

Traitement et analyse d'images numériques

# Partie 5: Transformée de Fourier pour l'imagerie numérique

Pierre Maurel

Visages, IRISA/INRIA

`pierre.maurel@irisa.fr`

<http://www.normalesup.org/~pmaurel/IMA/>

# Introduction

Joseph Fourier, 1768-1830

THÉORIE  
 ANALYTIQUE  
 DE LA CHALEUR,

PAR M. FOURIER.



A PARIS,

CHEZ FIRMIN DIDOT, PÈRE ET FILS,  
 CI-devant chez les Mathématiciens, Cartographe Intendant  
 de la Marine, vis-à-vis N° 12

1822.

on parvient de cette manière à un résultat très-remarquable exprimé par l'équation suivante :

$$\frac{1}{2} \pi p x \sin x = S(\sin. x. q. x. d. x) + \sin. x. S(\sin. x. r. q. x. d. x) + \sin. 3. x. S(\sin. 3. x. q. x. d. x) + \dots + \sin. i. x. S(\sin. i. x. q. x. d. x) + \dots \quad (D)$$

Le second membre donnera toujours le développement cherché de la fonction  $px$ , si l'on effectue les intégrations depuis  $x=0$ , jusqu'à  $x=\pi$ .

On peut aussi vérifier l'équation précédente (D) (art. 220), en déterminant immédiatement les quantités  $a, a_1, a_2, \dots$ , etc., dans l'équation

$$q x = a_0 \sin x + a_1 \sin 2x + a_2 \sin 3x + \dots + a_i \sin i x + \dots \text{ etc.}$$

pour cela on multipliera chacun des membres de la dernière équation, par  $\sin. i. x. dx$ ,  $i$  étant un nombre entier, et l'on prendra l'intégrale depuis  $x=0$  jusqu'à  $x=\pi$ , on aura

$$S(q x \sin. i. x. dx) = a_0 S(\sin. x \sin. i. x. dx) + a_1 S(\sin. 2x \sin. i. x. dx) + \dots + a_i S(\sin. i. x \sin. i. x. dx) + \dots$$

Or on peut facilement prouver, 1° que toutes les intégrales qui entrent dans le second membre, ont une valeur nulle, excepté le seul terme  $a_i S(\sin. i. x \sin. i. x. dx)$ ; 2° que la valeur de  $S(\sin. i. x \sin. i. x. dx)$  est  $\pi$ ;

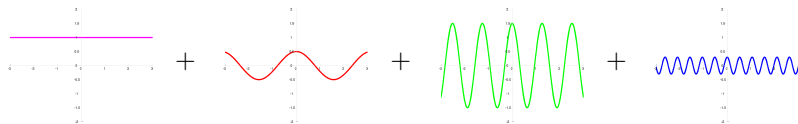
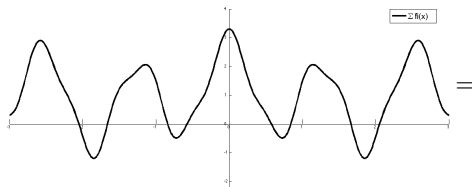


Il s'agissait initialement de résoudre un problème lié à l'équation de la chaleur : Fourier a proposé d'écrire les solutions comme une somme de fonctions sinus avec des périodes/fréquences différentes.

## Transformée de Fourier 1D

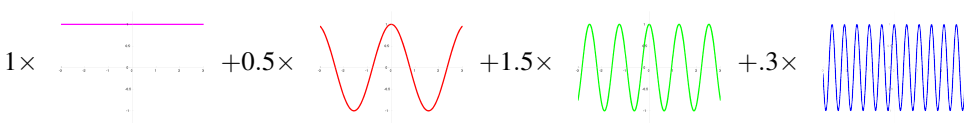
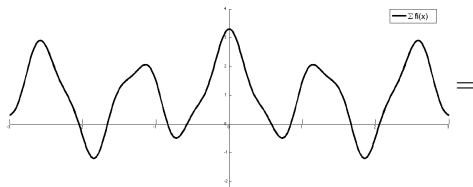
# Série de Fourier (signaux périodiques)

- Soit un signal périodique  $f(x)$  de période  $T = \frac{1}{f_0}$ .
- $f(x)$  peut se représenter par une somme dénombrable (p-e infinie) de sinus/cosinus



# Série de Fourier (signaux périodiques)

- Soit un signal périodique  $f(x)$  de période  $T = \frac{1}{f_0}$ .
- $f(x)$  peut se représenter par une somme dénombrable (p-e infinie) de sinus/cosinus



# Série de Fourier (signaux périodiques)

- Soit un signal périodique  $f(x)$  de période  $T = \frac{1}{f_0}$ .
- $f(x)$  peut se représenter par une somme dénombrable (p-e infinie) de sinus/cosinus

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n(f) e^{i2\pi \frac{n}{T} x} \quad \text{avec} \quad c_n(f) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-i2\pi \frac{n}{T} t} dt$$

# Série de Fourier (signaux périodiques)

- Soit un signal périodique  $f(x)$  de période  $T = \frac{1}{f_0}$ .
- $f(x)$  peut se représenter par une somme dénombrable (p-e infinie) de sinus/cosinus

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n(f) e^{i2\pi \frac{n}{T} x} \quad \text{avec} \quad c_n(f) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-i2\pi \frac{n}{T} t} dt$$

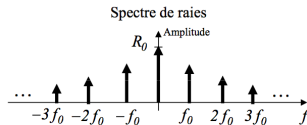
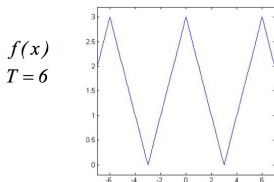
- Remarque :  $c_0(f) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt = R_0 =$  moyenne de  $f$

# Série de Fourier (signaux périodiques)

- Soit un signal périodique  $f(x)$  de période  $T = \frac{1}{f_0}$ .
- $f(x)$  peut se représenter par une somme dénombrable (p-e infinie) de sinus/cosinus

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n(f) e^{i2\pi \frac{n}{T} x} \quad \text{avec} \quad c_n(f) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-i2\pi \frac{n}{T} t} dt$$

- Remarque :  $c_0(f) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt = R_0 =$  moyenne de  $f$

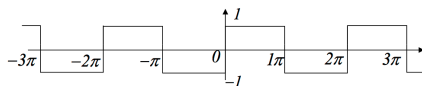


$R_0$  Signal moyen (ici 1.5)  
 $f_0$  Fréquence fondamentale (ici 1/6)  
 $nf_0$  Fréquence harmonique



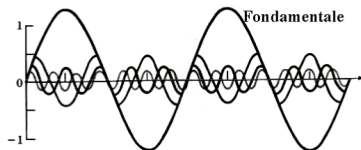
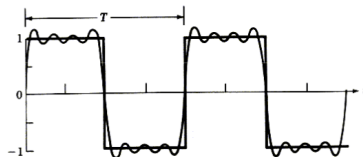
# Série de Fourier (signaux périodiques)

- Exemple, signal carré impair :



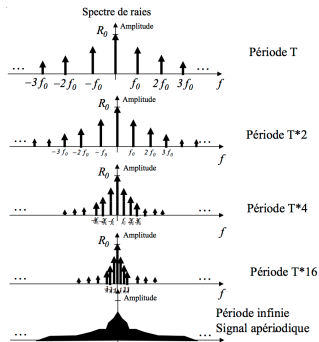
- $$f(x) = \frac{4}{\pi} \left( \frac{\sin x}{1} + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots \right)$$

- $$R_0 = 0 \text{ et } f_0 = \frac{1}{2\pi}$$



# Des séries de Fourier aux transformées de Fourier

- Plus la période,  $T$ , du signal est grande, plus la fréquence fondamentale  $f_0 = \frac{1}{T}$  est faible (ainsi que la distance entre harmoniques)
- On peut donc voir un signal **apériodique** comme un signal de **période infinie**
- La série de Fourier d'un tel signal est une suite infinie et **indénombrable** de sinusoïdes. La somme est remplacée par l'intégrale.



# Transformées de Fourier 1D

- $f$  peut maintenant être un signal **apériodique**

## Transformée de Fourier et son inverse

$$\text{Transformée de Fourier : } \mathcal{F}(f) : \xi \mapsto \hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} dx$$

$$\text{Transformée inverse : } \mathcal{F}^{-1}(\hat{f}) : x \mapsto f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\xi) e^{+i\xi x} d\xi$$

- $x \rightarrow$  coordonnée spatiale,  $\xi \rightarrow$  coordonnée fréquentielle (ou spectrale)
- les constantes (ici,  $\frac{1}{2\pi}$  sur la transformée inverse) peuvent varier suivant les définitions.

