Master SIF - REP (6-8/20)
Image Transform and dictionaries

Thomas Maugey
thomas.maugey@inria.fr

UNIVERSITÉ DE RENNES 1

informatics mathematics Inria

Fall 2020

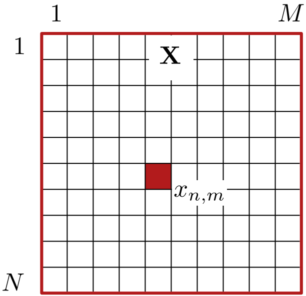
Table of Contents

Table of Contents

Notations

Let $\mathbf{X} = \{x_{n,m}\}$ be a matrix of size $N \times M$ $\rightarrow$ **the image**

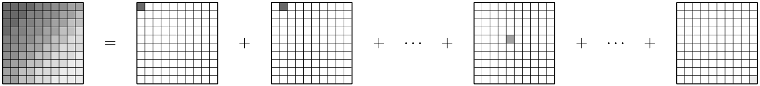
We assume that $x_{n,m} \in \mathbb{R}$ $\rightarrow$ **the pixels**



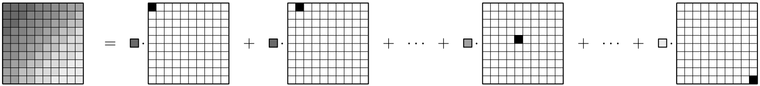
The image is seen as a vector of NM dimensions in which the dimensions are arranged with a specific geometry (*i.e.*, a 2D grid of size $N \times M$).

Pixel basis

The image $\mathbf{X}$ can be seen as the sum of each pixel



And as a linear combination of the **Pixel-basis**



In this basis, the position of the pixels is not taken into account, while, it is known that neighboring pixels are generally correlated.

Transform's objectives

A transform is simply **another basis** $\{\mathbf{U}_u\}_{u \in [\![0, NM-1]\!]}$:

$$\mathbf{X} = \hat{x}_0 \mathbf{U}_0 + \ldots + \hat{x}_u \mathbf{U}_u + \ldots + \hat{x}_{NM-1} \mathbf{U}_{NM-1}$$

in which

- the "2D grid" shape of the image is taken into account
- the representation is sparse (*i.e.*, the number of non-zero $\hat{x}_u$ is small)
- the representation is more suited for processing (*e.g.*, analysis,filtering, denoising)

Table of Contents

1D Fourier transform

Let x be a 1D function with an infinite support ($t \in \mathbb{R}$).

The 1D Fourier transform is:

$$\forall \omega \in \mathbb{R}, \quad \hat{x}(\omega) = \int_{t\in\mathbb{R}} x(t) e^{-2i\pi\omega t} dt$$

The inverse transform is:

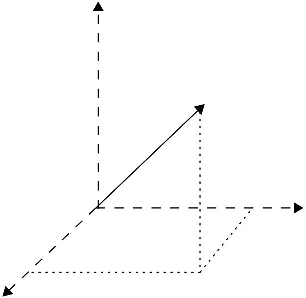
$$\forall t \in \mathbb{R}, \quad x(t) = \int_{w\in\mathbb{R}} \hat{x}(w) e^{2i\pi\omega t} d\omega$$

1D Fourier transform

The transform is the scalar product of the function and a oscillating basis

$$\forall \omega \in \mathbb{R}, \quad \hat{x}(\omega) =< x, u_\omega >$$

where $\forall t, \quad u_\omega(t) = e^{2i\pi\omega t}$



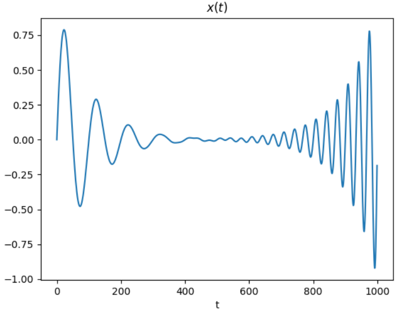
Transform's outputs

From the output of the Fourier transform, we define:

- The frequency spectrum: $Real\left(\hat{x}(\omega)\right) + iImg\left(\left(\hat{x}(\omega)\right)\right)$
  The Fourier transform of a function produces a frequency spectrum which contains all of the information about the original signal, but in a different form.
- Magnitude spectrum: $\left|Real\left(\hat{x}(\omega)\right) + iImg\left(\hat{x}(\omega)\right)\right|$
- Phase spectrum: $Arctg\left(\frac{Img(\hat{x}(\omega))}{Real(\hat{x}(\omega))}\right)$
- Power spectrum: $Real\left(\hat{x}(\omega)\right)^2 + Img\left(\hat{x}(\omega)\right)^2$

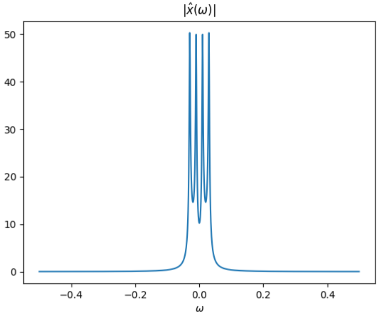
Example


```
import numpy as np
import matplotlib.pyplot as plt

fo=0.01
fo2=0.03
a = 0.01
N=1000
t=np.arange(N)
x=np.sin(2*np.pi*fo*t)*np.exp(-a*t)+np.sin(2*np.pi*fo2*t)*np.exp(-a*(N-t))
plt.plot(t,x)
```



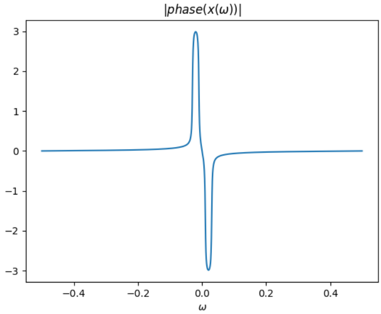
Example

```
import numpy.fft as fft

om = (t-N/2)/N
xom = fft.fft(x)
plt.plot(om,np.abs(fft.fftshift(xom)))
```



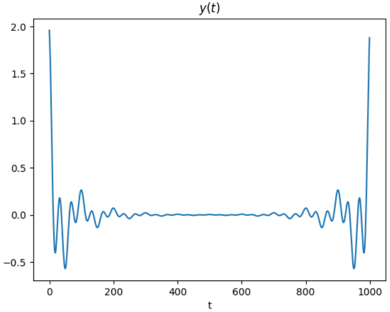
Example

```
plt.plot(om,np.angle(fft.fftshift(xom)))
```



Example

```
y = fft.ifft(np.abs(xom))
plt.plot(t,y)
```



Properties

- Linearity:

$$
\begin{aligned}
ax(t) &\xrightarrow{\mathcal{F}} a\hat{x}(\omega) \\
ax_1(t) + bx_2(t) &\xrightarrow{\mathcal{F}} a\hat{x}_1(\omega) + b\hat{x}_2(\omega)
\end{aligned}
$$

- Complex conjugate: $x^*(t) \xrightarrow{\mathcal{F}} \hat{x}^*(-\omega)$

$$
\begin{aligned}
x^*(t) &= \left(\int_{-\infty}^{+\infty} \hat{x}(\omega)e^{2i\pi\omega t}d\omega\right)^* \\
x^*(t) &= \int_{-\infty}^{+\infty} \hat{x}^*(\omega)e^{-2i\pi\omega t}d\omega
\end{aligned}
$$

In the same way, $x^*(-t) \xrightarrow{\mathcal{F}} \hat{x}^*(\omega)$ and $x(-t) \xrightarrow{\mathcal{F}} \hat{x}(-\omega)$.

Fourier transformation

- Hermitian symmetry: if $x(t) \in \mathbb{R}$, we deduce $\hat{x}(-\omega) = \hat{x}^*(\omega)$

$$\begin{aligned}
\hat{x}(\omega) &= \int_{-\infty}^{+\infty} x(t)e^{-2i\pi\omega t}dt \\
\hat{x}^*(\omega) &= \left(\int_{-\infty}^{+\infty} x(t)e^{-2i\pi\omega t}dt\right)^* \\
\hat{x}^*(\omega) &= \int_{-\infty}^{+\infty} x^*(t)e^{2i\pi\omega t}dt
\end{aligned}$$

Given that $x(t) \in \mathbb{R}$, we have $x^*(t) = x(t)$ that implies $\hat{x}^*(\omega) = \hat{x}(-\omega)$.

