

Rapport bibliographique
Le Codage Audio et ses Différentes Étapes

Gilles GONON
Laboratoire d'Informatique de l'Université du Maine

6 décembre 1999

Table des matières

Introduction	1
1 Présentation des différents types de codeurs	3
1.1 Généralités sur les codeurs	3
1.2 Évaluation des performances	4
1.2.1 Qualité de restauration	5
1.2.2 Complexité	5
1.2.3 Temps de codage-décodage	5
1.2.4 Robustesse	5
1.2.5 Évolutivité	6
1.2.6 Effets	6
1.3 Présentation de codeurs	6
1.3.1 Codeurs basés sur la forme d'onde	6
1.3.2 Codeurs paramétriques	7
1.3.3 Codeurs par transformée	8
2 Transformation du Signal	11
2.1 Rappels et définitions	11
2.2 Transformées à recouvrement	12
2.2.1 Rappel sur les transformées linéaires	13
2.2.2 Ondelettes de Malvar	14
2.3 Analyse multirésolution	16
2.4 Transformée en ondelettes	17
2.4.1 Transformée en Ondelettes Continue	17
2.4.2 Transformée en Ondelettes Discrète	19
2.5 Bancs de filtres	20
2.5.1 Bancs de filtres à deux voies	21
2.6 Décompositions en paquets	23
2.6.1 Structure dyadiques	24
2.6.2 Structure s-dyadiques	25
3 Intégration de critères perceptuels	27
3.1 Excitation	28
3.2 Taux de masquage	29

3.3	Seuil de masquage	29
4	Allocation de bit	31
4.1	Principe	31
4.2	Allocation paramétrique basée sur des critères perceptuels	32
5	Quantification	35
5.1	Quantification scalaire	36
5.2	Quantification prédictive	36
5.3	Quantification Vectorielle	37
6	Codage entropique	39
6.1	Quantité d'information et Entropie	39
6.2	Recherche du code optimal	40
7	Objectifs de la thèse	43

Abréviations utilisées¹

- ✦ TF : Transformée de Fourier
- ✦ PCM -MIC : Pulse Coded Modulation, Modulation par Impulsion et Codage
- ✦ TOC : Transformée en Ondelette Continue
- ✦ TOD : Transformée en Ondelette Discrète
- ✦ TOR : Transformée en Ondelette Rapide
- ✦ LPC : Linear Predictive Coder
- ✦ kbps : KiloBits par seconde, unité de débit binaire
- ✦ TB : Transformée par blocs
- ✦ TRO : Transformée à recouvrement orthogonal
- ✦ POM : Paquets d'Ondelettes de Malvar

¹Ces abréviations sont explicitées au cours du rapport au moment où elles interviennent

Introduction

Depuis quelques années, la quantité de données circulant sur les réseaux ne cesse d'augmenter. De fait, le besoin de méthodes de compression des données toujours plus efficaces se fait ressentir dans le domaine des communications numériques. Dans des applications telles que la téléphonie mobile ou les visioconférences via internet, des taux de compressions élevés sont requis pour alléger les réseaux et accélérer la transmission. A ces contraintes d'encombrement s'ajoutent aujourd'hui des critères perceptifs (auditifs et visuels) pour lesquels l'utilisateur réclame toujours plus de confort. Autant d'impératifs motivent la recherche dans le domaine du codage et les différentes normes établies sont fréquemment remises à jour pour répondre aux attentes de plus en plus exigeantes des utilisateurs (qualité, applications multimédias).

Le codage de la parole et d'autres signaux en général consiste en une mise en forme du signal pour l'adapter à diverses applications. Les deux applications les plus courantes sont la transmission et le stockage. La première, la transmission, nécessite souvent des contraintes de temps-réel (communications numériques par exemple) - aussi appelé *full-duplex* - qui se répercutent sur le temps disponible pour le codage, et donc la complexité du codeur. Dans le cas du stockage, on dispose souvent de plus de temps pour la mise en forme du signal, le but étant principalement de le comprimer, et on repousse cette fois-ci le problème du "*temps-réel*" au décodeur qui doit alors assurer la décompression du signal de manière transparente à l'utilisateur.

Ce rapport présente en détail les différents éléments constituant un codeur audio. La chaîne de codage peut en effet être scindée en cinq blocs indépendants réalisant chacun une tâche spécifique. La figure 1 montre la succession des différents des blocs dans le cas d'un codeur *complet* puis du décodeur associé. Le signal en entrée est supposé échantillonné au préalable et le codeur réalise donc sa conversion en une trame binaire destinée à être transmise ou stockée. La chaîne de décodage réalise globalement le chemin inverse de la chaîne de codage. Dans le cadre des codeurs *complexes*, intégrant par exemple des courbes de masquage, le décodeur comprend souvent un étage de moins pour la reconstitution du signal.

La première partie présente les codeurs audio de manière générale et donne des exemples de solutions choisies pour différents types de codeurs. Les parties

