TP 3 - Ondelettes et débruitage

Pour le 30/04/14

1 - Ondelettes de Haar

1. On suppose qu'un entier N est fixé et qu'il est de la forme $N = 2^J$, pour un certain $J \in \mathbb{N}^*$. Pour tout $j \in \{1, ..., J\}$ et pour tout $n \in \{0, ..., 2^{J-j} - 1\}$, on définit des signaux $\phi_{j,n}[k]$ et $\psi_{j,n}[k]$ avec $0 \leq k < N = 2^J$ par :

$$\begin{aligned}
\phi_{j,n}[k] &= 2^{-j/2} \text{ si } 2^j n \leq k < 2^j(n+1) \\
&= 0 \text{ sinon} \\
\psi_{j,n}[k] &= 2^{-j/2} \text{ si } 2^j n \leq k < 2^j n + 2^{j-1} \\
&= -2^{-j/2} \text{ si } 2^j n + 2^{j-1} \leq k < 2^j(n+1) \\
&= 0 \text{ sinon}
\end{aligned}$$

a) Calculer, pour tous j_1, j_2, n_1, n_2 tels que $j_1 \leq j_2$, $\langle \psi_{j_1,n_1}, \psi_{j_2,n_2} \rangle$ et $\langle \psi_{j_1,n_1}, \phi_{j_2,n_2} \rangle$. Calculer également $||\phi_{j,n}||_2$ pour tous j et n.

On définit maintenant, pour tout $j \in \{1, ..., J\}$ et tous $n_1, n_2 \in \{0, ..., 2^{J-j} - 1\}$, des signaux bidimensionnels $\psi^1_{j,n_1,n_2}[k_1, k_2], \psi^2_{j,n_1,n_2}[k_1, k_2], \psi^3_{j,n_1,n_2}[k_1, k_2]$ avec $0 \leq k_1, k_2 < N$:

$$\begin{aligned}
\psi^1_{j,n_1,n_2}[k_1, k_2] &= \phi_{j,n_1}[k_1]\psi_{j,n_2}[k_2] \\
\psi^2_{j,n_1,n_2}[k_1, k_2] &= \psi_{j,n_1}[k_1]\phi_{j,n_2}[k_2] \\
\psi^3_{j,n_1,n_2}[k_1, k_2] &= \psi_{j,n_1}[k_1]\psi_{j,n_2}[k_2]
\end{aligned}$$

On définit également :

$$\phi[k_1, k_2] = \phi_{J,0}[k_1]\phi_{J,0}[k_2] = 2^{-J}$$

b) Montrer que $\{\phi\} \cup \{\psi^1_{j,n_1,n_2}, \psi^2_{j,n_1,n_2}, \psi^3_{j,n_1,n_2}\}_{0<j\leq J,\, 0\leq n_1,n_2<2^{J-j}}$ est une base orthonormée de $\mathbb{R}^{N\times N}$.

2 - Transformée en ondelettes de Haar

Puisque l'ensemble considéré à la question précédente est une base orthonormée, tout signal $f \in \mathbb{R}^{N\times N}$ peut s'écrire sous la forme :

$$f = a_\phi[f]\phi + \sum_{j=1}^{J} \sum_{s=1,2,3} \sum_{n_1,n_2} a_{s,j,n_1,n_2}[f]\psi^s_{j,n_1,n_2}$$

où $a_\phi[f] = \langle f, \phi \rangle$ et $a_{s,j,n_1,n_2}[f] = \langle f, \psi^s_{j,n_1,n_2} \rangle$.
Dans les questions suivantes, on construit un algorithme qui calcule les coefficients a.

2. a) Montrer que, pour tous $n_1, n_2 \in \{0, ..., 2^J - 1\}$:

$$\begin{aligned}
a_{1,1,n_1,n_2} &= \frac{1}{2}\left(f[2n_1, 2n_2] + f[2n_1+1, 2n_2] - f[2n_1, 2n_2+1] - f[2n_1+1, 2n_2+1]\right) \\
a_{2,1,n_1,n_2} &= \frac{1}{2}\left(f[2n_1, 2n_2] - f[2n_1+1, 2n_2] + f[2n_1, 2n_2+1] - f[2n_1+1, 2n_2+1]\right) \\
a_{3,1,n_1,n_2} &= \frac{1}{2}\left(f[2n_1, 2n_2] - f[2n_1+1, 2n_2] - f[2n_1, 2n_2+1] + f[2n_1+1, 2n_2+1]\right)
\end{aligned}$$

b) Écrire une fonction `last_scale`, qui prend en argument un signal f, de taille $N \times N$, où N est de la forme 2^J avec $J \geq 1$, et renvoie les $a_{s,1,n_1,n_2}[f]$.

