Traitement de l'information et vision artificielle **Guillaume Obozinski**
Ecole des Ponts ParisTech, 2ème année guillaume.obozinski@imagine.enpc.fr

TP 8 : ONDELETTES 2D, COMPRESSION ET DÉBRUITAGE D'IMAGE

Pour commencer la séance

1. Créez un dossier `TIVA_TP8` sur le bureau.
2. Lancer ensuite `Matlab` et modifier le répertoire de travail en choisissant le répertoire `Bureau/TIVA_TP8` que vous avez créé.

1. Installation de Wavelab

1. Nous ferons ce TP avec le même package Wavelab developpé par l'université de Stanford que pour la dernière séance sur les ondelettes. Si vous l'avez déjà installé vous pouvez aller directement à la section 2. Il est peut être téléchargé à partir de la page : `https://statistics.stanford.edu/~wavelab`
   Il faut suivre les instructions sur la page web.
2. Nous allons utiliser un certain nombre de scripts et de fonctions du package. Pour rappel, vous pouvez taper `help myfunction` pour afficher l'aide de la fonction `myfunction.m` . Vous pouvez aussi taper `type myfunction` pour afficher directement dans la ligne de commande le code source de la fonction. Finalement, en tapant `open myfunction` vous demanderez à Matlab d'ouvrir le code source dans l'éditeur.
3. Le code contient un certain nombre de démos qui nous seront utiles, en particulier celles contenues dans
   `/Wavelab850/Workouts/Toons/`
   Pour aller dans ce dossier à partir de votre répertoire `Bureau/TIVA_TP8`, faites
   `cd(matlabroot)` puis `cd toolbox/Wavelab850/Workouts/Toons/`
   La commande `ls` vous permet d'avoir la liste de tous le fichiers présents dans le dossier. Vous verrez un certain nombre de fichiers `toon_xyz.m` qui sont des petites démos. Si vous tapez `help Contents` dans ce répertoire, vous aurez le sommaire des démos. Utilisez `help toon0111` pour avoir une présentation de la démo 0111, etc. Je signalerai au fur et à mesure du TP quelles sont les démos utiles.
   N'oubliez pas de retournez dans votre propre dossier si vous voulez imprimer des figures et sauver ou utiliser vos scripts.

2. Visualisation des ondelettes 2D

Le but de cette première partie est de comprendre comment est organisée la transformée en ondelettes (ou décomposition sur une base d'ondelettes) d'une image.

1. Commençons par rappeler les fonctions principales de Wavelab que nous allons utiliser.
   - Nous avons déjà utilisé la fonction `MakeONFilter`. Par exemple, le filtre miroir pour D4 s'obtient avec
     ```
     qmf=MakeONFilter('Daubechies',4);
     ```
   - La fonction `FWT2_PO` calcule la transformée en ondelettes périodisée (c'est-à-dire en considérant que l'image de départ est périodique comme pour la TFD).
     Sa syntaxe est
     ```
     tw=FWT2_PO(mon_image,L,qmf);
     ```
     où `mon_image` est une matrice de niveaux de gris, `L` défini la résolution 2^{-L} la plus grossière de la TO et `qmf` est le filtre miroir associé à l'ondelette choisie.
   - La transformée inverse a quasiment la même syntaxe. En effet,
     ```
     mon_image=IWT2_PO(tw,L,qmf);
     ```
     permet de reconstruire l'image `mon_image` à partir de la transformée `tw`.
2. Pour commencer visualisez les transformées en ondelettes des `toon0231-0233`.
3. On considère la transformée en ondelettes d'une image 256×256. C'est aussi une matrice de taille 256×256. On suppose qu'on a calculé une transformée en ondelette de l'image (donc une transformée 2D) en s'arrêtant à la résolution 16×16.

   Quelles sont les valeurs de j, k et (n_1, n_2) pour lesquels l'image admet un coefficient de décomposition sur $\psi^k_{j,(n_1,n_2)}$? sur $\phi_{j,(n_1,n_2)}$?
   Quelles sont les coordonnées $(m_1, m_2) \in \{1, \dots, 256\}^2$ dans cette matrice du coefficient de décomposition de l'image sur l'ondelette $\psi^k_{j,(n_1,n_2)}$?
4. Proposez une manière d'obtenir l'image 2D à la résolution 256×256 correspondant à la fonction $\psi^k_{j,(n_1,n_2)}$ en utilisant un simple appel à la fonction qui calcule la transformée inverse de la transformée en ondelettes, c'est-à-dire la fonction `IWT2_PO`.
5. Ecrivez une fonction qui représente successivement les éléments de la base d'ondelettes des échelles les plus grossières aux échelles les plus fines. Concrètement, la fonction devra pour chaque j variant de 0 à 6, et pour j fixé en faisant varier k et pour (j, k) fixés en faisant varier (n_1, n_2), représenter l'ondelette $\psi^k_{j,(n_1,n_2)}$ (dans cet ordre) en faisant une pause après chaque image. On pourra visualiser l'ondelette d'une part comme une image avec la fonction `imagesc` d'autre part comme une function en 2D avec la fonction `surf`.