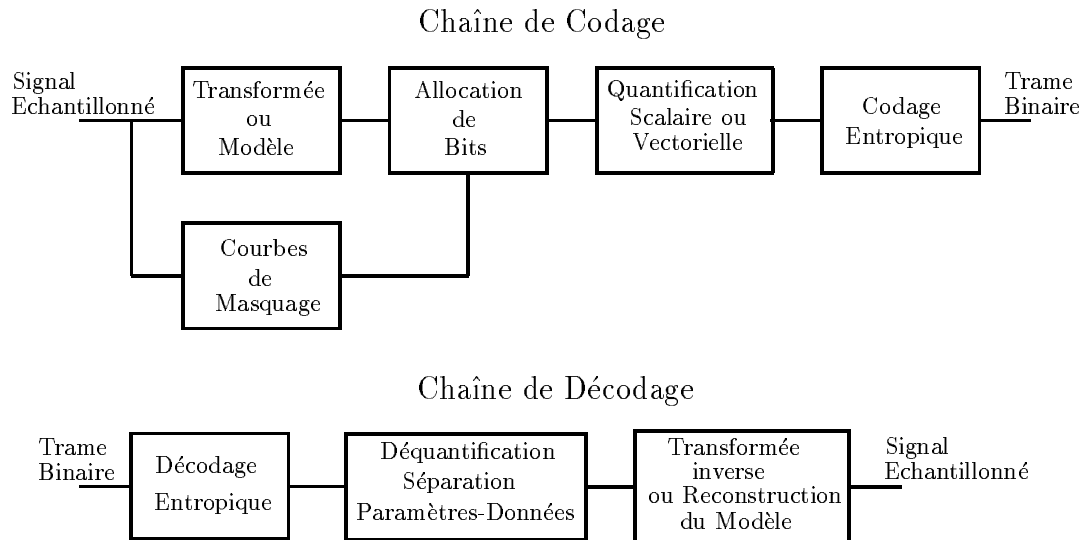


FIG. 1: Schéma bloc de chaîne complètes de codage et de décodage

suivantes détaillent séparément chacun des blocs constituant la chaîne de codage dans le cas d'un codeur par transformée² à savoir :

- La *transformation du signal* qui vise à décorrélérer les échantillons et concentrer ainsi l'énergie du signal dans le domaine de la transformée,
- les *phénomènes psychoacoustiques* de masquage utilisés pour ne garder que l'information perceptibles et compresser ainsi le signal en ne gardant que la partie subjectivement perceptible,
- la procédure d'*allocation de bits* pondérant les échantillons retenus suivant leur importance pour restituer la meilleure qualité possible à coût fixé,
- la *quantification* permettant de formater les échantillons compte tenu de leur poids respectif,
- et enfin une phase de *codage entropique* ordonnant le contenu binaire de la trame avant sa transmission.

Enfin, en conclusion nous exposons les améliorations envisageables pour les codeurs par transformée, notamment au niveau de la transformation du signal, qui peut être optimisé par une meilleure segmentation du signal dans les domaines temporels et fréquentiels.

²les codeurs paramétriques ne sont pas traités en détail, voir [12]

Chapitre 1

Présentation des différents types de codeurs

1.1 Généralités sur les codeurs

Le terme générique *audio* regroupe plusieurs classes de signaux échantillonnés correspondant à différentes qualités d'écoute. Pour chacune de ces classes, un type de codeurs est souvent plus approprié et permet, à **qualité égale**, de réduire le débits de codage donné usuellement en KiloBits par seconde (kbps). Le débit d'un codeur est sa grandeur de référence et est associé au type de signal encodé. Il permet de déduire le gain final de codage. Par exemple, pour les codeurs MPEG, un signal échantillonné à $44,1\text{kHz}$ sur 16 bits (ce qui représente en stéréo un débit de 1411kbps) est compressé à 128kbps , ce qui représente une compression d'un facteur 11.

Parmi la multitude de codeurs existants, il est possible de distinguer trois grandes catégories qui ont une approche différente du signal à coder :

- Les codeurs basés sur la **forme d'onde**. Ils traitent directement le signal échantillonné et le principal travail de codage s'effectue au niveau de la quantification. S'ils présentent l'avantage d'être très rapides (enregistrement-lecture en temps réel), ils n'effectuent presque aucune compression. Le codage par modulation d'amplitude (**P**ulse **C**oded **M**odulation), appelé Modulation d'Impulsion et Codage (**MIC**) en français, réalise une quantification uniforme de chaque échantillon du signal sur un nombre de bits fixé. Le codage PCM sert de référence pour évaluer les gains de compression des autres types de codeurs, du fait qu'il se contente de coder le signal sans le compresser.
- Les codeurs **paramétriques**, comme leur nom l'indique, se basent sur un modèle du signal pour en extraire les paramètres. L'exemple le plus typique

de signal modélisé est la parole dont les principaux paramètres à extraire sont l'excitation et la fonction de transfert du conduit vocal. L'inconvénient majeur de ces codeurs est leur impossibilité à coder des signaux ne rentrant pas dans le modèle (2 voix en même temps, musique ...).

- Les codeurs par **transformée** (non paramétriques) qui traitent indifféremment tous les types de signaux audio. Ils conduisent toutefois à des débits plus importants quand les signaux sont modélisables.

Le tableau 1.1 présente les formats de signaux utilisés dans les transmissions et pour le stockage ainsi que les types de codeurs qui leurs sont associés. Dans tous les cas, il est possible de calculer le débit associé au code PCM, qui sert ensuite pour évaluer le taux de compression.

Au niveau de la compression, un autre choix à effectuer est celui de la reconstruction parfaite ou non du signal. Pour accéder à de faibles débits de codage, il faut souvent enlever une partie de l'information jugée moins importante pour la qualité d'écoute. La compression est alors **avec perte** par opposition à la compression **sans perte** pour laquelle le signal décompressé est identique au signal de départ. La phase de codage entropique effectuée par exemple une compression sans perte tandis que la prise en compte des phénomènes psychoacoustiques se fait avec perte. Dans le cadre du codage audio, deux grands principes (voir [12]) sont à retenir :

Il est inutile de transmettre $\left\{ \begin{array}{l} \text{une information prédictible} \\ \text{une information présente mais non perçue} \end{array} \right.$

Classe de signal	Format	Type de codeur
Bande téléphonique	$F_e = 8kHz$, mono, 8 – 16 bits	paramétrique
Bande élargie	$F_e = 16kHz$, mono, 8 – 16 bits	paramétrique
Bande radio	$F_e = 32kHz$, stéréo, 8 – 16 bits	PCM ou transformée
Haute fidélité (CD)	$F_e = 44.1kHz$, stéréo, 16 bits	PCM ou transformée
DVD Audio	$192kHz$, stéréo, 16 – 24 bits	transformée
	$92kHz$, multicanal, 16 – 24 bits	transformée

TAB. 1.1: Différents types de codeurs pour différents types de signaux

1.2 Évaluation des performances

La qualité du codeur ne tient pas seulement à la qualité du signal reconstruit et un certain nombre de critères sont à prendre en compte pour destiner le codeur à une application spécifique. Les principaux critères sont présentés ici mais d'autres peuvent intervenir dans des cas plus particuliers (cryptage des données par exemple).

