

Transformée de Fourier discrète 1D et 2D

Bruno Galerne

`bruno.galerie@univ-orleans.fr`

Université d'Orléans

Modélisation :

Modèles déterministes et stochastiques pour le traitement d'images

Master de Mathématiques Approfondies

Plan du cours

- 1 Transformée de Fourier discrète
- 2 Transformée de Fourier discrète des images numériques
- 3 TFD 2D et transformations géométriques des images
- 4 Composante périodique d'une image

Outline

- 1 Transformée de Fourier discrète
- 2 Transformée de Fourier discrète des images numériques
- 3 TFD 2D et transformations géométriques des images
- 4 Composante périodique d'une image

Signaux discrets : Cadre et notation

- Soit $N \in \mathbb{N}^*$ un entier naturel non nul que l'on supposera être pair.
- On se place sur l'espace vectoriel \mathbb{C}^N des vecteurs $u = (u_0, u_1, \dots, u_{N-1})$ à coefficients complexes à N coefficients **dont les indices sont numérotés de 0 à $N - 1$** .
- Notation des coordonnées : $u = (u_0, u_1, \dots, u_{N-1})$ ou notation fonctionnelle $u = (u(0), u(1), \dots, u(N - 1))$.
- On notera $\Omega_N = \{0, 1, \dots, N - 1\}$ les valeurs des indices, *i.e.* le domaine spatial.

Signaux discrets : Périodicité

- Un vecteur $u \in \mathbb{C}^N$ est étendu sur \mathbb{Z} par N -périodicité, c'est-à-dire :

$$\forall n \in \mathbb{Z}, \quad u(n) = u(n \bmod N)$$

- On verra que c'est naturel vu la définition de la transformée de Fourier discrète...

Signaux discrets : Echantillonnage de fonctions périodiques

- Etant donné une fonction $f : \mathbb{R} \rightarrow \mathbb{C}$ 2π -périodique, on définit le vecteur échantillonné à N points de f par

$$\forall n \in \Omega_N, \quad u(n) = f\left(\frac{2\pi n}{N}\right).$$

- C'est compatible avec la N -périodicité des signaux :

$$f \text{ } 2\pi\text{-périodique} \implies u \text{ } N\text{-périodique}$$

Transformée de Fourier discrète : Définition

Définition (Transformée de Fourier discrète)

La transformée de Fourier discrète (TFD) est l'application linéaire

$$\begin{aligned} \mathcal{F} : \mathbb{C}^N &\rightarrow \mathbb{C}^N \\ u &\mapsto \hat{u} = (\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1}) \end{aligned}$$

où pour tout $k \in \Omega_N$,

$$\hat{u}_k = \frac{1}{N} \sum_{n=0}^{N-1} u_n e^{-\frac{2ikn\pi}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} u_n \omega_N^{-kn}, \quad (1)$$

en posant

$$\omega_N = e^{\frac{2i\pi}{N}}.$$

- On remarque que \hat{u}_0 n'est autre que la moyenne de u .

Transformée de Fourier discrète inverse

Proposition

La TFD $\mathcal{F} : u \mapsto \hat{u}$ est un isomorphisme de \mathbb{C}^N . La matrice de la TFD dans la base canonique est égale à

$$A = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N^{-1} & \omega_N^{-2} & \dots & \omega_N^{-(N-1)} \\ 1 & \omega_N^{-2} & \omega_N^{-4} & \dots & \omega_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{-(N-1)} & \omega_N^{-2(N-1)} & \dots & \omega_N^{-(N-1)(N-1)} \end{pmatrix} = \frac{1}{N} (\omega_N^{-kl})_{0 \leq k, l \leq N-1}$$

Elle est donc proportionnelle à la matrice de Vandermonde associée au vecteur $(1, \omega_N^{-1}, \omega_N^{-2}, \dots, \omega_N^{-(N-1)})$. Cette matrice est inversible et

$$A^{-1} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N^1 & \omega_N^2 & \dots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \dots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \dots & \omega_N^{(N-1)(N-1)} \end{pmatrix} = (\omega_N^{kl})_{0 \leq k, l \leq N-1}.$$

Transformée de Fourier discrète inverse

Definition (TFD inverse)