# Transformées de Fourier : propriétés

- La transformée de Fourier est à valeurs complexes :  $\hat{f}(\xi) \in \mathbb{C}$

# Transformées de Fourier : propriétés

- La transformée de Fourier est à valeurs complexes :  $\hat{f}(\xi) \in \mathbb{C}$

## Transformée de Fourier d'un signal réel

- $f(x) \in \mathbb{R}$ , on peut écrire  $f(x) = f_p(x) + f_i(x)$  avec
  - $f_p$  fonction paire,  $f_p(-x) = f_p(x) \rightarrow f_p(x) = \frac{1}{2}(f(x) + f(-x))$
  - $f_i$  fonction impaire,  $f_i(-x) = -f_i(x) \rightarrow f_i(x) = \frac{1}{2}(f(x) - f(-x))$

# Transformées de Fourier : propriétés

- La transformée de Fourier est à valeurs complexes :  $\hat{f}(\xi) \in \mathbb{C}$

## Transformée de Fourier d'un signal réel

- $f(x) \in \mathbb{R}$ , on peut écrire  $f(x) = f_p(x) + f_i(x)$  avec
  - $f_p$  fonction paire,  $f_p(-x) = f_p(x) \rightarrow f_p(x) = \frac{1}{2}(f(x) + f(-x))$
  - $f_i$  fonction impaire,  $f_i(-x) = -f_i(x) \rightarrow f_i(x) = \frac{1}{2}(f(x) - f(-x))$
- $$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} dx = \int_{-\infty}^{+\infty} f_p(x) e^{-i\xi x} dx + \int_{-\infty}^{+\infty} f_i(x) e^{-i\xi x} dx$$

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f_p(x) \cos(\xi x) dx + \int_{-\infty}^{+\infty} f_i(x) \cos(\xi x) dx$$

$$- i \int_{-\infty}^{+\infty} f_p(x) \sin(\xi x) dx - i \int_{-\infty}^{+\infty} f_i(x) \sin(\xi x) dx$$

# Transformée de Fourier d'un signal réel 1D

- or 
$$\int_{-\infty}^{+\infty} f_i(x) \cos(\xi x) dx = \int_{-\infty}^0 f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$

# Transformée de Fourier d'un signal réel 1D

- or 
$$\int_{-\infty}^{+\infty} f_i(x) \cos(\xi x) dx = \int_{-\infty}^0 f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$
$$= \int_0^{+\infty} f_i(-x) \cos(-\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$



## Transformée de Fourier d'un signal réel 1D

- or 
$$\int_{-\infty}^{+\infty} f_i(x) \cos(\xi x) dx = \int_{-\infty}^0 f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$
$$= \int_0^{+\infty} f_i(-x) \cos(-\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$
$$= - \int_0^{+\infty} f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx = 0$$

## Transformée de Fourier d'un signal réel 1D

- or 
$$\int_{-\infty}^{+\infty} f_i(x) \cos(\xi x) dx = \int_{-\infty}^0 f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$
$$= \int_0^{+\infty} f_i(-x) \cos(-\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$
$$= - \int_0^{+\infty} f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx = 0$$
- De la même manière, on obtient 
$$\int_{-\infty}^{+\infty} f_p(x) \sin(\xi x) dx = 0$$

## Transformée de Fourier d'un signal réel 1D

- or 
$$\int_{-\infty}^{+\infty} f_i(x) \cos(\xi x) dx = \int_{-\infty}^0 f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$

$$= \int_0^{+\infty} f_i(-x) \cos(-\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx$$

$$= - \int_0^{+\infty} f_i(x) \cos(\xi x) dx + \int_0^{+\infty} f_i(x) \cos(\xi x) dx = 0$$

- De la même manière, on obtient 
$$\int_{-\infty}^{+\infty} f_p(x) \sin(\xi x) dx = 0$$

- et finalement

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f_p(x) \cos(\xi x) dx - i \int_{-\infty}^{+\infty} f_i(x) \sin(\xi x) dx$$

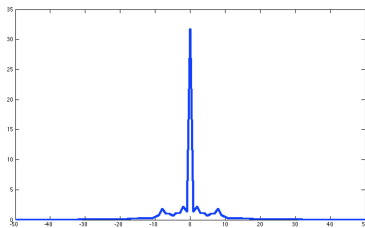
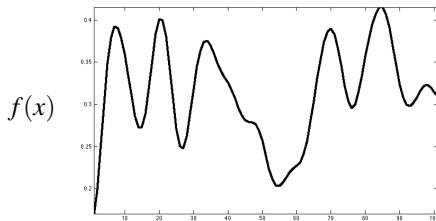
# Transformée de Fourier d'un signal réel 1D

- La partie réelle de  $\hat{f}(\xi)$  est donc la transformée de la partie paire  $f_p(x)$
- La partie imaginaire de  $\hat{f}(\xi)$  est la transformée de la partie impaire  $f_i(x)$
- Si  $f(x)$  est pair alors  $\hat{f}(\xi)$  est réel
- Si  $f(x)$  est impair alors  $\hat{f}(\xi)$  est imaginaire

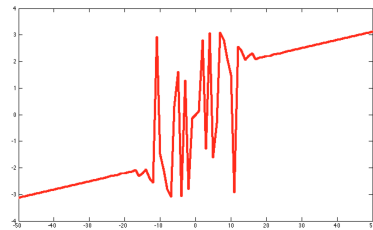
## Propriétés

- Le spectre d'amplitude est une **fonction paire**
- Le spectre de phase est une **fonction impaire**
- Ou encore :  $\hat{f}(-\xi) = \overline{\hat{f}(\xi)}$  (conjugué)

# Transformée de Fourier d'un signal réel 1D



$$|\hat{f}(\xi)|$$



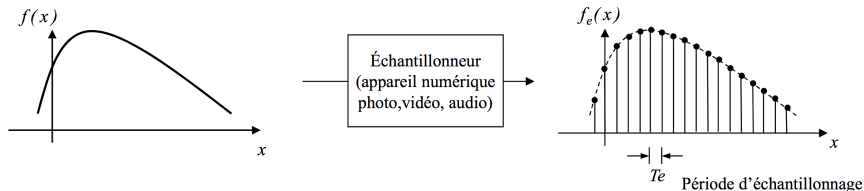
$$\arg(\hat{f}(\xi))$$

# Transformées de Fourier : propriétés

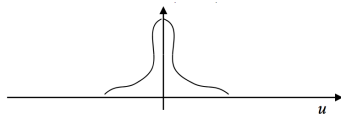
- Linéarité :  $\mathcal{F}[\alpha f + \beta g] = \alpha \mathcal{F}[f] + \beta \mathcal{F}[g]$
- Translation :  $\mathcal{F}[f(x + x_0)](\xi) = e^{i\xi x_0} \mathcal{F}[f](\xi)$
- Changement d'échelle :  $\mathcal{F}[f(ax)] = \frac{1}{|a|} \mathcal{F}[f]\left(\frac{\xi}{a}\right)$
- Convolution :  $\mathcal{F}[f * g] = \mathcal{F}(f) \cdot \mathcal{F}(g)$     et     $\mathcal{F}[f \cdot g] = \mathcal{F}(f) * \mathcal{F}(g)$

# Transformée de Fourier d'un signal échantillonné

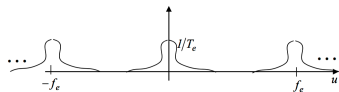
- En pratique : signal échantillonné



- On peut montrer qu'échantillonner un signal spatial revient à le **dupliquer à l'infini dans le domaine spectral**



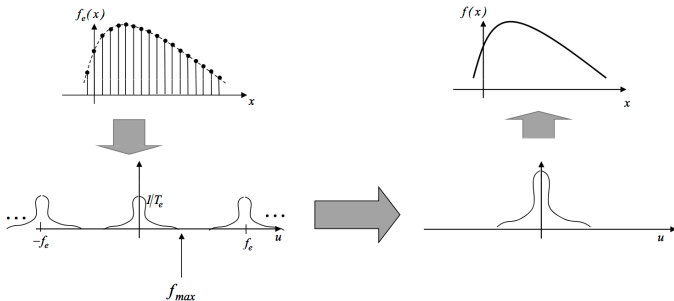
$$\hat{f}(\xi) = \mathcal{F}[f(x)](\xi)$$



$$\mathcal{F}[f_e(x)](\xi) = \frac{1}{T_e} \sum_{n=-\infty}^{\infty} \hat{f}(\xi - n f_e)$$

# Théorème de Nyquist-Shannon

- Étant donné  $f_e(x)$  il est donc possible de retrouver  $f(x)$

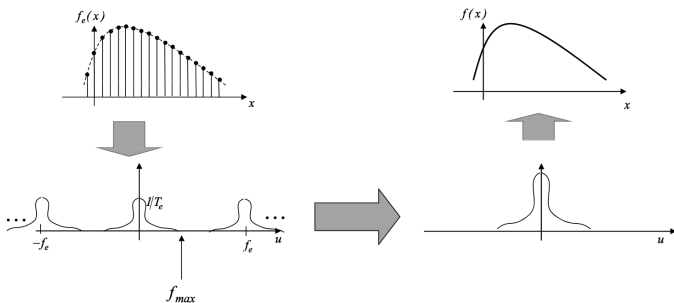


- À condition que  $f_e \geq 2f_{max}$  (fréquence de Nyquist)

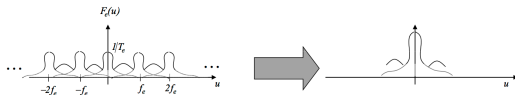


# Théorème de Nyquist-Shannon

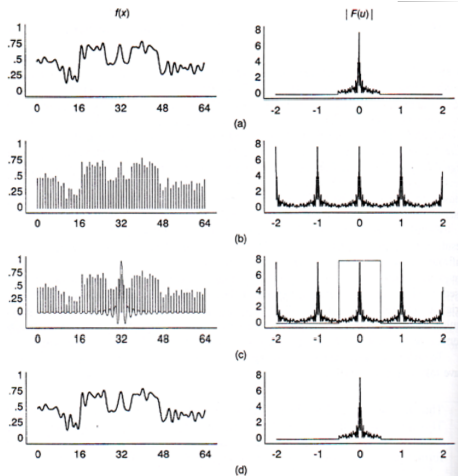
- Étant donné  $f_e(x)$  il est donc possible de retrouver  $f(x)$



- À condition que  $f_e \geq 2f_{max}$  (fréquence de Nyquist)
- Si  $f_e < 2f_{max}$ , problème de repliement de spectre  $\rightarrow$  **aliasing**



