TP7 : décomposition pyramidale et initiation aux ondelettes.

Fabien PIERRE
fabien.pierre@math.u-bordeaux1.fr
http://sites.google.com/site/fabienpierre/enseignements

Automne 2013.

1 Décomposition pyramidale.

On décrit l'algorithme pyramidal de la manière suivante :

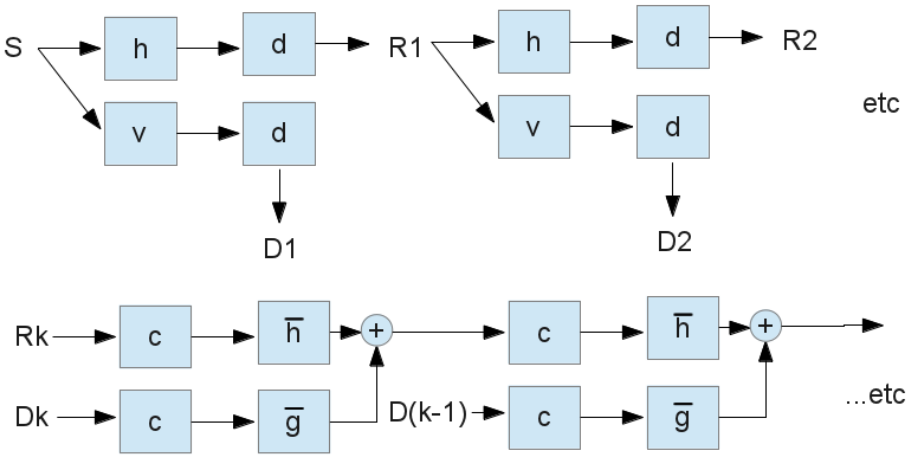

FIGURE 1 – Schéma de bloc de l'algorithme pyramidal et reconstruction.

h représente un filtre linéaire et g son filtre miroir. $\overline{h}$ et $\overline{g}$ sont les versions retournées des deux filtres précédents. d est un opérateur de décimation et c est l'opérateur inverse de la décimation qui comble les données manquantes par 0. R_k s'appelle le résumé et D_k le détail d'ordre k.

Le résultat d'une telle transformation est conventionnellement classé de la manière suivante :

$$(R_k, D_k, D_{k-1}, \ldots, D_1).$$

La construction de tels filtres, efficaces en traitement du signal et de l'image, est le sujet d'une UE de Master 2. Ici on utilisera les filtres de réponse impulsionnelle

$$h = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \ , \ g = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \ , \ \overline{h} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \ , \ \overline{g} = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right).$$

Ces filtres correspondent à l'ondelette de *Haar*, et la transformation associée s'appelle *transformation en ondelette de Haar*. La représentation dans le domaine de cette transformation est particulièrement adaptée aux signaux constants par morceaux.

2 Implémentation de la transformée en ondelettes de Haar.

– Implémenter une telle transformation et son inverse dans deux fonctions séparées. Une première prendra en entrée le signal et le niveau de décomposition et retournera la transformée en ondelettes. La deuxième prendra en entrée la transformée en ondelettes et le niveau de décomposition et retournera le signal reconstruit.

3 Utilisation élémentaire des ondelettes en traitement du signal.

– Créer un signal `'Blocks'` avec la fonction `MakeSignal`.
– Afficher sa transformée en ondelette. Commentez. Comparer avec la transformation du signal `Piece−Regular`.
– Afficher une approximation linéaire de `Piece−Regular` en mettant à zéro les coefficient des plus petits détails de la transformation en ondelette de Haar. Observer. Commenter.
– Ajouter un bruit gaussien au signal `'Blocks'`. Débruiter par un seuillage doux de la transformée en ondelettes de Haar pour un seuil bien choisi. Commenter. Comparer avec la même opération effectuée avec la DCT.
– Réaliser les mêmes opérations sur `Piece−Regular`. Commenter.

4 Inpainting d'un signal constant par morceaux.

Nous n'aborderons pas en détail la théorie permettant de justifier la construction de cet algorithme, nous donnons seulement ici une utilisation de la transformée en ondelette de Haar.

L'inpainting vise à retrouver des données perdues d'un signal. On peut justifier heuristiquement cet algorithme en remarquant que la transformée en ondelette de Haar d'un signal constant par bloc contient beaucoup de zéros, il suffit alors de tronquer la transformée en ondelette pour éliminer les petites composantes et ne conserver que les plus grandes.

On suppose que l'on a l'image masquée i.e. l'image donnée par l'opérateur de masquage défini comme suit :

$$M(I_{i,j}) = \begin{cases} I_{i,j} & \text{si l'image est non masquée,} \\ 0 & \text{sinon.} \end{cases}$$

L'algorithme est le suivant :

Algorithme 1 Iterative soft thresholding.

$\gamma > 0$
$x_0 \leftarrow$ Image masquée.
for $n > 0$ **do**
    $x_n \leftarrow SD\left[x_n + A^t\left(y - Ax_n\right), \gamma/2\right]$
end for
Résultat $= Ax_n$

Avec $A^t = \mathrm{WT}\,.M$ et $A = M.\,\mathrm{IWT}$, M l'opérateur de masquage et $SD[x, \gamma]$ le seuillage doux de x par le seuil γ. WT et IWT représentent respectivement la transformée en ondelettes et son inverse.

– Créer un signal `'Bolcks'` avec la fonction `MakeSignal`.
– Créez le masque et l'image masquée en masquant 50% des valeurs du signal aléatoirement.
– Mettez en place l'algorithme de seuillage doux itéré.