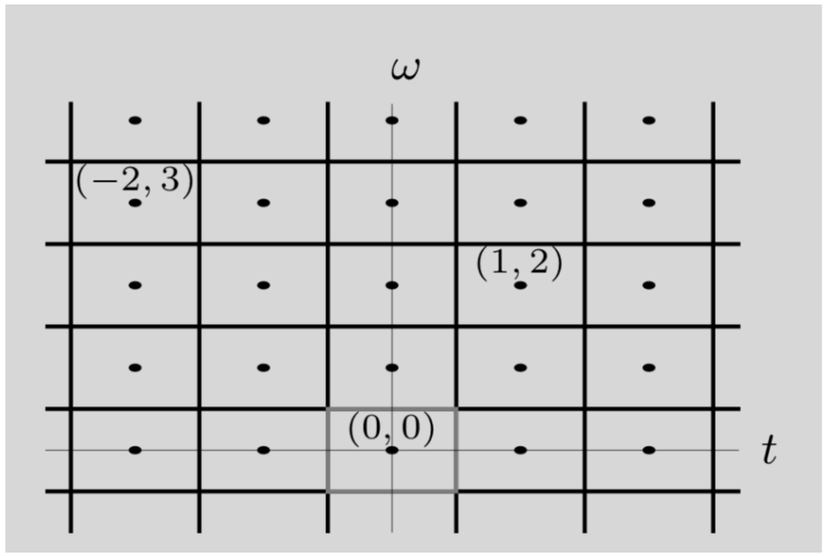
STFT



Rubinstein et al., "Dictionaries for sparse representation modeling," Proceedings of the IEEE, 2010.

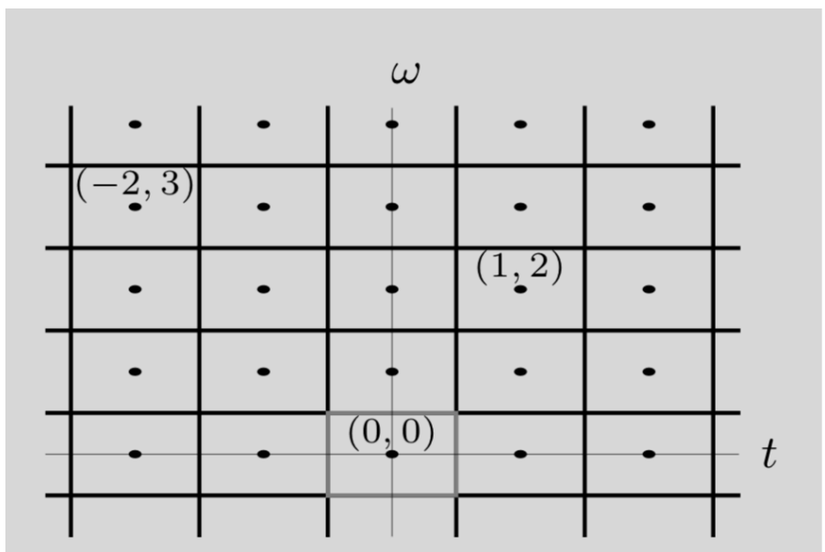
1980s: Wavelet transform

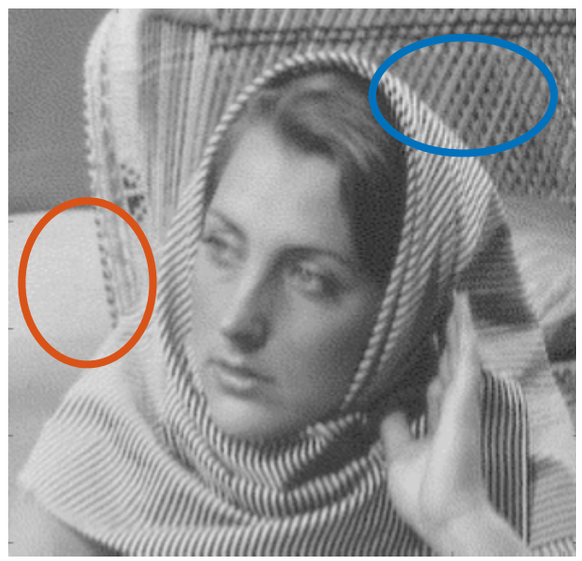
- STFT atoms have **fixed time-frequency resolution**



1980s: Wavelet transform

- STFT atoms have **fixed time-frequency resolution**
- often times a **multiresolution** representation is needed to capture various scales in natural signals

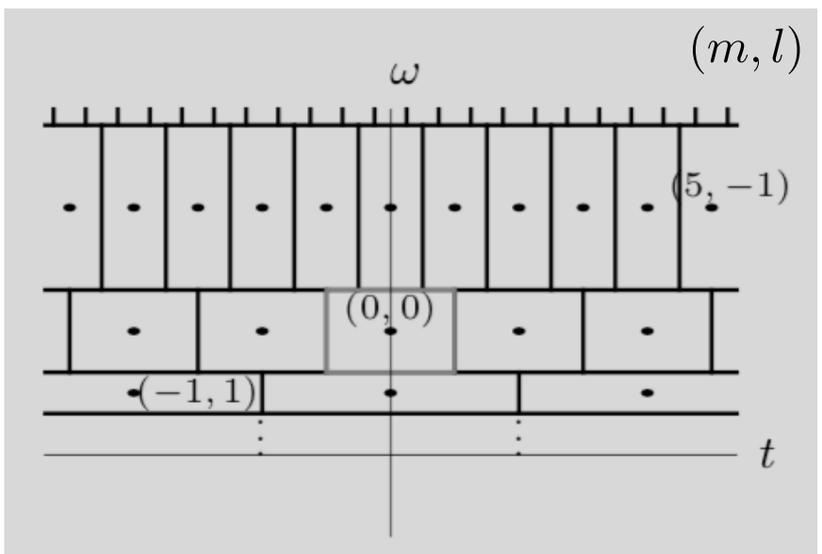




1980s: Wavelet transform

- consider a set of shifted and scaled versions of a band-pass function

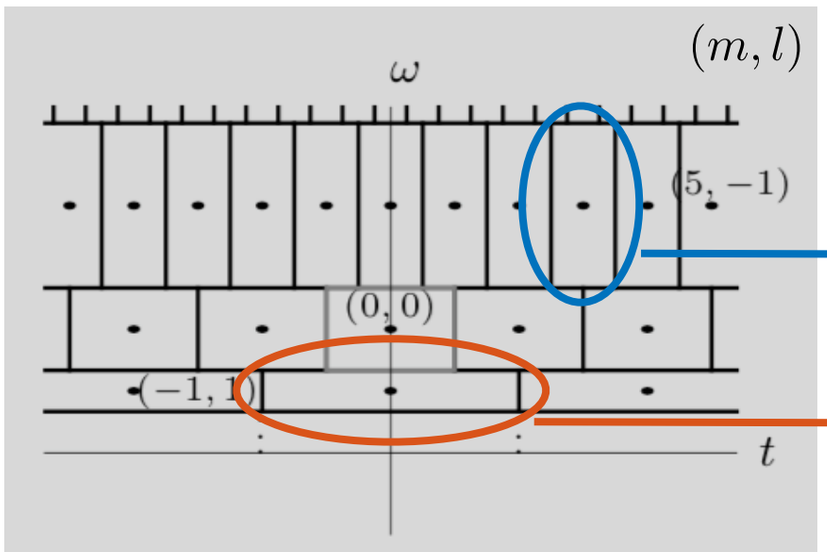
$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) = \varphi(\frac{t - ma^l t_0}{a^l}) \quad l, m \in \mathbb{Z}$$



1980s: Wavelet transform

- consider a set of shifted and scaled versions of a band-pass function

$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) = \varphi(\frac{t - ma^l t_0}{a^l}) \quad l, m \in \mathbb{Z}$$



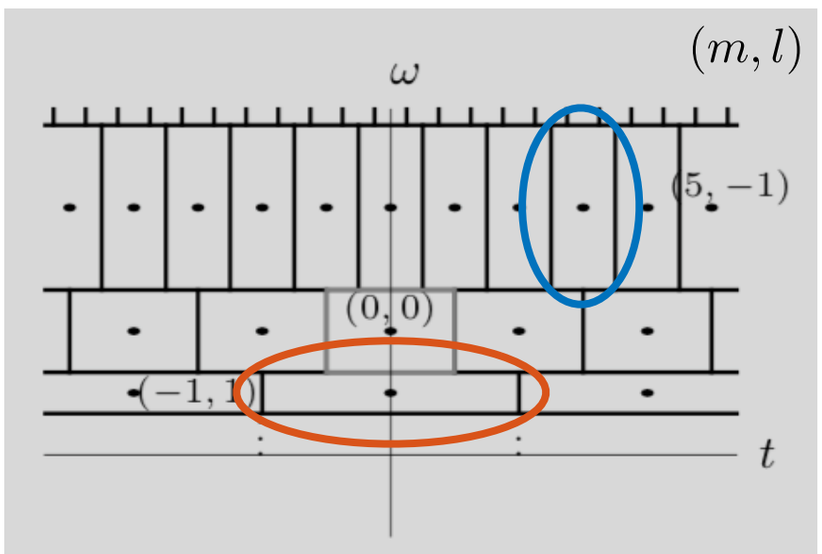
good time resolution (short-term)

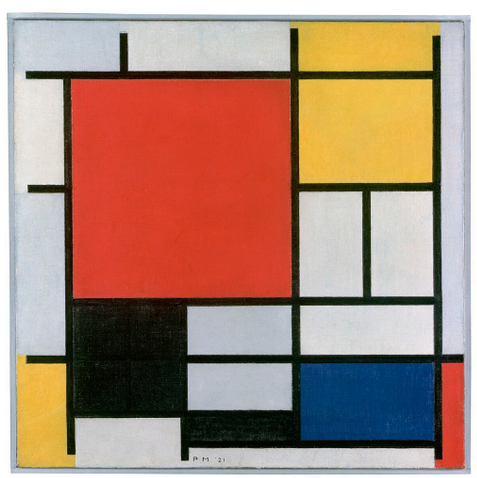
good frequency resolution (long-term)

1980s: Wavelet transform

- consider a set of shifted and scaled versions of a band-pass function

$$\varphi_{l,m}(t) = \varphi(a^{-l}t - mt_0) = \varphi(\frac{t - ma^l t_0}{a^l}) \quad l, m \in \mathbb{Z}$$