La transformée de Fourier discrète inverse (TFD inverse) est l'application linéaire

$$\begin{aligned} \mathcal{F}^{-1} : \mathbb{C}^N &\rightarrow \mathbb{C}^N \\ u &\mapsto \mathcal{F}^{-1}(u) = \check{u} = (\check{u}_0, \check{u}_1, \dots, \check{u}_{N-1}) \end{aligned}$$

où pour tout $k \in \Omega_N$,

$$\check{u}_k = \sum_{n=0}^{N-1} u_n \omega_N^{kn}.$$

Remarques

- La TFD et son inverse sont très proches. En effet, on a pour tout $u \in \mathbb{C}^N$,

$$\mathcal{F}^{-1}(u) = N \overline{\mathcal{F}(\overline{u})}.$$

- Il n'y a d'ailleurs pas de convention universelle pour la distinction entre le TFD et la TFD inverse (ou encore savoir quelle transformée est divisée par N). Cette ambiguïté se retrouve dans les différents algorithmes calculant la DFT.

TP Transformée de Fourier discrète

- En python, les fonctions relatives à la fft sont dans `numpy.fft`.
 - FFT = Fast Fourier transform [Cooley & Tukey, 1965]
 - La TFD se calcule par la commande `numpy.fft.fft`, son inverse par `numpy.fft.ifft`.
 - **TP: Questions:**
- 1 Consulter la page : <https://docs.scipy.org/doc/numpy/reference/routines.fft.html>
 - 2 Vérifier que `ifft` est bien l'inverse de `fft` (attention ce n'est pas toujours le cas dans certaines librairies !).
 - 3 Est-ce la même définition que dans le cours ?

Module et phase

Vocabulaire :

- Le vecteur $k \mapsto |\hat{u}_k|$ est appelé le **module** de la TFD de u , et noté $|\hat{u}|$.
On parle aussi du module de u par abus de langage.
- Le vecteur $k \mapsto \arg(\hat{u}_k)$ est appelé la **phase** de la TFD de u .

Proposition

Si $u \in \mathbb{R}^N$ est un vecteur à coefficients réels alors \hat{u}_0 et $\hat{u}_{-\frac{N}{2}} = \hat{u}_{\frac{N}{2}}$ sont réels et pour tous $k \in \Omega_N \setminus \{0, \frac{N}{2}\}$, $\hat{u}_{-k} = \overline{\hat{u}_k}$.

En particulier, le module de u est une fonction paire et la phase de u est une fonction impaire.

Etant donné ces conditions de symétries, il est d'usage de travailler avec un domaine d'indice centré autour de 0 pour les indices des vecteur de la TFD. On distinguera donc le domaine spatial $\Omega_N = \{0, 1, \dots, N-1\}$ et le domaine spectral

$$\hat{\Omega}_N = \left\{ -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} - 1 \right\}$$

(N indices dont la fréquence nulle est au centre).

TP Transformée de Fourier discrète

On considère la fonction f 2π -périodique égale à l'indicatrice de l'intervalle $[\frac{\pi}{2}, \frac{3\pi}{5}]$ sur $[0, 2\pi[$.

Questions :

- 1 Définir la fonction f .
- 2 Calculer le vecteur échantillonné u de f pour $N = 256$.
- 3 Afficher le graphe de u ainsi que le graphe du module de u et celui de la phase de u . Que remarque-t-on ?
- 4 Recalculer la DFT de u à l'aide de `tfu = np.fft.fftshift(np.fft.fft(u))` et afficher à nouveau le graphe du module de u .
- 5 Afficher à nouveau le module de u en précisant les bonnes valeurs sur l'axe des abscisses.

Polynômes trigonométriques

Definition (Polynômes trigonométriques)

Un polynôme trigonométrique de degré inférieur ou égal à N est une fonction $p : \mathbb{R} \rightarrow \mathbb{C}$ de la forme

$$p(x) = \sum_{k=-N}^N c_k e^{ikx}.$$

où les c_k sont des nombres complexes appelés coefficients du polynôme trigonométrique p . On dit que p est de degré N si de plus $c_{-N} \neq 0$ ou $c_N \neq 0$.

Rappel : Formule de Fourier

- Les coefficients d'un polynôme trigonométrique sont donnés sous forme intégrale par ses coefficients de Fourier:

$$\forall k \in \mathbb{Z}, \quad c_k = c_k(p) = \frac{1}{2\pi} \int_0^{2\pi} p(x) e^{-ikx} dx.$$

(où on étend la définition des c_k à tout \mathbb{Z} par des 0)

Polynômes trigonométriques et TFD

Proposition

Soit $u \in \mathbb{C}^N$. Il existe un unique polynôme trigonométrique de la forme

$$p(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} c_k e^{ikx}$$

et dont les échantillons de p valent u , c'est-à-dire: $\forall n \in \Omega_N, p\left(\frac{2n\pi}{N}\right) = u_n$.