1.2.1 Qualité de restauration

C'est évidemment le critère essentiel. Il reste subjectif car on ne recherche pas la reconstruction parfaite mais la transparence, c'est-à-dire une différence inaudible entre le signal *avant et après* codage. Ainsi, un signal de bande téléphonique restera médiocre même s'il passe au travers d'un codeur transparent. La transparence peut toujours être obtenue, mais souvent au dépend d'un autre critère : la complexité.

1.2.2 Complexité

Pour obtenir la meilleure qualité au débit le plus faible possible, il faut augmenter la complexité de l'algorithme de codage/décodage. Même s'il existe aujourd'hui des calculateurs puissants, certains algorithmes ne peuvent pas être destinés à des applications *temps-réel* comme les transmissions mais donnent de très bons résultats pour ce qui est du stockage (le standard MPEG-III en est un exemple flagrant). Aussi, dans les deux cas, la charge de calcul est répartie entre le codeur et le décodeur avec cependant un poids plus important du côté du codeur de manière à ce que le décodage se fasse le plus simplement et le plus rapidement possible.

1.2.3 Temps de codage-décodage

Il est intimement lié à la complexité de l'algorithme global. Les codeurs destinés à des applications *temps-réel* (téléphonie mobile par exemple) requièrent des temps de codage-décodage inférieurs à 100ms pour lesquels les algorithmes doivent être allégés et optimisés au maximum. Pour les applications de stockage, le temps de codage n'est pas fixé, ce qui donne lieu à des temps assez longs¹ pour la mise en forme du signal. Par contre, la charge de calcul est quasi-entièrement reportée sur le codeur et le décodage peut être effectué en temps réel (en même temps que la lecture du fichier).

1.2.4 Robustesse

La robustesse est d'une part la capacité d'un codeur à garder un signal intelligible quand le débit diminue et d'autre part l'aptitude du décodeur à corriger le bruit du canal de transmission. Tous les codeurs proposent un choix de débits procurant des qualités différentes. Par exemple, les codeurs de parole garantissent la transparence à des débits de 48kbps et proposent des débits allant jusqu'à 1.2kbps. A 1.2kbps, le contenu reste intelligible mais on ne reconnaît pas le locuteur (procédé utilisé dans les communications militaires où le locuteur se présente, le gain

¹ dans le cas du MPEG-III, sur un PC-200MHz, le codage dure environ la longueur du morceau à coder

de bits étant utilisé pour sécuriser la transmission, ou dans des jouets comme la fameuse *dictée magique*©).

1.2.5 Évolutivité

De plus en plus, et ceci peut se généraliser à tous les systèmes informatiques, les différentes normes de codeurs laissent un degré de liberté quant à l'évolution de la partie codage. Ainsi, un certain nombre de paramètres sont transmis au décodeur de sorte qu'il construise chaque bloc, voir figure 1, utile au décodage sans avoir de modèle fixe. Les paramètres peuvent par exemple définir la forme du banc de filtre, ou la longueur de trame analysée ... Ceci est dû au fait que le codage se fait de manière adaptative et que de nombreuses améliorations sont encore possibles au niveau de l'analyse du signal.

1.2.6 Effets

Depuis peu, les décodeurs proposent une série d'effets reliés aux applications dans lesquels interviennent des sons. Outre la synthèse de sons à partir de fichier de type texte (MiDi, Text To Speech ...) les décodeurs proposent des effets de spatialisation, d'égalisation et de changement de vitesse sans changement de hauteur. Ce dernier sert par exemple à rechercher des sons dans une base de données ou à voir un film en accéléré avec le son. Ceci ne concerne pas directement le codage audio mais il est tout de même intéressant de noter l'interaction désormais possible entre les différentes sources de signaux codés².

1.3 Présentation de codeurs

Cette section décrit de manière succincte les techniques utilisées dans quelques codeurs standards. L'approche reste ici globale, compte tenu que chacun des différents blocs est détaillé dans les chapitres suivant, le but étant de présenter les solutions employées pour chaque type de codeurs.

1.3.1 Codeurs basés sur la forme d'onde

Ce type de codeur n'utilise pas de transformation du signal et mets en forme directement l'allure temporelle du signal. Le codage est ramené uniquement à une étape de quantification. Les codes PCM et DPCM sont détaillés au chapitre 5.

Pulse Coded Modulation (PCM)

Chaque échantillon est codé séparément à l'aide d'un mot binaire de longueur fixe. C'est une quantification. Le nombre de niveaux de quantification est une

²les effets cités sont proposés dans la norme MPEG4

puissance de 2, de sorte que toutes les valeurs codables par le mot binaire soient utilisées. Différentes lois de quantification sont utilisées. Une loi uniforme est utilisée pour coder la musique, tandis que la loi A est utilisée pour coder la parole. La loi A réalise une compression par rapport à la loi uniforme en attribuant moins de bits pour les fortes valeurs d'amplitude car elles portent moins d'information que les zones d'attaque dans les signaux de parole.

Le codage PCM, dans sa version uniforme, sert de référence pour évaluer les performances de compression d'un codeur. Il est utilisé dans le codage des CD de manière à couvrir les exigences de l'oreille humaine. Ainsi, la bande passante de l'oreille est $[20Hz, 20kHz]$ et sa dynamique est de 120 dB. Le codage haute fidélité des CD est basé sur ces résultats : il a pour fréquence d'échantillonnage $44,1kHz$ et offre une dynamique de $96dB$ (les mots binaires font 16 bits sachant qu'un bit augmente la dynamique de $+6dB$, voir chapitre 5).