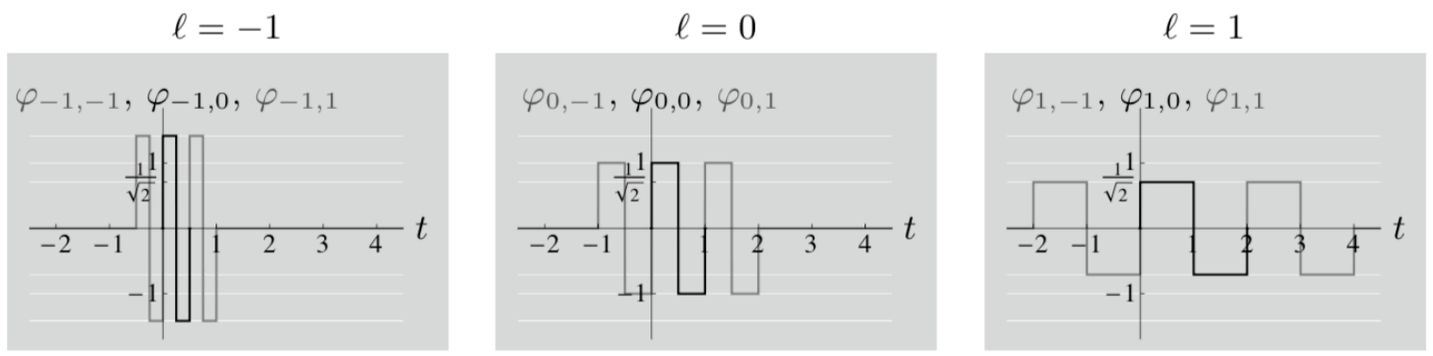


Piet Mondrian (1921)

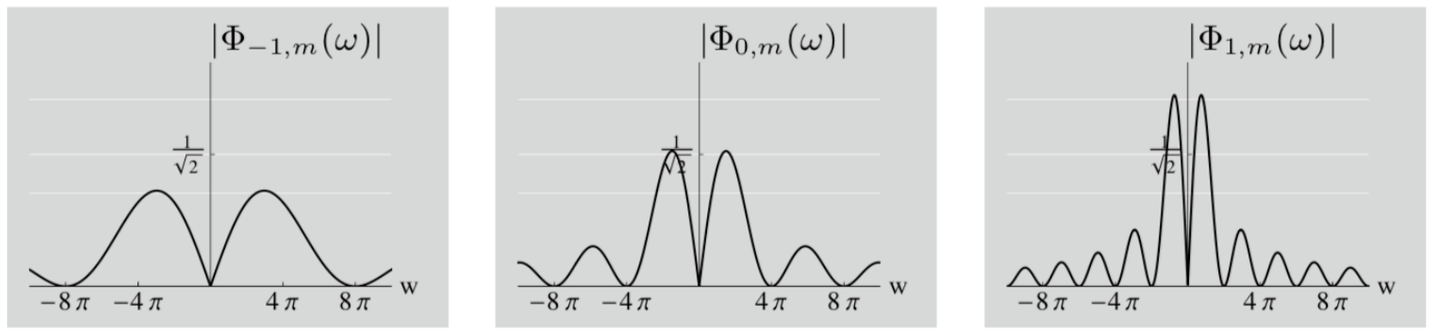
1980s: Wavelet transform

- example: consider a square wave function and $t_0 = 1$, $a = 2$

$$\varphi_{l,m}(t) = \varphi(\frac{t - 2^l m}{2^l}), \quad \varphi(t) = \begin{cases} 1, & \text{for } 0 \le t < \frac{1}{2}; \\ -1, & \text{for } \frac{1}{2} \le t < 1; \\ 0, & \text{otherwise.} \end{cases}$$



Basis functions.



Magnitudes of the Fourier transform.

1980s: Wavelet transform

- consider a more general function and define the following transform

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s}) \Rightarrow X(s,\tau) = \int_{-\infty}^{\infty} x(t)\psi^*_{s,\tau}(t)dt = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{s}}\psi^*(\frac{t-\tau}{s})dt$$

1980s: Wavelet transform

- consider a more general function and define the following transform

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s}) \Rightarrow X(s,\tau) = \int_{-\infty}^{\infty} x(t)\psi^*_{s,\tau}(t)dt = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{s}}\psi^*(\frac{t-\tau}{s})dt$$

- the prototype function $\psi(t)$
  - has a compact support (small or "-let")
  - is band-pass with zero mean ("wave"): $\int_{-\infty}^{\infty} \psi(t)dt = 0$

1980s: Wavelet transform

- consider a more general function and define the following transform

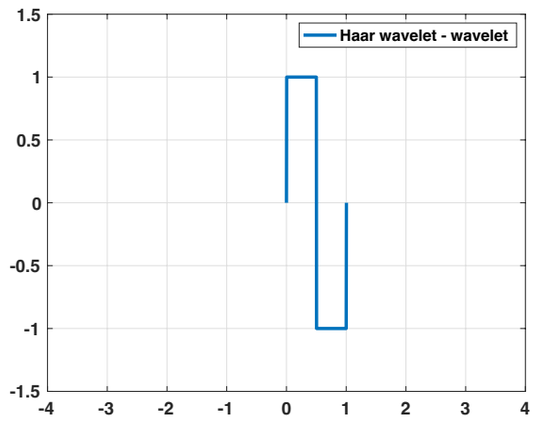
$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s}) \Rightarrow X(s,\tau) = \int_{-\infty}^{\infty} x(t)\psi^*_{s,\tau}(t)dt = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{s}}\psi^*(\frac{t-\tau}{s})dt$$

- the prototype function $\psi(t)$
  - has a compact support (small or "-let")
  - is band-pass with zero mean ("wave"): $\int_{-\infty}^{\infty} \psi(t)dt = 0$
- this is called the **continuous wavelet transform (CWT)**
- CWT maps a 1-D function into a 2-D function (**overcomplete**)

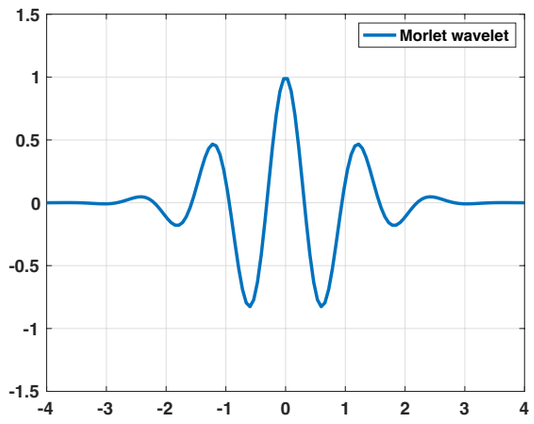
1980s: Wavelet transform

- examples of prototype function (mother wavelet)

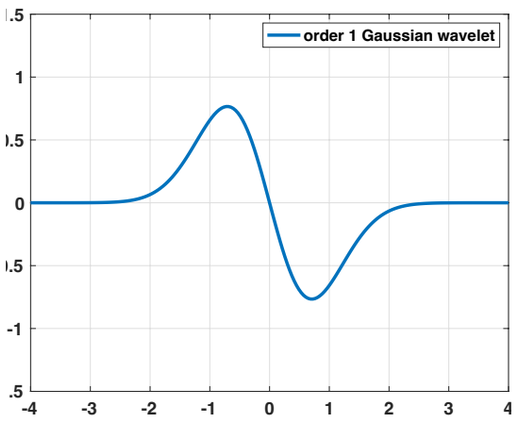
Haar



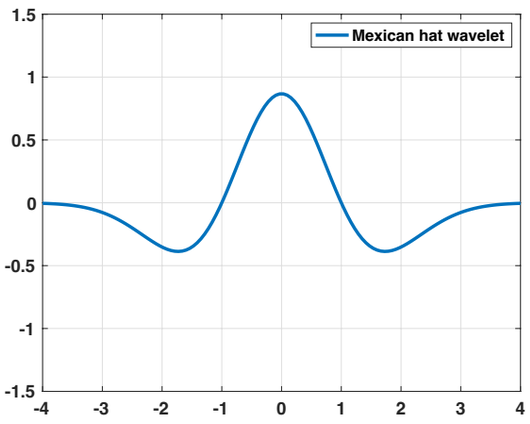
Morlet



Derivative of Gaussian



Marr (Mexican hat)



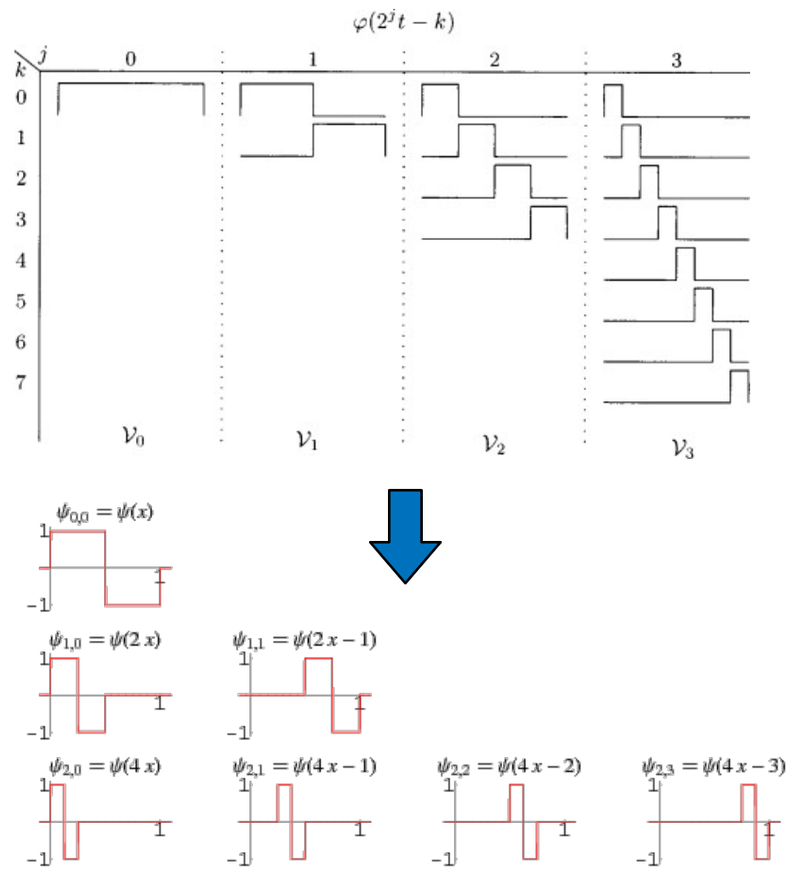
1980s: Wavelet transform

- CWT is an overcomplete transform; however, we can design an orthogonal wavelet transform through a **multiresolution analysis**

$$\psi_{l,m}(t) = \frac{1}{\sqrt{2^l}}\psi(\frac{t-2^l m}{2^l})$$

design principle

- form nested multiresolution spaces using scaling function φ
- obtain wavelets ψ by difference between nested subspaces