Les coefficients de ce polynôme sont donnés par la TFD de u : $\forall k \in \hat{\Omega}_N, c_k = \hat{u}_k$.

Polynômes trigonométriques et TFD

Corollaire (Calcul des coefficients d'un polynôme)

Si p est un polynôme trigonométrique de la forme

$$p(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} c_k e^{ikx},$$

alors les coefficients de p sont exactement calculables à partir des N échantillons $u_n = p\left(\frac{2n\pi}{N}\right)$.

- On a donc montré que

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} p(x) e^{-ikx} dx = \frac{1}{N} \sum_{n=0}^{N-1} p\left(\frac{2n\pi}{N}\right) e^{-\frac{2ikn\pi}{N}}.$$

- Interprétation : Égalité entre l'intégrale sur le domaine continu $[0, 2\pi]$ de la fonction $x \mapsto \frac{1}{2\pi} p(x) e^{-ikx}$ et la somme de Riemann de cette fonction associée à la subdivision régulière $\left\{ \frac{2n\pi}{N}, n \in \{0, \dots, N-1\} \right\}$.

Interpolation par un polynôme trigonométrique

- Soit $u \in \mathbb{C}^N$. Alors, $p(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k e^{ikx}$ est l'unique polynôme

trigonométrique de cette forme tel que $u_n = p\left(\frac{2n\pi}{N}\right)$.

- Si u est réel, alors $\text{Re}(p(x)) = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} \hat{u}_k e^{ikx} + c_{-\frac{N}{2}} \cos\left(\frac{Nx}{2}\right)$ est un

polynôme trigonométrique à valeurs réelles de même degré (mais un coefficient de plus) qui interpole u .

TP/TD:

- 1 Créer un signal (aléatoire ou autre) de taille $N = 16$.
- 2 Définir une fonction `interp_trigo(u, x)` qui évalue $\text{Re}(p(x))$ et vérifier sur un graphe que l'on a bien une interpolation.
- 3 On considère un entier $r \geq 2$, et on pose on définit $v \in \mathbb{R}^{rN}$ le rN -échantillonné de p :

$$v_n = p\left(\frac{2\pi n}{rN}\right).$$

Exprimer la TFD de $v \in \mathbb{R}^{rN}$ en fonction de la TFD de $u \in \mathbb{R}^N$.

- 4 En déduire un algorithme pour évaluer v .

Séries de Fourier, TFD et échantillonnage

- On a vu que la TFD permet de calculer les coefficients d'un polynôme trigonométrique à partir de N valeurs:

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} p(x) e^{-ikx} dx = \frac{1}{N} \sum_{n=0}^{N-1} p\left(\frac{2n\pi}{N}\right) e^{-\frac{2ikn\pi}{N}}.$$

- S'étend à toutes les fonctions 2π -périodiques ?

Proposition

Soit f une fonction 2π -périodique intégrable sur $[0, 2\pi]$, $N \in \mathbb{N}^*$, et $u \in \mathbb{C}^N$ le vecteur échantillonné de f , c'est-à-dire le vecteur u tel que

$\forall n \in \Omega_N$, $u_n = f\left(\frac{2n\pi}{N}\right)$. Alors, pour tout $k \in \hat{\Omega}_N$, le coefficient \hat{u}_k d'indice k de la TFD est l'approximation par la méthode des trapèzes du coefficient de Fourier $c_k(f)$.

- Vu la périodicité, on peut voir l'approximation comme la formule du point milieu sur l'intervalle $[-\frac{1}{2N}, 2\pi - \frac{1}{2N}]$ donc l'erreur est de l'ordre de $\frac{1}{N^2}$ pour une fonction suffisamment régulière.
- En fait on peut relier explicitement la TFD de l'échantillonné u de f avec les coefficients de Fourier de f ...

Séries de Fourier, TFD et échantillonnage

Proposition

Soit f une fonction 2π -périodique dont la série de Fourier converge normalement, i.e. $\sum_{n \in \mathbb{Z}} |c_n(f)| < +\infty$, et $u \in \mathbb{C}^N$ le vecteur échantillonné de f . Alors la TFD $(\hat{u}_k)_{k \in \hat{\Omega}_N}$ de u est la N -périodisé de la suite de coefficients de Fourier

$$\forall k \in \hat{\Omega}_N, \quad \hat{u}_k = \sum_{q \in \mathbb{Z}} c_{k+qN}(f).$$

Cette proposition permet d'expliciter le **phénomène de repliement du spectre (aliasing)** lorsque l'on sous-échantillonne un signal.

