

II.1 Introduction

La compression des données est aujourd'hui devenue indispensable dans le cadre de la transmission et de l'archivage des données. Deux grandes familles d'algorithmes de compression existent : ceux qui peuvent reconstituer l'information exacte (algorithmes sans pertes) et ceux qui tolèrent une perte d'information (algorithmes avec pertes).

L'enjeu de la recherche sur la compression des signaux ECG est de trouver un moyen de diminuer la taille d'une image, tout en essayant de limiter la dégradation due à la compression. Pour cela, nous allons le long de ce chapitre étudier les différentes méthodes de compression.

II.2 Types de compression

Les techniques de compression sont classées dans deux types :

1. compression sans pertes d'informations (sans distorsion ou réversibles).
2. compression avec pertes d'informations (avec distorsion ou irréversibles).

Les expérimentations menées montrent que généralement les méthodes qui atteignent des taux de compression très élevés sont les méthodes avec distorsion. Ils codent et compriment, mais reforment un signal altéré par rapport à l'originale. Par contre, les méthodes sans distorsion engendrent des taux de compression très faibles codent et compriment puis permettent de décompresser en reconstituant exactement le fichier d'origine et ne sont utilisées que dans des applications sensibles telles que les images médicales et les images satellites.

II.2.1 Compression sans pertes

Puisque non destructives, les compressions du même nom ne modifient en rien le signal : ils la codent. C'est-à-dire que tous les 0 et 1 ou plus exactement les séquences de 0 et 1 qui décrivent l'image vont être codés. Les modes les plus employés sont : RLE (Run Length Encoding), LZW (Lempel Ziv Welch), Shanon-Fano, codage Huffman et le codage Arithmétique [12].

II.2.1.1 Codage RLE

Cette méthode se base sur le comptage de motifs identiques dans la plage des données étudiée. Ainsi, plus les motifs sont répétitifs, meilleurs sont les résultats obtenus. RLE est l'une des méthodes les plus anciennes, les plus simples et la plus utilisée. Tout son secret consiste à identifier et supprimer des redondances d'informations en les codant sous une

forme plus compacte. RLE s'emploie à réduire la taille physique d'une répétition de chaîne de caractère. Cette chaîne répétée est appelée un passage (Run) et est typiquement codée avec deux bytes. Le premier byte représente le nombre de caractères dans le passage et est appelé le compteur de passage (Run Count). Il peut prendre une valeur comprise entre 0h et 128h ou 256h. Le second byte est la valeur du caractère dans le passage qui peut prendre la valeur 0h à FFh. Ce dernier byte est appelé la valeur du passage (run value). Non compressé, le passage comprenant 15 caractères "A" devrait normalement prendre 15 bytes à stocker AAAAAAAAAAAAAAAAAA La même chaîne après codage ne prend plus que 2 bytes "15A". Le "15A" généré pour représenter la chaîne de caractère est appelé un paquet RLE (RLE packet). Ici, le compteur de passage contient le nombre de répétitions soit 15. Le deuxième byte "A", la valeur du passage, contient la valeur répétée dans le passage Un nouveau paquet est généré à chaque fois que le caractère change ou chaque fois que le nombre de caractère dans le passage excède la valeur maximum que peut prendre le compteur. Si l'on suppose que notre chaîne de 15 caractères contient maintenant quatre passages différents : AAAAAAbbbbXXXXXt

En utilisant le codage RLE, cette chaîne pourra être compressée en quatre paquets de deux bytes : 6A3b5X1t

Ainsi, après le codage, la chaîne de 15 bytes de départ prendra seulement 8 bytes de données pour représenter la chaîne. Le codage RLE est simple et efficace, mais cette efficacité dépend fortement du type de données à encoder, il se peut même que les données compressées prennent plus de place que les données originales.

II.2.1.2 Méthode LZW

L'algorithme LZW (Lempel Ziv Welch) est semblable à la méthode RLE, mais appliquée à une suite d'octets. C'est un algorithme de compression puissant car il est capable de travailler sur presque n'importe quel type de données. Il est rapide aussi bien en phase de compression que de décompression.