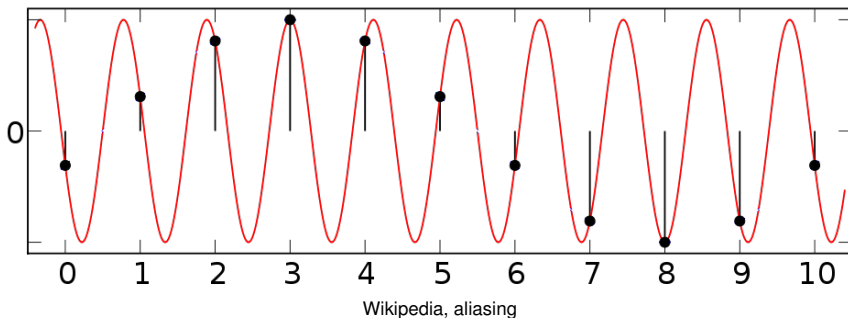
# Théorème de Nyquist-Shannon



Foley-VanDam

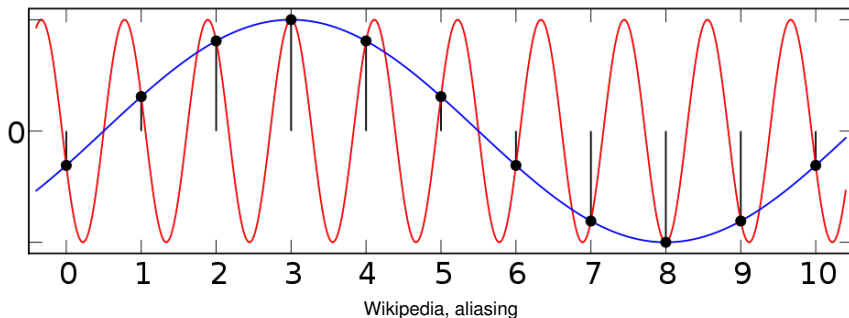
# Aliasing / repliement de spectre

- L'échantillonnage du signal (à une fréquence inférieure à la fréquence de Nyquist) fait artificiellement apparaître une fréquence différente de la fréquence d'origine.

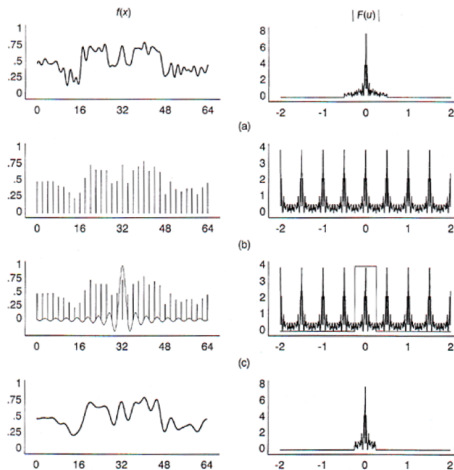


# Aliasing / repliement de spectre

- L'échantillonnage du signal (à une fréquence inférieure à la fréquence de Nyquist) fait artificiellement apparaître une fréquence différente de la fréquence d'origine.

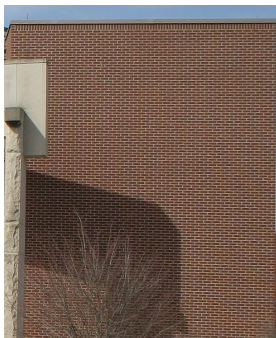


# Aliasing / repliement de spectre



Foley-VanDam

# Aliasing (en 2D)



Bon échantillonnage



Mauvais échantillonnage → aliasing

[Wikipedia-en : aliasing](#)

# Transformée de Fourier discrète

## Rappel

Transformée de Fourier continue :  $\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} dx$

Transformée inverse :  $f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\xi) e^{+i\xi x} d\xi$

## Définition TFD

- équivalent discret de la transformée de Fourier continue
- pour un signal  $s$  de  $N$  échantillons, on définit la TFD et son inverse

$$S(u) = \sum_{n=0}^{N-1} s(n) \cdot e^{-2i\pi u \frac{n}{N}}$$

$$s(n) = \frac{1}{N} \sum_{u=0}^{N-1} S(u) \cdot e^{2i\pi n \frac{u}{N}}$$

# Transformée de Fourier rapide (FFT)

- (Re-)Découverte majeure en 1965
  - Algorithme initialement trouvé par Gauss en 1805
  - Redécouvert de nombreuses fois au XIXème et XXème siècle
  - Redécouvert en 1965 par Cooley et Tukey
- La transformée de Fourier rapide (FFT : Fast Fourier Transform) est un algorithme de calcul de la transformée de Fourier discrète (TFD).
- complexité :  $\mathcal{O}(n \ln n)$ , alors que la complexité du calcul de base :  $\mathcal{O}(n^2)$
- Le plus utilisé : algorithme de Cooley-Tukey  $\rightarrow$  approche de type "diviser pour régner" (récursion) en utilisant les propriétés des racines n-èmes de l'unité.



## Transformée de Fourier 2D

# Transformée de Fourier 2D

## TF 1D

$$\text{Transformée de Fourier : } \hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} dx$$

$$\text{Transformée inverse : } f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\xi) e^{+i\xi x} d\xi$$

## TF 2D

$$\text{Transformée de Fourier : } \hat{f}(\xi_1, \xi_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-i\xi_1 x} e^{-i\xi_2 y} dx, dy$$

$$\text{Transformée inverse : } f(x, y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\xi_1, \xi_2) e^{+i\xi_1 x} e^{+i\xi_2 y} d\xi_1 d\xi_2$$

- $(x, y)$  coordonnées spatiales,  $(\xi_1, \xi_2)$  coordonnées fréquentielles

# Transformée de Fourier 2D discrète

## TFD 1D

$$S(u) = \sum_{n=0}^{N-1} s(n) \cdot e^{-2i\pi u \frac{n}{N}}$$

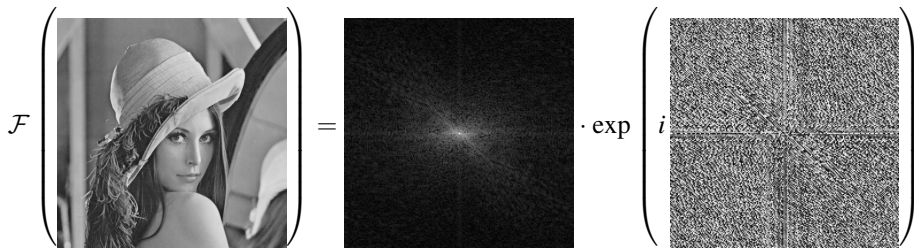
$$s(n) = \frac{1}{N} \sum_{u=0}^{N-1} S(u) \cdot e^{2i\pi n \frac{u}{N}}$$

## TFD 2D

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

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

# TF 2D sur des images

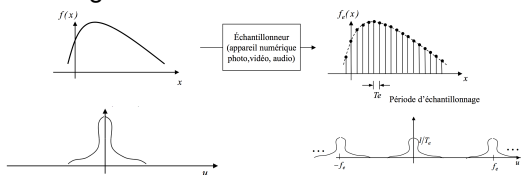


DEMO MATLAB

- On se contente souvent de visualiser le module de la transformée de Fourier
- Puisque les hautes fréquences sont beaucoup plus faibles que les basses fréquences, on utilise souvent une échelle logarithmique pour la visualisation
- On place la fréquence nulle au centre de l'image par "recalage cyclique"

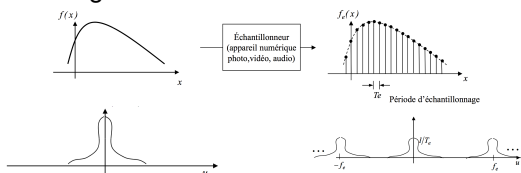
# TF 2D : périodicité et recalage cyclique

- Rappel, échantillonnage et Fourier en 1D :

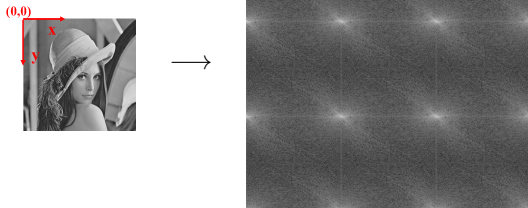


# TF 2D : périodicité et recalage cyclique

- Rappel, échantillonnage et Fourier en 1D :

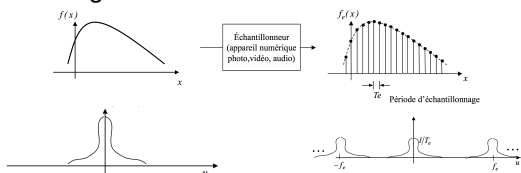


- En 2D, une image est également un signal échantillonné

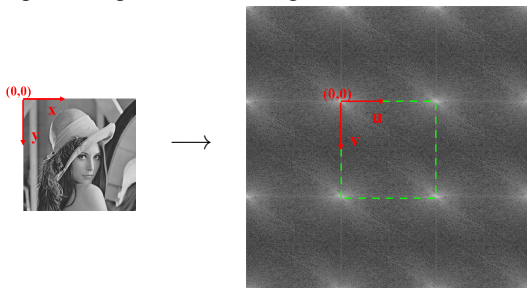


# TF 2D : périodicité et recalage cyclique

- Rappel, échantillonnage et Fourier en 1D :

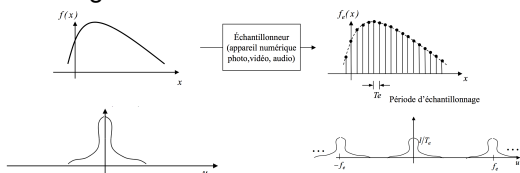


- En 2D, une image est également un signal échantillonné

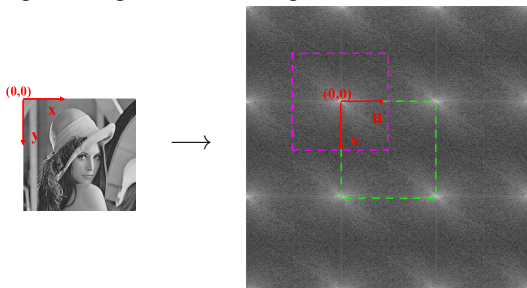


# TF 2D : périodicité et recalage cyclique

- Rappel, échantillonnage et Fourier en 1D :