1980s: Wavelet transform

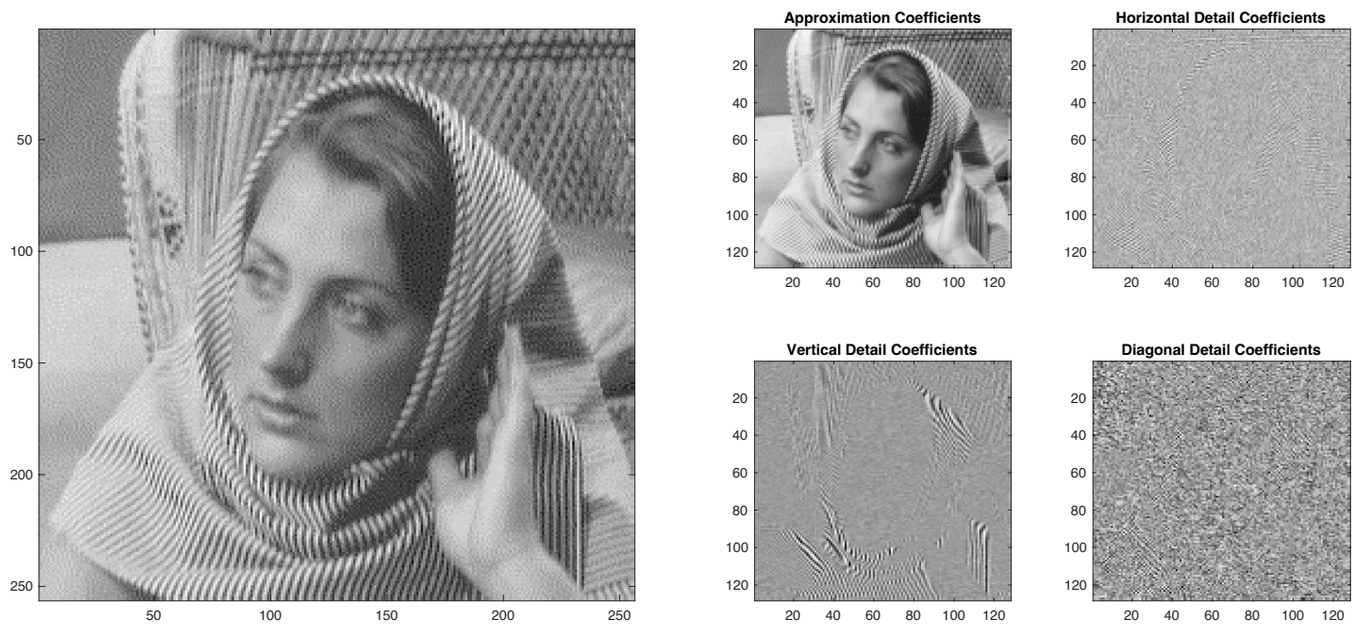
- this leads to the **discrete wavelet transform (DWT)** which provides an **orthogonal** dictionary

1980s: Wavelet transform

- this leads to the **discrete wavelet transform (DWT)** which provides an **orthogonal** dictionary

1980s: Wavelet transform

- this leads to the **discrete wavelet transform (DWT)** which provides an **orthogonal** dictionary

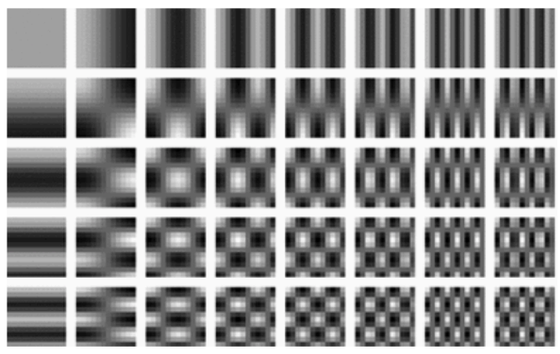


- DWT is behind the JEPG 2000 image compression standard

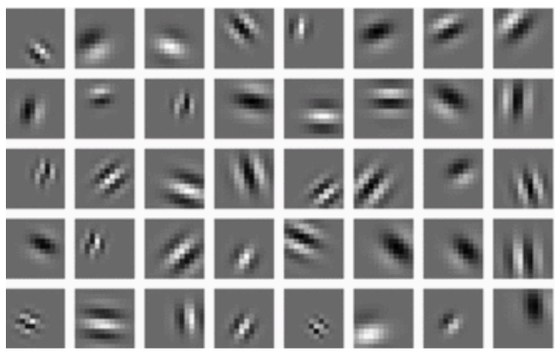
DCT vs STFT vs DWT

- comparison of the dictionaries we looked at so far

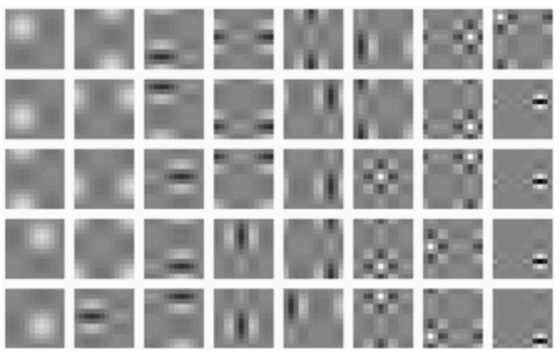
DCT



STFT



DWT

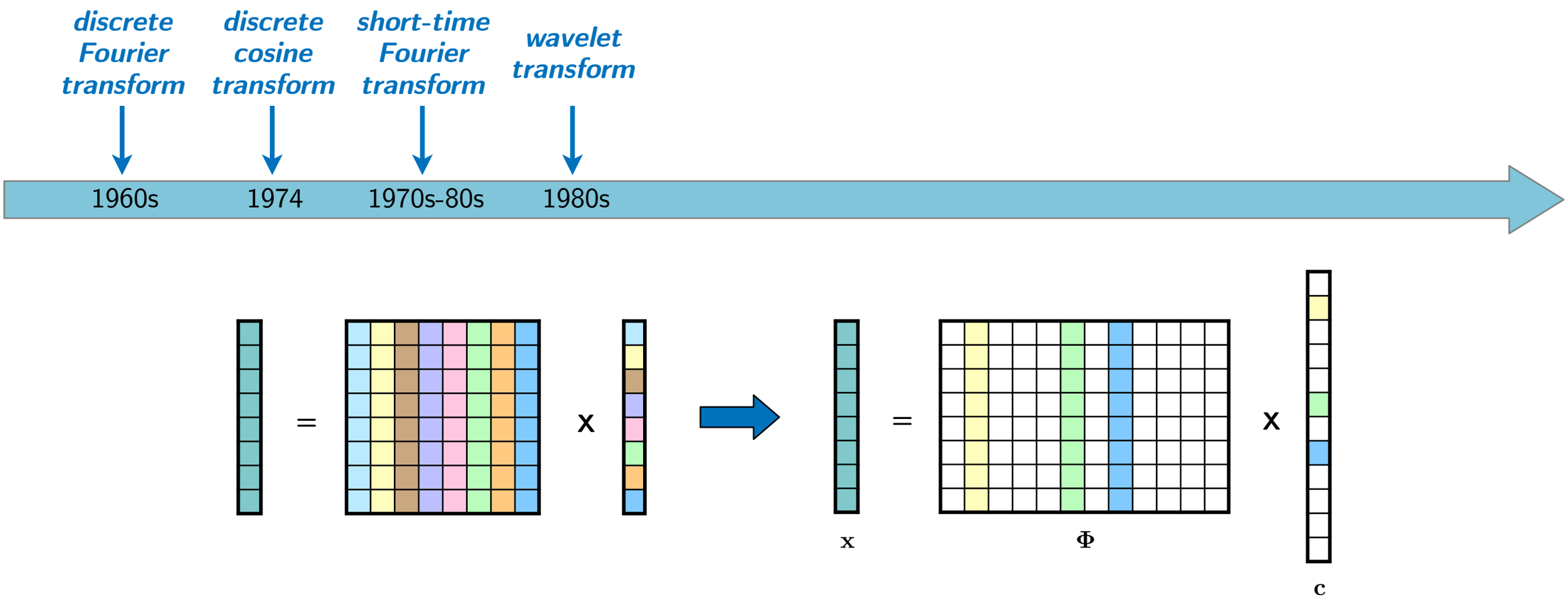


Rubinstein et al., "Dictionaries for sparse representation modeling," Proceedings of the IEEE, 2010.

Transform/analytic dictionary design

- summary
  - modelling data by a simpler class of mathematical functions
    - smooth functions (DFT, DCT)
    - piecewise-smooth functions (wavelets)
  - desired properties
    - localisation (STFT, wavelets)
    - multiresolution (wavelets)
    - adaptivity (wavelet packets)
  - fast implementation is usually available
  - limited expressiveness

A paradigm shift in dictionary design



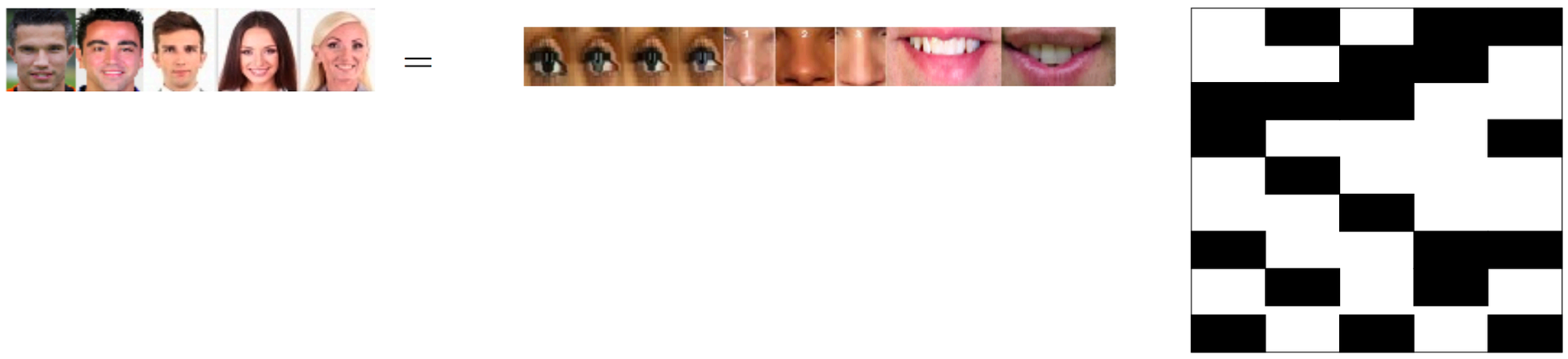
orthogonal atoms	non-orthogonal atoms
complete dictionary	overcomplete dictionary
all signals use all atoms	different signals use different atoms
dense coefficients	sparse coefficients
mathematical modelling	adaptation to data realisations

An illustrative example

- **Modelling assumption:** Each data point is a combination of only a few (sparse) fundamental elements, i.e., dictionary atoms