Fourier transformation

The norm is given by:

$$\|x\|^2 = \langle x, x \rangle = \int_{-\infty}^{+\infty} |x(t)|^2 \, dt$$

$$\int_{-\infty}^{+\infty} |x(t)|^2 \, dt \quad =$$

Fourier transformation

The norm is given by:

$$\|x\|^2 = \langle x, x\rangle = \int_{-\infty}^{+\infty} |x(t)|^2 \, dt$$

$$\begin{aligned}
\int_{-\infty}^{+\infty} |x(t)|^2 \, dt &= \int_{-\infty}^{+\infty} x(t)x^*(t)dt \\
&= \int_{-\infty}^{+\infty} x(t) \left[\int_{-\infty}^{+\infty} \hat{x}^*(\omega) e^{-2i\pi\omega t} d\omega \right] dt \\
&= \int_{-\infty}^{+\infty} \hat{x}^*(\omega) \left[\int_{-\infty}^{+\infty} x(t) e^{-2i\pi\omega t} dt \right] d\omega \\
&= \int_{-\infty}^{+\infty} \hat{x}^*(\omega)\hat{x}(\omega) d\omega \\
\int_{-\infty}^{+\infty} |x(t)|^2 \, dt &= \int_{-\infty}^{+\infty} |\hat{x}(\omega)|^2 \, d\omega
\end{aligned}$$

Fourier transformation

$$\int_{-\infty}^{+\infty} |x(t)|^2 \, dt = \int_{-\infty}^{+\infty} |\hat{x}(\omega)|^2 \, d\omega$$

This formula (Parseval's theorem or energy conservation) proves that the energy is conserved by the Fourier transform.

Loosely, the sum (or integral) of the square of a function is equal to the sum (or integral) of the square of its transform.

Fourier transformation

- Translation in time/space domain: $x(t-t_0) \xrightarrow{\mathcal{F}} e^{-2i\pi\omega t_0}\hat{x}(\omega)$

$$x(t-t_0) \xrightarrow{\mathcal{F}} \int_{-\infty}^{+\infty} x(t-t_0)e^{-2i\pi\omega t}dt$$

$$k = t - t_0$$

$$x(t-t_0) \xrightarrow{\mathcal{F}} \int_{-\infty}^{+\infty} x(k)e^{-2i\pi\omega(k+t_0)}dk$$

$$x(t-t_0) \xrightarrow{\mathcal{F}} e^{-2i\pi\omega t_0}\left[\int_{-\infty}^{+\infty} x(k)e^{-2i\pi\omega k}dk\right]$$

Displacement in time or space induces a phase shift proportional to frequency and to the amount of displacement.

- Frequency shift: $x(t)e^{\pm 2i\pi\omega_0 t} \xrightarrow{\mathcal{F}} \hat{x}(\omega \pm \omega_0)$

Displacement in frequency multiplies the time/space function by a unit phasor which has angle proportional to time/space and to the amount of displacement. Amplitude modulation.

1D discrete Fourier transform

The transform is said "discrete" when:

- the function x is finite: $\mathbf{x} = (x(n))_{n\in[\![0,N-1]\!]}$
- the frequencies ω are finite $(\frac{u}{N})_{u\in[\![0,N-1]\!]}$

The 1D DFT is defined by

$$\forall u \in [\![0, N-1]\!], \quad \hat{x}(u) = \sum_{n=0}^{N-1} x(n) e^{-2i\pi \frac{u}{N} n}$$

The inverse transform is

$$\forall n \in [\![0, N-1]\!], \quad x(n) = \frac{1}{N} \sum_{u=0}^{N-1} \hat{x}(u) e^{2i\pi \frac{u}{N} n}$$

2D discrete Fourier transform

Let $\mathbf{X}$ be a 2D matrix of size $N \times M$

The 2D discrete Fourier transform is $\forall (u,v) \in [\![0, N-1]\!] \times [\![0, M-1]\!]$,:

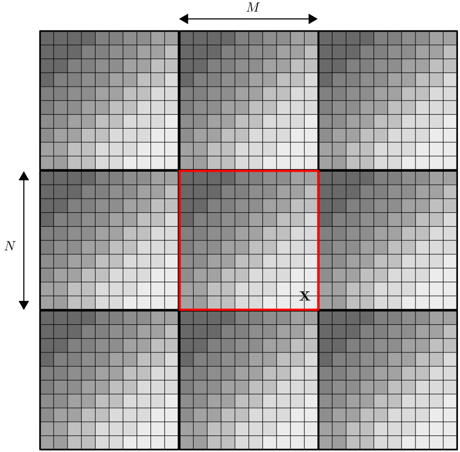
$$\hat{x}(u,v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n,m) e^{-2i\pi\left(\frac{u}{N}n + \frac{v}{M}m\right)}$$

The inverse transform is $\forall (n,m) \in [\![0, N-1]\!] \times [\![0, M-1]\!]$

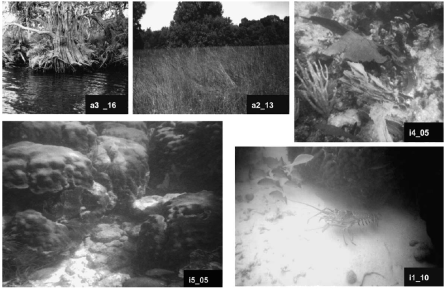
$$x(n,m) = \frac{1}{MN} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \hat{x}(u,v) e^{2i\pi\left(\frac{u}{N}n + \frac{v}{M}m\right)}$$

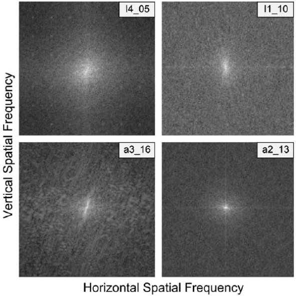
Implicit assumption

When the discrete Fourier transform is defined, the signal is supposed to be periodic, with a period of N vertically and M horizontally.
Which can cause some problems:



Examples



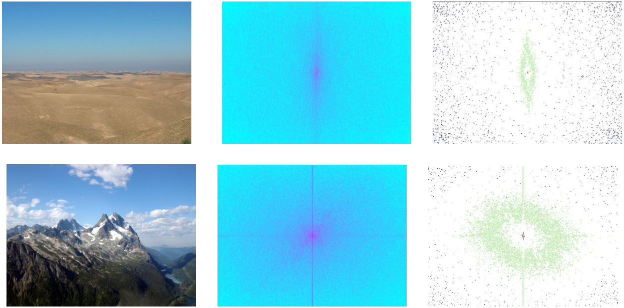


The spectra may be more or less anisotropic.

[From R.M. Balboa, N.M. Grzywacz, Power spectra and distribution of contrasts of natural images from different habitats, Vision Research, 43, pp. 2527-2537, 2003.]

Examples

From left hand-side to right: original picture, spectrum, contour lines.



Remarks:

- Fourier modulus of real images is even;
- Fourier phase of real images is odd;
- For display purpose, a logarithm is applied on the spectrum.

Examples

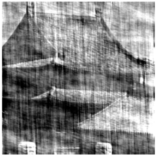
- Geometric contours are mostly contained in the phase;
- Textures are mostly contained in the modulus.

Exchanging the modulus and the phase of two images:

Image 1

Image 2

Modulus of 1 & phase of 2



Modulus of 2 & phase of 1

[A. V. Oppenheim and J. S. Lim, The importance of phase in signals, Proceedings of the IEEE, 1981 and adapted from B. Galerne's lecture.]

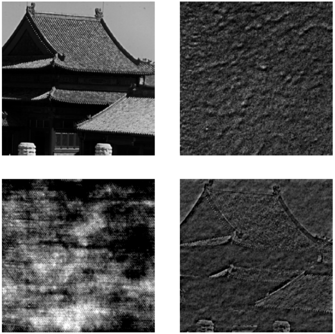
Examples

- Geometric contours are mostly contained in the phase;
- Textures are mostly contained in the modulus.

Exchanging the modulus and the phase of two images:

Image 1

Image 2



Modulus of 1 & phase of 2

Modulus of 2 & phase of 1

[A. V. Oppenheim and J. S. Lim, The importance of phase in signals, Proceedings of the IEEE, 1981 and adapted from B. Galerne's lecture.]

Table of Contents

Discrete Cosine Transform

The DFT transforms a complex signal into its complex spectrum. However, if the signal is real as in most of the applications, half of the data is redundant:

- In time domain: the imaginary part of the signal is all zero;
- In frequency domain: the real part of the spectrum is even symmetric and imaginary part odd.

How to avoid this high redundancy? We would need a real unitary transform that transforms a sequence of real data points into its real spectrum.

Discrete Cosine Transform

Let $x(n)$ a real signal defined over N sample ($n = \{0, \ldots, N-1\}$).

- Construction of a new sequence of $2N$ samples:

$$x_p(n) = \begin{cases} x(n) & 0 \leq n < N \\ x(-n-1) & -N \leq n \leq -1 \end{cases}$$

  $x_p(n)$ is now even symmetric with respect to the point $n = -\frac{1}{2}$.