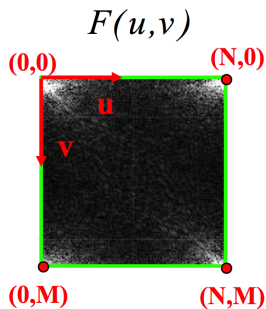


- En 2D, une image est également un signal échantillonné



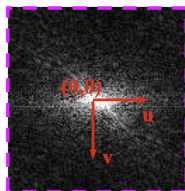


# TF 2D : périodicité et recalage cyclique



$$F(u - N/2, v - M/2)$$

recalage cyclique (fftshift dans matlab)

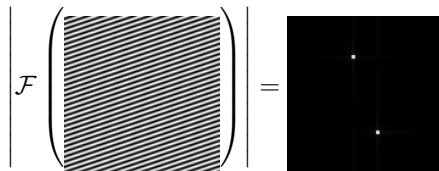
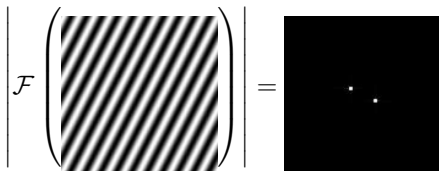


# Interprétation fréquentielle

- On a  $\hat{f}(u, v) = \overline{\hat{f}(-u, -v)}$  ("symétrie" centrale)

# Interprétation fréquentielle

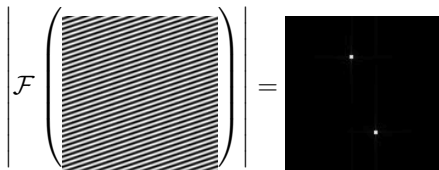
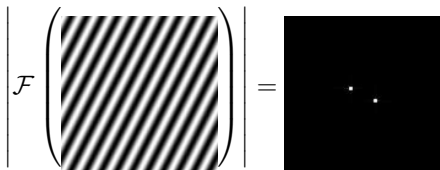
- On a  $\hat{f}(u, v) = \overline{\hat{f}(-u, -v)}$  ("symétrie" centrale)
- $|\hat{f}(u, v)|$  représente l'importance des motifs périodiques de l'image  $f$  dans la direction donnée par le vecteur  $((0, 0) \rightarrow (u, v))$  de fréquence proportionnelle à  $1/(u^2 + v^2)$ .



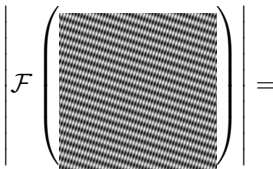
DEMO MATLAB

# Interprétation fréquentielle

- On a  $\hat{f}(u, v) = \overline{\hat{f}(-u, -v)}$  ("symétrie" centrale)
- $|\hat{f}(u, v)|$  représente l'importance des motifs périodiques de l'image  $f$  dans la direction donnée par le vecteur  $((0, 0) \rightarrow (u, v))$  de fréquence proportionnelle à  $1/(u^2 + v^2)$ .

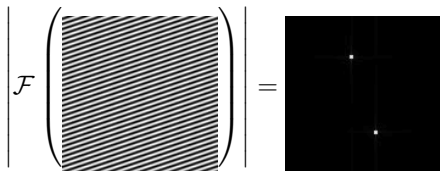
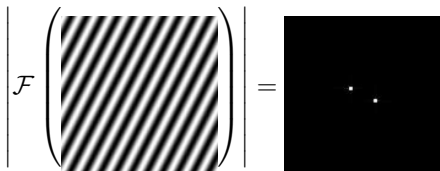


DEMO MATLAB

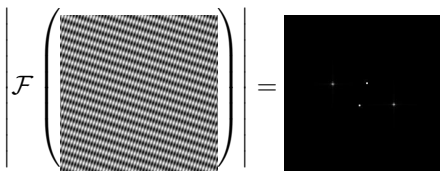


# Interprétation fréquentielle

- On a  $\hat{f}(u, v) = \overline{\hat{f}(-u, -v)}$  ("symétrie" centrale)
- $|\hat{f}(u, v)|$  représente l'importance des motifs périodiques de l'image  $f$  dans la direction donnée par le vecteur  $((0, 0) \rightarrow (u, v))$  de fréquence proportionnelle à  $1/(u^2 + v^2)$ .

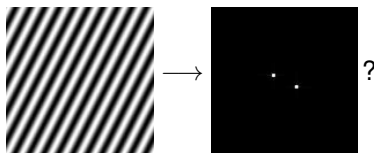


DEMO MATLAB



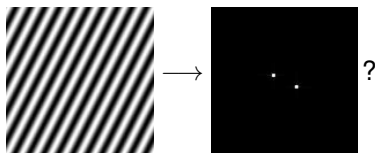
# Interprétation fréquentielle

- moyenne de l'image ?



# Interprétation fréquentielle

- moyenne de l'image ?

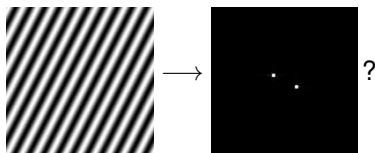


- si l'image n'est pas à moyenne nulle (cas plus réaliste) :

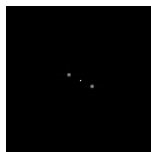


# Interprétation fréquentielle

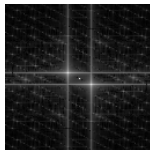
- moyenne de l'image ?



- si l'image n'est pas à moyenne nulle (cas plus réaliste) :



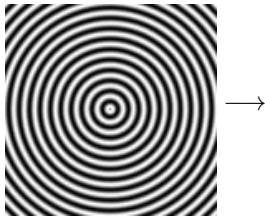
- et en fait (avec échelle logarithmique) :



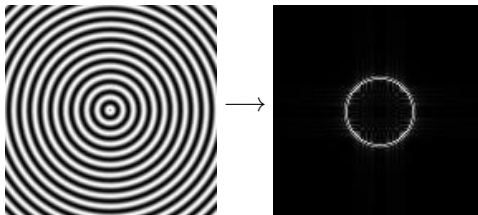
DEMO MATLAB



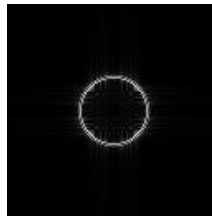
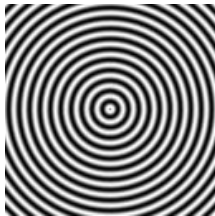
# Interprétation fréquentielle



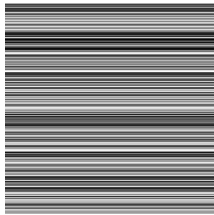
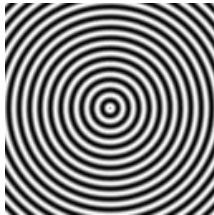
# Interprétation fréquentielle



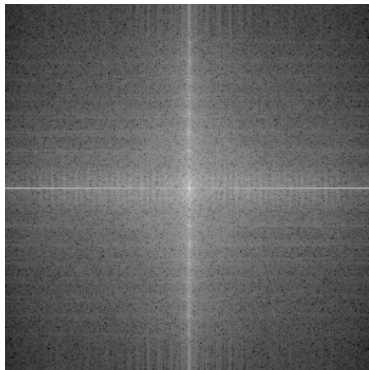
# Interprétation fréquentielle



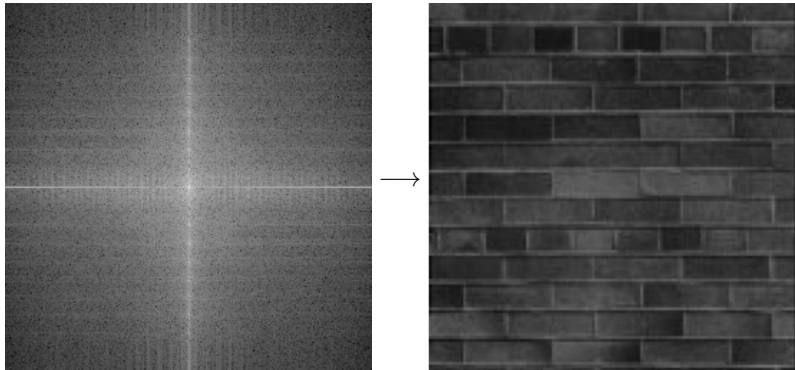
# Interprétation fréquentielle



# Interprétation fréquentielle



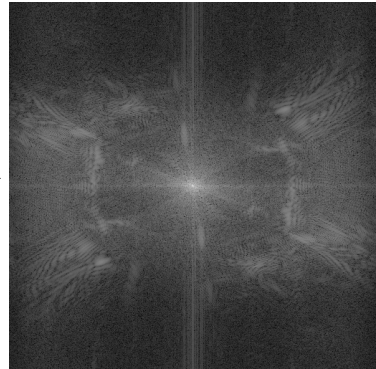
# Interprétation fréquentielle



# Interprétation fréquentielle

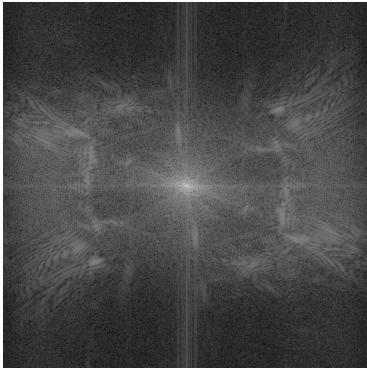


# Interprétation fréquentielle

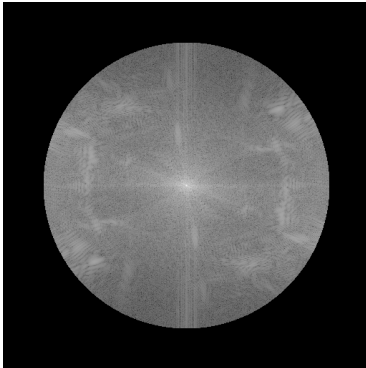




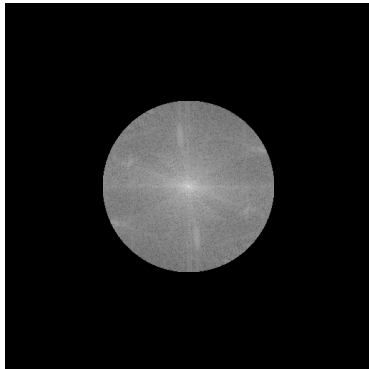
# Interprétation fréquentielle : filtre passe-bas