Adaptative Differential PCM (ADPCM)

C'est une version améliorée du code PCM qui quantifie la différence entre un échantillon et une valeur prédite de cet échantillon. L'idée de base est que, par continuité de la pression acoustique, deux échantillons voisins ont des amplitudes proches. Plutôt que de quantifier le signal et ses 120 dB de dynamique, on va quantifier la différence, de sorte à réduire la dynamique à encoder. Le schéma de ce type de codeur est présenté et discuté au chapitre 5. Outre, l'intérêt de compression que présente ce codeur, il est extrêmement facile à mettre en œuvre et sa complexité faible en permet l'implémentation temps réel.

1.3.2 Codeurs paramétriques

Ces codeurs servent essentiellement à coder les signaux de parole. On les appellent donc Vocoder (de l'anglais Voice Coder). Ils sont très utilisés dans le domaine des télécommunications car la conversation en est aujourd'hui encore le principal contenu. Ainsi, tous les codeurs intégrés dans les téléphones portables sont des vocodeurs car l'application d'une personne seule parlant dans un micro correspond au modèle qu'ils proposent.

Code Excited Linear Prediction (CELP)

Ce codeur, basé sur une prédiction linéaire de la parole, propose une amélioration sensible par rapport aux Codeurs Prédicatifs Linéaires (LPC). Le principe étant de tenter de reconstruire l'échantillon de parole courant à partir des p échantillons passés, le CELP effectue premièrement une analyse à court terme du signal pour extraire les paramètres de la parole puis une analyse à long terme pour prédire par exemple des changements d'amplitude où des redondances de périodicité. Le modèle est alors défini et l'erreur résiduelle est quantifiée vectoriel-

lement³ (vecteurs d'environ 40 échantillons pour un dictionnaire de 1024 mots) et transmise. Ceci améliore grandement la qualité du signal décodé en terme de timbre de voix. Le dernier paramètre à déterminer est l'excitation, c'est à dire la fonction que l'on va mettre en entrée des filtres au décodage. Grossièrement, cette excitation est un bruit blanc pour un son non-voisé et un train d'impulsions pour un son voisé. Elle est très importante pour la qualité de restitution de la parole. Aussi, le CELP dispose d'un dictionnaire contenant un multitude d'excitations de types différents. Le CELP propose différents débits mais chaque débit nécessite un dictionnaire différent pour la quantification.

Dans ce codeur, l'essentiel de la charge de calcul est situé à l'encodage, au moment de la quantification vectorielle qui consiste à rechercher la meilleure solution dans un dictionnaire. Il conduit à des débits de l'ordre de $4kbps$ pour une bonne qualité de parole échantillonnée à $8kHz$.

MultiBand Excited coder (MBE)

Ce codeur est complètement paramétrique et ne transmet pas l'erreur résiduelle. Il décompose le signal en plusieurs sous-bandes fréquentielles et pour chacune de ces sous-bandes effectue la décision voisé/non voisé. De cette manière l'excitation appliquée dans chaque sous-bande durant la synthèse est soit un bruit blanc soit un train d'impulsion suivant la décision prise. Cela améliore la qualité de la parole reconstruite et prodigue au codeur une grande robustesse au bruit de fond.

L'intelligibilité et la personnalité de la parole restent intactes pour un débit de $2,4kbps$. Étant donné qu'il ne nécessite pas de dictionnaire comme le CELP, la complexité s'en voit énormément réduite. Ce codeur et ses versions améliorées (Improved MBE) sont utilisées dans de nombreuses application de télécommunication, et le codeur IMBE a été sélectionné comme norme de codage de parole en Amérique du Nord.

1.3.3 Codeurs par transformée

Les principaux codeurs par transformée sont développés par les grands groupes de recherche. Ainsi, le MPEG (Motion Picture Experts Group), les Dolby Laboratories et les laboratoires des grands groupes industriels (Sony, Philips, Microsoft ...) proposent les principaux standards que l'on retrouve aujourd'hui à disposition du grand public : MPEG-I,II,III, Dolby AC3 et ATRAC. Les trois utilisent des stratégies proches de transformation du signal et incorporent les phénomènes de masquage de l'oreille pour aboutir à des taux de compression plus élevés.

Nous présentons ici le principe du codeur MPEG audio qui permet une bonne introduction au travail effectué dans le cadre du codage par transformée⁴. Il se

³voir section 5.3

⁴voir chapitre 2

décompose en trois *couches* qui correspondent aux standards connus sous les noms de MPEG-I, MPEG-II et MPEG-III. Il est à noter que cette version du codeur audio proposé par MPEG a été améliorée pour donner naissance aux trois couches de la norme MPEG2 audio. La norme MPEG2 Advanced Audio Coding propose un codage haute fidélité, multivoies à 64kbps par voies, et ce grâce à une prédiction rétrograde permettant une meilleure adaptation temporelle et l'intégration de critères psychoacoustiques.

L'appellation MPEG 3 en vogue en ce moment correspond en fait à la troisième couche de cette norme MPEG2 (MPEG2-III). Chacune des couches représente une amélioration de sa version précédente mais au prix d'une complexité chaque fois plus importante. Le MPEG-I est l'algorithme de base de transformation du signal, qui est réutilisé dans les couches II et III.