Proposition

Soit $u \in \mathbb{C}^N$ et soit $v \in \mathbb{C}^{\frac{N}{2}}$ le vecteur sous-échantillonné de u par un facteur 2, i.e.

$$\forall n \in \Omega_{\frac{N}{2}}, \quad v(n) = u(2n).$$

Alors la TFD de v est donnée par

$$\forall k \in \hat{\Omega}_{\frac{N}{2}} = \left\{ -\frac{N}{4}, \dots, \frac{N}{4} - 1 \right\}, \quad \hat{v}(k) = \begin{cases} \hat{u}(k) + \hat{u}(k + \frac{N}{2}) & \text{si } k \leq 0 \\ \hat{u}(k) + \hat{u}(k - \frac{N}{2}) & \text{si } k > 0 \end{cases}$$

Séries de Fourier, TFD et échantillonnage

Phénomène de repliement du spectre (aliasing) : Il faut enlever les hautes fréquences avant de sous-échantillonner (zoom arrière pour les images), sinon les hautes fréquences induisent de faux signaux basse fréquence.

Convolution

Definition (Convolution périodique)

Le produit de convolution entre deux vecteurs u et v de \mathbb{C}^N est le vecteur $u * v \in \mathbb{C}^N$ défini pour tout $n \in \Omega_N$ par

$$(u * v)_n = \sum_{m=0}^{N-1} u_m v_{n-m}$$

Proposition (Convolution périodique et TFD)

Soient u et $v \in \mathbb{C}^N$. Alors pour tout $k \in \hat{\Omega}_N$,

$$(\widehat{u * v})_k = N \hat{u}_k \hat{v}_k.$$

Cette proposition vaut pour algorithme. Pour calculer la convolution $u * v$, on effectue les opérations suivantes:

- 1 Calcul de \hat{u} et \hat{v} (par transformée de Fourier rapide (FFT)).
- 2 Calcul du produit coordonnées à coordonnées $N \hat{u} \cdot \hat{v} = \widehat{u * v}$.
- 3 Calcul de $u * v$ par TFD inverse.

Ce passage par le domaine spectral n'a d'intérêt que parce que la TFD peut être calculée de manière rapide en $O(N \log(N))$ opérations.

Outline

- 1 Transformée de Fourier discrète
- 2 Transformée de Fourier discrète des images numériques**
- 3 TFD 2D et transformations géométriques des images
- 4 Composante périodique d'une image

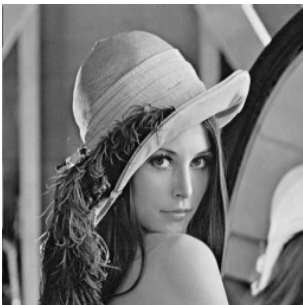
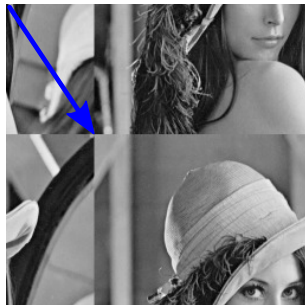
Images numériques : Cadre

- On considère des images numériques $u \in \mathbb{R}^{M \times N}$ dont les indices sont dans $\Omega_{M,N} = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$.
- Chaque image est étendue par périodicité

$$u(k, l) = u(m \bmod M, n \bmod N) \quad \forall (m, n) \in \mathbb{Z}^2.$$

- Conséquence: Translation d'une image : $\forall \tau \in \mathbb{Z}^2$,

$$\theta_{\tau} u(m, n) = u(m - \tau_1, n - \tau_2).$$


 u

 $\theta_{\tau} u$

Transformée de Fourier discrète des images numériques

- Domaine spatial : $\Omega_{M,N} = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$
- Domaine spectral : $\hat{\Omega}_{M,N}$: la fréquence nulle (0, 0) est placée au centre

$$\hat{\Omega}_{M,N} = \left\{ -\frac{M}{2}, \dots, \frac{M}{2} - 1 \right\} \times \left\{ -\frac{N}{2}, \dots, \frac{N}{2} - 1 \right\}.$$

(correspond aux vraies fréquences du polynôme trigonométrique qui interpole u ...)

