

# Traitement d'Image et du Signal - TP3

Laurent Cetinsoy, Karim Kouki, Aris Tritas

8 octobre 2016

## Résumé

L'objectif de ce TP est de calculer des convolutions, vérifier la validité de filtres linéaires, extraire leur réponse impulsionnelle. Par ailleurs nous mettons en pratique la méthode de rotation par distortion de Yaroslavsky.

## Introduction

Nous avons donné, lors du TP précédent, l'idée d'algorithme faisant une rotation efficace dans l'espace réel par succession de translations dans le domaine de Fourier. Par soucis de complétude, nous la ré-écrivons ci-suit.

## Rotation d'image

Pour chaque point  $(x, y)$  de l'image originale nous souhaitons utiliser la TFD pour calculer le point  $(x', y')$  résultant d'une rotation d'angle  $\theta \in [0, 2\pi]$ . L'on peut définir la rotation par la matrice  $\mathbf{M}$  ci-dessous :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \end{pmatrix}$$

où  $\mathbf{M}$  peut être ré-exprimée comme suit (e.g. [3]) :

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

L'idée de [1] [2] est d'utiliser trois convolutions linéaires (et donc séparables) qui distordent l'image successivement selon les axes  $x$ ,  $y$  et  $x$ . Chacune s'écrit comme une translation dans le domaine de Fourier. La distortion  $u_{dx}(x, y) = u(x + ay, y)$  pour l'axe  $x$  (où  $a$  contrôle l'angle) s'exprime par l'opérateur suivant :

$$D_x(\xi) = \mathcal{F}\{u_d(x, y)\}_1 = \mathcal{F}\{u(x, y)\}e^{-2i\pi\xi ay}$$

Le signal réel distordu sur  $x$  par la transformée inverse :  $u_{dx}(x, y) = \mathcal{F}^{-1}\{D_x(\xi)\}$   
 Si l'on répète cette opération pour  $y$  et encore une fois pour  $x$  l'on retrouve le signal réel pivoté. Il s'agit bien de trois translations, donc un algorithme efficace qui les effectue peut être donné. A noter que cette méthode induit une perte d'information près des bords, il devient donc nécessaire d'envisager une convolution plus robuste pour parer à cet effet-là.

## Exercices

Notons le signal d'entrée  $e(t)$  et le signal de sortie  $s(t)$ .

**Définition** Système linéaire :

Si  $e_1(t) \rightarrow s_1(t)$  et  $e_2(t) \rightarrow s_2(t)$ , alors  $\alpha_1 e_1(t) + \alpha_2 e_2(t) \rightarrow \alpha_1 s_1(t) + \alpha_2 s_2(t)$

**Définition** Système invariant :

Si  $e(t) \rightarrow s(t)$  alors  $e(t - \tau) \rightarrow s(t - \tau)$ ,  $\tau$  étant une constante de décalage.

**Définition** Réponse impulsionnelle :

La sortie d'un système linéaire invariant est égale au produit de convolution de l'entrée par la réponse impulsionnelle  $h$ .

$$e(t) \longrightarrow s(t) = (e \otimes h)(t)$$

**Définition** Produit de convolution de deux suites  $u_n$  et  $v_n$  :

$$(u \otimes h)(n) = \sum_{k \in \mathbb{Z}} u_{n-k} h_k$$

## Exercice 1

Pour les questions (1)-(4), l'entrée est la suite  $u_n$  et la sortie la suite  $v_n$  pour  $n \in \mathbb{Z}$ .  
 Pour les questions (5) et (6) l'entrée est une fonction  $f \in L^1 \cap L^2$  définie sur  $\mathbb{R}$  et la sortie est une fonction  $g(x) : \mathbb{R} \rightarrow \mathbb{R}$

1.  $v_n = u_n - u_{n-1} + 3u_{n+1}$  :

La relation est une somme de termes décalés et comporte des produits avec des scalaires, elle est donc linéaire.  $\alpha(u_n - u_{n-1} + 3u_{n+1}) \rightarrow \alpha v_n$

invariant car  $u_{n-\tau} - u_{n-1-\tau} + 3u_{n+1-\tau} \rightarrow v_{n-\tau}$

réponse impulsionnelle : par identification  $h_{-1} = 3, h_0 = 1, h_1 = -1$

2.  $v_n = u_{2n}$  :

La relation impulsionnelle est un sous échantillonnage qui est linéaire : sous-échantillonner la somme de deux signaux revient à sommer les sous échantillons des signaux. linéaire car  $\alpha u_{2n} \rightarrow \alpha v_n$