Première couche MPEG-I

Le signal est passé dans un banc uniforme de 32 filtres, à échantillonnage critique. Ce banc de filtres assure un bon compromis entre les résolutions temporelle et fréquentielle du signal. Les courbes de masquages⁵ sont alors appliquées sur la base de ce banc de filtres, même s'il ne reflète pas exactement le comportement en bandes critiques de l'oreille. La largeur d'un filtre dépasse la largeur d'une bande critique dans les basses fréquences si bien que la bande critique supportant le plus petit niveau de bruit de quantification fixe l'allocation de bits pour la sous-bande. Le banc de filtre n'est pas à reconstruction parfaite mais l'erreur de reconstruction est inaudible. Enfin, un recouvrement fréquentiel non négligeable entre les filtres peut entraîner qu'une même fréquence apparaisse dans deux filtres adjacents. Dans la première couche l'allocation de bits⁶ porte pour chaque sous-bande sur un vecteur de 12 échantillons (un bloc de 32 échantillons du signal d'entrée donne 1 coefficient pour chaque sous-bande). L'allocation de bits détermine le nombre de bits utilisés pour chaque échantillon et alloue aussi un facteur d'échelle pour maximiser la résolution du quantificateur.

Deuxième couche MPEG-II

La compression globale est améliorée en prenant non plus un groupe de 12 échantillons mais trois groupes, puis en regardant les similitudes de manière à optimiser l'allocation de bits. L'allocation peut ainsi porter séparément ou conjointement sur les trois groupes suivant la similitude entre ces groupes, le tout prenant en compte les critères psychoacoustiques de masquage du bruit de quantification. Le type de code utilisé pour la quantification est lui aussi optimisé.

⁵les courbes de masquages et bandes critiques sont définies au chapitre 3

⁶voir chapitre 4

Troisième couche MPEG-III

Partant du même banc de filtre que la première couche, le MPEG-III compense les défauts des filtres au niveau de la reconstruction en utilisant une transformée améliorée, la MDCT (modified discrete cosine transform, voir [15]).

Plusieurs débits sont disponibles suivant les couches. Le débit le plus bas pour les trois couches est 32 kbps. La transparence est obtenue pour 448kbps pour la couche I, 384 kbps pour la couche II et 320 kbps pour la couche III.

Chapitre 2

Transformation du Signal

La transformation d'un signal vise à donner un autre point de vue de ce signal pour condenser ou clarifier l'information qu'il contient. C'est principalement parce que l'oreille capte de manière temporelle les différences de pression que nous avons besoin de la représentation des signaux dans le domaine temporel, mais cette représentation est loin d'être la plus concise. Dans le cadre du codage audio, de nombreuses transformées sont utilisées pour éliminer au maximum les redondances entre les coefficients dans le domaine de la transformée. Les principaux outils du codage sont d'une part les transformées à recouvrement orthogonal pour segmenter le signal sans effets de bord et d'autre part l'analyse en ondelettes et l'analyse par banc de filtres qui se regroupe en un seul outil dans le formalisme mathématique de l'**analyse multirésolution**.

L'idée en est simple, segmenter le signal temporellement (fenêtres d'analyse) et fréquentiellement (bandes d'octaves ou sous-bandes) de telle sorte que chaque pavé temps-fréquence isole des coefficients non redondants du signal (une fréquence durant un intervalle de temps).

2.1 Rappels et définitions

Du point de vue mathématique, un signal est soit une fonction dans le cas continu, soit une suite de coefficients dans le cas discret. Dans le cas du codage, le signal est échantillonné et cela revient à ne s'intéresser qu'aux suites de coefficients, définies sur un espace qui est souvent $l_2(\mathbb{Z})$ l'espace des séquences (suites) de carrés intégrables¹, hilbertien. Cet espace sur lequel évolue le signal peut être caractérisé par des bases, c'est-à-dire des familles de fonctions (de séquences) qui permettent de générer tout signal de l'espace par combinaison linéaire (une fa-

¹le carré d'un signal est homogène à son énergie. Le fait qu'il soit intégrable entraîne que le signal est d'énergie finie

mille complète dont les vecteurs sont linéairement indépendants est une base). Il existe une infinité de bases et chacune donne un point de vue différent du signal. Soit $x \in S$, on cherche une famille $\{\varphi_i\}_{i \in Z} / x = \sum_i \alpha_i \varphi_i$.

Si $\{\varphi_i\}$ est une base de S , il existe alors une famille *duale* $\{\tilde{\varphi}_i\}$ telle que :
 $\alpha_i = \sum_n \tilde{\varphi}_i[n] x[n]$.

Le dual est donc la famille qui va permettre d'assurer la correspondance entre deux transformées, dite alors duales. La pertinence d'une famille complète tient principalement à la non redondance des vecteurs la composant. L'orthogonalité des vecteurs reflète cette propriété et c'est pourquoi les transformées associées au traitement du signal sont souvent **orthogonales**. L'intérêt d'une telle base réside dans le fait qu'elle est elle-même son dual.

Il arrive toutefois que tous les vecteurs d'une base, s'ils sont linéairement indépendant entre eux, ne soient pas orthogonaux entre eux. Dans ce cas là, il existe une base duale dont tous les vecteurs sont orthogonaux avec l'autre base. Les bases duales sont dites **Bi-orthogonales**.

Définition 1 :

Une base orthogonale de S est une famille orthonormale complète, et alors :

$$\forall i, j \in \mathbb{N}, \quad \varphi_i = \tilde{\varphi}_i \quad \text{et} \quad \langle \varphi_i, \varphi_j \rangle = \delta_{ij}$$

Définition 2 :

Une paire de bases biorthogonales de S est une paire de familles complètes mais non orthonormales, dont les vecteurs sont linéairement indépendants, qui vérifie :

$$\langle \varphi_i, \tilde{\varphi}_j \rangle = \delta_{ij}, \quad \forall i, j \in \mathbb{N}.$$

2.2 Transformées à recouvrement

Ces transformées constituent un des grands types de transformations utilisées pour le codage. La première étape du codage est la segmentation temporelle du signal car l'analyse va se faire sur des blocs fenêtrés. Les transformées en blocs (**TB**) qui traitent des séquences disjointes du signal sont peu utilisées dans les applications autre que l'analyse. Elles induisent en effet lors de la synthèse des ruptures aux frontières des blocs créant un cliquetis ou un ronflement suivant la taille de ces blocs.