LZW réduit la taille des chaînes de caractères (c'est à dire les mots récurrents). Cette méthode est peu efficace pour les images mais donne de bons résultats pour les textes et les données informatiques, en général le taux de compression est de l'ordre de 50%.

Cet algorithme utilise une bibliothèque c'est-à-dire une table de données contenant des chaînes de caractères. Au cours du traitement de l'information, les chaînes de caractères sont placées une par une dans une bibliothèque. Lorsqu'une chaîne est déjà présente dans la bibliothèque, son code de fréquence élevée est remplacé par un «mot» ayant un nombre de

caractères le plus petit possible et le code de correspondance est inscrit dans la bibliothèque. On obtient ainsi une information codée et sa bibliothèque [13].

II.2.1.3 Technique de Shanon-Fano

Les principaux pionniers de la compression de données furent sans doute C. Shanon et R. M. Fano. Ils démontrent que l'on pouvait coder un message en utilisant la probabilité d'apparition d'un symbole ainsi que la base 2. Ils ont établi trois propriétés qui permettent de compresser un message :

- Chaque code (ou presque) a un nombre de bits différent.
- Le code correspond a un symbole dont la probabilité d'apparition dans le message est faible sera codé avec plus de bits, le code dont le symbole respectif a une plus grande probabilité d'apparaître aura un nombre de bits moins élevées.
- Même si les codes ont des longueurs différentes, cela ne doit pas empêcher de pouvoir les décoder de façon distincte.

Nous voyons que pour reconstituer les codes attribués à chaque caractère, il faut parcourir l'arbre depuis la racine jusqu'à ce que nous arrivions à une feuille représentant un caractère. L'algorithme de Shanon-Fano n'utilise pas un arbre pour créer les codes mais s'en sert uniquement d'une table dont les entrées auront été préalablement triées en fonction de la fréquence d'apparition des symboles. L'algorithme de Shanon-Fano est des plus simples:

- Dresser une table triée par ordre croissant des fréquences d'apparition des symboles.
- Diviser cette table en deux parties. la somme des fréquences de la deuxième partie.
- Affecter le chiffre binaire 0 à la moitié supérieure, la moitié inférieure prenant le chiffre binaire 1.
- Répéter les opérations 2 et 3 aux parties, jusqu'à ce que chaque symbole ne représente plus qu'une partie de la table [14].

II.2.1.4 Méthode d'Huffman

C'est un principe très intéressant et assez performant. Il s'agit non plus de coder nos données sur un nombre fixe de bits mais de coder les valeurs qui reviennent souvent sur peu de bits et au contraire sur beaucoup plus les valeurs rares. L'algorithme d'Huffman est complètement différent de celui de Shanon-Fano (on part toujours d'une table dont les entrées sont triées selon les fréquences d'apparition des caractères). Chaque ligne de la table

correspond a un nœud de l'arbre d'Huffman dont le poids est égal au nombre d'apparition du caractère dans le message:

- Identifier les deux nœuds de plus faible poids.
- On crée un nœud parent dont le poids est égal à la somme des deux poids des nœuds identifiés en 1. On rajoute ce nœud dans la table et on élimine les deux nœuds fils.
- Un des deux nœuds fils est désigné pour décoder un bit 0 et l'autre un bit 1 [15].

II.2.1.5 Codage arithmétique

Le codage arithmétique est un codage récent utilisant un modèle statistique, tout comme le codeur de Huffman. Contrairement à ce dernier, il produit un code pour une séquence de symboles tout entière, et non pas un code par symbole. Chaque nouveau symbole lu modifie de façon incrémentale le code de sortie. Ce code de sortie est un nombre à virgule flottante compris entre 0 et 1, dont le nombre de chiffres après la virgule correspond au nombre de symboles. Contrairement à Huffman, il n'est pas obligatoire que chaque code ait un nombre entier de bits. Par exemple un symbole de probabilité 0.9 a pour entropie 0.15, mais Huffman affectera probablement un code de un bit (ou plus), et la séquence codée aura un nombre de bits plus long qu'en théorie.