La relation est non-invariante par translation car  $u_{2n-\tau} \not\rightarrow v_{n-\tau} (= u_{2(n-\tau)})$

3.  $v_n = \max(u_n, u_{n-1}, u_{n+1})$  :

L'opérateur max n'est pas linéaire. Considérons les suites  $u_n$  et  $u'_n$  définies par :

- $u_k = u'_k = 0 \forall k \in \mathbb{Z} \setminus \{0, 1, 2\}$
- $u_0 = -1, u_1 = -1, u_2 = -10$
- $u'_0 = -1, u'_1 = 1, u'_2 = 10$

Il est clair que

$$\max(-1, -1, 10) + \max(-1, 1, 10) = 10 \neq 0 = \max(-1 - 1, -1 + 1, -10 + 10)$$

4.  $v_n = u_{n-1}$  :  
 linéaire car  $\alpha u_{n-1} \rightarrow \alpha v_n$   
 invariant par translation car  $u_{n-1-\tau} \rightarrow v_{n-\tau}$   
 réponse impulsionnelle :  $h_{-1} = 1$
5.  $g(x) = \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} f(t) dt$  :  
 linéarité de l'intégration :  $\int_{x-\frac{1}{2}}^{x+\frac{1}{2}} f_1(t) + f_2(t) dt = \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} f_1(t) dt + \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} f_2(t) dt$   
 invariant par translation : soit l'entrée  $f(t - \tau)$ , en posant  $t' = t - \tau \implies dt' = dt$   
 $\int_{x-\frac{1}{2}-\tau}^{x+\frac{1}{2}-\tau} f(t') dt' \rightarrow g(x - \tau)$
6.  $g(x) = \max\{f(t), t \in [x - 1, x + 1]\}$  :  
 ni linéaire, ni invariant par translation : contre-exemple  $f(t) = t^2$

## Exercice 2

1.  $u_0 = 1, u_n = 0 \forall n \neq 0$  :  
 $(u \otimes v)(n) = (u \otimes v)(n) = v_n = \sqrt{\log(\cos(3n) + 2)}$
2. L'on donne les termes de  $w$  non-nuls :
  - $w(0) = (u \otimes v)(0) = u(0)v(0)$
  - $w(1) = (u \otimes v)(1) = u(0)v(1) + u(1)v(0)$
  - $w(2) = (u \otimes v)(2) = u(1)v(1) + u(0)v(2)$
  - $w(3) = (u \otimes v)(3) = u(1)v(1) + u(1)v(2)$
3. De même, les termes non-nuls sont :
  - $w(0) = (u \otimes v)(0) = u(0)v(0)$
  - $w(1) = (u \otimes v)(1) = u(0)v(1) + u(1)v(0)$
  - $w(2) = (u \otimes v)(2) = u(1)v(1) + u(0)v(2)$
  - $w(3) = (u \otimes v)(3) = u(1)v(1) + u(1)v(2)$
4. Encore, les termes non-nuls sont :
  - $w(0) = (u \otimes v)(0) = u(0)v(0)$
  - $w(1) = (u \otimes v)(1) = u(0)v(1) + u(1)v(0)$
  - $w(2) = (u \otimes v)(2) = u(1)v(1) + u(0)v(2)$
  - $w(3) = (u \otimes v)(3) = u(1)v(1) + u(1)v(2)$

5.  $u_n = (-\frac{1}{2})^n$ ,  $n \in \mathbb{N}$ , sinon  $u_n = 0$  et  $v_0 = 1, v_1 = \frac{1}{2}$  :

$$(u \otimes h)(n) = \sum_{k \in \mathbb{Z}} u_{n-k} h_k$$

## Résultats de rotation

Ci-dessous de gauche à droite se trouvent : l'image originale de dimension 502px  $\times$  502px, la même image après une rotation d'angle  $\theta = \frac{\pi}{4}$ .

FIGURE 1 – Image originale, image après rotation

## Références

- [1] Unser, Michael, Philippe Thevenaz, and Leonid Yaroslavsky. "Convolution-based interpolation for fast, high-quality rotation of images." IEEE Transactions on Image Processing 4.10 (1995) : 1371-1381.
- [2] Larkin, Kieran G., Michael A. Oldfield, and Hanno Klemm. "Fast Fourier method for the accurate rotation of sampled images." Optics communications 139.1 (1997) : 99-106.
- [3] Paeth, Alan W. "A fast algorithm for general raster rotation." Graphics Interface. Vol. 86. 1986.