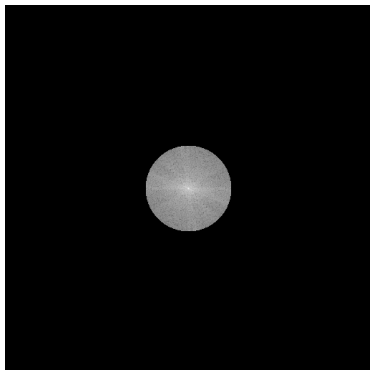
# Interprétation fréquentielle : filtre passe-bas



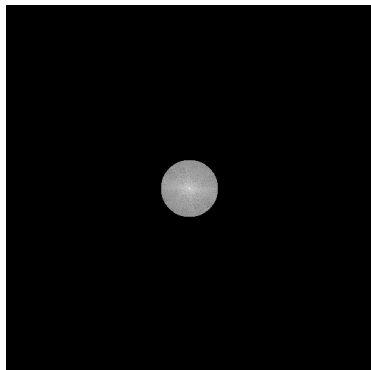
# Interprétation fréquentielle : filtre passe-bas



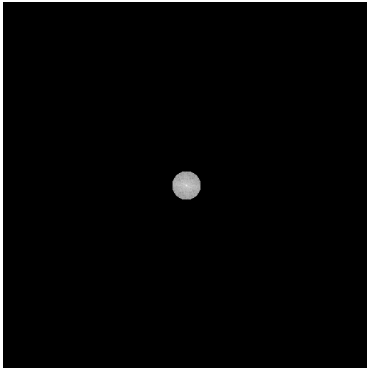
# Interprétation fréquentielle : filtre passe-bas



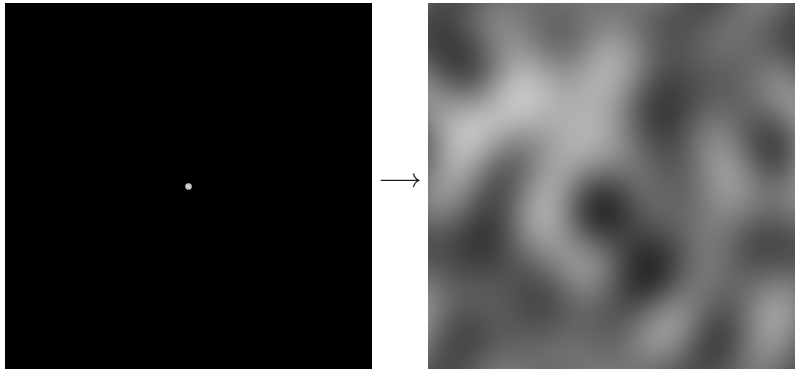
# Interprétation fréquentielle : filtre passe-bas



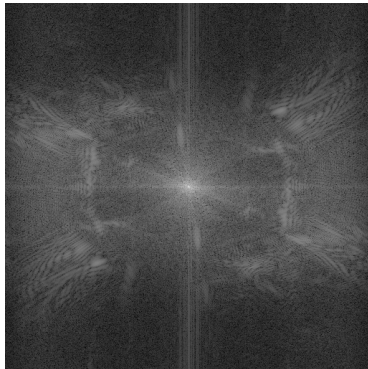
# Interprétation fréquentielle : filtre passe-bas



# Interprétation fréquentielle : filtre passe-bas

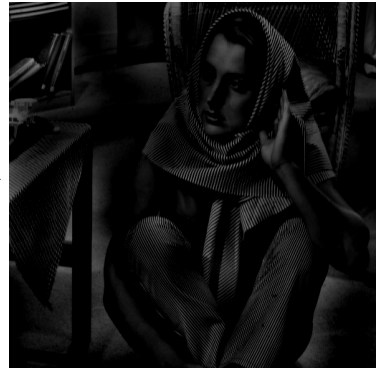
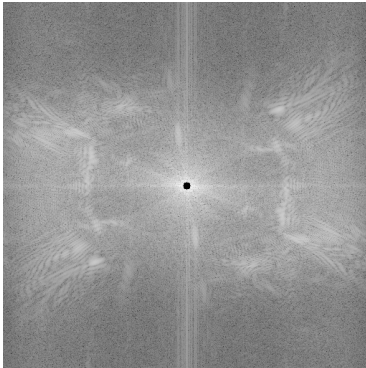


# Interprétation fréquentielle : filtre passe-haut

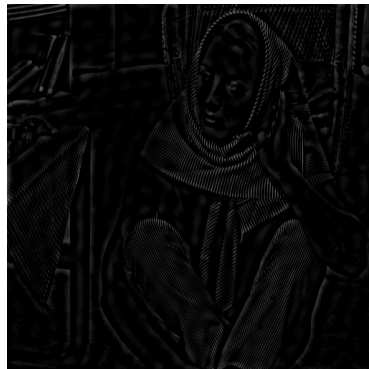
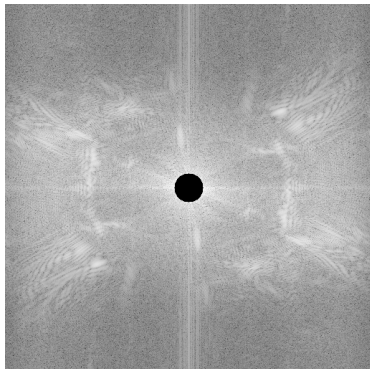




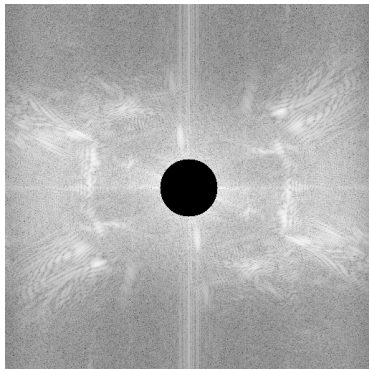
# Interprétation fréquentielle : filtre passe-haut



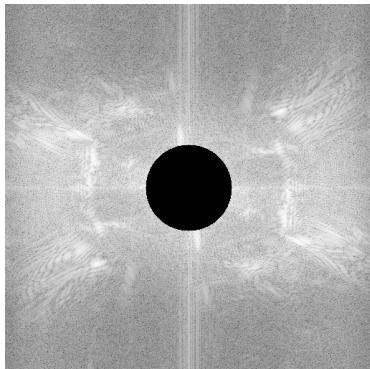
# Interprétation fréquentielle : filtre passe-haut



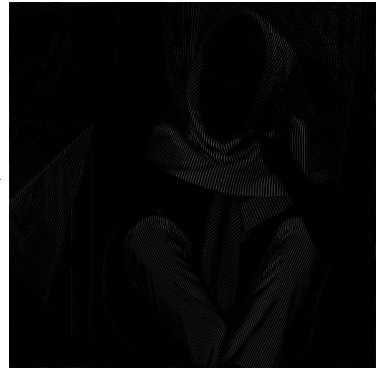
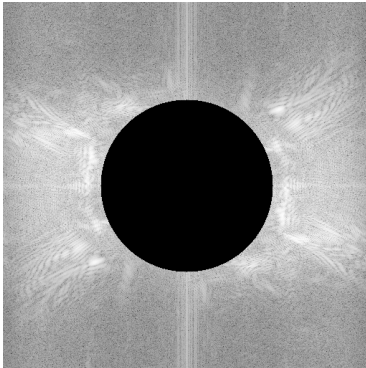
# Interprétation fréquentielle : filtre passe-haut



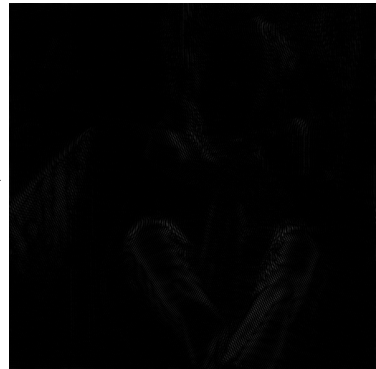
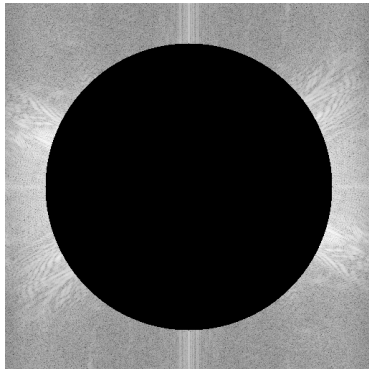
# Interprétation fréquentielle : filtre passe-haut



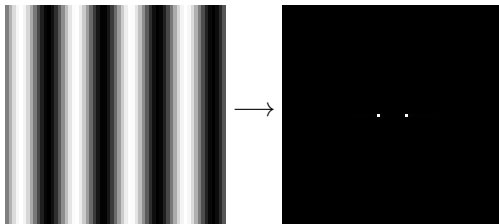
# Interprétation fréquentielle : filtre passe-haut



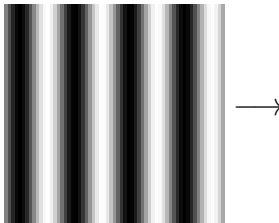
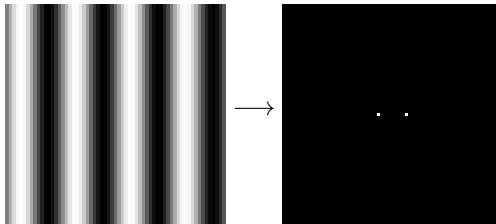
# Interprétation fréquentielle : filtre passe-haut



# Interprétation fréquentielle

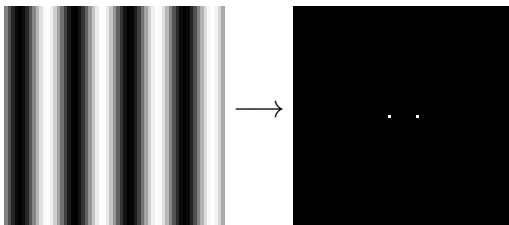
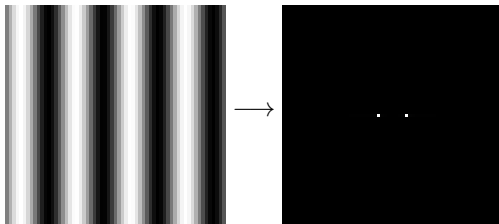


# Interprétation fréquentielle



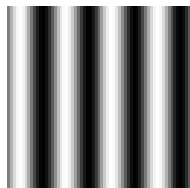


# Interprétation fréquentielle

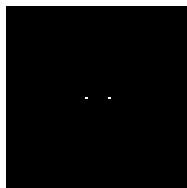


?