Les Transformées à Recouvrement Orthogonales (**TRO**) permettent d'éliminer ces ruptures inhérentes à la construction des TB en traitant des séquences se recouvrant. Bien que l'analyse porte sur N coefficients reflétant le comportement fréquentiel du bloc de même taille, la TRO transforme un nombre $L > N$ de coefficients pour prendre en compte les effets de bord. La transformée inverse, à

partir de N coefficients, donne L coefficients qui ne correspondent pas encore aux échantillons du signal. Il faut en effet ajouter toutes les séquences se recouvrant pour avoir la reconstruction parfaite.

Deux familles de TRO sont principalement utilisées en codage, la MLT (Modulated Lapped Transform) et les Ondelettes de Malvar. Les deux réalisent une transformation linéaire du signal, comme la transformée en Cosinus Discret (**DCT**).

Après avoir rappelé l'intérêt de la DCT, les ondelettes de Malvar sont présentées car elles ont de bonnes propriétés d'adaptativité pour le codage. Le lecteur désireux d'en savoir plus sur la MLT et ses différentes améliorations peut se reporter à [15, 1, 9].

2.2.1 Rappel sur les transformées linéaires

Leur rôle est de décorréler les coefficients. En d'autres termes, dans le domaine de la transformée, la matrice d'autocorrélation du signal doit être diagonale. En théorie, la transformée de Karhunen-Løve (**KLT**) réalise cette diagonalisation. Malheureusement, pour des raisons d'implémentation, elle est très difficile à réaliser en pratique car elle dépend uniquement du signal considéré. Aussi, d'autres transformées comme la DCT réalisent des approximations de la KLT pour un modèle probabiliste de signal.

Transformée de Karhunen-Løve

La matrice T définissant la KLT est la matrice du changement de base qui diagonalise la matrice d'autocorrélation de x . Soit X^{KLT} le vecteur transformé d'une séquence x et soit R_x la matrice d'autocorrélation de x . Alors la KLT est définie par

$$R_{X^{KLT}} = E(X^{KLT} X^{KLT^T}) = E(T x x^T T^T) = T R_x T^T. \quad (2.1)$$

Ainsi T est constituée des vecteurs propres (normalisés) et $R_{X^{KLT}}$ contient les valeurs propres de x rangées dans l'ordre croissant. De cette manière, la meilleure approximation de x à l'ordre k est obtenue en ne gardant que les k premiers coefficients.

La KLT, de par l'ordonnement des coefficients trouvés, fournit la concentration d'énergie optimale pour x . Cette transformée dépend malheureusement très fortement du signal analysé et aucun algorithme rapide n'a encore été trouvé.

Transformée en Cosinus Discret

Elle réalise simplement l'approximation de la KLT si le signal est un processus Gaussien-Markovien du premier ordre. Plus son coefficient de corrélation sera grand (tendra vers 1), meilleure sera l'approximation.

la DCT d'une séquence x de longueur N est notée X^{DCT} et elle est définie par

$$\begin{cases} X^{DCT}[0] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \\ X^{DCT}[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi(2n+1)k}{4N}\right), \quad k = 1, \dots, N-1. \end{cases} \quad (2.2)$$

Cette transformée offre une bonne approximation de la KLT en général mais elle présente deux inconvénients majeurs :

- Les effets de blocs en synthèse qui n'en font pas un outil applicable directement au codage
- La segmentation fixe qui ne permet pas de s'adapter au signal.

C'est donc pour pallier ces deux inconvénients qu'ont été mis en place d'autres outils. La MLT permet d'effacer les effets de blocs mais la taille de la fenêtre reste fixée et non adaptative. Ce sont les ondelettes de Malvar qui proposent de traiter conjointement ces deux problèmes inhérents à la DCT.

2.2.2 Ondelettes de Malvar

La transformée en ondelettes de Malvar permet une segmentation temporelle adaptative car les fenêtres sont de taille variable. Du point de vue mathématique cela revient à dire que les vecteurs d'analyse sont de taille variable. L'analyse va porter successivement sur des blocs de taille N_j , $j \in \mathbb{N}$ dont les frontières sont notées $[a_j, a_j + 1]$. La fenêtre d'analyse est composée de trois zones qui s'adaptent indépendamment au signal. La figure 2.1 illustre les recouvrements des fenêtres et présente les notations.

- La première est appelée zone d'attaque et constitue l'établissement de la fenêtre. Elle commence à la valeur zéro et finit à un. Soit $2N_j^g$ sa longueur.
- La seconde est une zone stationnaire qui a la valeur 1. Sa longueur vaut $N_j - N_j^d - N_j^g$.
- La dernière est une zone d'extinction qui termine la fenêtre en la ramenant à 0. Soit $2N_j^d$ sa longueur.

Lors de la transformée, on va calculer les similitudes de la fonction avec l'ondelette fenêtrée, si bien qu'on aura un indice du transitoire du bloc analysé.

Les trois zones peuvent être définies indépendamment sur une fenêtre mais pour obtenir la condition de reconstruction parfaite en synthèse, il faut respecter entre deux fenêtres successives plusieurs conditions de régularité. En notant $h_j[n]$ l'allure temporelle de la fenêtre d'analyse du bloc j d'une séquence x de longueur N_j , alors h_j et h_{j+1} doivent vérifier :

$$\begin{cases} h_j^2[n] + h_{j+1}^2[n] = 1, & \forall n \\ h_{j-1}[a_j + k] = h_j[a_j - k], & \text{pour } |k| \leq N_j^g \end{cases} \quad (2.3)$$

Le support $h_j[n]$ est donc supérieur à la taille des blocs analysés (utile pour gérer le recouvrement) et la fenêtre n'est pas non plus constante sur ce bloc.