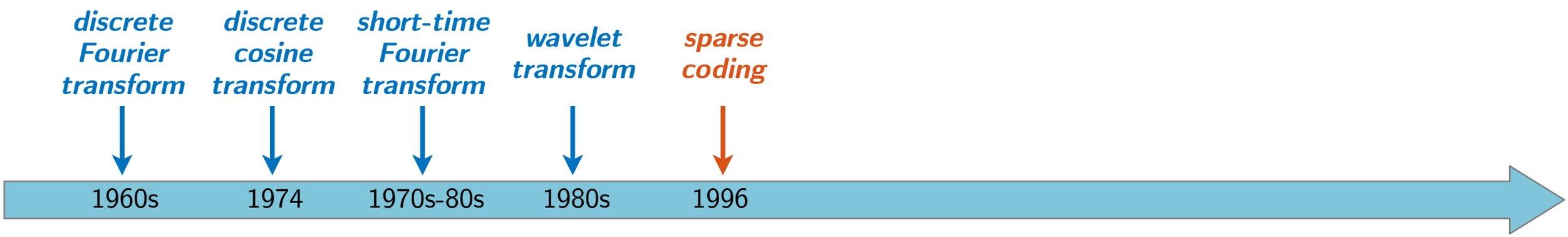
An illustrative example

- **Modelling assumption:** Each data point is a combination of only a few (sparse) fundamental elements, i.e., dictionary atoms



Signals **Dictionary** **Coefficients**

Dictionary learning via sparse coding



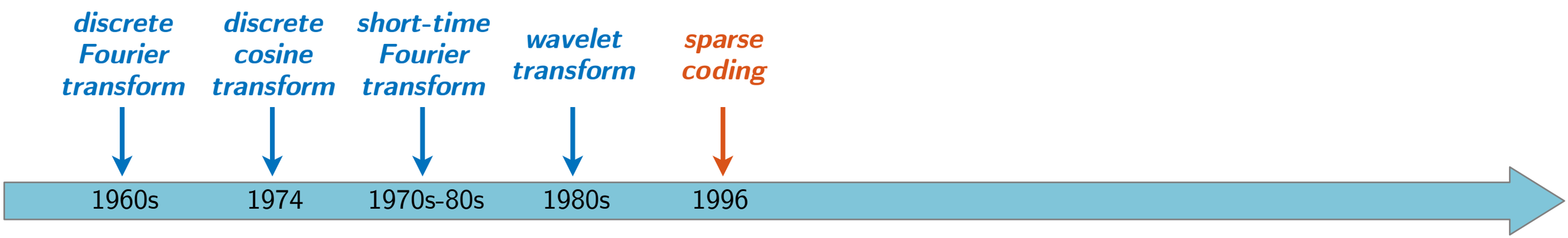
- the dictionary learning problem can be formulated as

$$\min_{\mathbf{\Phi},\mathbf{c}} ||\mathbf{x} - \mathbf{\Phi c}||_2^2 + \lambda ||\mathbf{c}||_1$$

- **sparse approximation:** given $\mathbf{\Phi}$, solve for $\mathbf{c}$ via Lasso
- **dictionary update:** given $\mathbf{c}$, update $\mathbf{\Phi}$ via gradient descent

Olshausen and Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, 1996.

Dictionary learning via sparse coding



- the dictionary learning problem can be formulated as

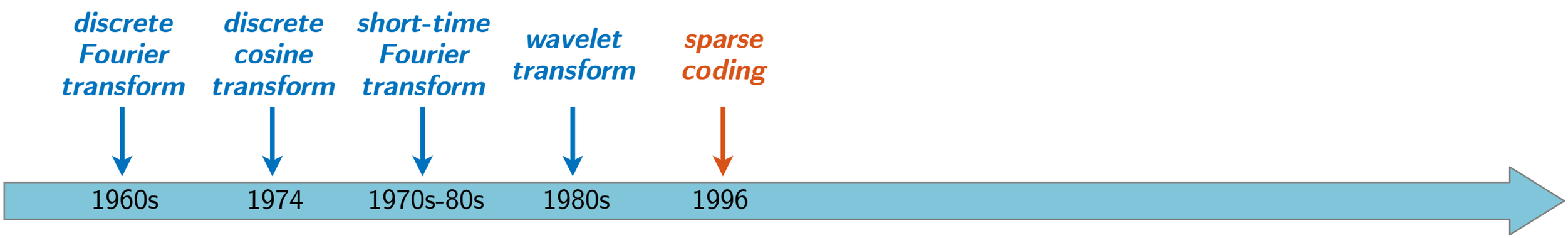
$$\min_{\mathbf{\Phi},\mathbf{c}} ||\mathbf{x} - \mathbf{\Phi c}||_2^2 + \lambda ||\mathbf{c}||_1$$

  - **sparse approximation:** given $\mathbf{\Phi}$, solve for $\mathbf{c}$ via Lasso
  - **dictionary update:** given $\mathbf{c}$, update $\mathbf{\Phi}$ via gradient descent

- works at patch level for efficiency
- does not necessarily find global optimum

Olshausen and Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, 1996.

Dictionary learning via sparse coding



- the dictionary learning problem can be formulated as

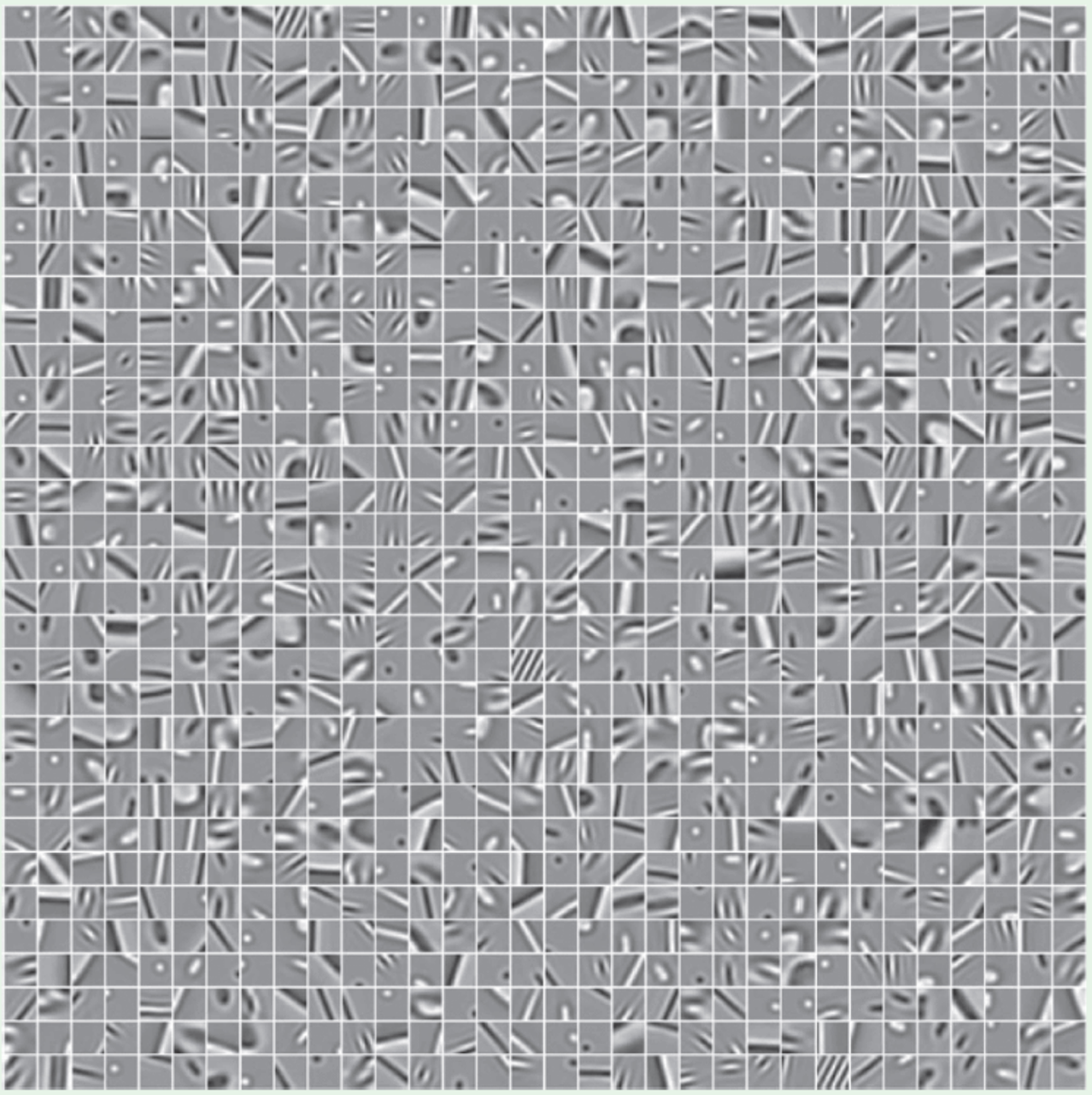
$$\min_{\mathbf{\Phi},\mathbf{c}} ||\mathbf{x} - \mathbf{\Phi c}||_2^2 + \lambda ||\mathbf{c}||_1$$

  - **sparse approximation:** given $\mathbf{\Phi}$, solve for $\mathbf{c}$ via Lasso
  - **dictionary update:** given $\mathbf{c}$, update $\mathbf{\Phi}$ via gradient descent

- works at patch level for efficiency
- does not necessarily find global optimum
- trained atoms are remarkably similar to mammalian simple-cell receptive fields

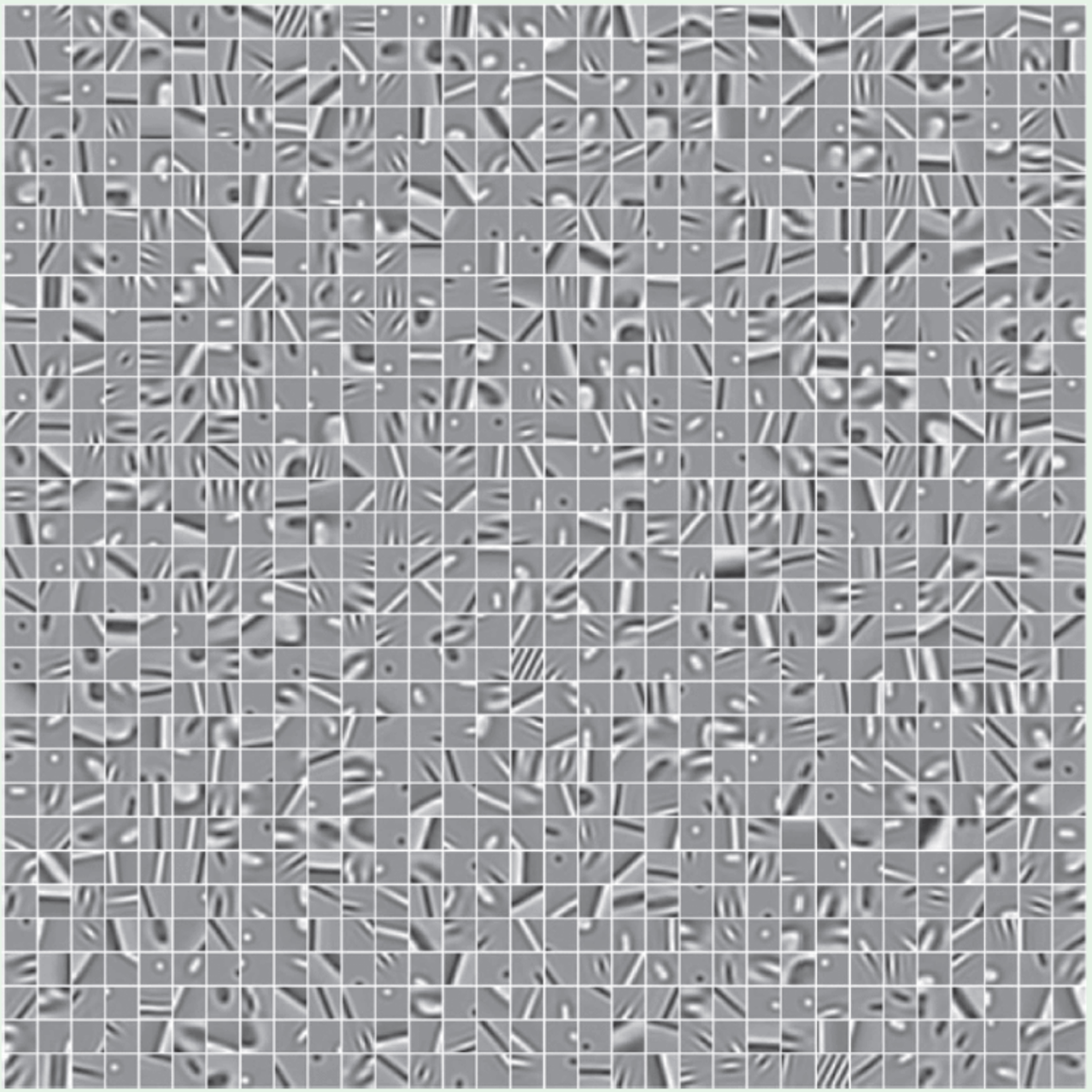
Olshausen and Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, 1996.