Le codeur arithmétique est plus performant que le codeur de Huffman, mais il est plus complexe à implémenter [13].

II.2.2 Compression avec pertes

Certains types de données peuvent bien se comprimer si l'on accepte de perdre une partie de l'information. C'est notamment le cas de sons ou d'images numérisés. Les taux de compression peuvent alors être énormes (par exemple un taux de 80 % pour une image).

Une image reconstruite après une compression avec perte contient des dégradations relatives à l'image originale qui sont généralement dues à l'élimination complète des redondances. Cependant les méthodes de compression avec perte sont capables de réaliser des taux de compression très élevés produisant des images de qualité acceptable pour l'œil humain ; on peut citer la quantification. D'autres approches de la compression avec perte, sont basées sur le codage des coefficients dans le plan transformé ; on peut citer JPEG basé sur la DCT (Discret Cosine Transform) qui est utilisé jusqu'à présent et cela depuis quelques années en raison de ses performances et de la facilité de son implémentation sur micro-

ordinateurs. En cherchant à améliorer les performances de la compression d'images avec perte, d'autres transformations ont été utilisées, telles que (la transformée en ondelettes et fractales) [16].

II.2.2.1 Quantification scalaire

La quantification scalaire consiste à remplacer un nombre très grand de symboles par un nombre restreint de codes. C'est une opération irréversible très largement employée en compression. Par exemple, le fait d'arrondir des nombres flottant à l'entier le plus proche est une quantification scalaire. En général, un quantificateur scalaire est une fonction en escalier. L'intervalle de toutes les valeurs possibles de symboles d'entrée est divisé en " n " intervalles adjacents appelés intervalles de décision. Chaque symbole appartient à l'un de ces intervalles, et sa valeur codée correspond au numéro de son intervalle. Tous les symboles d'entrée appartenant à un même intervalle sont codés et reconstruits avec la même valeur. La valeur reconstruite est déduite directement des bornes de l'intervalle d'entrée. Pour cela on essaie d'adapter le mieux possible la position des bornes des intervalles de décision et la règle de reconstruction aux propriétés de la séquence de symboles. Tout le problème consiste à minimiser la distorsion introduite par la quantification tout en restreignant au maximum le nombre de codes.

II.2.2.2 Quantification vectorielle

La quantification vectorielle découpe la matrice de symboles (pixels ou coefficients du plan des fréquences) en petits blocs. Chaque bloc est codé par son indice dans un dictionnaire de blocs types. Le codage consiste à repérer dans le dictionnaire des mots-codes disponibles celui qui ressemble le plus au bloc d'entrée. Le critère consiste en général à minimiser l'erreur quadratique. Le décodage consiste simplement à remplacer l'indice par le mot-code correspondant dans le dictionnaire. La création du dictionnaire de bloc types est une étape préalable et cruciale. Il est construit par apprentissage à l'aide d'un ensemble d'images type à compresser, ou à l'aide d'un treillis qui utilise les propriétés de la distribution statistique des symboles à coder [17].

II.3 Compression par transformations

Dans la pratique, la plupart des signaux sont des signaux dépendant du temps (du domaine Temporel) sous leur format brut. La représentation du signal est une représentation

temps-amplitude. Cette représentation n'est pas toujours la meilleure pour la plupart des applications de traitement du signal. Dans beaucoup de cas, l'information la plus pertinente est cachée dans la composante de fréquence du signal. Le spectre de fréquence d'un signal est constitué par les composantes de fréquence de ce signal. Le spectre de fréquence d'un signal indique quelles sont les fréquences qui existent dans le signal.

Le principe de la compression par transformation est de décomposer par exemple dans le cas des images les pixels fortement corrélés en ensembles de coefficients spectraux partiellement décorrélés, dont l'énergie est concentrée dans un nombre restreint de coefficients. Ce compactage de l'énergie permet d'affecter en priorité aux coefficients spectraux les plus énergétiques un nombre de bits plus élevé qu'aux autres [18].