- we define $n' = n + \frac{1}{2}$, to get an even symmetry with respect to $n' = 0$. The DFT of this 2N-point even symmetric sequence:

$$\hat{x}(u) =$$

Discrete Cosine Transform

Let $x(n)$ a real signal defined over N sample ($n = \{0, \dots, N-1\}$).

- Construction of a new sequence of $2N$ samples:

$$x_p(n) = \begin{cases} x(n) & 0 \leq n < N \\ x(-n-1) & -N \leq n \leq -1 \end{cases}$$

  $x_p(n)$ is now even symmetric with respect to the point $n = -\frac{1}{2}$.

- we define $n' = n + \frac{1}{2}$, to get an even symmetry with respect to $n' = 0$. The DFT of this 2N-point even symmetric sequence:

$$\begin{aligned} \hat{x}(u) &= \sum_{n'=-N+\frac{1}{2}}^{N-\frac{1}{2}} x_p(n' - \frac{1}{2}) e^{-\frac{2i\pi}{2N} n' u} \\ &= \sum_{n'=-N+\frac{1}{2}}^{N-\frac{1}{2}} x_p(n' - \frac{1}{2}) \cos(\frac{2\pi n' u}{2N}) \\ & \quad -i \sum_{n'=-N+\frac{1}{2}}^{N-\frac{1}{2}} x_p(n' - \frac{1}{2}) \sin(\frac{2\pi n' u}{2N}) \end{aligned}$$

$x_p(n)$ is even and $sin(\frac{\pi(2n+1)u}{2N})$ is odd. The second term is then null.

Discrete Cosine Transform

$$\hat{x}(u) = \sum_{n'=-N+\frac{1}{2}}^{N-\frac{1}{2}} x_p(n' - \frac{1}{2}) \cos(\frac{2\pi n' u}{2N})$$

$\hat{x}(u)$ is then real and even $\hat{x}(u) = \hat{x}(-u)$. We replace n' with $n + \frac{1}{2}$.

$$\hat{x}(u) = 2 \sum_{n=0}^{N-1} x(n) \cos(\frac{\pi(2n+1)u}{2N})$$

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

Discrete Cosine Transform

Direct DCT:

$$\hat{x}(u) = \lambda_N(u) \sum_{n=0}^{N-1} x(n) \cos(\frac{\pi(2n+1)u}{2N})$$

with $u = \{0, \ldots, N-1\}$.

$$\lambda_N(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u \neq 0 \end{cases}$$

Inverse DCT:

$$x(n) = \sum_{u=0}^{N-1} \lambda_N(u)\hat{x}(u) \cos(\frac{\pi(2n+1)u}{2N})$$

Discrete Cosine Transform

Direct 2D DCT:

$$\hat{x}(u,v) = \lambda_N(u)\lambda_M(v)\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m)cos(\frac{\pi(2n+1)u}{2N})cos(\frac{\pi(2m+1)v}{2M})$$

$$\lambda_N(u) = \begin{cases}\sqrt{\frac{1}{N}} & u=0\\ \sqrt{\frac{2}{N}} & u\neq 0\end{cases}$$

similar for $\lambda_M(v)$

Inverse 2D DCT:

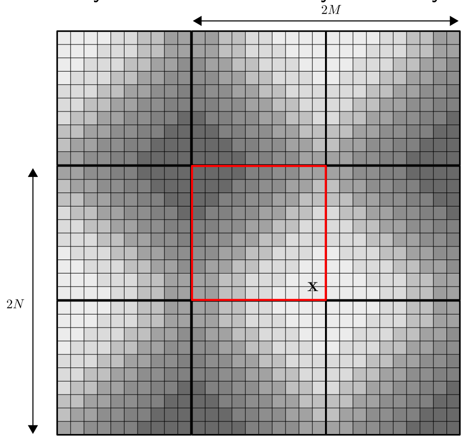
$$x(n,m) = \sum_{u=0}^{N-1}\sum_{v=0}^{M-1} \lambda(u)\lambda(v)\hat{x}(u,v)cos(\frac{\pi(2n+1)u}{2N})cos(\frac{\pi(2m+1)v}{2M})$$

$$\lambda_N(u) = \begin{cases}\sqrt{\frac{1}{N}} & u=0\\ \sqrt{\frac{2}{N}} & u\neq 0\end{cases}$$

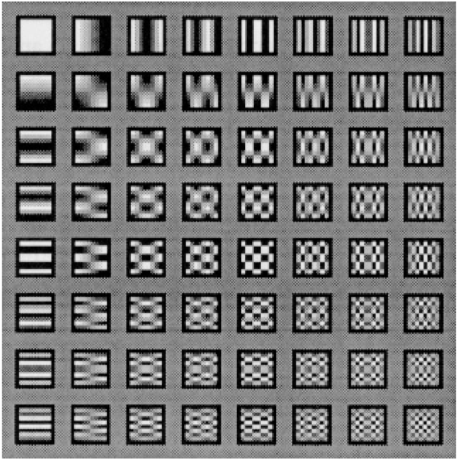
simalr for $\lambda_M(v)$

Properties

The implicit periodicity does not create any boundary

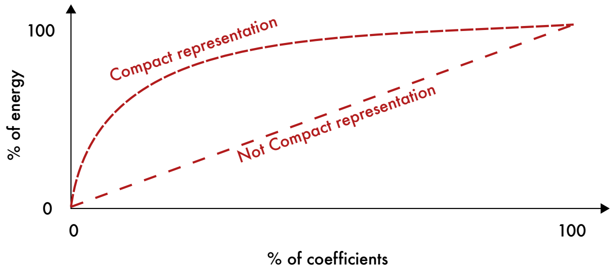


DCT2 basis



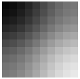
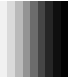
Energy compaction

The compaction is measured by the energy that remains after putting part of the coefficients to zero

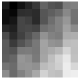
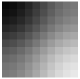
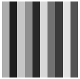
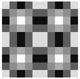


Compaction

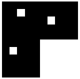
For this smooth images, all the energy is compacted on 3 coefficients.

 $=$ $1200\times$  $-$ $400\times$  $-$ $400\times$ 

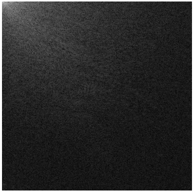
And even more complex shapes can be generated with 5 coefficients:

 $=$  $+$ $100\times$  $+$ $100\times$ 

The energy is compacted as long as the image variation are not too localized.

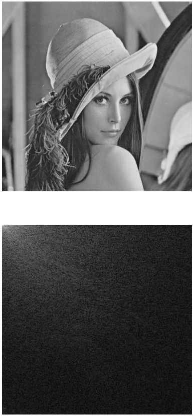
The DCT of  is 

DCT and compression

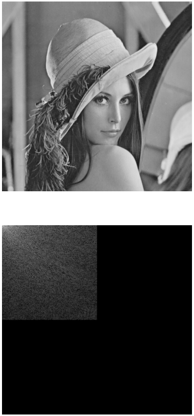


100%

DCT and compression



100%

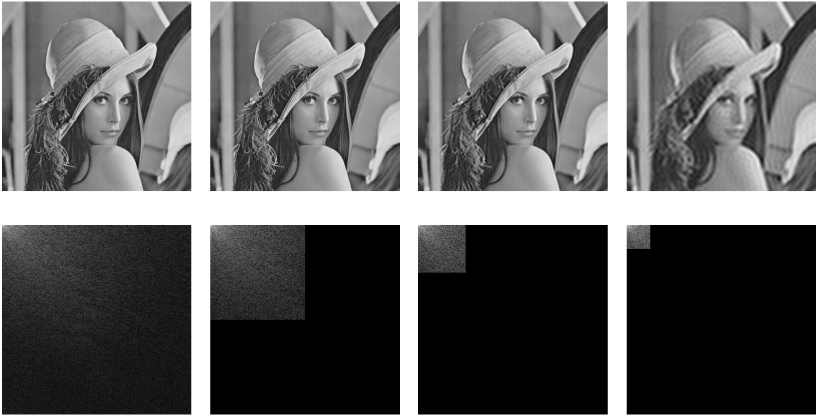


25%

DCT and compression

100% 25% 6.25%

DCT and compression



100% 25% 6.25% 1.56%

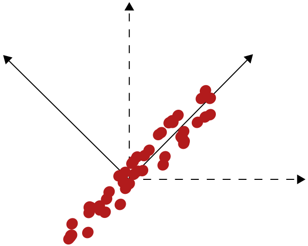
DCT and compression

6.25% of DCT coefficients

6.25% of pixels

"Optimal" Transform

In order to Compact the energy as much as possible, one needs to take into account the statistics of the signal to transform.