Dictionary learning via sparse coding

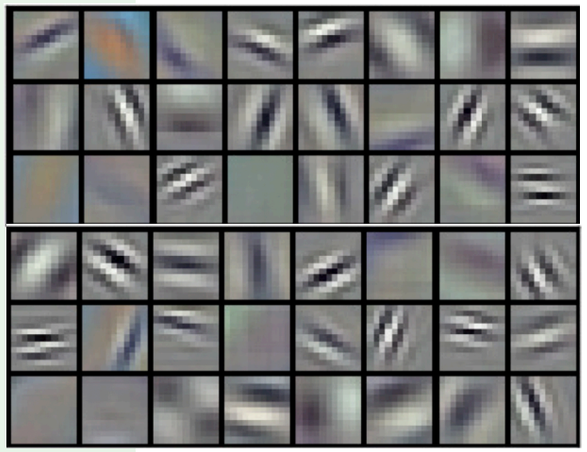


Olshausen et al., " "Learning real and complex overcomplete representations from the statistics of natural images," SPIE, 2009.

Dictionary learning via sparse coding

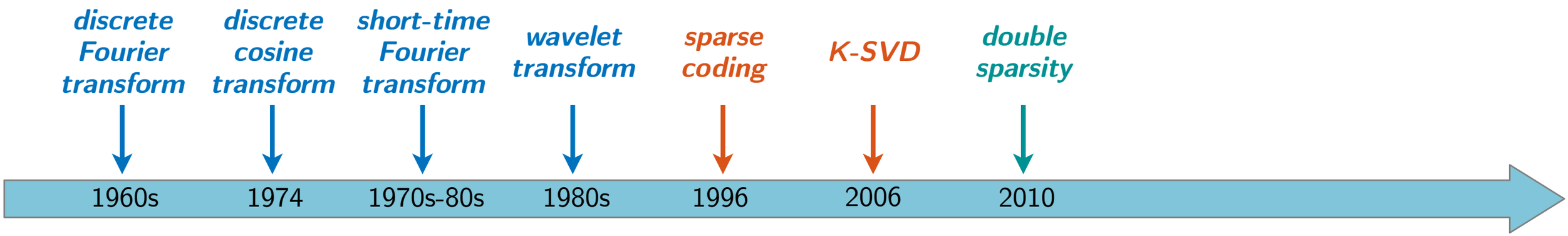


AlexNet



Olshausen et al., “ “Learning real and complex overcomplete representations from the statistics of natural images,” SPIE, 2009.

Dictionary learning

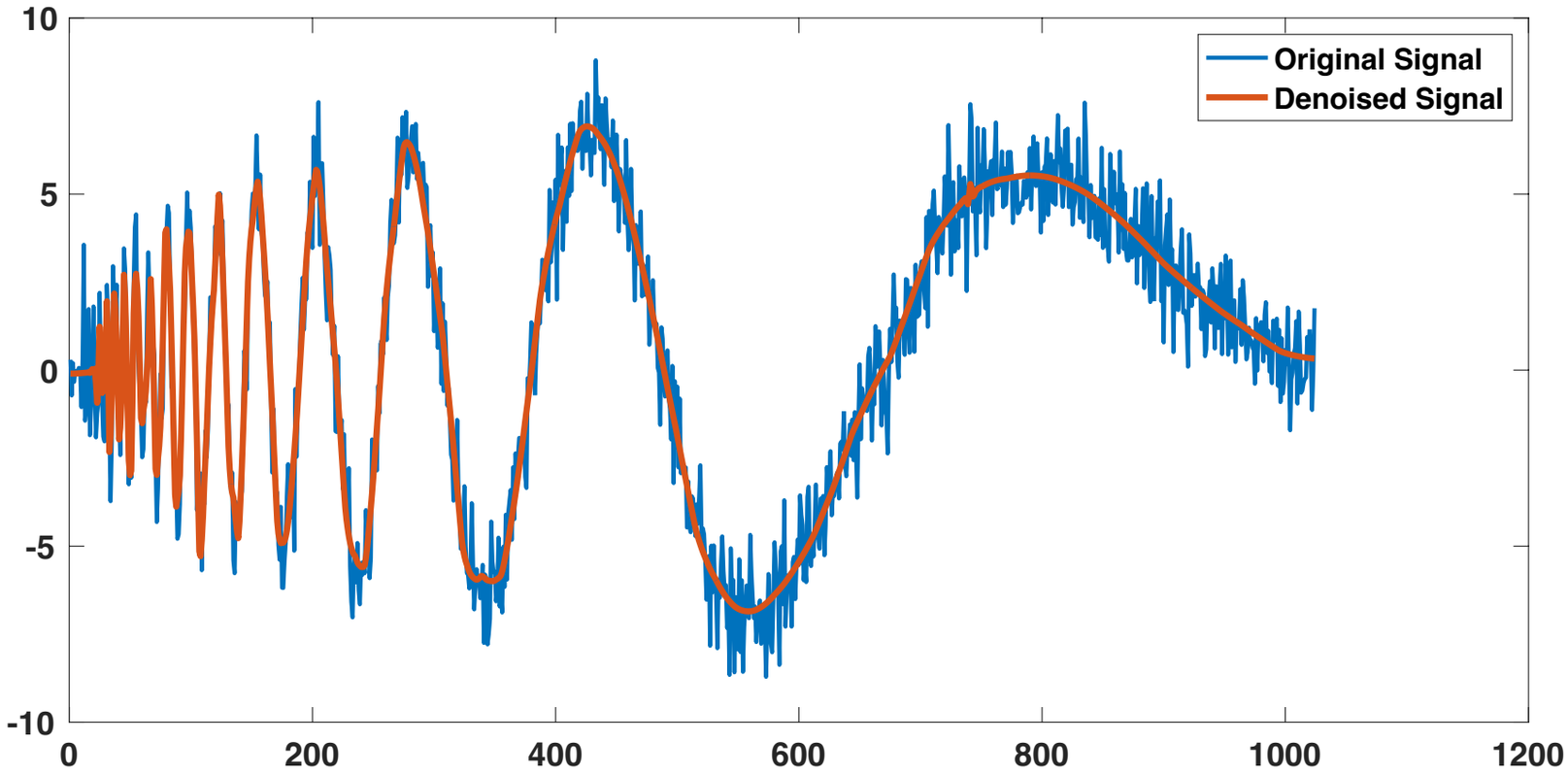


- summary
  - learning representations directly from data realisations
  - desired properties
    - overcompleteness
    - sparse representations
    - efficiency in training
  - may be combined with analytical dictionary design
    - trained dictionary with structures (e.g., **parametric dictionary learning** via double sparsity)

Lecture 1

- Introduction & Basic concepts and tools
- A historical overview of signal representation techniques
- Applications & Discussion

Application I: Signal denoising



denoising using the order 4 symlet wavelets

Application II: Image compression

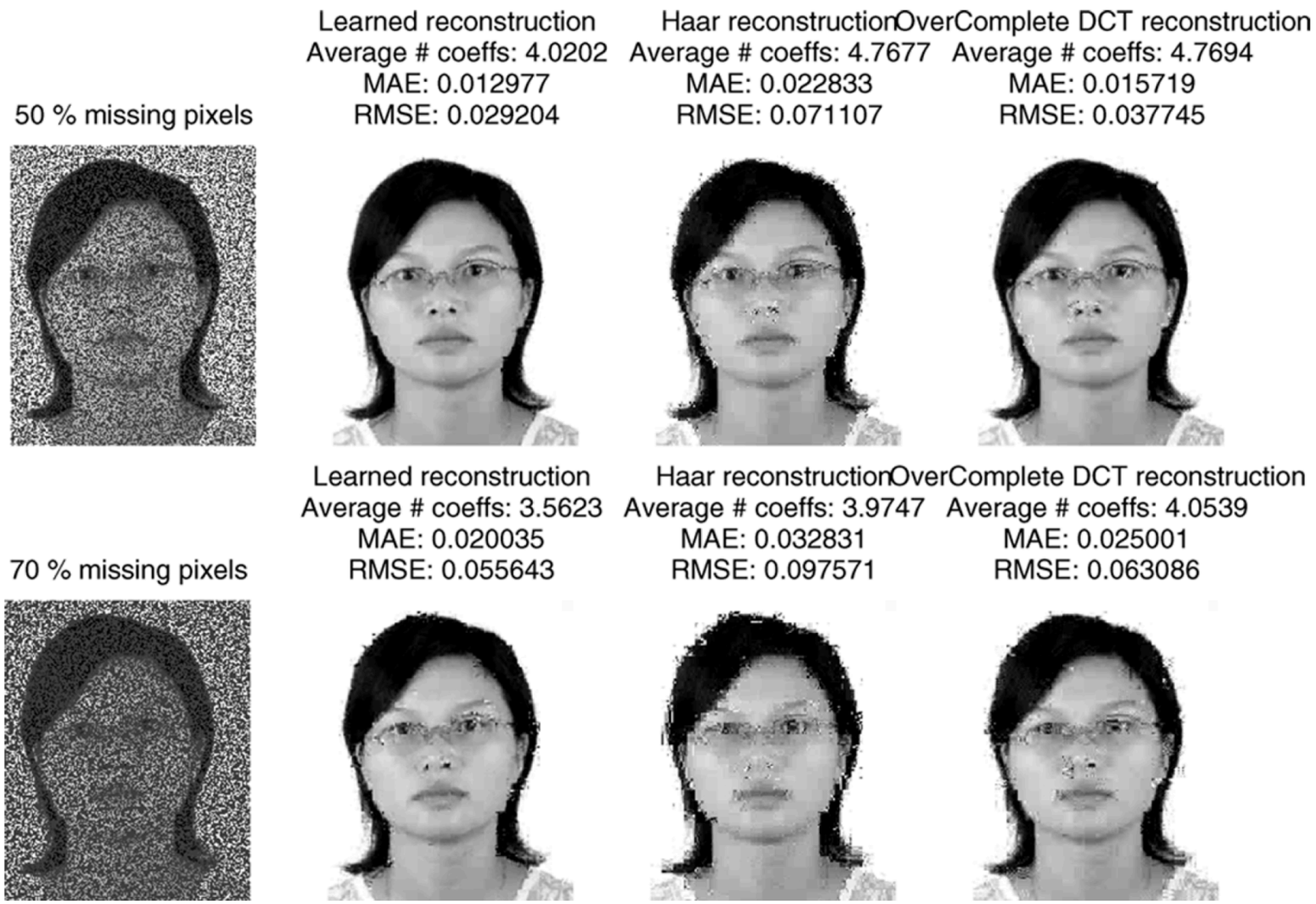
original

JPEG 2000 (10% in size)