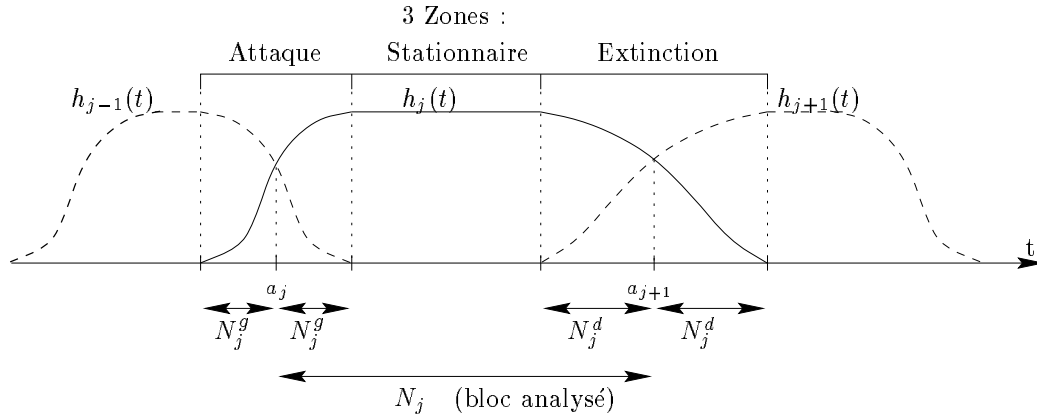


FIG. 2.1: Recouvrement et notation des fenêtres pour la transformée en ondelettes de Malvar

Dans ces conditions, les ondelettes de Malvar sont définies par la base orthonormée de $l_2(\mathbb{Z})$ $\{u_{j,k}\}_{j \in \mathbb{Z}, 0 \leq k \leq N_j}$ où

$$u_{j,k}[n] = \sqrt{\frac{2}{N_j}} h_j[n] \cos \left[\pi \left(k + \frac{1}{2} \right) \left(\frac{n - a_j}{N_j} \right) \right]. \quad (2.4)$$

La transformée en ondelettes de Malvar s'écrit alors, sous forme matricielle, en considérant un vecteur \mathbf{v}_j composé du bloc à analyser \mathbf{x}_j de taille N_j et des zones couvertes par le support de la fenêtre h_j . Soit \mathbf{X}_j le vecteur transformé, de taille N_j .

$$\mathbf{v}_j = \left[\{\mathbf{x}_{j-1}[n]\}_{n \in [a_j - N_j^g]} \quad \mathbf{x}_j \quad \{\mathbf{x}_{j+1}[n]\}_{n \in [a_{j+1} + N_j^g]} \right]$$

Alors

$$\mathbf{X}_j = \mathbf{P}_j \cdot \mathbf{v}_j,$$

où \mathbf{P}_j est la matrice constituée des vecteurs de la base $\{u_{j,k}\}_{j \in \mathbb{Z}, 0 \leq k \leq N_j}$

$$\mathbf{P}_{j,n,k} = u_{j,k}[n].$$

Cette matrice peut aussi être vue comme un banc de filtre d'analyse, composé des cosinus, modulé par un passe-bas dont la réponse impulsionnelle est la fenêtre temporelle.

L'appellation d'ondelettes est un peu abusive car les fonctions analysantes n'ont pas le même nombre d'oscillations et ne sont donc pas des dilatées les unes des autres. Cependant ce terme est utilisé car les ondelettes de Malvar sont des versions duales des ondelettes dans le domaine temporel et la décomposition en paquet d'ondelettes de Malvar est similaire à la formation des paquets d'ondelettes traditionnels.

La matrice de synthèse est obtenue en transposant \mathbf{P}_j . Les vecteurs obtenus par synthèse sont de taille $N_j + N_j^g + N_j^d$ et ne représentent pas encore le signal d'origine car ils incluent les recouvrements avec les blocs précédent et suivant. Le signal peut être parfaitement reconstruit lorsque tous les blocs sont synthétisés en ajoutant les parties se recouvrant. La reconstruction sera parfaite si les matrices d'analyses $\{\mathbf{P}_j\}_{0 \leq j \leq M}$ correspondant aux différents blocs sont unitaires entre elles, c'est à dire si

$$\sum_{i=0}^{M-l-1} \mathbf{P}_j \cdot \mathbf{P}_{j+l}^T = \delta_{0,l} \mathbf{I}_{N \times N} \quad \text{pour } l = 0, \dots, M-1.$$

Dans le cadre du codage par transformée, les ondelettes de Malvar permettent d'éliminer quasi complètement les problèmes d'effets de blocs (ruptures) et constituent donc un bon outil de synthèse pour l'étape de décodage.

2.3 Analyse multirésolution

C'est la théorie mathématique qui unifie les algorithmes pyramidaux de décomposition en sous-bandes et la théorie des ondelettes. Ces théories ont pour but de pallier aux différents problèmes rencontrés avec l'analyse de Fourier, à savoir étudier les non-régularités des fonctions ainsi que leurs propriétés locales. Dans ce paragraphe est simplement présenté le principe de l'analyse multirésolution et le lecteur intéressé pourra se référer à [3, 11] pour une présentation complète sur le sujet. À partir d'une suite croissante de sous-espaces fermés emboîtés $\{V_j\}_{j \in \mathbb{Z}}$ ou V_j est obtenu par dilatation² - typiquement d'un facteur 2 - de V_{j+1} .

L'idée est donc de distiller la précision du signal jusqu'au résultat souhaité, tout en gardant une trace des fluctuations précédentes.

Les deux notions importantes sont celles d'**échelle** et de **résolution** :

- L'échelle est la grandeur proportionnelle à la durée des fonctions (bases) utilisées pour décomposer le signal.
- La résolution est le nombre minimum d'échantillons nécessaires pour représenter le signal, en relation avec la quantité d'information porté par le signal.