The goal is to align the basis along the most significant directions of the signals.

Karhunen-Loeve Transform (KLT)

Similar than in the Principal Component Analysis, one considers the covariance of the signal $\mathbf{x}$ which is defined as

$$\mathbf{\Sigma} = \mathbb{E}((\mathbf{x} - \overline{\mathbf{x}})^\top (\mathbf{x} - \overline{\mathbf{x}}))$$

The covariance matrix is diagonalized

$$\mathbf{\Sigma} = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U}$$

The signal is projected on the eigenvectors $\mathbf{U} = \{u(n,u)\}$

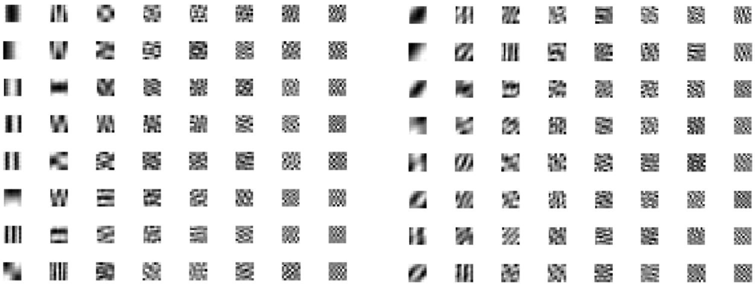
$$\forall u \in \{0, \dots, N\} \quad \hat{x}(u) = \sum_{n=0}^{N-1} x(n) u(n,u)$$

The inverse transform is $\mathbf{U} = \{u(n,u)\}$

$$\forall n \in \{0, \dots, N\} \quad x(n) = \sum_{u=0}^{N-1} \hat{x}(u) u(n,u)$$

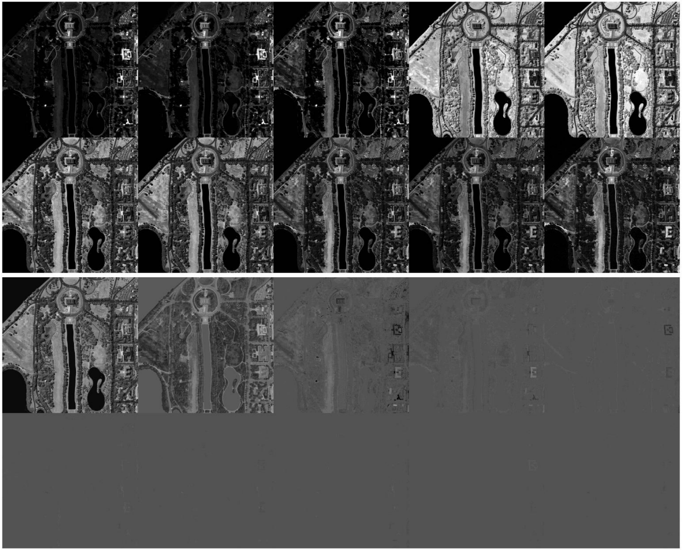
Examples of KLT

When a KLT is calculated on edges of different directions



[https://web.stanford.edu/class/ee398a/projects/reports/Hampapur_Ni.pdf]

KLT application



[http://fourier.eng.hmc.edu/e161/lectures/klt/node9.html]

Table of Contents

Time-frequency; Spatial-frequency problem

Frequential representations of signals, such as Fourier transform, are widely used. However, they suffers from a localization problem:

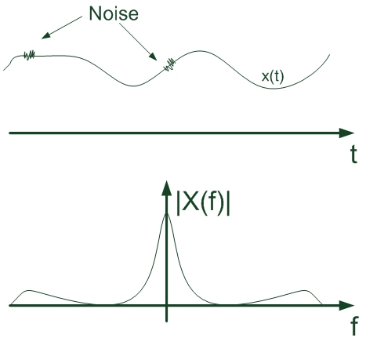
Time vs frequency for 1D signal

Space vs frequency for 2D signal

Time vs frequency localization for 1D signal

$x(t)$ is a continuous signal, corrupted by noise.
We compute its Fourier transform:

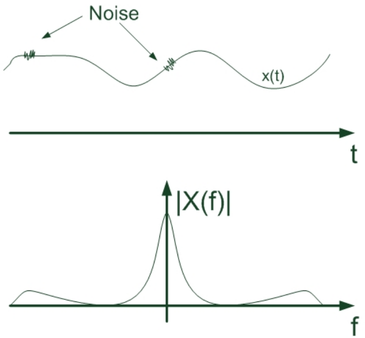
- Are we able to detect the noise in the Fourier spectrum?
- Are we able to remove it?
- Are we able to identify when the noise occured over time?



Time vs frequency localization for 1D signal

$x(t)$ is a continuous signal, corrupted by noise.
We compute its Fourier transform:

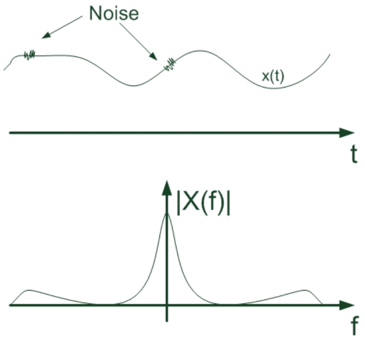
- Are we able to detect the noise in the Fourier spectrum? YES
- Are we able to remove it?
- Are we able to identify when the noise occured over time?



Time vs frequency localization for 1D signal

$x(t)$ is a continuous signal, corrupted by noise.
We compute its Fourier transform:

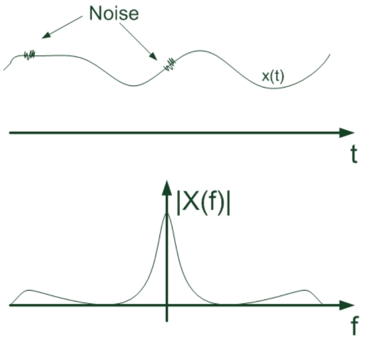
- Are we able to detect the noise in the Fourier spectrum? YES
- Are we able to remove it? YES
- Are we able to identify when the noise occured over time?



Time vs frequency localization for 1D signal

$x(t)$ is a continuous signal, corrupted by noise.
We compute its Fourier transform:

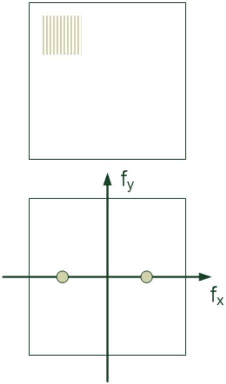
- Are we able to detect the noise in the Fourier spectrum? YES
- Are we able to remove it? YES
- Are we able to identify when the noise occured over time? NO



Space vs frequency localization for 2D signal

$\mathbf{X}(n, m)$ is an image of size $N \times M$.
We compute its Fourier transform:

- Are we able to detect the patch of texture in the Fourier spectrum? YES;
- Are we able to remove it? YES;
- Are we able to localize it spatially in the picture? NO



Problem

Fourier transform correlates the signals with a family of waveformes that are well localized in frequency (but nothing in time):

$$\hat{x}(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-2i\pi\omega t}dt$$

To be able to examine the properties of a signal in a time (or space)-frequency domain, a trade-off between the two representations must be found.

How to define a transform that correlates the signal with a family of waveforms that are well concentrated in time (or space) and in frequency?

Uncertainty principle

Can we construct a function well localized in time and frequency?

- Dirac δ_T:
  - $\rightarrow$ well localised in time $t = T$;
  - $\rightarrow$ $\hat{\delta}_T(\omega) = e^{-2i\pi\omega T}$, energy uniformly spread over all frequencies.

- Time scaling: $x_s(t) = \frac{1}{\sqrt{s}} x(\frac{t}{s})$, $s > 1$.
  - $\rightarrow$ we gain in time localization;
  - $\rightarrow$ $\hat{x}_s(\omega) = \sqrt{s}\hat{x}(sf)$, the Fourier transform is dilated.

Uncertainty principle

Heisenberg uncertainty principle

In quantum mechanics, the Heisenberg uncertainty principle states that certain pairs of physical properties, like position and momentum, cannot both be known to arbitrary precision.
That is, the more precisely one property is known, the less precisely the other can be known.

Time and frequency energy concentrations are then restricted by this principle.

If f is $\mathcal{L}^2$, then its time root deviation σ_t and its Fourier root deviation σ_f are defined. Then the Heisenberg uncertainty principle states that

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4\pi}$$

- σ_t is the standard deviation of the function in the temporal domain;
- σ_ω is the standard deviation of the function in the frequency domain.