II.3.1 Transformée en Cosinus Discrète (DCT)

C'est une méthode proposée en 1974 par le professeur Rao de l'université du Texas en 1974. La DCT peut-être vue comme une version discrète des séries de Cosinus de Fourier. Par ses propriétés, elle est plus proche de la transformée optimale «KLT » (Karhunen –Loève Transform) calculée en utilisant une transformée de Fourier rapide « FFT » (Fast Fourier Transform). A la différence de la FFT, la DCT produit des coefficients à valeur réelle et fournit une meilleure approximation du signal avec moins de coefficients. La DCT c'est une transformation mathématique qui transforme un ensemble de données d'un domaine spatial en un spectre de fréquence et inversement (DCTI). C'est la plus utilisée parmi les transformations citées. Elle permet schématiquement de changer d'échelle de mesure, passant d'une échelle définissant un pixel en fonction de sa position en x et en y à une échelle définissant la fréquence d'apparition de ce pixel dans un bloc de pixels. En effet, il est dès lors possible de supprimer des informations sans pour autant altérer le résultat final, contrairement à un bloc de pixels où la disparition brute de plusieurs éléments est immédiatement visible. La DCT est effectuée sur une matrice carrée $N \times N$ de valeurs de pixels et donne une matrice carrée $N \times N$ de coefficients de fréquence. Le temps de calcul requis pour chaque élément dans la DCT dépend de la taille de la matrice. Vu la difficulté d'appliquer la DCT sur la matrice entière, celle-ci est décomposée en blocs de taille 8×8 pixels. A la sortie de la matrice de la DCT, la valeur de la position (0,0) est appelée le coefficient continu. Cette valeur représente une moyenne de la grandeur d'ensemble de la matrice d'entrée. Ce coefficient est plus grand d'un ordre de grandeur à toute valeur dans la matrice de la DCT, par convention, les 64 valeurs transformées (de chaque bloc) sont

positionnées d'une certaine manière, Ainsi la valeur moyenne de tous ces coefficients est placée en haut à gauche de ce bloc. Plus on s'éloigne des coefficients continus plus leur grandeurs tendent à diminuer. Ce qui signifie que la DCT concentre la représentation de l'image en haut à gauche de la matrice de sortie, les coefficients en bas et à droite de cette matrice contiennent moins d'information utile. Les équations qui suivent, donnent respectivement la transformée en cosinus discrète directe et inverse.

Transformée Directe

$$F(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u \pi}{2N} \right] \cos \left[\frac{(2y+1)v \pi}{2N} \right] \quad (\text{II.1})$$

Pour $u, v = 0, 1, 2, \dots, N-1$.

Transformée Inverse

$$F(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) f(u, v) \cos \left[\frac{(2x+1)u \pi}{2N} \right] \cos \left[\frac{(2y+1)v \pi}{2N} \right] \quad (\text{II.2})$$

Pour $u, v = 0, 1, 2, \dots, N-1$.

Avec

$$\alpha(u) = \begin{cases} \left(\frac{1}{N}\right)^{\frac{1}{2}} & \text{pour } u = 0; \\ \left(\frac{2}{N}\right)^{\frac{1}{2}} & \text{pour } u = 0, 1, 2, \dots, N-1 \end{cases} \quad (\text{II.3})$$

$f(x, y)$: Valeur de l'image initiale pour x et y donnés.

$F(u, v)$: Coefficients de la DCT.

N : Taille d'un bloc [19].