JPEG 2000 (1% in size)

compression using the Cohen-Daubechies-Feauveau wavelets

Application III: Image reconstruction



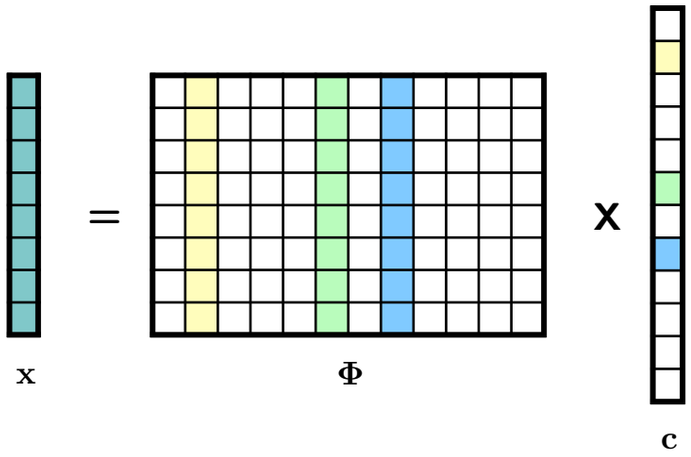
Aharon et al., "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," IEEE TSP, 2006.

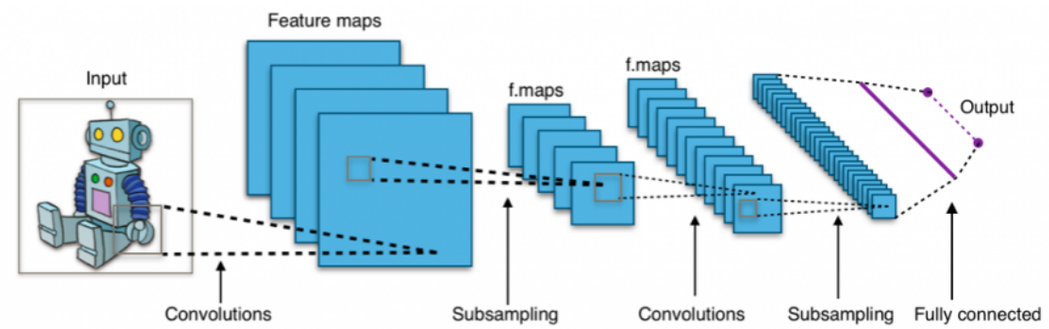
Application IV: Image restoration

Mairal et al., "Sparse representation for color image restoration," IEEE TIP, 2008.

Connection with machine learning

- Dictionary learning vs Deep learning

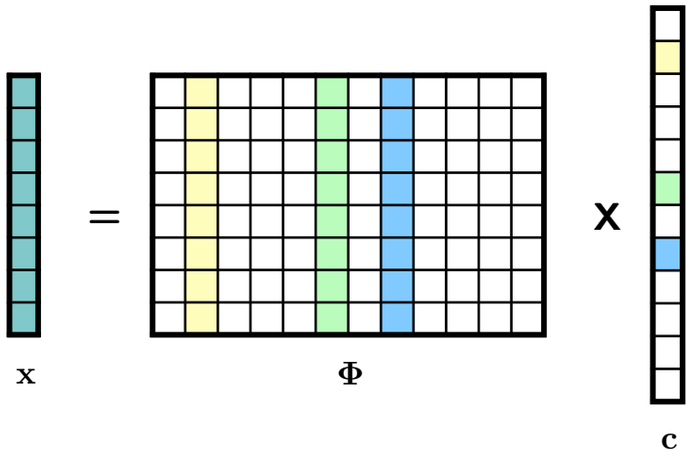


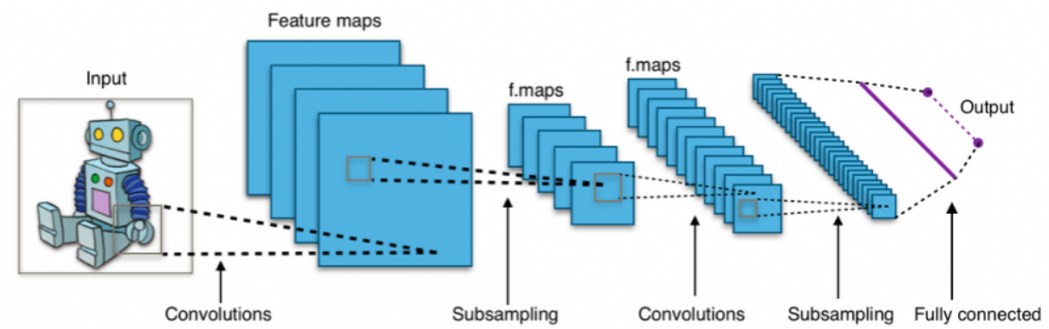


https://en.wikipedia.org/wiki/File:Typical_cnn.png

Connection with machine learning

- Dictionary learning vs Deep learning
  - both extract **feature representations** from data realisations

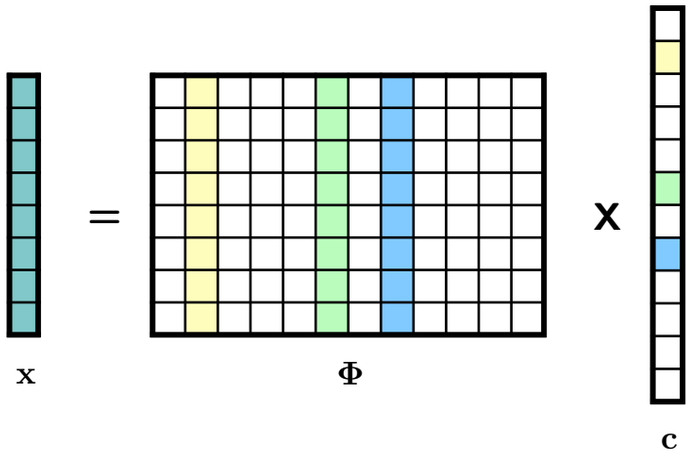


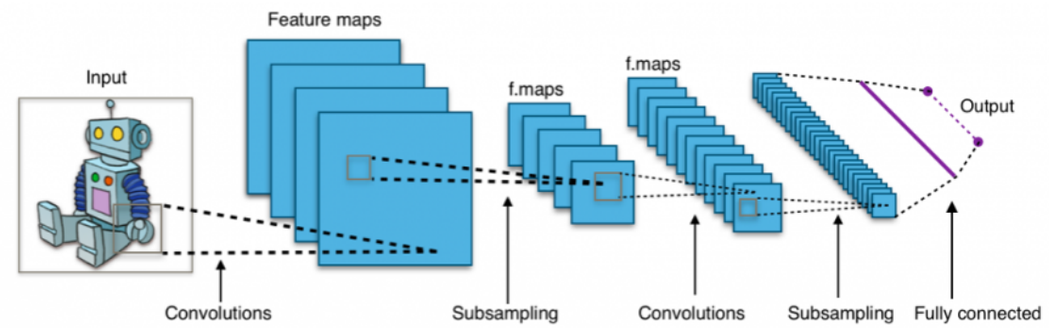


https://en.wikipedia.org/wiki/File:Typical_cnn.png

Connection with machine learning

- Dictionary learning vs Deep learning
  - both extract **feature representations** from data realisations
  - both apply **sparsifying operations** such as shrinkage or rectified linear units

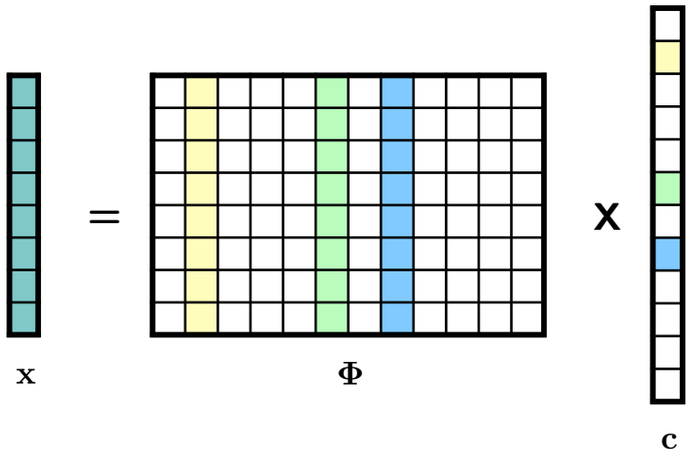


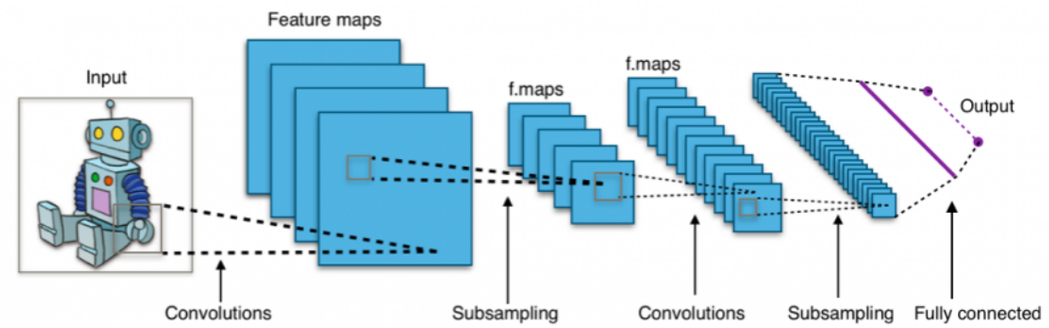


https://en.wikipedia.org/wiki/File:Typical_cnn.png

Connection with machine learning

- Dictionary learning vs Deep learning
  - both extract **feature representations** from data realisations
  - both apply **sparsifying operations** such as shrinkage or rectified linear units
  - the former often leads to **shallow** representations while the latter to **hierarchical** representations (via multi-layered convolution and pooling)



