Automne 2014

Modèles Mathématiques pour l'Image

Raphaëlle Chaine

Master Professionnel Image

Université Claude Bernard – Lyon 1

Transformée de Fourier discrète d'un signal 1D

- Soit $s_e(t)$ un signal échantillonné avec la fréquence f_e uniquement dans l'intervalle [0, .., NT_e[
- On dispose donc de N échantillons
- Soit $s_{ep}(t)$ le signal obtenu à partir de $s_e(t)$ en le reproduisant après translation de $T_0=NT_e$
  - $s_{ep}(t)$ est un signal discret et périodique
  - Sa transformée de Fourier est donc périodique de période $1/T_e=f_e$
  - Sa transformée de Fourier est également discrète, avec un spectre n'intégrant que les fréquences multiples de $1/T_0=f_0$

- Plus précisément :
  - $S_{ep}(f)$ est une somme d'impulsions de Dirac,

$$S_{ep}(f) = \sum_{k=-\infty}^{k=+\infty} S_{ep}(kf_0)\delta(f-kf_0)$$

$$\text{(ie } s_{ep}(t) = \sum_{k=-\infty}^{k=+\infty} S_{ep}(kf_0)e^{i2\pi kf_0t} \text{)}$$

  - pondérées par des coefficients $S_{ep}(kf_0)$ qui se retrouvent être périodique et correspondent à des moyennes sur une période T_0

$$S_{ep}(kf_0) = \frac{1}{N}\sum_{n=0}^{n=N-1} s_{ep}(nT_e)e^{-i2\pi kf_0nT_e}$$

- Que pouvez-vous dire de $2\pi kf_0nT_e$?

- Que pouvez-vous dire de $2\pi kf_0nT_e$?
  - Simplifications menant à $2\pi kn/N$ ne faisant intervenir ni f_0 ni T_e
- On a donc :

$$S_{ep}(kf_0) = \frac{1}{N}\sum_{n=0}^{n=N-1} s_{ep}(nT_e)e^{-i2\pi kn/N}$$

Transformée de Fourier discrète d'un signal 1D

- On peut donc s'affranchir de T_e et T_0, la seule quantité importante étant N
- Soit s un signal discret périodique composé de N valeurs réelles ou complexes

$$s(n) = \sum_{k=0}^{N-1} S(k)e^{i2\pi kn/N} \qquad O \leq n < N$$

$$s(n) = \sum_{k=-N/2}^{k<N/2} S(k)e^{i2\pi kn/N} \quad O \leq n < N$$

Transformée de Fourier discrète d'un signal 1D

- Avec :

$$S(k) = \frac{1}{N} \sum_{n=0}^{n=N-1} s(n) e^{-i2\pi kn/N}$$

$$O \leq k < N$$

$$\text{ou bien } -N/2 \leq k < N/2$$

- S correspond au produit du vecteur s avec une matrice de Vandermonde

$$M_{kn} = e^{-i2\pi kn/N}$$

Transformée de Fourier discrète d'un signal 2D

- x(l,c) signal périodique, composé d'une période de L lignes et C colonnes
- On a

$$x(l,c) = \sum_{l=-L/2}^{l<L/2} \sum_{c=-C/2}^{c<C/2} X(u,v) e^{i2\pi(ul/L+vc/C)}$$

- Avec

$$X(u,v) = \frac{1}{LC} \sum_{u=-L/2}^{u<L/2} \sum_{v=-C/2}^{v<C/2} x(l,c) e^{-i2\pi(ul/L+vc/C)}$$

Transformée de Fourier discrète d'un signal 2D

- Faire le calcul de la transformée de Fourier des lignes et remplacer chaque ligne par sa transformée de Fourier
- Faire le calcul de la transformée de Fourier des colonnes ainsi obtenues
- Cela est équivalent à faire directement une transformée de Fourier discrète 2D

Représentation d'une Transformée de Fourier discrète 2D

- Le spectre X de l'image x est à valeurs complexes
- On considère
  - Spectre d'amplitude
  - Spectre de phase
- Sur les images du monde réel,
  - Représenter log(1+|X(u,v)|) plutôt |X(u,v)| pour mieux distinguer les variations au voisinage de (0,0) (fréquences basses de module beaucoup trop grand)
  - Ce spectre a généralement une forme proche de 1/f ou de $1/f^3$

Représentation d'une Transformée de Fourier Discrète 2D

- Pour la compression, le spectre de phase contient donc plus d'information spécifique au contenu de l'image

Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- La transformée de Fourier d'un signal échantillonné sur N valeurs nécessite N^2 multiplications et additions (L^2C^2 multiplications et additions en 2D)
- Il existe une stratégie Diviser pour Régner permettant de diminuer cette complexité : Algorithme en O(nlgn)
- Biblio : Cormen, Leiserson, Rivest

Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- Idée de l'algorithme :
  - Le calcul d'une TFD de taille N se ramène au calcul de 2 TFD de taille N/2 suivi de N/2 multiplications

$$NS(m) = \sum_{n=0}^{n=N-1} s(n) e^{-i2\pi mn/N}$$

  - On pose n=2k si n est pair, n=2k+1 si n est impair et on suppose que N est pair : N=2M

$$NS(m) = \sum_{k=0}^{N/2-1} s(2k) e^{-i2\pi 2km/N} + \sum_{k=0}^{N/2-1} s(2k+1) e^{-i2\pi(2k+1)m/N}$$

Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- On adopte les notations suivantes qui correspondent à l'introduction de nouveaux signaux :
  - Signal s_{2M} de taille 2M

$$s_{2M}(k) = s(k) \text{ pour } k = 0, .., 2M-1$$

  - De TFD S_{2M} de taille 2M

$$2MS_{2M}(k) = 2MS(k) \text{ pour } k = 0, .., 2M-1$$

  - Signal s^p_M de taille M

$$s^p_M(n) = s(2n) \text{ pour } n = 0, .., M-1$$

  - Signal s^i_M de taille M

$$s^i_M(n) = s(2n+1) \text{ pour } n = 0, .., M-1$$

Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- On partage donc le signal s_{2M} en 2 signaux de taille moitié s^p_M et s^i_M
- On a alors :

pour $m = 0, .., M-1$

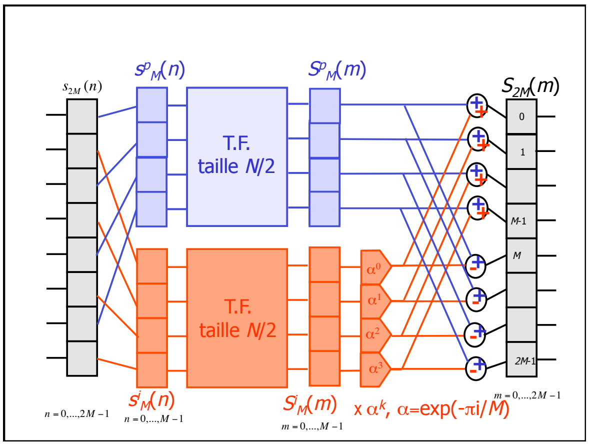
1ère moitié du signal

$$2MS_{2M}(m) = MS^p_M(m) \boxed{+} e^{-i\pi m/M} MS^i_M(m)$$

2ème moitié du signal

$$2MS_{2M}(M+m) = MS^p_M(m) \boxed{-} e^{-i\pi m/M} MS^i_M(m)$$

- Démontrons ce résultat



Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- On partage donc le signal s_{2M} en 2 signaux de taille moitié s^p_M et s^i_M
- L'opération de composition des deux TDFD de taille moitié s'appelle l'opération Papillon
  - M multiplications, additions et soustractions
- Cas d'arrêt de la récursion
  - Si M=1 $S^p_M(m=0) = s^p_M(m=0)$
    $S^i_M(m=0) = s^i_M(m=0)$
  - Algorithme récursif valable pour N correspondant à une puissance de 2
  - Profondeur de récursion $lg_2(N)$

Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- Dérécursification de l'algorithme
  - A la descente, simple réorganisation de l'ordre des éléments du signaux
  - L'opération Papillon se fait à la remontée
- Simulation sur un signal de taille 8
- Réorganisation des éléments du signal dans leur ordre d'apparition aux feuilles
  - Inversion de la décomposition binaire
  - En effet à la première récursion, les indices dont la décomposition binaire se termine par 0 se retrouvent à gauche et les autres à droite, et ainsi de suite…

Algorithme de Transformée de Fourier Discrète Rapide (FFT)

- Après réorganisation de l'ordre des éléments du signal (dans un tableau)
- Pour s de 1 à $lg_2(N)$
  - Pour k de 0 à N-1 pas de 2^s faire
    - Combiner les deux TFD à 2^{s-1} éléments stockées entre les indices k et $k+2^{s-1}-1$ puis $k+2^{s-1}$ et $k+2^s-1$
    - En une TFD à 2^s éléments que l'on stocke entre les indices k et et $k+2^s-1$
- Le tableau final contient la TFD