II.3.2 Transformée en ondelettes

Cet outil mathématique s'ajoute aux méthodes classiques d'analyse du signal. Il met l'accent sur les caractéristiques importantes du signal et semble en outre correspondre à des réalités physiques du traitement des signaux acoustiques et chez l'homme [21][22]. Quand on

cherche à analyser un signal, il est très fréquent qu'on établisse, de manière explicite ou implicite, une représentation temps-fréquence de ce signal. L'image qu'on peut avoir d'une telle opération est celle de la transcription d'une partition de musique, qui indique au musicien les notes (donc l'information fréquentielle) qu'il doit produire à un moment donné. La transformée de Fourier n'est pas l'outil approprié pour mener cette analyse puisqu'elle masque l'évolution temporelle du signal. Par contre, comme nous le montrerons, la Transformée en ondelettes et ses extensions fournissent des solutions intéressantes dans ce contexte. Les ondelettes sont issues de l'intuition d'un ingénieur en géophysique, J. Morlet, dans les années 1980.

Les ondelettes se sont imposées comme des outils fondamentaux de l'analyse harmonique moderne. D'un point de vue applicatif, les ondelettes ont eu une influence importante dans divers domaines : physique, analyse numérique (par exemple, pour la résolution d'équations aux dérivées partielles, statistiques, traitement du signal et des images, vision par ordinateur,...

Dans le contexte de traitement du signal dans lequel nous nous placerons, le lien existant entre les décompositions en ondelettes et des outils plus traditionnels que sont les bancs de filtres, confère une certaine légitimité à ces transformations. Les bancs de filtres considérés agissent en divisant le spectre des signaux de manière logarithmique et constituent ainsi d'assez bonnes approximations du mode de fonctionnement des systèmes perceptuels visuel ou auditif humains. Les ondelettes et les techniques multirésolutions ont connu un grand succès en traitement d'images pour des problèmes tels que l'estimation de mouvement, la reconnaissance de formes, la recherche dans des bases de données et la transmission progressive d'informations. La propriété essentielle qui est exploitée dans ces applications est la possibilité d'approximer les signaux à plusieurs échelles, en partant d'une vue « grossière » qu'on vient affiner au cours de traitements successifs [21][22].

La plupart des signaux ne sont pas stationnaires, l'essentiel de l'information qu'ils contiennent réside dans cette non stationnarité. L'analyse de Fourier propose une approche globale du signal. Toute notion temporelle dans l'espace de Fourier (espace fréquentiel) disparaît. Il faut trouver une transformation qui nous renseigne sur le contenu fréquentiel tout en préservant la localisation afin d'obtenir une représentation temps / fréquence.

Plusieurs solutions ont été proposées [22]. Ces solutions sont : la transformée de Fourier à fenêtre glissante, et la transformée de Gabor. Mais ces deux méthodes donnent une même résolution temporelle pour les hauts et les basses fréquences. Donc l'analyse n'est pas idéale.

C'est dans ce contexte qu'intervient la transformée en ondelettes qui propose une solution de compromis entre la résolution temporelle et la résolution fréquentielle [22][21].

Une Ondelette mère Ψ est une fonction f de base que l'on va translater et dilater pour recouvrir le plan temps fréquences et analyser le signal [22,24]. On peut définir la transformée en Ondelettes d'un signal $f(t)$ comme une projection sur la base des fonctions ondelettes $\psi(\frac{t-b}{a})$:

$$TO(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \cdot \psi\left(\frac{t-b}{a}\right) dt \quad \text{avec } a, b \in \mathbb{R}, a \neq 0 \quad (\text{II.4})$$

Dans cette expression, a est le facteur d'échelle, b est le paramètre de translation. En posant

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (\text{II.5})$$

Où $\psi_{(a,b)}(t)$ sont une famille d'ondelettes analysantes générales à partir d'une ondelette « mère » $\psi(t)$ par dilatation (facteur a) et translation (paramètre b). L'équation (2.1) devient : [22][25].

$$TO(a, b) = \int_{-\infty}^{+\infty} f(t) \cdot \psi_{a,b}(t) dt = \langle f, \psi_{a,b} \rangle \quad (\text{II.6})$$

$\langle g, f \rangle$: produit scalaire entre deux fonctions g et f .

Le changement d'échelle sert à compresser ou dilater l'onde mère, ce qui même a analyser les hautes fréquence ou les basse respectivement continu dans un signal [21].