   Précisément, si `im_wav` est la matrice-image de l'ondelette, on pourra utiliser :
   ```
   figure(2)
   surf(im_wav);
   colormap('copper');
   shading interp %donne à la surface un aspect lisse par interpolation
   light %ajoute une source lumineuse pour illuminer la surface
   lighting gouraud; %rendu de la diffusion/réflexion de la lumière
   ```

6. Incluez le code de cette fonction dans votre rapport et des exemples d'images/surfaces d'ondelettes représentées pour Haar, D4 et Symmlet 8.
7. Dans les toons `0521-0524`, on s'intéresse à la compression d'une image cérébrale. Quelles sont les conclusions qui ressortent de ces expériences ?
8. Dans les toons `0541-048`, on s'intéresse à la compression d'une photo de visage. Quelles sont les conclusions qui ressortent de ces expériences ?

3. Débruitage d'image.

On utilisera au choix pour cette partie soit l'image `Lenna` soit l'image `Canaletto` qui sont fournies avec Wavelab et que l'on peut ouvrir avec la fonction `ReadImage`. Voir l'aide de cette fonction pour plus de détails. Nous allons artificiellement bruiter ces images et essayer de les débruiter au mieux.

1. Il est très répandu en traitement d'image d'utiliser le SNR (*Signal to Noise Ratio*) et le PSNR (*Peak Signal to Noise Ratio*) comme des mesures de qualité d'approximation d'une image I par une image approchée I_1. Les définitions du SNR et du PSNR sont :

$$\mathrm{PSNR} = 10\log_{10}\frac{MN\max_{i,j} I(i,j)^2}{\sum_{i,j}|I(i,j)-I_1(i,j)|^2}, \qquad \mathrm{SNR} = 10\log_{10}\frac{\sum_{i,j} I(i,j)^2}{\sum_{i,j}|I(i,j)-I_1(i,j)|^2},$$

   où I et I_1 sont des images de taille $M \times N$ et $\sum_{i,j}$ désigne la somme pour $i \in \{1,\dots,M\}$ et $j \in \{1,\dots,N\}$.

   Écrire une fonction, qui prend comme entrée les deux images I et I_1 et qui calcule les deux valeurs SNR et PSNR correspondantes. Attention, comme le traitement des boucles est très lent en Matlab, il faudra utiliser exclusivement des opérations matricielles.
2. On ajoute à chaque pixel de l'image un bruit gaussien indépendant de celui présent aux autres pixels et d'écart-type σ que l'on choisira de sorte que le bruit soit suffisamment important pour dégrader l'image sans que le signal soit noyé dans le bruit. On pourra utiliser la fonction `randn` pour obtenir une matrice d'entrées gaussiennes indépendantes. (En fait le bruit des images a plutôt une distribution de Poisson, mais par souci de simplicité on travaillera avec un bruit gaussien).
3. Comme première méthode pour débruiter on calcule la transformée en ondelette de l'image bruitée et on enlève les coefficients d'amplitudes inférieures à un seuil T. Implémentez cette méthode avec l'ondelette de votre choix et tester avec différentes valeurs du seuil T. A chaque fois calculer le SNR et PSNR par rapport à l'image originale.
4. La transformée en ondelette n'est pas invariante par translation. En conséquence, si l'on applique une méthode débruitage à la transformée en ondelettes d'une image et à sa version translatée, les résultats peuvent être très différents. Pour pallier cet inconvénient, on utilise la transformée en ondelette *invariante par translation*. Il s'agit de stocker

non-seulement les coefficients de la transformée en ondelette de l'image I, mais aussi de toutes les images que l'on obtient à partir de I par une translation. Si l'on utilise une transformée en ondelette avec un niveau de résolution minimale J_0, alors il suffit de considérer les translations par un vecteur $(x, y) \in \{1, \ldots, 2^{J_0} - 1\}^2$. En pratique, on peut éviter de considérer toutes les translations possibles et se limiter à un nombre de translations $m^2 \ll (2^{J_0})^2$.