Définitions:

- La **transformée de Fourier discrète (TFD)** de u est l'image à **valeurs complexes** image \hat{u} définie par :

$$\forall (k, l) \in \hat{\Omega}_{M,N}, \quad \hat{u}(k, l) = \hat{u}_{k,l} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} u_{m,n} \omega_M^{-km} \omega_N^{-ln}.$$

- $|\hat{u}|$: **Module de Fourier** de u .
- $\arg(\hat{u})$: **Phase de Fourier** de u .
- La **transformée de Fourier discrète (TFD) inverse** de u est l'image \check{u} définie par :

$$\forall (m, n) \in \Omega_{M,N}, \quad \check{u}_{m,n} = \sum_{k=-\frac{M}{2}}^{\frac{M}{2}-1} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_{k,l} \omega_M^{km} \omega_N^{ln}.$$

Transformée de Fourier discrète des images numériques

Propriétés de symétrie

- Si $u \in \mathbb{R}^{M \times N}$ alors $\hat{u}(-k, -l) = \overline{\hat{u}(k, l)}$.
- $|\hat{u}|$: Le **module de Fourier** de u est **paire**.
- $\arg(\hat{u})$: Le **phase de Fourier** de u est **impaire**.

Visualisation de la TFD:

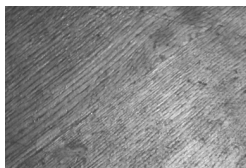
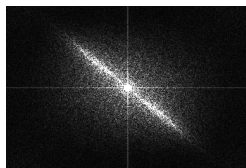


Image u



Module $|\hat{u}|$



Phase $\arg(\hat{u})$

Calcul :

- La transformée de Fourier rapide (*Fast Fourier Transform (FFT)*) calcule \hat{u} en $\mathcal{O}(MN \log(MN))$ opérations (en non $\mathcal{O}((MN)^2)$).
- Commandes python: `np.fft.fft2` ou plutôt `np.fft.fftshift(np.fft.fft2(u))`

Convolution 2D

Definition (Convolution 2D)

Le produit de convolution entre deux images u et v de $\mathbb{C}^{M \times N}$ est l'image $u * v \in \mathbb{C}^{M \times N}$ définie pour tout $(m, n) \in \Omega_{M, N}$ par

$$(u * v)_{m, n} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} u_{p, q} v_{m-p, n-q}.$$

En dimension 2 la TFD transforme encore la produit de convolution en produit coordonnées par coordonnées.

Proposition (Convolution périodique et TFD 2D)

Soient u et v deux images de $\mathbb{C}^{M \times N}$. Alors, pour tout $(k, l) \in \hat{\Omega}_{M, N}$,

$$(\widehat{u * v})_{k, l} = MN \hat{u}_{k, l} \hat{v}_{k, l}.$$

Convolution 2D

- Tout opérateur linéaire invariant par translation est une convolution.

Nombreux exemples :

- Flou avec un noyau quelconque (moyenne locale)
- Gradient discret, gradient régularisé,...
- Corrélation entre images

TP convolution et flou gaussien

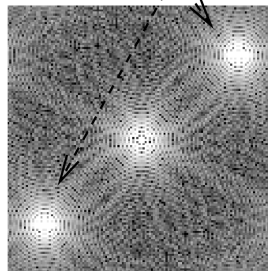
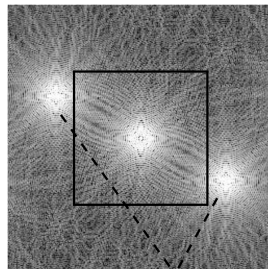
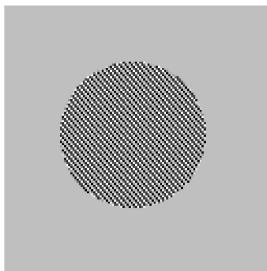
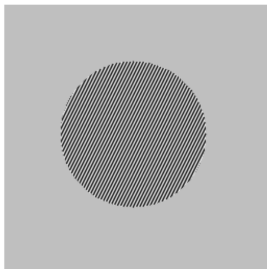
```
def convolfft2(u,v):
    tfu = np.fft.fft2(u)
    tfv = np.fft.fft2(v)
    tfw = tfu*tfv
    w = np.real(np.fft.ifft2(tfw))
    return(w)

def gaussianblur(u, sigma):
    s = int(np.ceil(4*sigma))
    kernel = np.zeros(2*s+1)
    for t in np.arange(-s,s+1):
        kernel[t] = np.exp(-t**2/(2*sigma**2))
    kernel /= sum(kernel)
    gausskernel = np.zeros(u.shape)
    for t1 in np.arange(-s,s+1):
        for t2 in np.arange(-s,s+1):
            gausskernel[t1,t2] = kernel[t1]*kernel[t2]
    return(convolfft2(u,gausskernel))
```

- 1 Tester cette fonction sur une image en variant σ .
- 2 Déterminer l'image u telle que `gaussianblur(u, sigma)` affiche le noyau gaussien au centre de l'image.

