Traitement d'Image et du Signal - TP3

Laurent Cetinsoy, Karim Kouki, Aris Tritas

6 octobre 2016

Résumé

Introduction

Translation d'image

On propose un algorithme (cf. code source) de translation d'image basé sur la Fast Fourier Transform (FFT). A partir de la transformée d'une image l'on calcule la transformée inverse avec l'interpolation suivante, en 2D : $\forall n \in \{0, ..., N-1\}$ $\forall m \in \{0, ..., M-1\}$

$$\mathcal{I}u(x+\tau_x, y+\tau_y) = \frac{1}{N}\frac{1}{M}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \hat{u}(n,m)\exp(2i\pi\frac{n}{N}(x+\tau_x))\exp(2i\pi\frac{m}{N}(x+\tau_x))$$

[1]

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 :

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

1. **Remarque :** Pour des raisons historiques, l'algorithme standard de FFT donne une représentation en fréquence différente de celle qui est manipulée par l'homme. Il faut tenir compte de ce fait lors de la manipulation de la représentation fréquentielle du signal.

où **M** peut être ré-exprimée comme suit (e.g. [**?**]) :

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

L'idée de [**?**] [**?**] 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 :

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

Le signal réel distordu sur x par la transformée inverse : $u_{dx}(x, y) = \mathcal{F}^{-1}\{D_x(\xi)\}$
Si l'on répête 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.

Résultats

Ci-dessous de gauche à droite se trouvent : l'image originale de dimension 502px $\times$ 502px, la même image translatée de 50.5px dans chaque direction, et un patch de 256px de côté extrait du centre de l'image et zoomé $\times$2.

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.