5. Ecrivez une fonction qui a partir d'une image I calcule l'image translatée de s_x et s_y respectivement selon l'axe des x et des y en considérant que l'image est périodique dans les deux directions. On pourra en particulier utiliser la fonction `mod`.
6. Écrivez une fonction, nommée `denoisingWTTI`,
   - qui prend comme entrée une image (bruitée) I, un niveau de seuillage T et un nombre de translation m,
   - pour chaque $i \in 1, \ldots, m^2$,
     - détermine l'image translatée I_i,
     - calcule la transformée en ondelette W_i de I_i,
     - effectue le seuillage de W_i
     - inverse la transformée en ondelette et applique la translation inverse pour obtenir l'image débruitée J_i,
   - fait la moyenne des images J_i :
   $$J = \frac{1}{m^2} \sum_{i=1}^{m^2} J_i.$$
7. Pour différentes valeurs de m (exemple, $m = 1, 2, 4, 8$), calculer le SNR et le PSNR des images débruitées. Le résultat obtenu par la transformée invariante par translation est-il meilleur que celui obtenu par la transformée originale ?
8. Dans cette dernière question on essaie de faire du débruitage en suivant le même principe mais avec la transformée en cosinus discret (DCT) plutôt que la transformée en ondelette. Remplacez dans votre code la transformée en ondelette par la DCT. On pourra utiliser la commande `dct2_iv` de Wavelab qui est aussi la commande à utiliser pour inverser la DCT (Voir l'aide de la fonction).

4. Relation entre les filtres miroirs.

1. D'après le théorème vu en cours l'ondelette père est associé au filtre miroir h et l'ondelette mère est associée au filtre miroir g. Dans le cours on a exprimé $\widehat{g}$ en fonction de $\widehat{h}$. Exprimez directement les filtre discret g en fonction du filtre discret h.

5. Interprétation de la transformée en ondelettes dans le domaine fréquentiel.

1. On s'intéresse dans cette question au support fréquentiel (en fait le support en pulsation pour être plus rigoureux) des différentes parties de la transformée en ondelette. En

effet, la projection d'une fonction f sur les espaces V_j pour j grand correspondant au grandes échelles garde plutôt les basses fréquences de f et les hautes fréquences sont essentiellement encodées dans les espaces de détails W_j pour j très négatif. Pour être plus précis les grandes échelles gardent seulement les basses fréquences quand l'ondelette est régulière (Ce n'est donc pas le cas pour l'ondelette de Haar, les ondelettes D4, etc). Le cas où la coïncidence est pour ainsi dire parfaite est celle des ondelettes de Shannon. Pour comprendre l'opération effectuée en Fourier par les ondelettes, on revient donc sur celles-ci, d'abord dans le cas 1D.

Pour l'ondelette de Shannon on a $\widehat{\phi}(\omega) = 1_{[-\pi,\pi]}$. En déduire la fonction $\omega \mapsto \widehat{\phi}(2\omega)$. Ensuite, en utilisant la relation $\sqrt{2}\widehat{\phi}(2\omega) = \widehat{h}(\omega)\,\widehat{\phi}(\omega)$, déduisez la forme de la fonction $\widehat{h}(\omega)$. Rappelez pourquoi la fonction $\widehat{h}$ est nécessairement 2π-périodique. En utilisant l'expression de $\widehat{g}$ en fonction de $\widehat{h}$, en déduire $\widehat{\psi}$. Du coup, quel est le support fréquentiel de $\psi_{j,n}$? Déduire de cette analyse que la projection d'une fonction f sur chaque sous-espace W_j correspond à garder l'information de f pour une bande fréquentielle bien précise. Laquelle ? Faire un dessin des différents bandes fréquentielles.

2. On se penche maintenant sur la même question mais en 2D. Quelles sont les transformées de Fourier de ψ^1, ψ^2 et ψ^3 si l'ondelette père choisie est celle de Shannon ? En déduire leur support en pulsation bidimensionelle et représenter ces supports sur un dessin comme des régions du plan $\mathbf{R}^2$. Quels sont les support en pulsation de $\psi^k_{j,(n_1,n_2)}$? En déduire que la décomposition d'une fonction f sur les différents espaces W_j^2 correspond à l'information contenue dans f pour des ensembles de pulsation précis du plan $\mathbf{R}^2$. Représenter sur un dessin une partition de $\mathbf{R}^2$ formée des régions correspondantes et indiquer quelles régions sont associées aux ondelettes des types 1, 2 et 3.

Le compte-rendu est à envoyer au format pdf à guillaume.obozinski@imagine.enpc.fr avant le cours du 20 janvier.