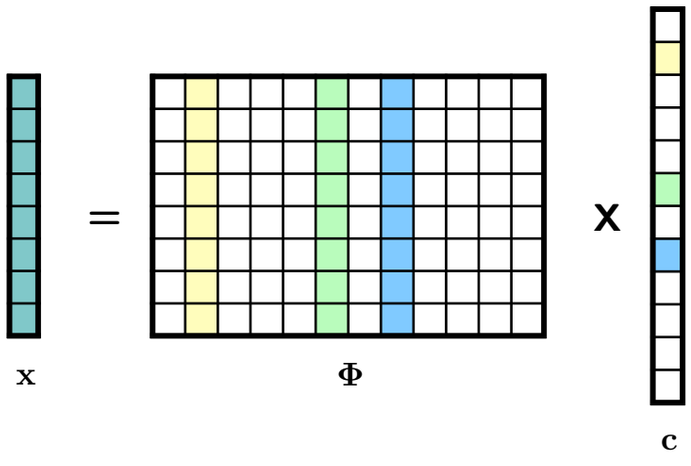


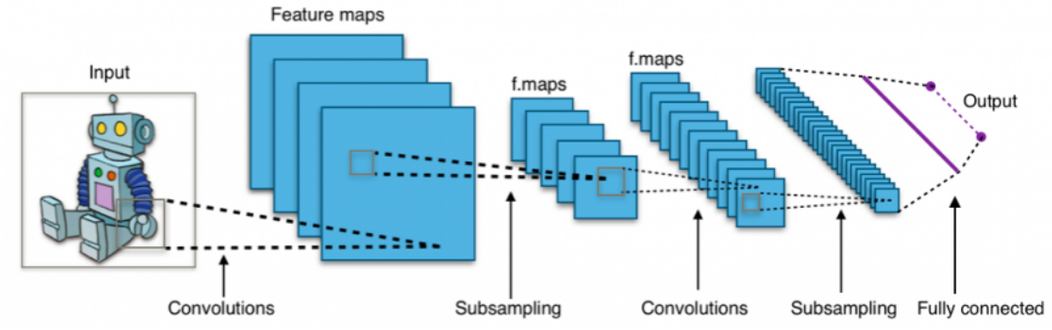
https://en.wikipedia.org/wiki/File:Typical_cnn.png

Connection with machine learning

- Dictionary learning vs Deep learning
  - both extract **feature representations** from data realisations
  - both apply **sparsifying operations** such as shrinkage or rectified linear units
  - the former often leads to **shallow** representations while the latter to **hierarchical** representations (via multi-layered convolution and pooling)
  - the former is mainly for **reconstruction/approximation** (similar to autoencoders)

while the latter is widely used for **classification**

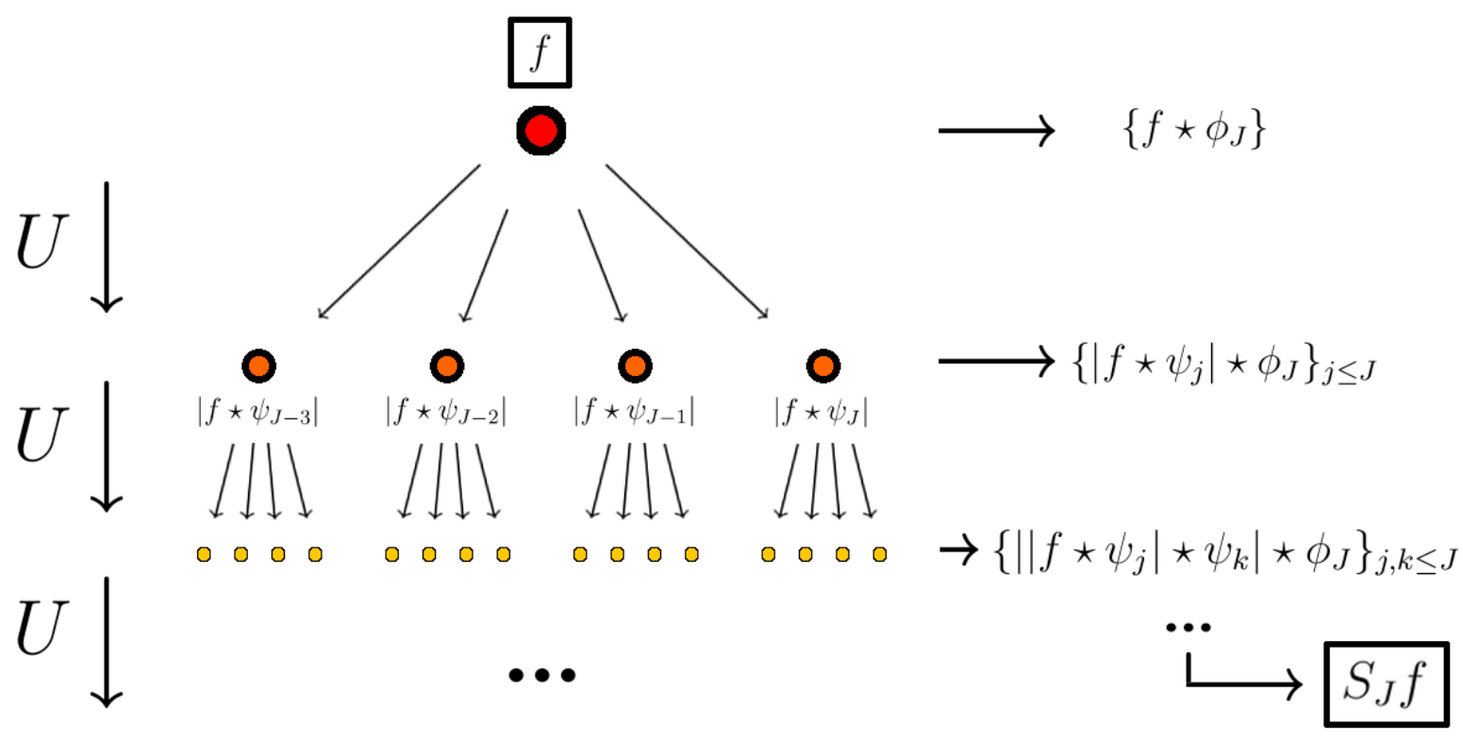




https://en.wikipedia.org/wiki/File:Typical_cnn.png

Dictionary-inspired deep architectures

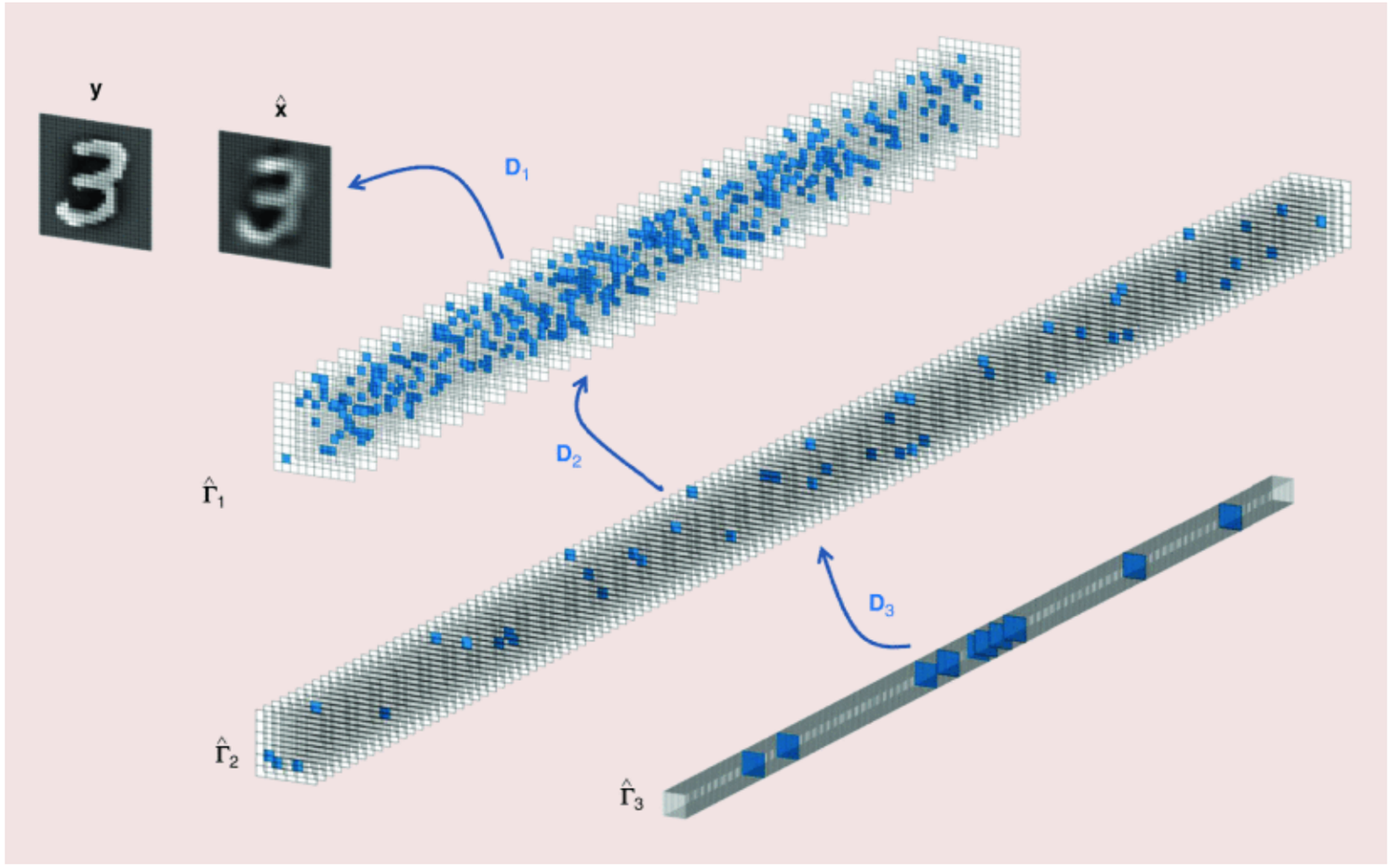
- Scattering transform



Bruna and Mallat, "Invariant scattering convolution networks," IEEE TPAMI, 2013.

Dictionary-inspired deep architectures

- Multi-layer convolutional sparse coding



Papyan et al., "Theoretical foundations of deep learning via sparse representations," IEEE SPM, 2018.

References

INVITED PAPER

Dictionaries for Sparse Representation Modeling

Digital sampling can display signals, and it should be possible to expose a large part of the desired signal information with only a limited signal sample.