TFD et échantillonnage

Problème de l'aliasing :



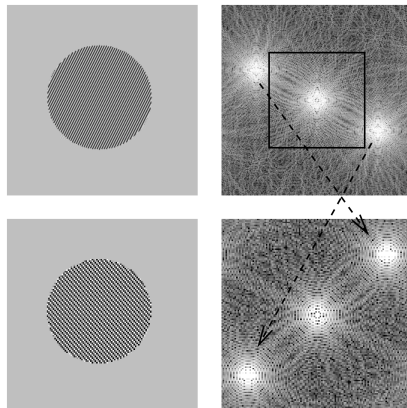
TFD et échantillonnage

Problème de l'aliasing :



TP TFD et échantillonnage

Créer une image permettant de reproduire cette expérience. Vérifier que la nouvelle fréquence obtenue s'obtient par un décalage de $\frac{N}{2}$.



Module et phase d'une image numérique

Que représentent le module et la phase d'une image ?

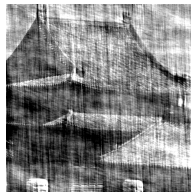
Module et phase d'une image numérique

Echange de la phase et du module de deux images

Image 1



Image 2

Module de 1
& phase de 2Module de 2
& phase de 1

- Les contours géométriques sont principalement contenus dans la phase.

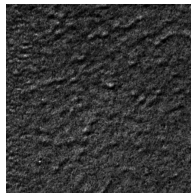
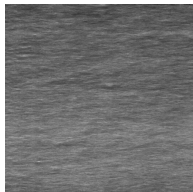
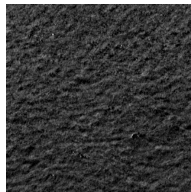
Module et phase d'une image numérique

Echange de la phase et du module de deux images

Image 1



Image 2

Module de 1
& phase de 2Module de 2
& phase de 1

- Les textures sont principalement contenus dans le module.

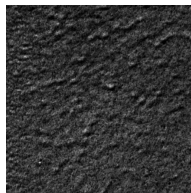
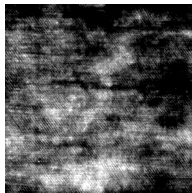
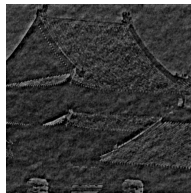
Module et phase d'une image numérique

Echange de la phase et du module de deux images

Image 1



Image 2

Module de 1
& phase de 2Module de 2
& phase de 1

- Les contours géométriques sont principalement contenus dans la phase.
- Les textures sont principalement contenus dans le module.

Outline

- 1 Transformée de Fourier discrète
- 2 Transformée de Fourier discrète des images numériques
- 3 TFD 2D et transformations géométriques des images**
- 4 Composante périodique d'une image

Rappel : TFD et polynôme trigonométrique interpolateur

La **transformée de Fourier discrète (TFD)** de u est l'image à **valeurs complexes** image \hat{u} définie par :

$$\forall (k, l) \in \hat{\Omega}_{M,N}, \quad \hat{u}(k, l) = \hat{u}_{k,l} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} u_{m,n} e^{-\frac{2i\pi km}{M}} e^{-\frac{2i\pi ln}{N}}.$$

Proposition

Soit $u \in \mathbb{C}^M \times N$. Il existe un unique polynôme trigonométrique de la forme

$$p(x, y) = \sum_{k=-\frac{N}{2}}^{\frac{M}{2}-1} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} c(k, l) e^{ikx} e^{ily}$$

et dont les échantillons de p valent u , c'est-à-dire:

$$\forall (m, n) \in \Omega_{M,N}, \quad p\left(\frac{2m\pi}{N}, \frac{2n\pi}{N}\right) = u(m, n).$$

Les coefficients de ce polynôme sont donnés par la TFD de u :

$$\forall (k, l) \in \hat{\Omega}_{M,N}, \quad c(k, l) = \hat{u}(k, l).$$

Rappel : TFD et polynôme trigonométrique interpolateur

- En résumé,

$$p(x, y) = \sum_{k=-\frac{N}{2}}^{\frac{M}{2}-1} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}(k, l) e^{ikx} e^{ily}$$

est un polynôme trigonométrique interpolateur de u .