3. On définit :

$$g = a_\phi[f]\phi + \sum_{j=2}^{J} \sum_{s=1,2,3} \sum_{n_1,n_2} a_{s,j,n_1,n_2}[f]\psi^s_{j,n_1,n_2}$$

[Remarquer dans la somme le changement d'indexation sur j.]

a) Montrer que $g[2k_1, 2k_2] = g[2k_1, 2k_2+1] = g[2k_1+1, 2k_2] = g[2k_1+1, 2k_2+1]$ pour tous $0 \leq k_1, k_2 < 2^{J-1}$.

On note $\tilde{f}$ le signal tel que $\tilde{f}[k_1, k_2] = 2g[2k_1, 2k_2]$, pour $0 \leq k_1, k_2 < 2^{J-1}$.

b) Montrer que, pour tous k_1, k_2 :

$$\tilde{f}[k_1, k_2] = \frac{1}{2}\left(f[2k_1, 2k_2] + f[2k_1+1, 2k_2] + f[2k_1, 2k_2+1] + f[2k_1+1, 2k_2+1]\right)$$

c) Modifier la fonction `last_scale` pour qu'elle renvoie également $\tilde{f}$.

On note $\{\tilde{\phi}\} \cup \{\tilde{\psi}^1_{j,n_1,n_2}, \tilde{\psi}^2_{j,n_1,n_2}, \tilde{\psi}^3_{j,n_1,n_2}\}_{0<j\leq J-1,\, 0\leq n_1,n_2<2^{J-1-j}}$ les ondelettes de Haar de dimension $2^{J-1} \times 2^{J-1}$ (c'est-à-dire définies de la même façon que dans la première section, en remplaçant N par $\tilde{N} = 2^{J-1}$). Il s'agit d'une base orthonormée de $\mathbb{R}^{2^{J-1}\times 2^{J-1}}$.

4. a) Montrer que :

$$\tilde{f} = a_\phi[f]\tilde{\phi} + \sum_{j=1}^{J-1} \sum_{s=1,2,3} \sum_{n_1,n_2} a_{s,j+1,n_1,n_2}[f]\tilde{\psi}^s_{j,n_1,n_2}$$

b) Exprimer la transformée en ondelettes de Haar de $\tilde{f}$ en fonction de celle de f.

c) Écrire une fonction récursive, `wavelet_transform`, qui prend en entrée un signal f et renvoie en sortie $a_\phi[f]$ et les $a_{s,j,n_1,n_2}[f]$.

3 - Reconstruction à partir de la transformée en ondelettes

Dans ce paragraphe, on écrit l'inverse de la fonction `wavelet_transform` : on suppose les $a_{s,j,n_1,n_2}[f]$ et $a_\phi[f]$ connu et on souhaite reconstruire f.

5. Si $N = 1$, exprimer f en fonction de $a_\phi[f]$.

6. On suppose maintenant que f est de taille $N = 2^J$ avec $N \geq 1$. On définit $g, \tilde{f}$ comme à la question 3. et on pose :

$$h = f - g = \sum_{s=1,2,3} \sum_{n_1,n_2} a_{s,1,n_1,n_2}[f]\psi^s_{1,n_1,n_2}$$

a) Montrer que, pour tous $k_1, k_2 \in \{0, ..., 2^{J-1} - 1\}$:

$$h[2k_1, 2k_2] = \frac{1}{2}(a_{1,1,k_1,k_2}[f] + a_{2,1,k_1,k_2}[f] + a_{3,1,k_1,k_2}[f])$$
$$h[2k_1+1, 2k_2] = \frac{1}{2}(a_{1,1,k_1,k_2}[f] - a_{2,1,k_1,k_2}[f] - a_{3,1,k_1,k_2}[f])$$
$$h[2k_1, 2k_2+1] = \frac{1}{2}(-a_{1,1,k_1,k_2}[f] + a_{2,1,k_1,k_2}[f] - a_{3,1,k_1,k_2}[f])$$
$$h[2k_1+1, 2k_2+1] = \frac{1}{2}(-a_{1,1,k_1,k_2}[f] - a_{2,1,k_1,k_2}[f] + a_{3,1,k_1,k_2}[f])$$

b) Écrire une fonction `inverse_wavelet_transform`, qui prend en entrée des coefficients $a_{s,j,n_1,n_2}[f]$ et $a_\phi[f]$ et renvoie f.
[Indication : utiliser à nouveau la question 4.b).]