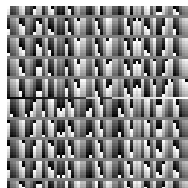
# Importance de la phase



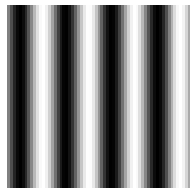
$f_1$



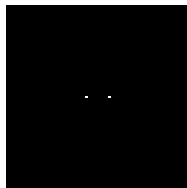
$|\hat{f}_1|$



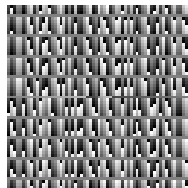
$\arg(\hat{f}_1)$



$f_2$

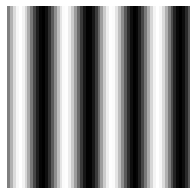


$|\hat{f}_2|$

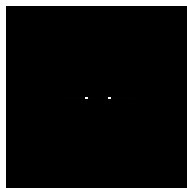


$\arg(\hat{f}_2)$

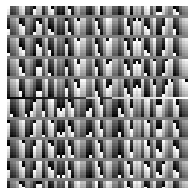
# Importance de la phase



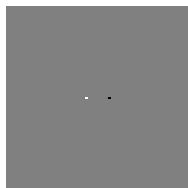
$f_1$



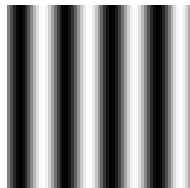
$|\hat{f}_1|$



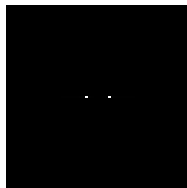
$\arg(\hat{f}_1)$



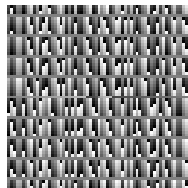
$|\hat{f}_1| \cdot \arg(\hat{f}_1)$



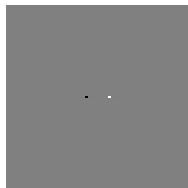
$f_2$



$|\hat{f}_2|$



$\arg(\hat{f}_2)$



$|\hat{f}_2| \cdot \arg(\hat{f}_2)$

On ne visualise souvent que le module car plus facile à interpréter mais phase très importante

# Importance de la phase

$$\mathcal{F} \left( \text{Image of peppers} \right) = \text{Magnitude Spectrum} \cdot \exp(i * \text{Phase Spectrum})$$

The equation shows the forward Fourier transform of a grayscale image of various peppers. The result is the product of the magnitude spectrum (a central bright spot with four symmetric side spots) and the phase spectrum (a complex interference pattern with four quadrants).

$$\mathcal{F}^{-1} \left( \text{Magnitude Spectrum} \cdot \exp(i * \text{Phase Spectrum}) \right) = \text{Image of peppers}$$

The equation shows the inverse Fourier transform of the product of the magnitude spectrum and the phase spectrum. The result is the original grayscale image of the peppers, demonstrating that the phase information is essential for reconstructing the image's structure.

# Importance de la phase

$$\mathcal{F} \left( \text{Image of peppers} \right) = \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right)$$

The equation shows the Fourier transform of a grayscale image of peppers. The left side is the original image. The right side is the product of the magnitude spectrum (a central bright spot with four symmetric dimmer spots) and the phase spectrum (a complex interference pattern).

$$\mathcal{F}^{-1} \left( \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right) \right) = \text{Image of peppers}$$

The equation shows the inverse Fourier transform of the product of the magnitude spectrum and the phase spectrum. The result is the original grayscale image of peppers, demonstrating that the phase information is essential for reconstructing the image's structure.

# Importance de la phase

$$\mathcal{F} \left( \text{Image of peppers} \right) = \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right)$$

The equation shows the forward Fourier transform of a grayscale image of peppers. The result is the product of the magnitude spectrum (a central bright spot with four symmetric side spots) and the phase spectrum (a complex interference pattern).

$$\mathcal{F}^{-1} \left( \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right) \right) = \text{Reconstructed Image of peppers}$$

The equation shows the inverse Fourier transform of the product of the magnitude spectrum and the phase spectrum. The result is a reconstructed grayscale image of the peppers, demonstrating that the phase information is essential for recovering the original image's structure.

# Importance de la phase

$$\mathcal{F} \left( \text{Image of peppers} \right) = \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right)$$

$$\mathcal{F}^{-1} \left( \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right) \right) = \text{Image of peppers}$$

# Importance de la phase

$$\mathcal{F} \left( \text{Image of peppers} \right) = \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right)$$

The equation shows the Fourier transform of a pepper image. On the left is the original image. In the middle is the magnitude spectrum, which is a grayscale image with a bright central peak and four symmetric side peaks. On the right is the phase spectrum, which is a grayscale image with a complex, noisy pattern. The entire right-hand side is enclosed in large parentheses.

$$\mathcal{F}^{-1} \left( \text{Magnitude Spectrum} \cdot \exp \left( i * \text{Phase Spectrum} \right) \right) = \text{Reconstructed Image}$$

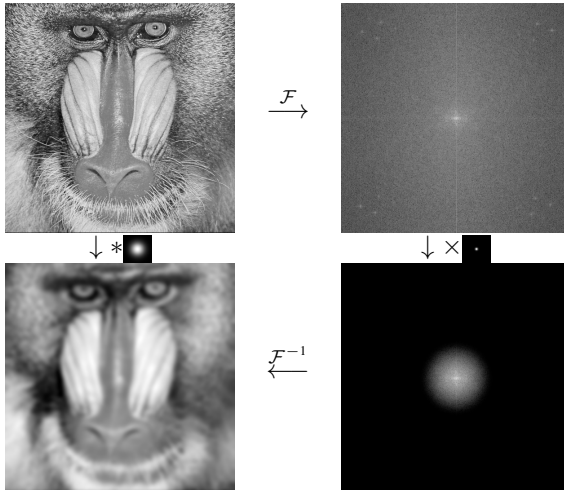
The equation shows the inverse Fourier transform of the magnitude and phase spectra. On the left is the magnitude spectrum and phase spectrum from the previous equation, enclosed in large parentheses. On the right is the reconstructed image, which is a noisy version of the original pepper image.



# Applications : convolution

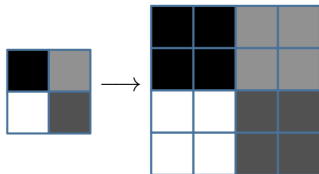
- On a vu que des opérations de filtrage pouvaient se modéliser par une convolution
- Exemple : filtre moyenneur, filtre gaussien
- Une convolution spatiale équivaut à une multiplication dans l'espace de Fourier
- Il peut être plus rapide de
  - 1 calculer la TF de l'image et du filtre
  - 2 effectuer la multiplication fréquentielle
  - 3 calculer la TF inverse du résultat
- efficacité : dépend de la taille des images et des filtre

# Applications : convolution



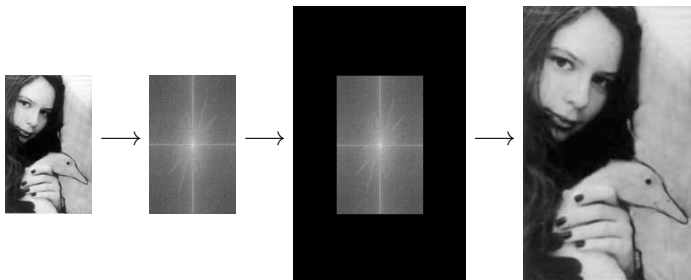
# Applications : zoom par zero-padding

- Une image  $I$  qu'on souhaite agrandir (exemple,  $n \times p \rightarrow 2n \times 2p$ )
- idée naïve, on duplique les pixels :



# Applications : zoom par zero-padding

- autre possibilité : zero-padding



# Applications : zoom par zero-padding

- Comparaison



Duplication des pixels



Zero-padding

# Applications : zoom par zero-padding

- Comparaison



Duplication des pixels

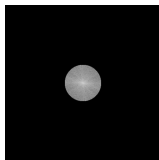


Zero-padding

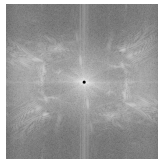


# Applications : compression

- L'oeil humain est moins sensible aux hautes fréquences. On remarquera donc moins une perte en hautes fréquences qu'en basses

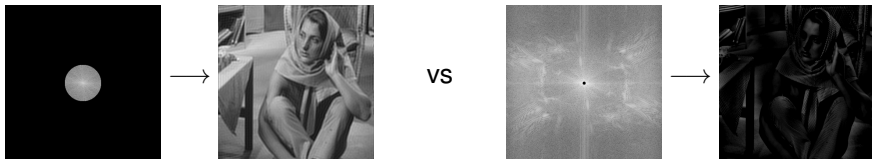


VS



# Applications : compression

- L'oeil humain est moins sensible aux hautes fréquences. On remarquera donc moins une perte en hautes fréquences qu'en basses