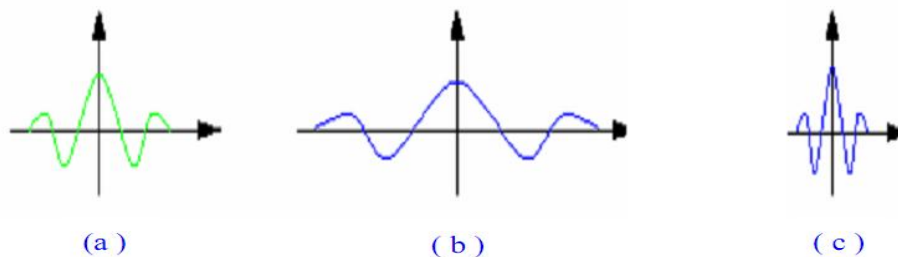


Figure II.1 : Illustration de la variation du facteur d'échelle, (a) l'Onde mère, (b) l'Onde pour $0 < a < 1$, (c) l'Onde pour $a > 1$

La procédure de calcul des coefficients d'ondelettes s'effectue comme suit : On multiplie le signal et la fonction analysante et l'on calcule l'intégrale du produit. C'est un processus assez simple ; en fait il se déroule en cinq étapes : [21]

1. On prend une ondelette et on l'a compare à une section au début du signal original,
2. On calcul le coefficient $C_x(a, b)$ qui représente le degré de corrélation de l'ondelette avec cette portion du signal,
3. On translate l'ondelette vers la droite et on répète les étapes (1) et (2) jusqu'à ce que le signal soit couvert en entier,
4. On dilate l'ondelette et on répète les étapes une (1) à trois (3),
5. On recommence l'opération pour toutes les étapes à différentes échelles.

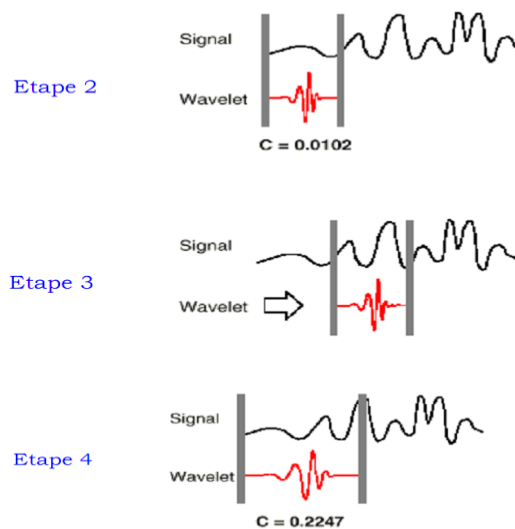


Figure II.2 : Etapes de calcul des Coefficients

II.3.2.1 Transformée en ondelettes continue (CWT)

Une fois que l'ondelette mère est choisie le calcul commence par $a = 1$ et la CWT est calculée pour toutes les valeurs de $a < 1$ et $a > 1$. Cependant, selon le signal, une transformée complète n'est habituellement pas nécessaire. Pour tous les besoins pratiques, les signaux sont limités en largeur de la bande (bandlimited) et donc, calcul de la transformation pour un intervalle limité d'échelles est habituellement adéquat.

Si le signal a une composante spectrale qui correspond à la valeur courante de a , le produit de l'ondelette mère avec le signal à l'endroit où cette composante spectrale existe donne une valeur relativement grande. Autrement ce produit donne une valeur relativement petite ou nulle.

II.3.2.2 Transformée en ondelettes discrète (DWT)

Dans le monde d'aujourd'hui, les ordinateurs sont presque tous les calculs. Il est évident que ni la WT, ni le STFT, ni le CWT ne puissent être pratiquement calculés en employant des équations analytiques, des intégrales, etc... Il est donc nécessaire de discrétiser les transformées. Comme dans la FT et le STFT, la manière la plus intuitive de faire ceci est simplement d'échantillonner le plan temps-fréquence (échelle). Encore intuitivement, l'échantillonnage du plan avec un taux d'échantillonnage uniforme semble le choix le plus normal. Cependant, dans le cas de la WT, le changement de l'échelle peut être employé, pour réduire le taux d'échantillonnage.