Uncertainty principle

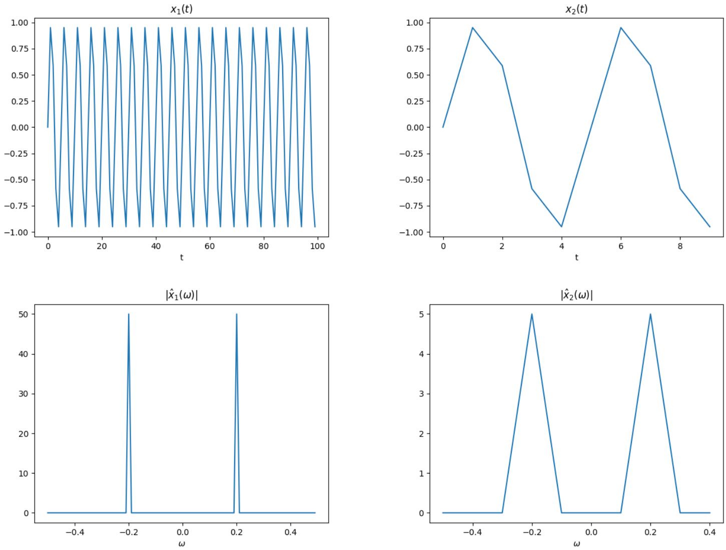
$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4\pi}$$

It means that there is no finite energy function which is compactly supported both in the time and frequency domains.

Example for Fourier:

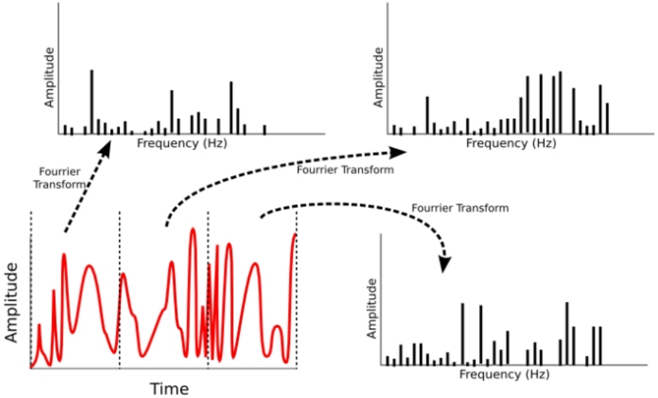
- $\sigma_t \longrightarrow +\infty$ (constant), $\sigma_\omega \longrightarrow 0$;
- $\sigma_\omega \longrightarrow +\infty$, $\sigma_t \longrightarrow 0$ (Dirac).

Example


$x_1(t)$
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
20
40
60
80
100
t
$x_2(t)$
0
2
4
6
8
t
$|\hat{x}_1(\omega)|$
50
40
30
20
10
0
−0.4
−0.2
0.0
0.2
0.4
ω
$|\hat{x}_2(\omega)|$
5
4
3
2
1
0
ω

Windowed Fourier Transform

The windowed Fourier transform replaces the Fourier transform's sinusoidal wave by the product of a sinusoid and a window which is localized in time. The windowing can be used to divide the signal in small pieces, and transform them separately. It takes two arguments: time and frequency.



Windowed Fourier Transform

Gabor defined in 1946 a new decomposition using a spatial window in the Fourier integral. The window is translated along the spatial axis in order to cover the whole signal.
At a position t_0 and for a frequency ω_0, the windowed Fourier transform of a function $x(t)$ ($\in L^2(\mathbb{R})$) is defined by

$$Sx(\omega_0, t_0) = \int_{-\infty}^{+\infty} x(t) \underbrace{g(t - t_0)}_{\text{Spatial window}} e^{-2i\pi\omega_0 t} dt$$

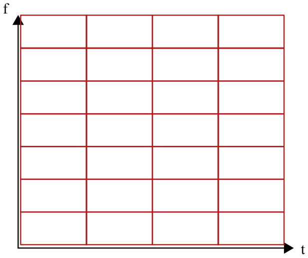
It measures locally, around the point t_0, the amplitude of the sinusoidal wave component of frequency ω_0.

Originally, the window function $g(t)$ is a Gaussian (Gabor transform).
However, different windows can be used: rectangle, Hamming, Blackman...
The resolution in time and frequency of the windowed Fourier transform depends on the spread of the window in time and frequency.

Windowed Fourier Transform

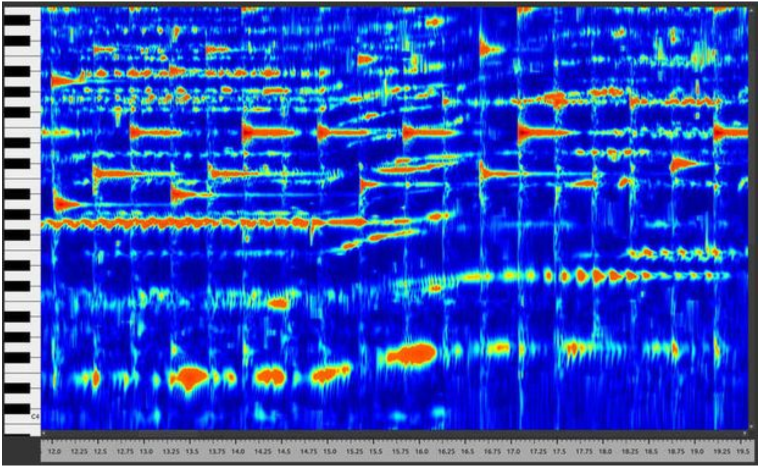
We can use the family $\{g_{\omega_z,t_z}(t)\}_{(\omega_z,t_z)\in\mathbb{R}^2}$ to cover the spatial-frequency domain with $g_{\omega_0,t_0}(t) = g(t-t_0)e^{-2i\pi\omega_0 t}$.

- t_z translation in the time domain;
- ω_z translation in the frequency domain.



Inconvenience = transform having a fixed resolution in the spatial and frequency domains (impossible to zoom into the irregularities of the signal).

Example: spectrogram



Definition of the CWT

To overcome the fixed resolution both in spatial and frequency domains, Jean Morlet defined the continuous wavelet transform (CWT) by decomposing the signal into a family of functions which are the translation and the dilatation of a unique function $\psi(x)$.

Definition of the CWT

The continuous wavelet transform of a function $x(t)$ ($\in L^2\mathbb{R}$) is defined by

$$\gamma(s,\tau) = \int_{-\infty}^{+\infty} x(t)\psi^*_{s,\tau}(t)dt$$

The inverse wavelet transform is defined by

$$x(t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \gamma(s,\tau)\psi_{s,\tau} ds d\tau$$

- $\psi(t)$ is the mother wavelet;
- $\psi_{s,\tau}$ is the family of functions $(s,\tau) \in \mathbb{R}^2$;

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

- s is the scale parameter, τ the translation parameter and $\frac{1}{\sqrt{s}}$ a normalization factor.

Definition of the CWT

Properties:

- To reconstruct the signal without loss of information, the function $\psi(t)$ must satisfy the admissibility conditions:

$$\int_{\mathbb{R}} \frac{\left|\psi(\hat{\omega})\right|^2}{|\omega|} dw < +\infty$$

where, $\psi(\hat{\omega})$ is the Fourier transform of $\psi(t)$;

- The admissibility condition implies that

$$\left|\psi(\hat{\omega})\right|^2{}_{\omega=0} = 0$$

This means that wavelets must have a band-pass like spectrum.

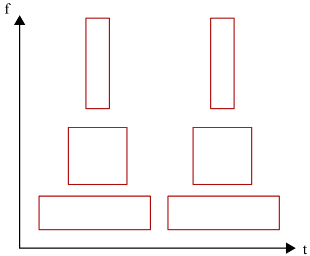
- The CWT is highly redundant (continuously shifting a scalable function over a signal):

a one-dimensional signal $\stackrel{CWT}{\rightarrow}$ a two-dimensional time-scale joint representation.

$$\gamma(s,\tau) = \int_{-\infty}^{+\infty} x(t)\psi^*_{s,\tau}(t)d\omega$$

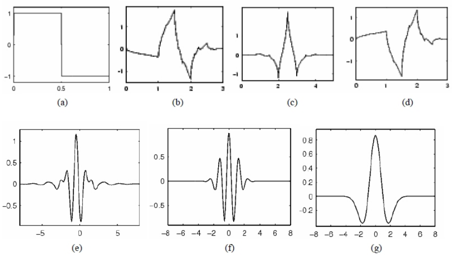
Definition of the CWT

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



Adapted resolution in the spatial and frequency domains.
(LF=long duration; HF=short duration)

Definition of the CWT



(a) Haar, (b) Daubechies4, (c) Coiflet1, (d) Symlet2, (e) Meyer, (f) Morlet, (g) Mexican hat.