By Ron Rubinstein, *Student Member IEEE*, Alfred M. Bruckstein, *Member IEEE*, and Michael Elad, *Senior Member IEEE*

ABSTRACT | Sparse and redundant representation modeling of data assumes an ability to describe signals as linear combinations of a few atoms from a pre-specified dictionary. As such, the choice of the dictionary that sparsifies the signals is crucial for the success of this model. In general, the choice of a proper dictionary can be done using one of two ways: i) building a sparsifying dictionary based on a mathematical model of the data, or ii) learning a dictionary to perform best on a training set. In this paper we describe the evolution of these two paradigms. As manifestations of the first approach, we cover topics such as wavelets, wavelet packets, contourlets, and curvelets, all aiming to exploit 1-D and 2-D mathematical models for constructing effective dictionaries for signals and images. Dictionary learning takes a different route, attaching the dictionary to a set of examples it is supposed to serve. From the seminal work of Field and Olshausen, through the MOD, the K-SVD, the Generalized PCA and others, this paper surveys the various options such training has to offer, up to the most recent contributions and structures.

KEYWORDS | Dictionary learning; harmonic analysis; signal approximation; signal representation; sparse coding; sparse representation

I. INTRODUCTION

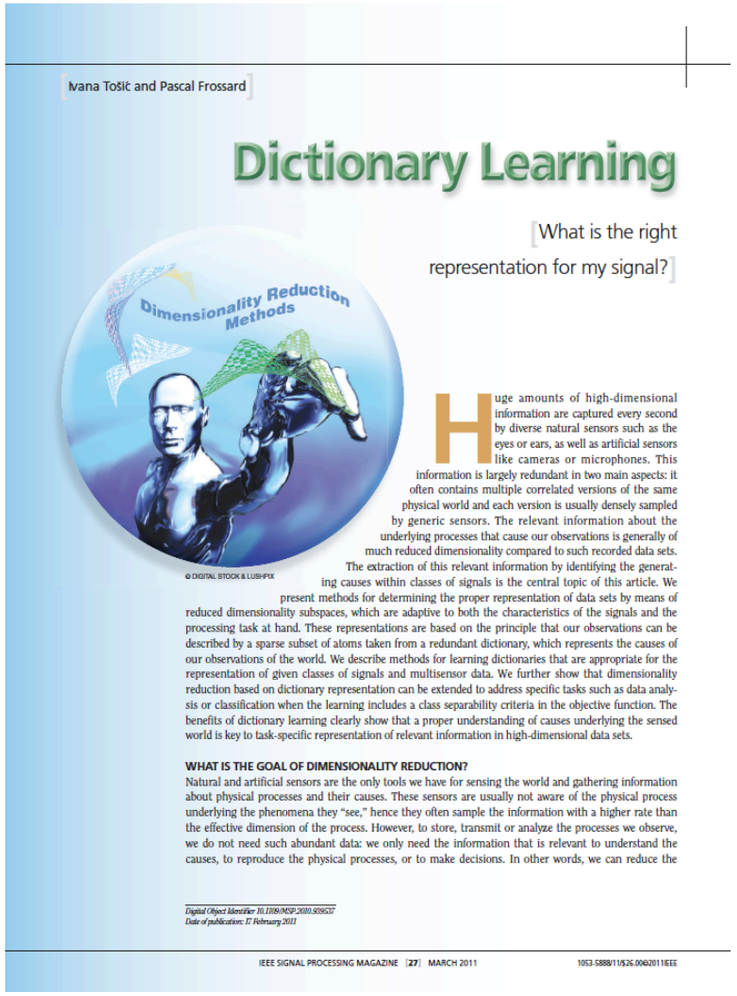
The process of digitally sampling a natural signal leads to its representation as the sum of Delta functions in space or time. This representation, while convenient for the purposes of display or playback, is mostly inefficient for analysis tasks. Signal processing techniques commonly require more meaningful representations which capture the useful characteristics of the signal—for recognition, the representation should highlight salient features; for denoising, the representation should efficiently separate signal and noise; and for compression, the representation should capture a large part of the signal with only a few coefficients. Interestingly, in many cases these seemingly different goals align, sharing a core desire for *simplification*.

Representing a signal involves the choice of a *dictionary*, which is the set of elementary signals—or *atoms*—used to decompose the signal. When the dictionary forms a basis, every signal is uniquely represented as the linear combination of the dictionary atoms. In the simplest case the dictionary is orthogonal, and the representation coefficients can be computed as inner products of the signal and the atoms; in the non-orthogonal case, the coefficients are the inner products of the signal and the dictionary inverse, also referred to as the bi-orthogonal dictionary.

For years, orthogonal and bi-orthogonal dictionaries were dominant due to their mathematical simplicity. However, the weakness of these dictionaries—namely their limited expressiveness—eventually outweighed their simplicity. This led to the development of newer *overcomplete* dictionaries, having more atoms than the dimensions of the signal, which promised to represent a wider range of signal phenomena.

The move to overcomplete dictionaries was done cautiously, in an attempt to minimize the loss of favorable properties offered by orthogonal transforms. Many dictionaries formed *tight frames*, which ensured that the representation of the signal as a linear combination of the atoms could still be identified with the inner products of the signal and the dictionary. Another approach, manifested by

Manuscript received April 5, 2009; accepted November 21, 2009. Date of publication April 22, 2010; date of current version May 19, 2010. This research was partly supported by the European Community's FP7-FET program, SMALL project, under grant agreement 225913, and by the ISF grant 599/08.
The authors are with the Department of Computer Science, The Technion-Israel Institute of Technology, Haifa 32000, Israel (e-mail: ronrubin@cs.technion.ac.il; freddy@cs.technion.ac.il; elad@cs.technion.ac.il; http://www.cs.technion.ac.il).
Digital Object Identifier 10.1109/JPROC.2010.2040551



Huge amounts of high-dimensional information are captured every second by diverse natural sensors such as the eyes or ears, as well as artificial sensors like cameras or microphones. This information is largely redundant in two main aspects: it often contains multiple correlated versions of the same physical world and each version is usually densely sampled by generic sensors. The relevant information about the underlying processes that cause our observations is generally of much reduced dimensionality compared to such recorded data sets. The extraction of this relevant information by identifying the generating causes within classes of signals is the central topic of this article. We present methods for determining the proper representation of data sets by means of reduced dimensionality subspaces, which are adaptive to both the characteristics of the signals and the processing task at hand. These representations are based on the principle that our observations can be described by a sparse subset of atoms taken from a redundant dictionary, which represents the causes of our observations of the world. We describe methods for learning dictionaries that are appropriate for the representation of given classes of signals and multisensor data. We further show that dimensionality reduction based on dictionary representation can be extended to address specific tasks such as data analysis or classification when the learning includes a class separability criteria in the objective function. The benefits of dictionary learning clearly show that a proper understanding of causes underlying the sensed world is key to task-specific representation of relevant information in high-dimensional data sets.

WHAT IS THE GOAL OF DIMENSIONALITY REDUCTION?

Natural and artificial sensors are the only tools we have for sensing the world and gathering information about physical processes and their causes. These sensors are usually not aware of the physical process underlying the phenomena they "see," hence they often sample the information with a higher rate than the effective dimension of the process. However, to store, transmit or analyze the processes we observe, we do not need such abundant data: we only need the information that is relevant to understand the causes, to reproduce the physical processes, or to make decisions. In other words, we can reduce the

Digital Object Identifier 10.1109/MSP.2010.939537
Date of publication: 17 February 2011

1053-5888/11/$26.00©2011IEEE

Representation Learning: A Review and New Perspectives

Yoshua Bengio, Aaron Courville, and Pascal Vincent

Abstract—The success of machine learning algorithms generally depends on data representation, and we hypothesize that this is because different representations can entangle and hide more or less the different explanatory factors of variation behind the data. Although specific domain knowledge can be used to help design representations, learning with generic priors can also be used, and the quest for AI is motivating the design of more powerful representation-learning algorithms implementing such priors. This paper reviews recent work in the area of unsupervised feature learning and deep learning, covering advances in probabilistic models, autoencoders, manifold learning, and deep networks. This motivates longer term unanswered questions about the appropriate objectives for learning good representations, for computing representations (i.e., inference), and the geometrical connections between representation learning, density estimation, and manifold learning.

Index Terms—Deep learning, representation learning, feature learning, unsupervised learning, Boltzmann machine, autoencoder, neural nets

1 Introduction

The performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied. For that reason, much of the actual effort in deploying machine learning algorithms goes into the design of preprocessing pipelines and data transformations that result in a representation of the data that can support effective machine learning. Such feature engineering is important but labor intensive and highlights the weakness of current learning algorithms: Their inability to extract and organize the discriminative information from the data. Feature engineering is a way to take advantage of human ingenuity and prior knowledge to compensate for that weakness. To expand the scope and ease of applicability of machine learning, it would be highly desirable to make learning algorithms less dependent on feature engineering so that novel applications could be constructed faster, and more importantly, to make progress toward artificial intelligence (AI). An AI must fundamentally *understand the world around us*, and we argue that this can only be achieved if it can learn to identify and disentangle the underlying explanatory factors hidden in the observed milieu of low-level sensory data.

This paper is about *representation learning*, i.e., learning representations of the data that make it easier to extract useful information when building classifiers or other predictors. In the case of probabilistic models, a good representation is often one that captures the posterior distribution of the underlying explanatory factors for the observed input. A good representation is also one that is useful as input to a supervised predictor. Among the various ways of learning representations, this paper focuses on deep learning methods: those that are formed by the composition of multiple nonlinear transformations with the goal of yielding more abstract—and ultimately more useful—representations. Here, we survey this rapidly developing area with special emphasis on recent progress. We consider some of the fundamental questions that have been driving research in this area. Specifically, what makes one representation better than another? Given an example, how should we compute its representation, i.e., perform feature extraction? Also, what are appropriate objectives for learning good representations?

• *The authors are with the Department of Computer Science and Operations Research, Université de Montréal, PO Box 6128, Succ. Centre-Ville, Montreal, Quebec H3C 3J7, Canada.*

Manuscript received 9 Apr. 2012; revised 17 Oct. 2012; accepted 24 Feb. 2013; published online 28 Feb. 2013.
Recommended for acceptance by S. Bengio, L. Deng, H. Larochelle, H. Lee, and R. Salakhutdinov.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2012-04-0260.
Digital Object Identifier no. 10.1109/TPAMI.2013.50.

 Published by the IEEE Computer Society

2 Why Should We Care about Learning Representations?

Representation learning has become a field in itself in the machine learning community, with regular workshops at the leading conferences such as NIPS and ICML, and a new conference dedicated to it, ICLR,[1] sometimes under the header of *Deep Learning* or *Feature Learning*. Although depth is an important part of the story, many other priors are interesting and can be conveniently captured when the problem is cast as one of learning a representation, as discussed in the next section. The rapid increase in scientific activity on representation learning has been accompanied and nourished by a remarkable string of empirical successes both in academia and in industry. Below, we briefly highlight some of these high points.

2.1 Speech Recognition and Signal Processing

Speech was one of the early applications of neural networks, in particular convolutional (or time-delay) neural

1. International Conference on Learning Representations.