4ETI, Majeure Image

TP Ondelettes 2D

Débruitage, classification

Dans ce TP on abordera deux applications majeures des ondelettes à l'imagerie. La première partie consiste à **débruiter une image**. Cette partie est basée sur l'utilisation des **ondelettes orthogonales** (comprendre, discrètes). La seconde partie vise à caractériser des **textures** et, pour une nouvelle image de texture donnée, à trouver celle qui lui est la plus proche dans une base de données. Cette partie fait appel à la transformation en ondelettes continues.

1 Débruitage d'une image : ondelettes orthogonales

Pour cette partie on utilisera la bibliothèque WaveLab développée par Buckheit, Chen, Donoho et Johnstone (http://www-stat.stanford.edu/~wavelab/). Commencez par télécharger les fichiers Matlab de cette librairie qui seront utiles pour le TP, depuis le site *e-campus*.

NB : pour les fonctions de WaveLab, taper help+nom de la fonction, pour une aide en ligne.

1.1 Analyse multi-résolution : reconstruction/reconstruction

On effectuera la décomposition puis la reconstruction d'une image à partir de l'ondelette «Coiflet». Cette ondelette a la particularité d'avoir N moments nuls de même que la fonction d'échelle qui lui est associée.

- Fabriquer l'ondelette Coiflet à 3 moments nuls :

```
CoifQMF = MakeONFilter('Coiflet',3);
```

- La visualiser.
- Lire l'image cameraman.tif. On la note I.
- A l'aide de la fonction FWT2_PO, effectuer l'analyse multi-résolution de l'image I. Le second paramètre de la fonction FWT2_PO donne l'échelle la plus grossière. Si ce paramètre est L alors la taille de l'image la plus grossière est de 2^L par 2^L.
- Visualiser l'analyse multi-résolution de l'image ; extraire l'image la plus grossière et la visualiser dans une autre fenêtre.
- Reconstruire l'image initiale à l'aide de la fonction IWT2_PO. Vérifier que celle-ci est bien identique à l'image originale.

1.2 Application au débruitage d'une image

Pour débruiter l'image, on effectue une analyse multi-résolution de cette première. On seuille alors tous les coefficients de la décomposition en ondelettes qui sont plus petits qu'une certaine valeur (bien sûre, on n'effectuera pas ce seuillage sur l'image basse résolution obtenue en fin de cascade des filtres). On utilisera ici un seuillage doux :

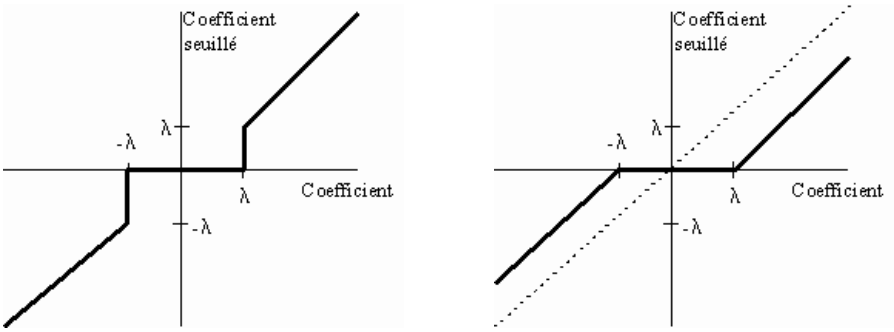

Figure 1 : seuillage dure (gauche) et seuillage doux (droite)

- Lire l'image Daubechies.raw à l'aide de la fonction ReadImage :

```
Ingrid = ReadImage('Daubechies');
```

- Bruiter l'image lue :

```
NoisyIngrid = Ingrid + 5*WhiteNoise(Ingrid);
```

- A l'aide de l'ondelette Coiflet, effectuer la décomposition multi-résolution de l'image bruitée, jusqu'à l'échelle 3 (l'image basse résolution sera de taille 8*8).
- Effectuer un seuillage doux des imagettes pour un seuil de 10.
- Reconstruire une image débruitée à partir de l'image multi-résolution seuillée.
- Visualiser l'image originale, l'image bruitée et l'image reconstruite de même que leur région d'intérêt autour de l'œil :

```
subplot(222), Imagesc(Ingrid), colormap gray, axis([110 160 110 160]);
axis square
```