Definition of the CWT

Haar wavelet (the oldest one):

Mother wavelet

Wavelet function

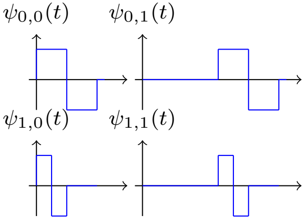
$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & otherwise \end{cases}$$

Scaling function

$$\phi(t) = 1_{[0,1]}$$

$$\{\psi_{s,\tau}(t) = \psi(2^s t - \tau)\}_{(s,\tau)\in\mathcal{Z}^2}$$

(Haar basis)



Wavelet series (frames)

The wavelet transform can be discretized by sampling the time and the scale parameters of a continuous wavelet transform. We must cover the time-frequency space. The goal is to decrease the redundancy (!) of the CWT.
A real continuous wavelet transform of $x(t)$ is given the function

$$\begin{aligned}\gamma(s,\tau) &= \int_{-\infty}^{+\infty} x(t)\psi^*_{s,\tau}(t)dt \\ \psi_{s,\tau}(t) &= \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s})\end{aligned}$$

Discrete wavelet family:

$$\psi_{j,n}(t) = \frac{1}{\sqrt{a^j}}\psi(\frac{t-n\tau_0 a^j}{a^j}),\ (j,n) \in \mathbb{N} \quad (1)$$

To cover the time-frequency plane with Heisenberg boxes:

- the parameter s is expressed as a^j $(j \in \mathcal{Z})$ (sampling);
- The parameter τ is sampled uniformly at intervals proportional to the scale a^j.

When the scale increases, the density of samples increases. Dyadic wavelets are wavelets which satisfy an additional scaling property: $a = 2$.

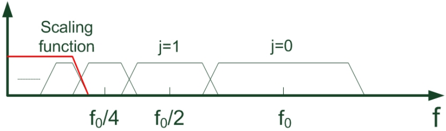
Scaling function

Discrete wavelet family:

$$\psi_{j,n}(t) = \frac{1}{\sqrt{a^j}}\psi(\frac{t - n\tau_0 a^j}{a^j}),\ (j,n) \in \mathbb{N}$$

To recover $x(t)$, we need a complement of information (also true for the CWT).

Every time the wavelet is scaled in the time domain with the factor a, the frequency bandwith is halved. It means that you will need an infinite number of wavelets to recover the signal (low-frequencies).



$$\xi(t) = \sum_{j,n} \gamma(j,n)\psi_{j,n}(t)$$

The scaling function ξ is a signal with a low-pass spectrum.

Note: the scaling function has nothing to do with the scaling parameter.

$$x(t) \xrightarrow{\text{Wavelet serie}} \begin{cases} \lambda_k \\ \gamma(j,n) \end{cases}$$

with,

$$\lambda_k = \int_{-\infty}^{+\infty} x(t)\xi_k^*(t)dt, \text{ low frequencies wavelet coefficients}$$

$$\gamma(j,n) = \int_{-\infty}^{+\infty} x(t)\psi_{j,n}^*(t)dt, \text{ high frequencies wavelet coefficients}$$

The signal $x(t)$ can be retrieve from the wavelet coefficients

$$x(t) = \sum_k \lambda_k \xi_k(t) + \sum_{j,n} \gamma(j,n)\psi_{j,k}(t)$$

if and only if

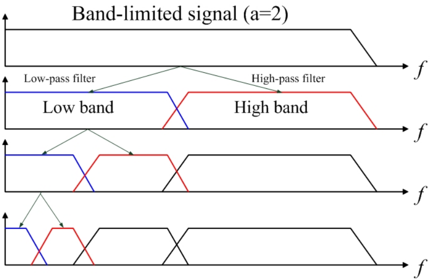
- ξ_k and $\psi_{j,k}$ are an orthogonal basis:

$$\sum_k |\lambda_k|^2 + \sum_{j,n} |\gamma(j,n)|^2 = \|x\|^2$$

- Bi-orthogonal wavelets: ξ and ψ are the wavelet used to decompose the signal and we define $\widetilde{\xi}$ and $\widetilde{\psi}$ to reconstruct the signal.

Iterated filter bank

Wavelet transform = band-pass filters + low-pass filter (a scaling function)
$$\Longleftrightarrow$$
a filter bank.

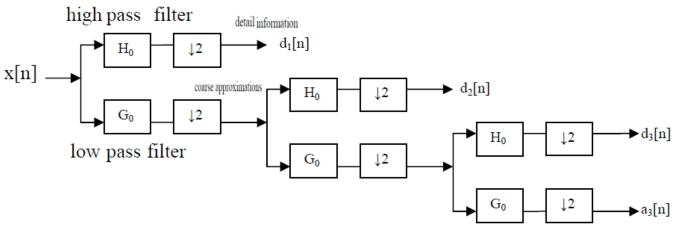


If we implement the wavelet transform as an iterated filter bank, we have just to specify a low-pass filter and a high-pass filter.
But, which scale should we choose?

Discrete Wavelet transform (DWT)

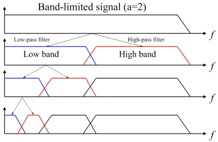
Warning.... Up to now, the input signal was continuous... only the scale and translation parameters were discrete.

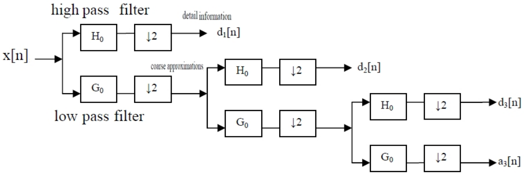
The original signal $x(n)$ passes through two complementary filters and emerges as two signals. The low pass filter is denoted by G_0 while the high pass filter is denoted by H_0. At each level, the high pass filter produces detail information, $d(n)$, while the low pass filter associated with scaling function produces coarse approximations, $a(n)$. Mallat-tree decomposition is shown below:



The DWT uses dyadic scales and positions (scales and positions based on powers of 2).

Discrete Wavelet transform (DWT)





Finally a wavelet decomposition of a signal $x(n)$ will provide:

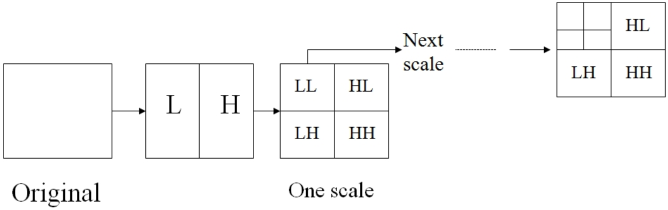
- A low resolution (low frequency) called $\lambda_j(n)$ $(a_3(n))$;
- A set of detailed signal (medium to high frequencies), $p \in \{j, j-1, \dots 1\}$ $(d_i(n))$.

multiresolution approach

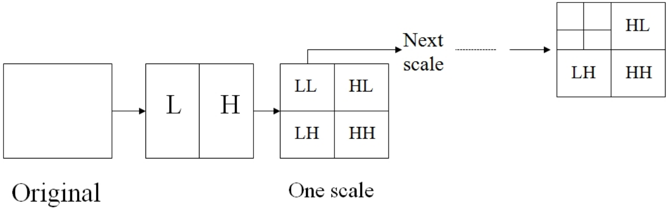
2D Discrete Wavelet transform

A 2D DWT is the combination of two 1D DWT:

1. Replace each row with its 1D DWT;
2. Replace each column with its 1D DWT;
3. Repeat steps (1) and (2) on the lowest subband for the next scale;
4. Repeat steps (3) until as many scales as desired have been completed.



2D Discrete Wavelet transform



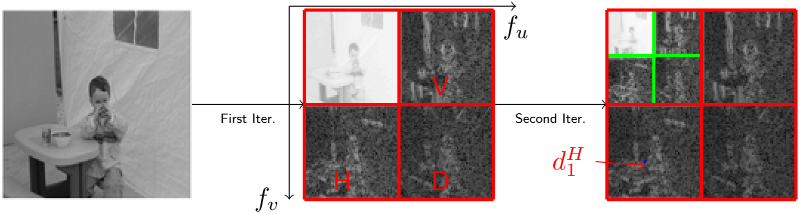
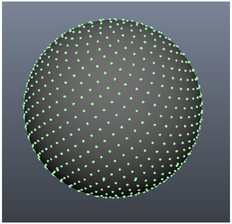
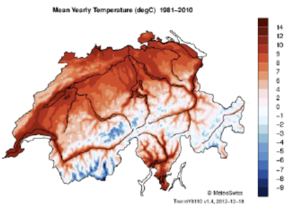


Table of Contents

Context

When the domain is not cartesian, the transformed above are not defined.