7. Tester les fonctions `wavelet_transform` et `inverse_wavelet_transform` : générer un signal f aléatoire de taille $N \times N$ avec $N = 512$ (par exemple un bruit blanc gaussien). Calculer sa transformée en ondelettes puis la reconstruction f_{rec} de f à partir de la transformée en ondelettes. Afficher la norme de $f_{rec} - f$:

$$\sqrt{\sum_{k_1,k_2} |f_{rec}[k_1, k_2] - f[k_2, k_2]|^2}$$

[Personnellement, pour un bruit blanc gaussien de variance 1, je trouve des valeurs de l'ordre de 10^{-13}. Si vous trouvez significativement plus, il y a probablement une erreur quelque part.]

4 - Débruitage d'une image

8. a) Charger l'image `batiment.png` (disponible à l'adresse `http://www.di.ens.fr/~waldspurger/tds/13_14_s2/batiment.png`), sous la forme d'un tableau `im`, de taille 512×512. A priori, c'est un tableau d'entiers. Le convertir en un tableau de flottants par l'instruction `im = double(im)`.
[Attention, si vous utilisez Matlab, l'image sera chargée sous la forme d'un tableau à 3 dimensions. Après l'avoir converti en un tableau de flottants, utilisez `im = mean(im,3)` pour le transformer en un tableau à seulement deux dimensions.]
b) Définir un tableau `noise` de même taille que `im`, dont chaque coordonnée est une réalisation d'une loi gaussienne centrée de variance $\sigma^2 = 100$.
c) Définir `im_noisy = im + noise`.
d) Afficher `im` et `im_noisy`.
[Pour que les images soient affichées en niveaux de gris, exécuter après chaque instruction d'affichage la commande `colormap(gray(256))`. Utiliser pour l'affichage `imagesc(im,[0,255])`.]

9. Dans cette question, on crée une fonction `denoise`, qui prend en entrée une image bruitée `im_noisy` et en renvoie une version débruitée `im_rec`. L'algorithme est le suivant :
a) Calculer les coefficients a_ϕ et a_{s,j,n_1,n_2} de la transformée en ondelettes de Haar de `im_noisy`.

b) On définit $\tau = 4\sigma$. Calculer les coefficients a'_ϕ et a'_{s,j,n_1,n_2} tels que :

$$\begin{aligned} a'_{s,j,n_1,n_2} &= a_{s,j,n_1,n_2} \text{ si } |a_{s,j,n_1,n_2}| \geq \tau \\ &= 0 \text{ sinon} \\ a'_\phi &= a_\phi \text{ si } |a_\phi| \geq \tau \\ &= 0 \text{ sinon} \end{aligned}$$

c) Renvoyer le signal `im_rec` dont la transformée en ondelettes de Haar a pour coefficients a'_ϕ et les a'_{s,j,n_1,n_2}.
10. a) Afficher la version débruitée de l'image `im_noisy` définie dans la question 8.c).
b) Normalement, vous devriez observer dans l'image débruitée des discontinuités selon des portions de lignes horizontales et verticales. À quoi sont-elles dûes ?

5 - Amélioration

11. Pour remédier au problème observé dans la question précédente, on définit une fonction `denoise2` qui prend à nouveau en entrée une image bruitée `im_noisy` et renvoie en sortie une version débruitée, plus satisfaisante que la précédente.
a) Pour tous les entiers l_1, l_2 tels que $-4 \leq l_1, l_2 \leq 4$, effectuer le calcul suivant : translater `im_noisy` de l_1 pixels dans la direction verticale et l_2 dans la direction horizontale (de manière circulaire). Débruiter l'image translatée à l'aide de la fonction `denoise`. Translater l'image débruitée de $-l_1$ pixels dans la direction verticale et $-l_2$ dans la direction horizontale.
b) Faire la moyenne des 81 images obtenues de cette manière. Renvoyer ce résultat.

12. a) Afficher l'image `im_noisy` définie dans la question 8.c), débruitée par la fonction `denoise2`.
b) Quelles sont les zones de l'image où l'effet du débruitage n'est pas satisfaisant ? Pouvez-vous expliquer ce qui pose problème dans ces zones ?