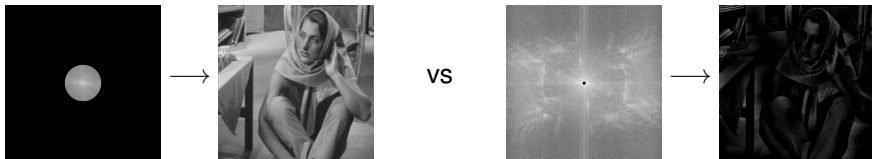
- On peut donc ne conserver que les "basses" fréquences (gain de place / compression) et espérer ne pas perdre trop d'informations à la reconstruction



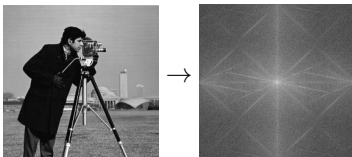


# Applications : compression

- L'oeil humain est moins sensible aux hautes fréquences. On remarquera donc moins une perte en hautes fréquences qu'en basses

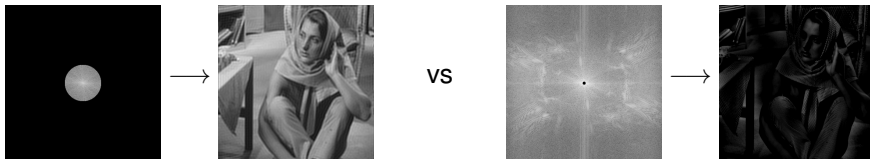


- On peut donc ne conserver que les "basses" fréquences (gain de place / compression) et espérer ne pas perdre trop d'informations à la reconstruction

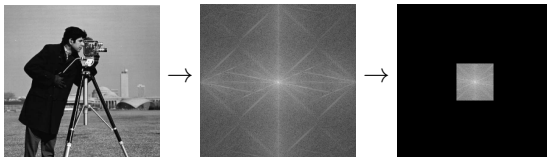


# Applications : compression

- L'oeil humain est moins sensible aux hautes fréquences. On remarquera donc moins une perte en hautes fréquences qu'en basses

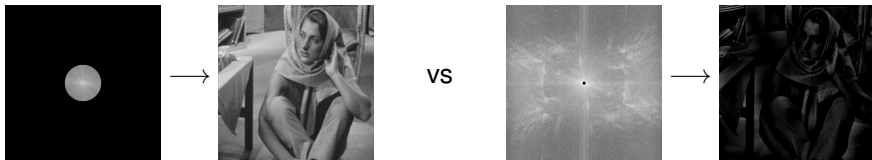


- On peut donc ne conserver que les "basses" fréquences (gain de place / compression) et espérer ne pas perdre trop d'informations à la reconstruction

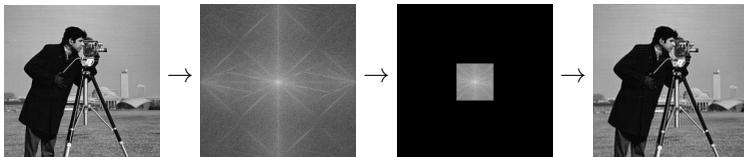


# Applications : compression

- L'oeil humain est moins sensible aux hautes fréquences. On remarquera donc moins une perte en hautes fréquences qu'en basses



- On peut donc ne conserver que les "basses" fréquences (gain de place / compression) et espérer ne pas perdre trop d'informations à la reconstruction



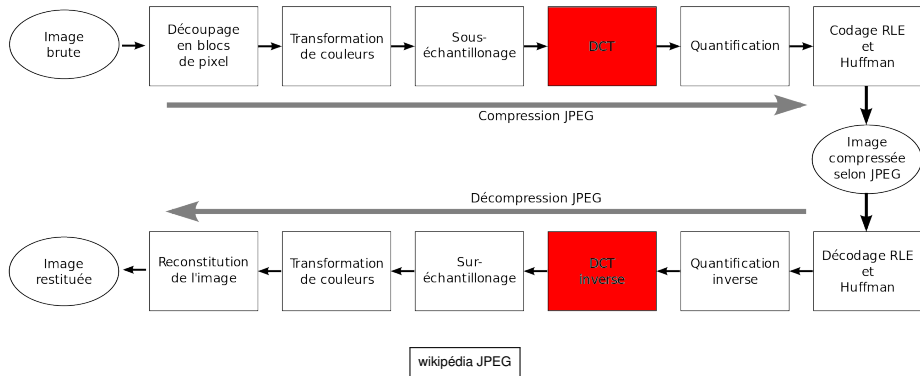
# Applications : compression JPEG

- Principe utilisé par la compression JPEG (sur les sous-blocs de l'image)
- Norme spécifiée en 1991 \* par le Joint Photographic Experts Group résultant de travaux menés dans les années 80.
- Méthode de compression avec pertes
- Permet des taux de compression de 3 à 100 (contre 3 pour des compressions sans pertes)

---

\*. <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>

# Applications : compression JPEG

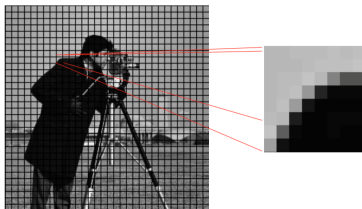


# Applications : compression JPEG

- *Transformation des couleurs* : meilleurs taux de compression avec luminance/chrominance car l'oeil humain est plus sensible à la luminance (luminosité) qu'à la chrominance (teinte)
- *Sous-échantillonnage de la chrominance*

# Applications : compression JPEG

- *Transformation des couleurs* : meilleurs taux de compression avec luminance/chrominance car l'oeil humain est plus sensible à la luminance (luminosité) qu'à la chrominance (teinte)
- *Sous-échantillonnage de la chrominance*
- *Découpage en blocs* : l'image est découpée en blocs de 64 ( $8 \times 8$ ) pixels.



# Applications : compression JPEG

- *Transformée en cosinus discrète (DCT)* : Variante de la transformée de Fourier
  - Transformée DCT directe :

$$\text{DCT}(i,j) = \frac{2}{N} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x,y) \cos \left[ \frac{(2x+1)i\pi}{2N} \right] \cos \left[ \frac{(2y+1)j\pi}{2N} \right]$$

- Transformée DCT inverse :

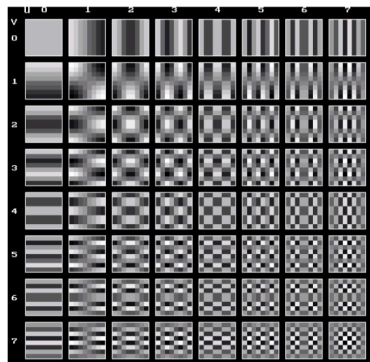
$$\text{pixel}(x,y) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j) \text{DCT}(i,j) \cos \left[ \frac{(2x+1)i\pi}{2N} \right] \cos \left[ \frac{(2y+1)j\pi}{2N} \right]$$



# Applications : compression JPEG

- *Transformée en cosinus discrète (DCT)* : Un bloc  $8 \times 8$  se décompose donc dans une base de 64 imagettes.

- En haut à gauche les basses fréquences
- En bas à droite les hautes fréquences



# Applications : compression JPEG

- *Transformée en cosinus discrète (DCT) :*
  - Avantage de cette base de décomposition : Certaines composantes  $DCT(i,j)$  (hautes fréquences) ont une contribution moins importante (que pour les basses fréquences)
  - Grâce à des techniques similaires à la FFT, on peut calculer la DCT efficacement

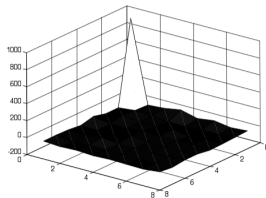
Exemple :



Bloc 8x8



DCT



DCT

# Applications : compression JPEG

- *Quantification* : c'est cette étape qui effectue la compression
  - Principe : "ne garder que les coefficients les plus significatifs" + "atténuer les hautes fréquences, l'oeil humain y est moins sensible."

# Applications : compression JPEG

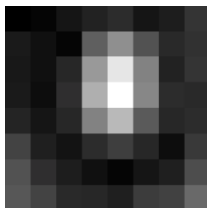
- *Quantification* : c'est cette étape qui effectue la compression
  - Principe : "ne garder que les coefficients les plus significatifs" + "atténuer les hautes fréquences, l'oeil humain y est moins sensible."