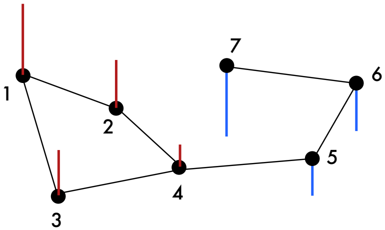




Introduction of Graph

If we want to define a transform, one needs to take into account the structure behind the data.

Graphs represent a pairwise relationship between the entities.



$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where

- $\mathcal{V}$ are the nodes (indexed from 1 to N)
- $\mathcal{E}$ are the edges
- $\mathcal{W}$ are the weights on the edges

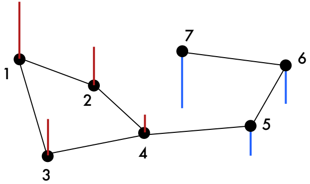
We define a function f on the graphs by assigning a value to each node: $f : \mathcal{V} \rightarrow \mathbb{R}$.

How to define the transforms on the graph?

Useful definitions

In order to represent the pairwise relation, one defines the **adjacency matrix $\mathbf{A}$**:

$$a_{ij} = \begin{cases} 1 \text{ if } e_{i,j} \in \mathcal{E} \\ 0 \text{ otherwise} \end{cases}$$

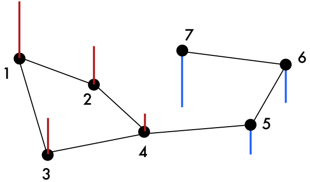


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Useful definitions

In order to represent the connectivity of a vertex, one defines the **degree matrix $\mathbf{D}$**:

$$d_{ij} = \begin{cases} \text{degree}(v_i) \text{ if} i = j \\ 0 \text{ otherwise} \end{cases}$$

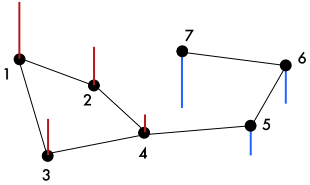


$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Useful definitions

One defines the **Laplacian matrix L**:

$$L = D - A$$



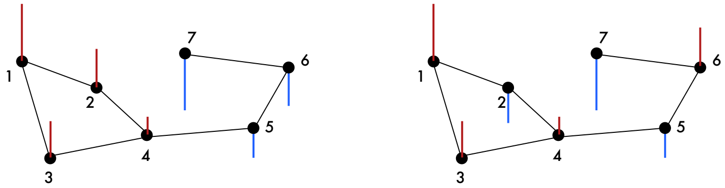
$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

It is called Laplacian because, this is the natural extension of laplacian operator to the graph. At each nodes, it calculates:

$$d_i f(i) - \sum_{j \in \text{Neighborhood}} f(j)$$

Smoothness

Laplacian matrix quantifies the smoothness of the signal on the graph. It says how much a $f(i)$ value can be estimated by the linear combination of its neighbors.



$\mathbf{f}^\top \mathbf{L} \mathbf{f} = 20$ $\mathbf{f}^\top \mathbf{L} \mathbf{f} = 60$

Laplacian and Fourier Transform

In temporal domain, Laplacian operator is defined as

$$\Delta(f(t)) = \frac{\partial^2 f(t)}{\partial t}$$

Fourier basis are eigenvectors of the Laplace operator:

$$\Delta(e^{2i\pi\omega t}) = \frac{\partial^2 e^{2i\pi\omega t}}{\partial t} = -(2\pi\omega)^2 e^{2i\pi\omega t}$$

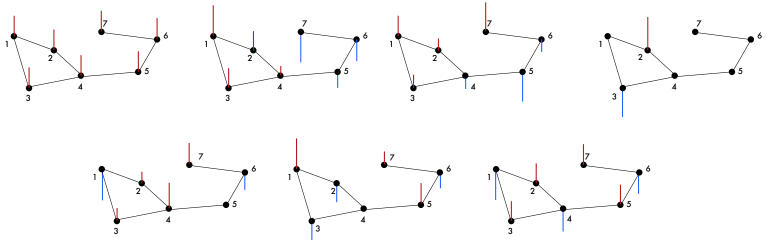
Since the Laplacian matrix $\mathbf{L}$ is positive semi-definite, it can be diagonalized:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

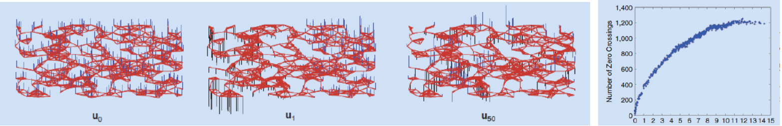
The orthogonal eigenvectors $\mathbf{U}$ are defined as the analogy of Fourier Transform on the graph, called **Graph Fourier Transform**.
The eigenvalues λ_i are the analog of the frequencies.

Frequency in the graph

Eigenvectors ranked in increasing eigenvalue order



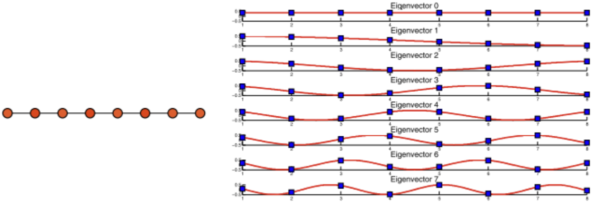
Frequencies as zero-crossing



[Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3), 83-98.]

Examples

If the graph represents a 1D cartesian space, th eigen decomposition fits with the DCT



[Shuman, D. I., Ricaud, B., and Vandergheynst, P. (2016). Vertex-frequency analysis on graphs. Applied and Computational Harmonic Analysis, 40(2), 260-291.]

Graph Fourier Transform

Compute the Laplacian matrix:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

Find the eigenvectors and the eigenvalues:

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$$

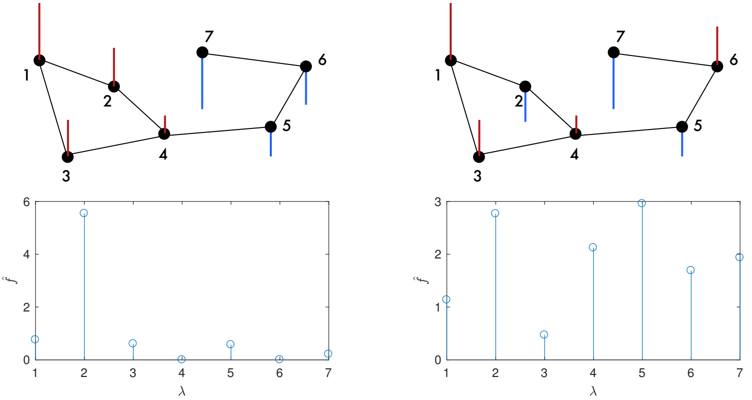
Project the signal f on the eigenvectors to get the transformed coefficients:

$$\hat{\mathbf{f}} = \mathbf{U}^\top\mathbf{f}$$

The inverse transform is:

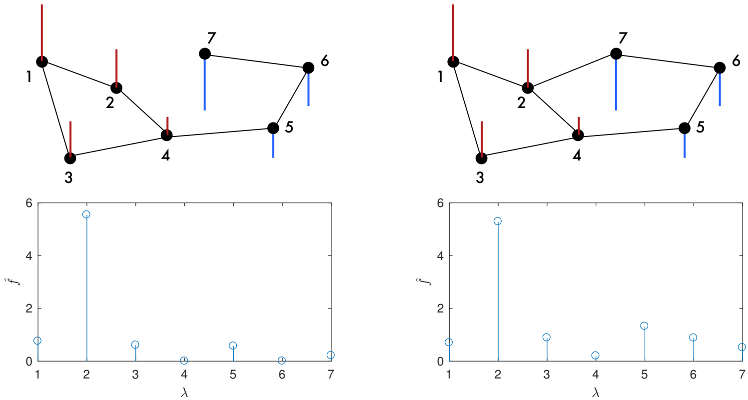
$$\mathbf{f} = \mathbf{U}\hat{\mathbf{f}}$$

Examples

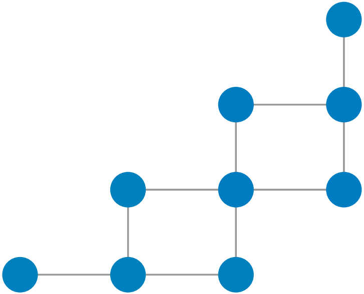


Importance of the graph

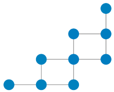
One connection can change the signal spectrum.