- Attention, même si $u \in \mathbb{R}^{M \times N}$ est réelle p peut prendre des valeurs complexes à cause des fréquences extrêmes qui n'ont pas de symétrie. Mais alors $\text{Re}(p)$ est aussi un polynôme trigonométrique interpolateur.
- Evaluer p en un point (x, y) coûte a priori de $O(MN)$ opérations.
- Mais **on peut évaluer p sur toute une grille grâce avec une seule FFT**, soit $O(MN \log(MN))$ opérations.

Zoom avant d'une image

Premier exemple : Zoom avant d'une image = Evaluation sur une grille deux fois plus fine

- Il existe plusieurs méthodes pour effectuer le zoom d'une image.
 - Interpolation plus proche voisin,
 - Interpolation bilinéaire,
 - Interpolation bicubique ...
- On présente ici l'interpolation liée au polynôme trigonométrique interpolateur.
- On parle de zoom par **zero padding**: Comme vu en 1D ceci est dû au fait que la TFD de l'image zoomée est obtenue en bordant de zéros la TFD de l'image originale.

TP Zoom avant d'une image

- 1 Implémenter une fonction

```
zoom_fourier(u, r)
```

qui implémente le zoom par zero padding avec un facteur entier $r \geq 2$.
Ne pas oublier de prendre la partie réelle du résultat.

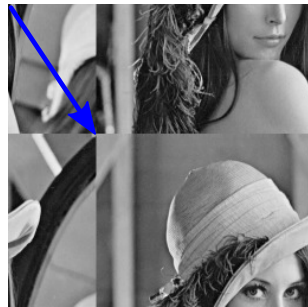
- 2 En utilisant une image naturelle u en entrée, vérifier que le sous-échantillonnage de l'image zoomée renvoie bien l'image originale.
- 3 Pour un facteur r important ($r = 8$) et une image petite (eg 64×64), vérifier qu'il y a des phénomènes d'oscillations le long des contours et sur les bords des images.
- 4 Comment expliquer ces oscillations ?

TP Zoom avant d'une image : Phénomène de Gibbs

Refaire un zoom de facteur important à partir d'une image représentant un rectangle gris clair sur un fond gris foncé.

Translations non entières

- Translater une image par un vecteur de translation à coordonnées entières revient à découper l'image en quatre et à replacer les morceaux.

 u  $\theta_{\tau} u$

- Grâce au polynôme interpolateur, on peut translater une image pour un vecteur de translation $\tau = (\tau_1, \tau_2)$ à coordonnées non entières.

Translations non entières

- Observation : Translater une exponentielle complexe $\varphi(x) = re^{i\theta} e^{i\alpha x}$ revient à changer l'argument θ (= phase à l'origine) du coefficient multiplicateur :

$$\varphi(x + t) = re^{i\theta} e^{i\alpha(x+t)} = re^{i(\theta+\alpha t)} e^{i\alpha x}.$$

TP Translations non entières

Soit $u \in \mathbb{R}^{M \times N}$ une image.

- 1 Si l'on souhaite traduire l'image u par un vecteur $\tau = (\tau_1, \tau_2) \in \mathbb{R}^2$, quel est le vecteur de translation pour le polynôme trigonométrique interpolateur ?
- 2 Soit v l'image traduite de u par le vecteur $\tau = (\tau_1, \tau_2) \in \mathbb{R}^2$ (évaluation sur les points $\tau + \Omega_{M,N}$). Déterminer la relation entre la TFD de v et la TFD de u .
- 6 Implémenter une fonction

```
v = translation_fourier(u, tau)
```

qui translate une image par la méthode étudiée. On n'oubliera pas de prendre la partie réelle des images traduite.

Rotation d'une image

- Effectuer la rotation d'une image revient à appliquer une matrice de rotation à la grille de pixel.
- Pour $\theta = \frac{\pi}{2}, \pi$, ou $-\frac{\pi}{2}$ il suffit de réordonner les coefficients de l'image.
- Pour un angle non multiple de $\frac{\pi}{2}$ on doit faire appel à une méthode d'interpolation.
- La méthode de Yaroslavsky repose sur l'observation suivante :

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} 1 & -\tan(\frac{\theta}{2}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin(\theta) & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan(\frac{\theta}{2}) \\ 0 & 1 \end{pmatrix}$$

- On décompose alors une rotation en translations successives sur les lignes et les colonnes des images.