- Quantification :  $F^*(u, v) = \text{round} \left( \frac{F(u, v)}{Q(u, v)} \right)$

où  $Q$  est la matrice de quantification (plus faible pour les basses fréquences que pour les grandes)

# Applications : compression JPEG

- Quantification : Exemple †



$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

**DCT**

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

†. <http://www.svcl.ucsd.edu/courses/ece161c/handouts/DCT.pdf>

# Applications : compression JPEG

- Quantification : Exemple

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

DCT		quantized DCT
$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$	→	$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

- Seuls les coefficients non-nuls sont transmis/sauvegardés

# Applications : compression JPEG

- Quantification : Exemple

- Avec  $Q =$ 

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 1 & \text{increasing freque} & & & \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 126 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

**DCT**

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

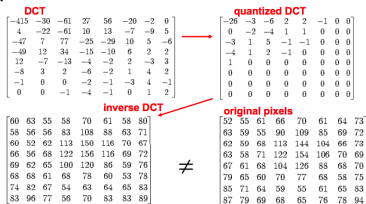
**quantized DCT**

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Seuls les coefficients non-nuls sont transmis/sauvegardés

# Applications : compression JPEG

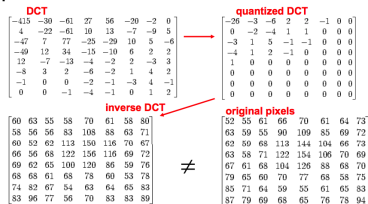
- *Quantification* :
  - Gain de place, mais perte d'informations



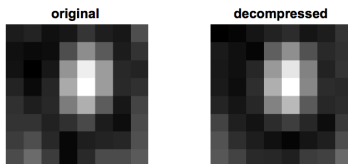


# Applications : compression JPEG

- *Quantification* :
  - Gain de place, mais perte d'informations



- visuellement, pas de grosses différences



- matrice  $Q$  permet de régler le niveau de compression = choisir la perte de qualité. Pas de spécification JPEG mais des recommandations

# Applications : compression JPEG

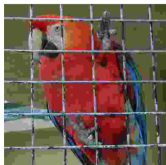
- *Organisation des données*
- *Compression sans pertes* RLE, Huffman

# Applications : compression JPEG

- *Qualité de la compression :*



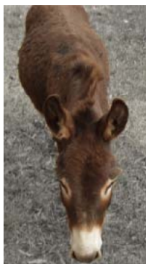
1:41



1:71



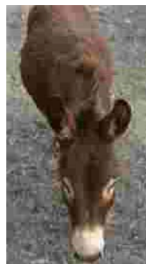
1:108



36KB



5.7KB



1.7KB

# Applications : compression JPEG

- *Qualité de la compression :*



taux de compression diminue de gauche à droite (wikipedia-en)

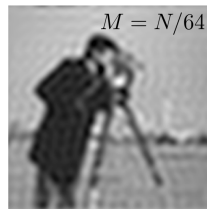
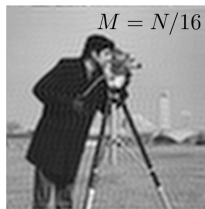
# Applications : compression JPEG

- *Bilan* : Pourquoi ça marche ?
  - propriété mathématique selon la régularité des signaux
  - on opère sur des blocs de petite taille : grandes variations peu probables
  - quantification détruit des détails peu perceptibles pour l'oeil humain
  - l'organisation des données augmente l'efficacité
  
- *Inconvénients*
  - Inadapté aux images contenant du texte
  - effet de "mosaïque" pour de forts taux de compressions (du à la décomposition en blocs)
  - Phénomène de Gibbs autour des contours

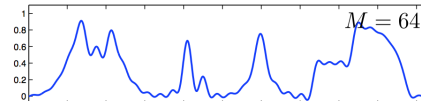
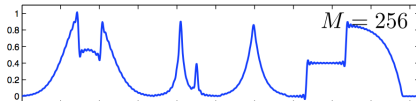
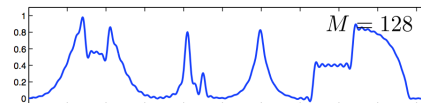
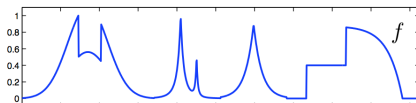
## Au delà de Fourier

# Limites de la transformée de Fourier

- Phénomène de Gibbs : effet de bord à proximité d'une discontinuité



[G.Peyré, Numerical Tours of Signal Processing]



# Limites de la transformée de Fourier

- La transformée de Fourier est **globale**
- Une erreur sur un sous-ensemble en spatial affecte la totalité de la TF
- Une erreur sur un sous-ensemble en fréquence affecte la totalité de l'image en spatial
- Les erreurs de phases sont désastreuses



# Transformée de Fourier locale

- **Transformée de Fourier à court terme (TFCT), ou transformée de Fourier locale** (en anglais Short-Time Fourier Transform (STFT)) ou encore **transformée de Fourier à fenêtre glissante**

$$\text{STFT} \{x(t)\} \equiv X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j\omega t} dt$$

$$\text{STFT} \{x[n]\} \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - m]e^{-j\omega n}$$

- $w$  est la fonction de fenêtrage  $\rightarrow$  non nulle uniquement sur un "petit" intervalle.

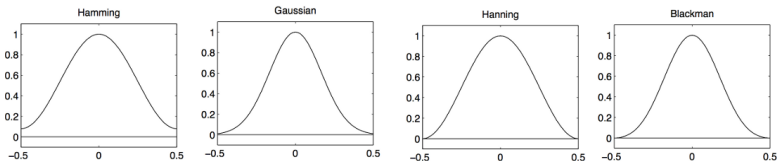
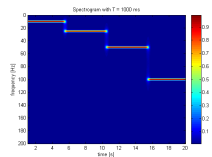
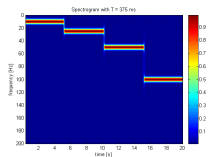
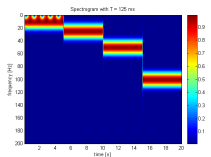
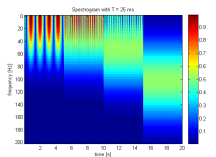


Fig. 4.5. A Wavelet: Tour of Signal Processing, 3<sup>rd</sup> ed. Graphs of four windows  $g$  whose support are  $[-1/2, 1/2]$ .

# Transformée de Fourier locale

- Problème : comment choisir la taille des fenêtres d'études ?
  - si la fenêtre est étroite, on localise bien les changements soudains mais on est aveugle aux basses fréquences
  - si la fenêtre est large, on ne peut pas préciser l'instant où se produit une discontinuité, un pic
- Exemple (wikipédia)

$$x(t) = \begin{cases} \cos(2\pi 10t) & 0 \text{ s} \leq t < 5 \text{ s} \\ \cos(2\pi 25t) & 5 \text{ s} \leq t < 10 \text{ s} \\ \cos(2\pi 50t) & 10 \text{ s} \leq t < 15 \text{ s} \\ \cos(2\pi 100t) & 15 \text{ s} \leq t < 20 \text{ s} \end{cases}$$



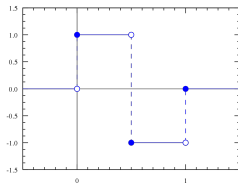
# Transformée en ondelettes

- Sorte de généralisation de la transformée de Fourier locale mais :
  - peut mettre en oeuvre une base différente, non forcément sinusoïdale
  - relation entre la largeur de l'enveloppe et la fréquence des oscillations : homothétie de l'ondelette-mère, et non seulement de l'oscillation.
- Ondelette-mère :  $\Psi$  fonction de carré sommable sur l'espace euclidien  $\mathbb{R} \times \mathbb{R}^+$ , le plus souvent oscillante et de moyenne nulle.
- Famille d'ondelettes définie à partir de l'ondelette-mère :

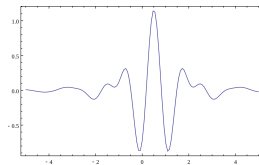
$$\forall t \in \mathbb{R}, \psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \Psi \left( \frac{t - \tau}{s} \right)$$

# Transformée en ondelettes : $\Psi$

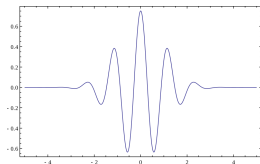
Haar



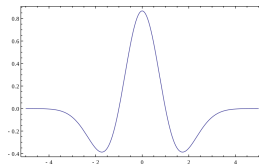
Meyer



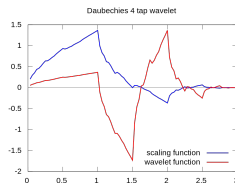
Morlet



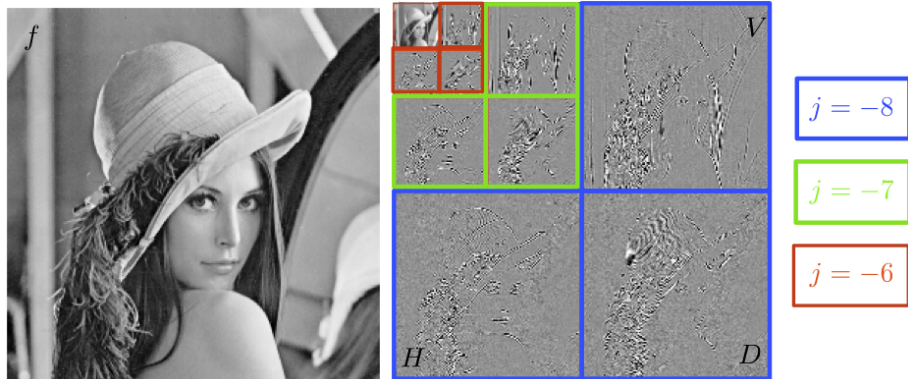
Mexican Hat



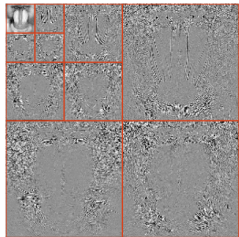
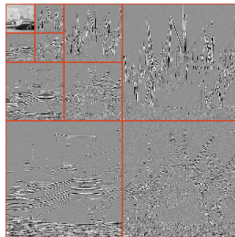
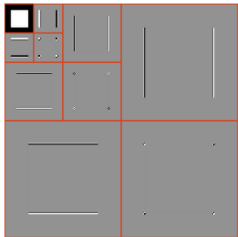
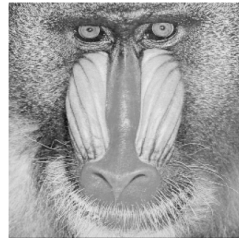
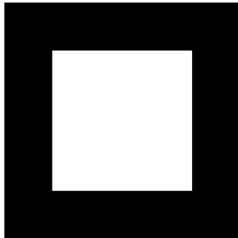
Daubechies



# Transformée en ondelettes de Haar 2D



# Transformée en ondelettes de Haar 2D



# Ondelettes pour compression



Image originale



Compression  
JPEG 2000 0.5 ko



Compression  
JPEG 0.5 ko

# Au delà de JPEG 2000

Les bandelettes...