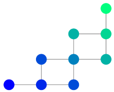
GFT basis on a toy graph



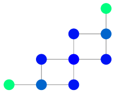
GFT basis on a toy graph



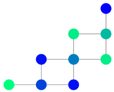
$\mathbf{u}_1$



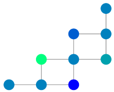
$\mathbf{u}_2$



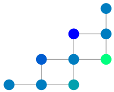
$\mathbf{u}_3$



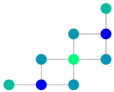
$\mathbf{u}_4$



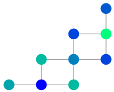
$\mathbf{u}_5$



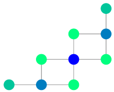
$\mathbf{u}_6$



$\mathbf{u}_7$

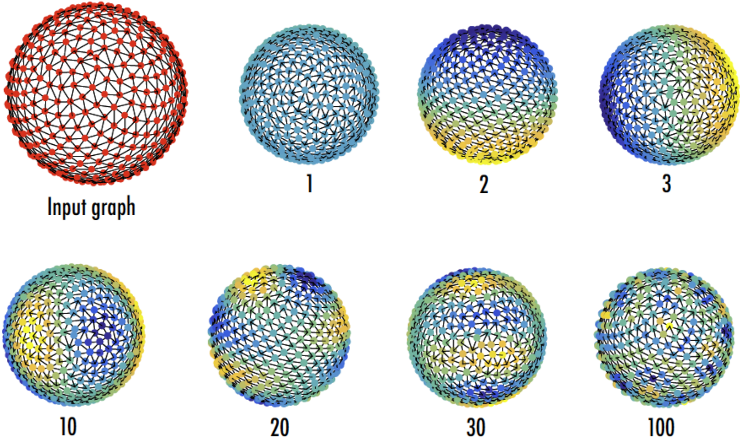


$\mathbf{u}_8$



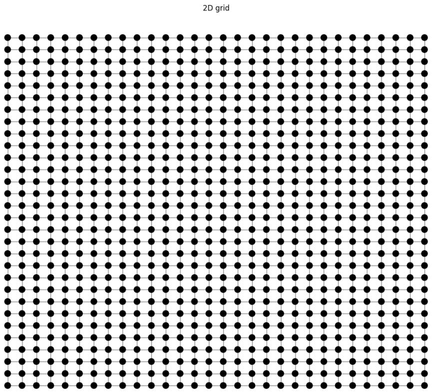
$\mathbf{u}_9$

Graph Transforms on the sphere



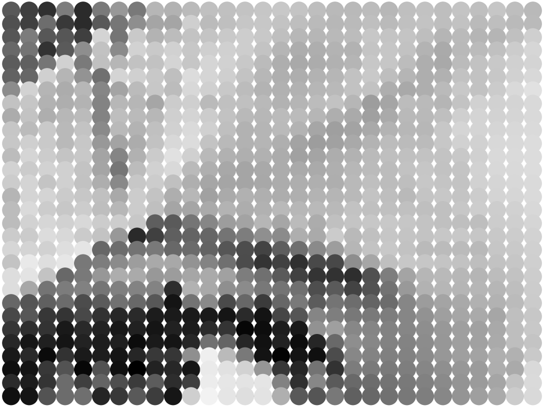
Graph-based transform on 2D image

The 2D image support can be seen as a 2D grid graph.



Graph-based transform on 2D image

The image is a signal on this graph.

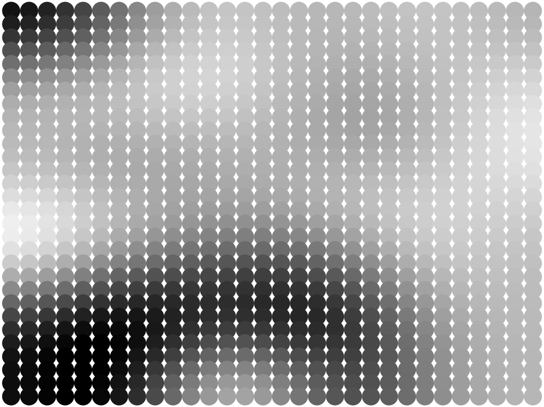


Graph-based transform on 2D image

```
import numpy as np
import pygsp as gp
import cv2

# We construct the 2D grid graph:
N = 30
G = gp.graphs.Grid2d(N1=N,N2=N)

# We create the signal on the graph from the image
img = cv2.imread('lena.jpg')
i1 = 105
i2 = 125
signal = img[i1:i1+N,i2:i2+N,1]
signalV = signal.flatten()

# We compute the graph transform (Equivalent to DCT)
G.compute_fourier_basis(N*N)
t = G.gft(signalV)

# We reconstruct the signal from the 3.33% first transform coefficients
comp = 30
t[comp:len(t)] = 0
signalR = G.igft(t)
```

Graph-based transform on 2D image

$w_{i,j} = 1$



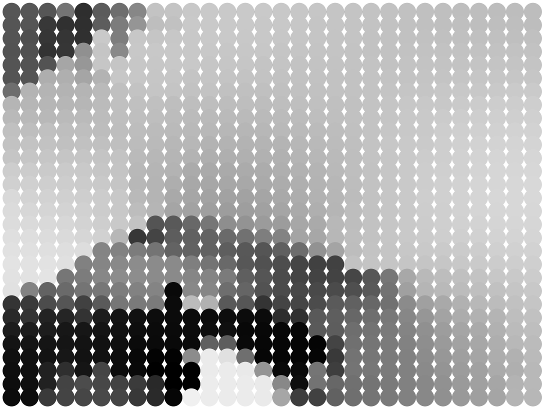
Graph-based transform on 2D image

The signal information can be used to adjust the weight

```
# We construct a new 2D grid graph
Gw = gp.graphs.Grid2d(N1=N,N2=N)

# We adjust the weights based on the signal
for i in range(N*N):
   for j in range(i,N*N):
     if Gw.A[i,j] == 1:
       Gw.W[i,j] = np.exp(-1*np.abs(int(signalV[i])-int(signalV[j]))/10)
       Gw.W[j,i] = Gw.W[i,j]

# We compute the graph transform (similar to the KLT)
Gw.compute_laplacian()
Gw.compute_fourier_basis(N*N)
tw = Gw.gft(signalV)

# We reconstruct the signal from the 3.33% first transform coefficients
 tw[comp:len(tw)] = 0
signalRw = Gw.igft(tw)
```

Graph-based transform on 2D image

$$w_{i,j} = exp(-|x_i - x_j|/\sigma)$$



Transforms

T. Maugey

Graph-based transform on 2D image

[Shen, G., Kim, W. S., Narang, S. K., Ortega, A., Lee, J., and Wey, H. (2010, December). Edge-adaptive transforms for efficient depth map coding. In 28th Picture Coding Symposium (pp. 566-569). IEEE.]

Table of Contents

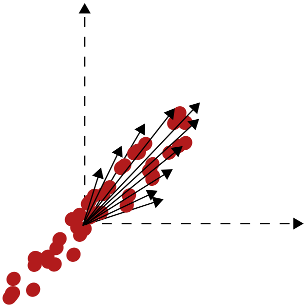
Basic Problem

Having a vector $\mathbf{x}$ of dimension N.
A basis change is

$$\mathbf{x} = \mathbf{A}\mathbf{c}.$$

In compression, the goal is to have $\mathbf{c}$ as sparse as possible.

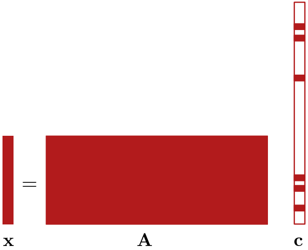
What if $\mathbf{A}$ is not an orthogonormal basis anymore ?



Over complete dictionary

$\mathbf{A}$ is an over-complete and has dimension $N \times P$ with $P > N$.

$$\mathbf{x} = \mathbf{A}\mathbf{c}.$$



The problem of finding the best dictionary is

$$(\mathbf{c}^*, \mathbf{A}^*) = \arg\min_{\mathbf{c},\mathbf{A}} ||\mathbf{c}||_0 \quad \text{s.t.} \quad P < P_{\max} \text{ and } \mathbf{A}\mathbf{c} = \mathbf{x}.$$

Non-convex problem, and depends on the application.

Table of Contents

References

Bracewell, R. N., and Bracewell, R. N. (1986). The Fourier transform and its applications (Vol. 31999). New York: McGraw-Hill.

Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete cosine transform. IEEE transactions on Computers, 100(1), 90-93.

Mallat, S. (1999). A wavelet tour of signal processing. Elsevier.

Taubman, D., and Marcellin, M. (2012). JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice (Vol. 642). Springer Science and Business Media.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3), 83-98.

Tekalp, A. M. (2015). Digital video processing. Prentice Hall Press.