La transformée en ondelettes telle qu'elle est définie est redondante, c'est-à-dire que l'on obtient plus de coefficients d'ondelettes qu'il n'en est nécessaire pour décrire le signal de manière exhaustive. En pratique, on a plus souvent affaire à des signaux discrets, mais même sans cela, on a intérêt à discrétiser les valeurs de a et b . Pour ce rendre compte d'une part, de l'intérêt d'utiliser les ondelettes et d'autre part, de la manière dont on va discrétiser les valeurs de a et b , on va regarder comment les ondelettes se déploient ou se répartissent en temps et en fréquence, par rapport, par exemple, à une transformée de Fourier à fenêtre glissante. Pour cela, on a représenté, en les juxtaposant, les supports temporels et fréquentiels des ondelette dans le plan défini en abscisse par l'axe temporel et en ordonnées par l'axe fréquentiel. On visualise ainsi comment est découpé le plan temps-fréquence pour chaque type de transformée (fig. II.3) [22][26].

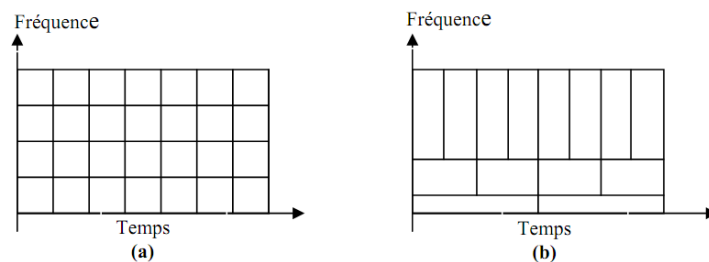


Figure II.3 : Plan Temps-Fréquence :

a) transformée de Fourier à fenêtre Glissante, b) transformée en ondelette.

On remarque que les résolutions temporelle et fréquentielle pour la transformée de Fourier à fenêtre glissant sont constants, alors que pour la transformée en ondelette les résolutions varient en sens inverse [21][22].

Le pavage temps-fréquence utilisé sur la figure II.3.a suggère une méthode de discrétisation exponentielle pour les échelles et pour le temps. Soit $a = a_0^j$ et $b = n \cdot b_0 \cdot a_0^j$ avec a_0 & $b_0 \in \mathbb{Z}$ on obtient alors une transformée en ondelettes discrète :

$$TO(n, j) = a_0^{-\frac{j}{2}} \int_{-\infty}^{+\infty} f(t) \cdot \psi(a_0^{-j} t - nb_0) dt \quad (\text{II.7})$$

Si on choisit $a_0 = 1$ & $b_0 = 2$, on parle alors d'une transformée en ondelette dyadique.

$$TO(n, j) = \sqrt{2}^{-j} \int_{-\infty}^{+\infty} f(t) \cdot \psi(2^{-j} t - n) dt \quad (\text{II.8})$$

II.3.2.3 Analyse multi-résolution

On construira une analyse multi résolutions à l'aide du son espace d'approximation V_j (généré par la fonction échelle) emboîtés les uns dans autres, tel que le passage de l'un à l'autre soit résultat d'un changement d'échelle (zoom) [21].

L'espace des détails W_j (généré par la fonction ondelette) vient compléter l'analyse. On peut définir pour V_j son complément orthogonal W_j dans V_{j-1} tel que :

$$V_{j-1} = V_j \oplus W_j \quad (\text{II.9})$$

Le schéma de la décomposition est représenté symboliquement sur la figure II.4, dans laquelle la largeur des rectangles symbolisant les sous espaces est proportionnelle à la densité de l'échantillonnage réalisé par la projection du signal dans le sous espace considéré [22][25].