Outline

- 1 Transformée de Fourier discrète
- 2 Transformée de Fourier discrète des images numériques
- 3 TFD 2D et transformations géométriques des images
- 4 Composante périodique d'une image**

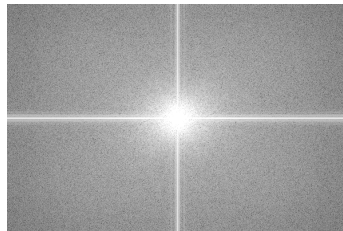
Motivation

- Why does the modulus of this image contain a cross ?

Image h



Modulus $|\hat{h}|$



Periodic component

- **Periodic component** $p = \text{per}(h)$, as proposed by [\[Moisan, 2011\]](#).
- Definition of the periodic component p of h : p unique solution of

$$\begin{cases} \Delta p = \Delta_i h \\ \text{mean}(p) = \text{mean}(h) \end{cases}$$

where, noting N_x the neighborhood of $x \in \Omega$ for 4-connectivity:

$$\Delta f(x) = -4f(x) + \sum_{y \in N_x} f(y) \quad \text{and} \quad \Delta_i f(x) = -|N_x \cap \Omega| f(x) + \sum_{y \in N_x \cap \Omega} f(y).$$

These two Laplacians only differ at the border:

- Δ : discrete Laplacian with periodic boundary conditions
 - Δ_i : discrete Laplacian without periodic boundary conditions (index i for interior)
- p is “visually close” to h (same Laplacian).
 - p is fastly computed using the FFT...

FFT-based Poisson Solver

Periodic Poisson problem: Find the image p such that

$$\begin{cases} \Delta p = \Delta_i h \\ \text{mean}(p) = \text{mean}(h) \end{cases}$$

In the **Fourier domain**, this system becomes:

$$\begin{cases} (-4 + 2 \cos(\frac{2s\pi}{M}) + 2 \cos(\frac{2t\pi}{N})) \hat{p}(s, t) = \widehat{\Delta_i h}(s, t), & (s, t) \in \hat{\Omega} \setminus \{(0, 0)\}, \\ \hat{p}(0, 0) = \text{mean}(h). \end{cases}$$

Algorithm to compute the periodic component:

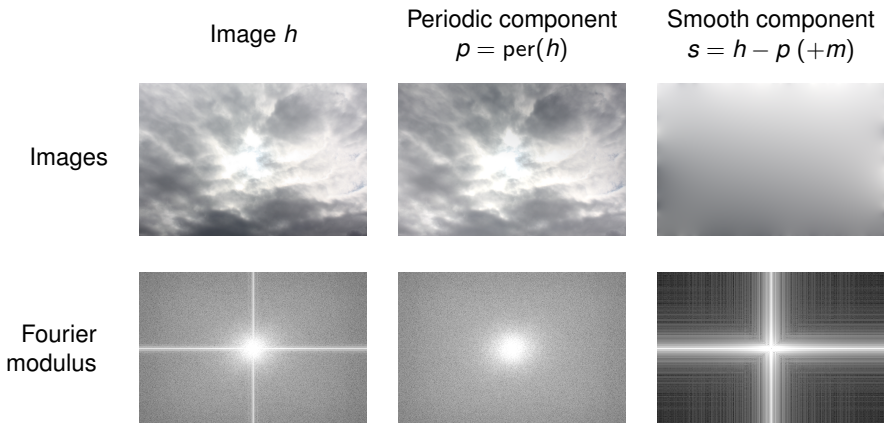
- 1 Compute $\Delta_i h$ the discrete Laplacian of h .
- 2 Compute $m = \text{mean}(h)$.
- 3 Compute $\widehat{\Delta_i h}$ the DFT of $\Delta_i h$ using the forward FFT.
- 4 Compute the DFT \hat{p} of p defined by

$$\begin{cases} \hat{p}(s, t) = \frac{\widehat{\Delta_i h}(s, t)}{-4 + 2 \cos(\frac{2s\pi}{M}) + 2 \cos(\frac{2t\pi}{N})} & \text{for } (s, t) \in \hat{\Omega} \setminus \{(0, 0)\} \\ \hat{p}(0, 0) = m \end{cases}$$

- 5 Compute p using the backward FFT (if necessary).

Periodic component: effects on the Fourier modulus

- p is “visually close” to h (same Laplacian).



- The application $\text{per} : h \mapsto p$ filters out the “cross structure” of the spectrum.



Cooley, J. W. and Tukey, J. W. *An algorithm for the machine calculation of complex Fourier series*, Math. Comput., 1965.



L. Moisan, *Periodic plus smooth image decomposition*, J. Math. Imag. Vis., 2011