

# TP6 : Design de filtre et filtrage d'image.

Fabien PIERRE

fabien.pierre@math.u-bordeaux1.fr

<http://sites.google.com/site/fabienpierre/enseignements>

Automne 2013.

## 1 Fractions rationnelles.

- Dans quelles couronnes du plan complexe  $\mathbb{C}$  la fraction rationnelle  $z \mapsto \frac{1}{1-z}$  est-elle développable. Donner le développement pour chaque couronne de  $\mathbb{C}$ .
- Décomposer la fraction rationnelle  $z \mapsto \frac{1}{z^2 - 3z + 1}$  en éléments simples. Vérifier votre résultat avec la commande `residue` de Matlab. Donner les couronnes de  $\mathbb{C}$  dans lesquelles elle est développable en série et ses développements dans les diverses couronnes.
- Vérifier la décomposition en éléments simples avec Matlab de  $z \mapsto \frac{1}{z^2 - 2z + 1}$  et donner son développement dans chaque couronne de  $\mathbb{C}$  possible.

## 2 Design de filtre.

- Créer un signal chirp linéaire de taille 1024 (que l'on supposera durer 2 sec.) à l'aide de la fonction `MakeSignal`.
- À l'aide de la commande `butter`, calculer les paramètres d'un filtre de Butterworth coupant à la fréquence normalisée  $F_e/4$  de taille de réponse impulsionnelle 10.
- Filtrer le chirp linéaire (généralisé par `MakeSignal`) à l'aide de la commande `filter` (la commande `poly` sera utile).
- Rappeler le lien entre la réponse fréquentielle et la transformée en  $z$ . À l'aide de la commande `polyval`, dessiner la réponse fréquentielle du filtre.

## 3 Filtrage de Tychonoff (ou de Wiener).

On définit le gradient d'une image de taille  $N \times N$  comme la champ de vecteur :

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < N \\ 0 & \text{si } i = N \end{cases}$$
$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N \\ 0 & \text{si } j = N \end{cases}$$

On veut résoudre le problème

$$\inf_u \|u - f\|_2^2 + \lambda \|\nabla u\|_2^2. \quad (1)$$

On rappelle que la TFD d'une image  $f(n, m)$  est donnée par

$$F[f](p, q) = \sum_{m,n} f(m, n) e^{-\frac{2im p \pi}{N}} e^{-\frac{2inq \pi}{N}}. \quad (2)$$

De plus,  $\|F[f]\|_2 = N^2\|f\|_2$ .

On peut également montrer que

$$\|F[\nabla f]\|_2^2 = \sum_{m,n} |F[\nabla f](m,n)|^2 = \sum_{m,n} 4|F[f](m,n)|^2 \left( \sin^2 \frac{\pi n}{N} + \sin^2 \frac{\pi m}{N} \right) \quad (3)$$

Par l'identité de Parseval, montrer que la solution de (1) est donnée par :

$$F[u](p,q) = \frac{F[f](p,q)}{1 + 8\lambda \left( \sin^2 \frac{\pi n}{N} + \sin^2 \frac{\pi m}{N} \right)}.$$

– Implémenter un tel filtre à l'aide de la formule ci-dessus.

## 4 Détection de contours par filtrage de Canny.

Le filtre de Canny permet de révéler les contours visibles sur une image.

**Lissage gaussien.** Le filtre de Canny est sensible au bruit. On va donc filtrer l'image par un noyau gaussien. On va effectuer la convolution d'une image par ce filtre (ces notions théoriques seront vues en cours ultérieurement).

Le filtre gaussien est une version discrétisée de la fonction

$$e^{-\frac{x^2 + y^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma},$$

sur l'intervalle  $[-3\sigma, 3\sigma]$ .

- Construire un tel filtre.
- Appliquez le sur l'image `lena` (commande `conv2`). Explorer les options de la commande `conv2`. Trouvez la plus appropriée.
- Commentez. (On appelle cette opération un lissage gaussien, ou flou gaussien.)

**Détection.** Il s'agit maintenant de trouver les contours de l'image. On approxime le gradient  $\nabla u$  de l'image par convolution avec le filtre  $d_x = [-1, 1]$  pour la dérivée dans la direction horizontale, et  $d_y = d_x^t$  pour la dérivée verticale.

Les contours sont donnés par les valeurs de  $\sqrt{\frac{\partial u^2}{\partial x} + \frac{\partial u^2}{\partial y}}$  supérieure à un seuil donné.

- Calculer et afficher les contours.
- Commentez.

On veut tester la robustesse au bruit de cet algorithme. Pour cela on crée une image bruitée par un bruit gaussien centré. Un bruit gaussien est la réalisation d'une suite de variables aléatoires indépendantes de loi normale centrée. La variance s'appelle aussi *puissance* du bruit.

- Utilisez la commande `randn` pour ajouter un tel bruit.
- Affichez l'image bruitée. Commentez. Faites varier la puissance du bruit.
- Appliquez le filtrage gaussien sur l'image bruitée.
- Réalisez une détection de contours sur l'image bruitée et sur l'image lissée.