

Part I

Algorithmes d'encodage et de décodage

1 Cadre théorique

D'un point de vue pratique, les fonctions sont représentées comme des fonctions en escalier, qui de plus sont à support bornés (disons sur $[0, 1]$). C'est-à-dire que l'on manipule des fonctions $f \in L^2(\mathbb{R})$ de la forme

$$f = \sum_{j=0}^{2^{N_0}-1} a_j \mathbb{1}_{[2^{-N_0}j, 2^{-N_0}(j+1)[}$$

Le calcul des coefficients s'en retrouve ainsi simplifié :

$$\begin{aligned} \langle f, \psi_{n,k} \rangle &= \int_0^1 f(t) \overline{\psi_{n,k}(t)} dt \\ &= \sum_{j=0}^{N-1} \int_j^{j+1} a_j \overline{\psi_{n,k}(t)} dt \\ &= \sum_{j=0}^{N-1} \int_j^{j+1} a_j \overline{\frac{1}{\sqrt{2^n}} \psi(2^n t - k)} dt \\ &= \sum_{j=0}^{N-1} a_j \frac{1}{\sqrt{2^{3n}}} \int_{2^n j - k}^{2^n(j+1) - k} \overline{\psi(u)} du \quad \text{où } 2^n t - k = u \\ &= \sum_{j=0}^{N-1} a_j \frac{1}{\sqrt{2^{3n}}} (F(2^n j - k) - F(2^n(j+1) - k)) \end{aligned}$$

où F est une primitive de $\overline{\psi}$ et $N = 2^{N_0}$.

Étant face à des fonctions en escalier avec un pas fixe, on peut les représenter comme des vecteurs de \mathbb{R}^N , où la $(j+1)$ -ième coordonnée est la valeur prise sur le $(j+1)$ -ième intervalle. On note $\tilde{f} = (a_j, a_1 \cdots a_N)$ et $\psi_{n,k} = \left(\frac{1}{\sqrt{2^{3n}}} (F(2^n j - k) - F(2^n(j+1) - k)) \right)_{j=0,1 \dots N-1}$, on a alors

$$\langle f, \psi_{n,k} \rangle = \tilde{\psi}_{n,k}^t \tilde{f}$$

De plus, en définissant $S_n = (\langle f, \psi_{n,k} \rangle)_{k=0,1 \dots s(n)}$, avec $s(n)$ le plus grand k que l'on souhaitera calculer, et

$$P_n = \begin{pmatrix} \tilde{\psi}_{n,0} \\ \tilde{\psi}_{n,1} \\ \vdots \\ \tilde{\psi}_{n,s(n)} \end{pmatrix}$$

on a enfin

$$S_n = P_n {}^t \tilde{f}$$

Chaque P_n est alors le projecteur de f sur l'espace de détails W_n , les coefficients sont alors stockés dans le vecteur S_n .

Remarque 1. La matrice \tilde{f} est de taille $1 \times N$, P_n de taille $s(n) \times N$ et S_n de taille $1 \times s(n)$.

L'algorithme de compression consiste alors en le remplissage des matrices de projection pour chaque niveau de détail et en la multiplication pour obtenir chaque S_n . Nous proposons l'algorithme naïf suivant :

```

1: procedure ENCODER( $F, \tilde{f}, N_0$ )
2:    $P_0 = \text{Construire}(F, 0)$ 
3:   for  $n = 1, 2 \dots N_0$  do
4:      $P_n = \text{Construire}(F, n)$ 
5:      $S_n = P_n {}^t \tilde{f}$ 
6:      $S_{-n} = P_{-n} {}^t \tilde{f}$ 
7:      $P_{-n} = \text{Construire}(F, -n)$ 
8:   end for
9:   return  $(P_0, P_{-1}, P_1 \dots P_{N_0}, P_{-N_0})$ 
10: end procedure

11: procedure DÉCODER( $\psi, A = (\alpha_{n,k})_{n,k}, N_0, s(n)$ )
12:   for  $x = 0, 2^{-1} \dots 2^{-N_0}$  do
13:      $f(x) = 0$ 
14:     for  $n = 0, 1 \dots N_0$  do
15:       for  $k = 0, 1 \dots s(n)$  do
16:          $f(x) = f(x) + \alpha_{n,k} \times \psi(2^{-n}x - k)$ 
17:       end for
18:     end for
19:   end for
20:   return  $f$ 
21: end procedure

```

1.1 Validation de l'algorithme

Vérifier que cet algorithme est correct consiste à vérifier que les approximations que l'on a du faire pour passer de valeurs infinies à des valeurs finies pour n et k ne sont pas préjudiciables à la validité des résultats produits. Il s'agit donc de vérifier que les valeurs de n et k enlevées sont négligeables. En effet, le reste de l'algorithme ne doit pas donner lieu à des erreurs : la formule utilisée est celle qui a été définie plus tôt, et il est certain de terminer.