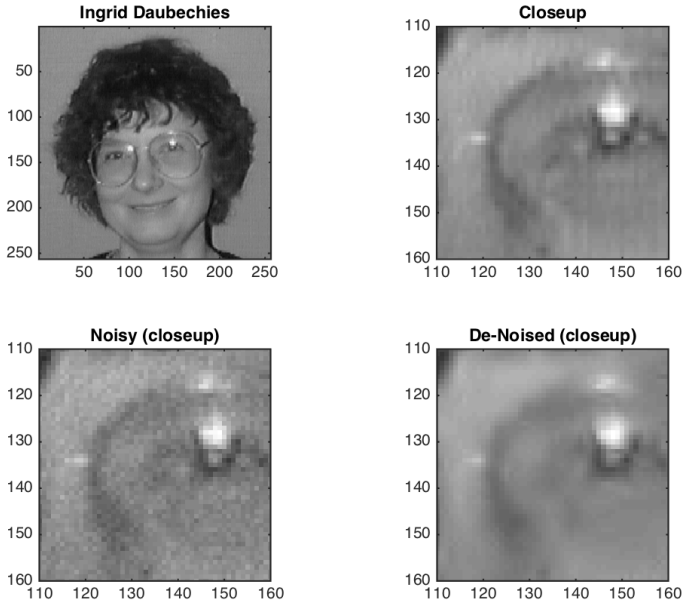

Figure 2: Image originale (haut), image bruitée et reconstruite (bas, gauche et droite)

2 Classification : transformée en ondelettes continues

Nous n'utilisons pas WaveLab pour cette partie.

2.1 Ondelettes orientées : dérivées d'une gaussienne

- Fabriquer l'ondelette mère dérivée de la gaussienne pour (x,y) variant entre -1 et 1 avec $N_x=N_y=64$ échantillons. On rappelle l'expression de la dérivée de la gaussienne :

$$\Psi(x,y)=xe^{-\frac{x^2+y^2}{2}}$$

- Fabriquer et visualiser la famille d'ondelettes $\Psi_{j,l}(X)=\frac{1}{2^j}\Psi(R_{ld\theta}\frac{X}{2^j})$ pour 4 directions entre 0 et π (d'où $d\theta=pi/4$) : l= {0,..3} et pour 4 échelles différentes : j={-4,-1}. On a noté $R_{l\,d\theta}$ la matrice de rotation d'angle l dθ.

2.2 Application à la caractérisation de textures

- Effectuer la transformée en ondelettes des 7 images de texture pour les 4 orientations et les 4 facteurs d'échelle suggérés précédemment. On utilisera la commande « conv2 » pour cette transformée (puisque, rappelons le, la transformée en ondelette continue revient à un produit de convolution). Pour chaque texture, on fabrique le vecteur caractéristique. Celui-ci sera formé de 4*4=16 composantes, chaque composante étant définie comme la moyenne de la norme au carrée de la transformée en ondelettes.
- Calculer les vecteurs caractéristiques des images : Image1, Image2 et rechercher la texture dont le vecteur caractéristique est le plus proche au sens de la norme euclidienne.

Dans la pratique, il faudrait calculer un vecteur caractéristique qui représente une classe de textures et non une texture unique.

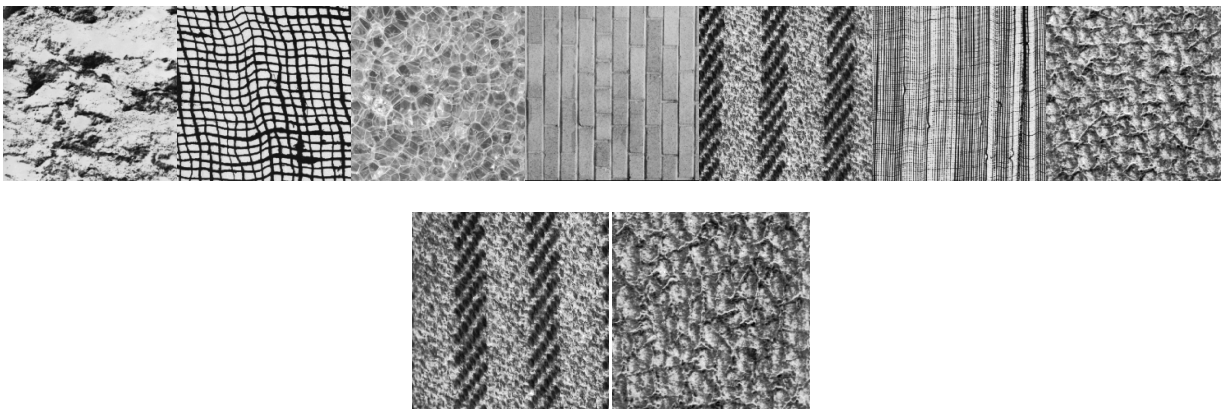

Figure 3 : images de 7 textures (haut) et deux images à "classer" (bas)