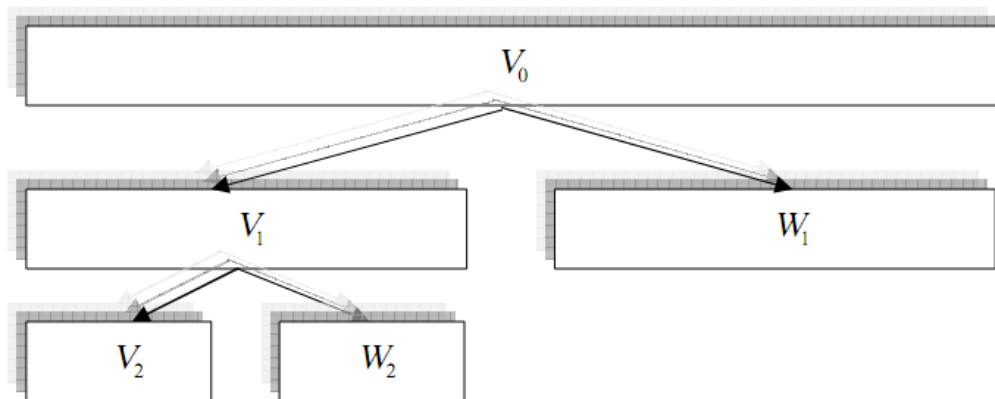


Figure II.4 : Schéma d'analyse Multi-résolution

Avec l'analyse par ondelettes, le signal est décomposé en fonctions élémentaires, engendrées par des Transformations simples d'une fonction de base qui est translatée et dilatée, (fig. II.5). Les ondelettes sont très étendues et dilatées pour étudier les basses

fréquences (les grandes échelles) et très fines pour étudier des phénomènes plus transitoires (hautes fréquences, ou petites échelles). Cette procédure, développée par S. Mallat et systématisée par I. Daubechies, pour le nom de multi-résolution, et suggère une interprétation différente de l'analyse par ondelettes, basée sur les idées de lissage, ou d'approximation des fonctions. Les décompositions en Ondelettes existent dans plusieurs versions, que l'on choisit en fonction de l'application visée [22].

La décomposition du signal en Ondelettes permet à l'utilisateur de s'adapter plus au signal selon son contenu fréquentiel pour extraire les informations utilisées. Dans ce cas tout changement de fréquence se traduira par un changement sur un plusieurs niveaux d'échelles. La classification sera appliquée sur certains signaux de détails et non sur le signal original [22].

$$\text{Détail}_m(t) = \text{approx}_{m-1}(t) - \text{approx}_m(t)$$

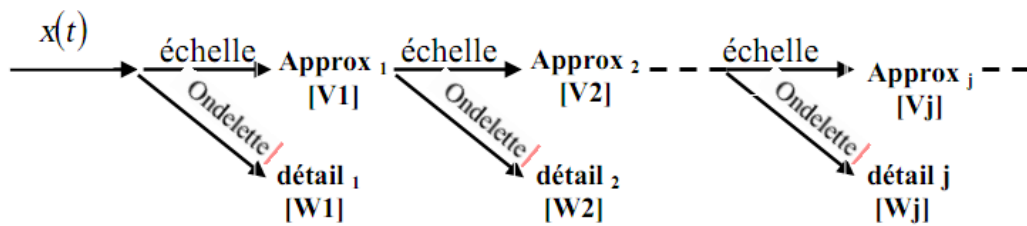


Figure II.5 : Analyse Multi-résolution : décomposition successive en approximation et détails.

II.4 Conclusion

Dans ce chapitre, nous avons montrée des techniques de compression utilisés dans la littérature ; soit les techniques directes ou par transformées , dans les transformées on trouve la plus fiable dans les recherche actuelles ; la transformé en ondelettes, qui permettent d'obtenir une distribution temps-fréquence et temps-échelle de l'énergie d'un signal. Nous avons également vu qu'à partir de cette distribution, les fréquences instantanées des composantes d'un signal peuvent être estimées.