1.1.1 Coordonnées $n \leq 0$

La valeur de $\langle f, \psi_{n,k} \rangle$ représente à peu près la variation dans l'intervalle $[2^n k, 2^n(k+1)]$.

Étant donné que f est échantillonnée avec une fréquence de 1, f est constante sur les intervalles $[2^n k, 2^n(k+1)]$ pour $n \leq 0$. On néglige donc la valeur des coordonnées à ces indices. Dans la pratique, cela correspond à des

intervalles qui sont plus petits que la fréquence d'échantillonnage: la fonction est donc constante sur l'intervalle, et comme l'ondelette est de moyenne nulle, les valeurs qui lui sont associées sont bien négligeables.

1.1.2 Coordonnées $n > \lceil \log_2 N \rceil$

De même que précédemment, $\langle f, \psi_{n,k} \rangle$ représente la variation dans l'intervalle $[2^n k, 2^n(k+1)]$. Or, cela veut dire que $2^n > 2^{\lceil \log_2 N \rceil} \geq N$, c'est à dire que les intervalles sont plus grands que le domaine de définition de la fonction. Dès lors, ils ne seront pas utiles.

1.1.3 Coordonnées k

On veut obtenir un recouvrement minimal de $[0, N]$ avec les intervalles disjoints $[2^n k, 2^n(k+1)]$. Pour que $[2^n k, 2^n(k+1)] \cap [0, N] \neq \emptyset$, il faut que $0 \leq k < 2^{-n} N$.

Exemple 1. Nous allons décomposer la fonction $f(t) = t$ définie sur $[0, 1]$ dans la base de l'ondelette de Haar, la famille donnée par

$$\psi_{n,k}(t) = \begin{cases} \sqrt{2^n} & 2^{-n}k \leq t < 2^{-n}(k+1/2) \\ -\sqrt{2^n} & 2^{-n}(k+1/2) \leq t < 2^{-n}(k+1) \end{cases}$$

Les coefficients sont

$$\begin{aligned} \alpha_{n,k} &= \int_0^1 \psi_{n,k}(t) f(t) dt \\ &= \sqrt{2^n} \int_{2^{-n}k}^{2^{-n}(k+1/2)} t dt - \sqrt{2^n} \int_{2^{-n}(k+1/2)}^{2^{-n}(k+1)} t dt \\ &= \sqrt{2^n} \frac{t^2}{2} \Big|_{2^{-n}k}^{2^{-n}(k+1/2)} - \sqrt{2^n} \frac{t^2}{2} \Big|_{2^{-n}(k+1/2)}^{2^{-n}(k+1)} \\ &= \frac{1}{\sqrt{2}^{3n+4}} \end{aligned}$$

2 Version alternative

[IDEE : économiser des coefficients]

Si l'on ne souhaite pas utiliser de coefficients pour des n négatifs, on peut l'éviter en intégrant φ dans la décomposition. En reprenant les calculs de la preuve XXX, on peut aisément retrouver

$$\bigcup_{n \in \mathbb{Z}} V_n = V_0 \oplus \left(\bigoplus_{n \in \mathbb{N}} W_n \right)$$

Autrement dit, on peut décomposer tout fonction $f \in L^2(\mathbb{R})$ ainsi :

$$f = \sum_{k \in \mathbb{Z}} \langle \varphi_k, f \rangle \varphi_k + \sum_{\substack{k \in \mathbb{Z} \\ n \in \mathbb{N}}} \langle \psi_{n,k}, f \rangle \psi_{n,k}$$

2.1 Optimisation et compression

L'intérêt de ce type d'algorithme est de pouvoir compresser les informations pour encoder le signal en perdant le moins d'informations possible. Pour cela, on peut agir à deux niveaux.

2.1.1 Niveaux de détail

Le premier point sur lequel on peut travailler est le niveau de détail. Lorsqu'on observe l'algorithme, on peut voir que les coefficients pour les premières valeurs de n sont moins nombreux, du fait qu'il suffise de quelques ondelettes pour parcourir le domaine de définition. Ce sont ces coefficients qui donnent la structure générale du signal et sont plus impactant que les niveaux de détails suivants. On peut donc penser qu'il est possible d'arrêter l'algorithme plus après moins d'itérations que ce que l'on a défini précédemment car les détails suivant seront négligeables.

2.1.2 Encodage des coefficients

Un autre point sur lequel on peut travailler est l'encodage des coefficients, ou tout du moins leur niveau de précision. En effet, il n'est pas forcément utile d'avoir une grande précision sur la valeur du coefficient. De plus, à partir d'un certain rang, comme on l'a pensé précédemment, la plupart des coefficients sont négligeables. Pour optimiser le volume des coefficients formés, il peut être utile de ne stocker que les coefficients non nuls, qui seront bien moins nombreux.