Ainsi, une division par deux de l'échelle divise par deux la résolution tandis qu'un doublement de l'échelle ne change pas la résolution.

$$\frac{\text{échelle}}{2} \implies \frac{\text{résolution}}{2} \quad \text{et} \quad \text{échelle} * 2 \implies \text{résolution inchangée}$$

Les conditions à vérifier pour qu'une famille $\{V_j\}_{j \in \mathbb{Z}}$ forme une analyse multirésolution sont les suivantes :

²Tous les exemples de la littérature sont traités dans le cas d'un facteur 2 mais rien ne semble prohiber un autre facteur

Définition 3 :

La suite $\{V_j\}_{j \in \mathbb{Z}}$ forme une *analyse multirésolution* si elle vérifie les propriétés suivantes :

1. $V_j \subset V_{j+1}$
2. $f(x) \in V_j \leftrightarrow f(2x) \in V_{j+1}$
3. $\bigcap_{j \in \mathbb{Z}} V_j = 0$
4. $\bigcup_{j \in \mathbb{Z}} V_j$ est dense dans $L^2(\mathbb{R})$
5. Il existe une fonction $g(x)$ dans V_0 telle que $\{g(x - k)\}_{k \in \mathbb{Z}}$ soit une base Riesz pour V_0 (toute combinaison linéaire finie est dense dans l'espace et la norme de toute suite finie est bornée).

La définition est donnée dans le cas continu car elle est plus facilement compréhensible que dans le cas discret et permet de comprendre tout l'intérêt d'une telle analyse. L'idée est donc d'observer la projection du signal sur chaque sous-espace dilaté, la propriété 4 assurant la reconstruction parfaite de toute fonction de $L^2(\mathbb{R})$. D'après les propriétés 2 et 5, si $\{\varphi_i^k\}_{k \in \mathbb{Z}}$ est une famille orthonormale, l'espace V_j est l'espace engendré par la base

$$\{\varphi_i^k\}_{k \in \mathbb{Z}} = \{2^{\frac{j}{2}} \varphi(2^j x - k)\}, \quad k \in \mathbb{Z},$$

la famille $\{\varphi(x - k)\}_{k \in \mathbb{Z}}$ engendrant V_0 .

2.4 Transformée en ondelettes

Les ondelettes ne sont autres que des familles de bases vérifiant les propriétés d'analyse multirésolution énoncées ci-dessus. Elles constituent donc l'application au traitement des signaux des principes mathématiques. Le principe de multirésolution est appliqué conjointement au temps et à la fréquence si bien que le pavage du plan temps-fréquence ne se fait plus à $\Delta(f)$ constant comme dans le cas de l'analyse de Fourier mais à $\frac{\Delta(f)}{f}$ constant. Ainsi, la précision fréquentielle est meilleure dans les basses fréquences et la localisation temporelle est meilleure dans les hautes fréquences.

2.4.1 Transformée en Ondelettes Continue

Dans $L^2(\mathbb{R})$ espace des signaux d'énergie finie, la transformée en ondelettes continue (**TOC**) d'une fonction f est définie par :

$$TOC_f(a, b) = \int_{-\infty}^{+\infty} \psi_{a,b}^*(t) f(t) dt. \quad (2.5)$$

La fonction $\psi_{a,b}$ est appelée *ondelette* et est définie par

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \text{ où } a \in \mathbb{R}^+ \quad b \in \mathbb{R} \quad (2.6)$$

Dans cette définition, les indices a et b ont respectivement un rôle de dilatation et de décalage temporel à partir d'une fonction ψ , dite ondelette mère, dont les propriétés restent à définir. Le facteur de normalisation $\frac{1}{\sqrt{a}}$ assure la conservation de la norme pour les ondelettes, soit $\|\psi_{a,b}(t)\| = \|\psi(t)\|$. Le produit scalaire de f avec la $\psi_{a,b}$ permet de mesurer les similitudes entre f et l'ondelette mère dilatée d'un facteur a et ce au temps b . Pour cette raison, la fonction d'ondelette se doit d'être une fonction élémentaire dont les propriétés sont bien connues. Olivier Rioul décrit par exemple les ondelettes comme les vibrations les plus élémentaires et les plus courtes envisageables.

L'ondelette mère et sa transformée de Fourier Ψ doivent vérifier les propriétés suivantes

$$\int_{-\infty}^{+\infty} \psi(t) dt = \Psi(0) = 0 \quad (\text{moyenne nulle}) \quad (2.7)$$

$$\int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega = C_\psi < +\infty \quad (\text{spectre à décroissance rapide}) \quad (2.8)$$

Il est alors à noter que ces propriétés restent valables pour toutes les ondelettes et définissent un comportement fréquentiel des ondelettes de type passe-bande.

Pour la reconstruction du signal, la conservation d'énergie entre la fonction et sa transformée assure la possibilité de reconstruction parfaite. Elle est analogue à l'égalité de Parseval pour la transformée de Fourier (**TF**). La propriété de conservation d'énergie se traduit par

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |TOC_f(a,b)|^2 \frac{da db}{a^2}, \quad (2.9)$$

et la reconstruction de f à partir de sa TOC s'écrit

$$f(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} |TOC_f(a,b)|^2 \psi_{a,b}(t) \frac{da db}{a^2}. \quad (2.10)$$

Dans le cas des ondelettes biorthogonales, deux ondelettes mères sont nécessaires pour l'analyse et la synthèse. La TOC est obtenue avec $\psi_1(t)$ et la synthèse

va se faire à l'aide de $\psi_2(t)$. La constante C_ψ intervenant dans la formule de conservation d'énergie fait maintenant intervenir les deux ondelettes et vaut

$$C_{\psi_1, \psi_2} = \int_{-\infty}^{+\infty} \frac{\Psi_1^*(\omega) \Psi_2(\omega)}{|\omega|} d\omega. \quad (2.11)$$

La formule de reconstruction s'écrit alors

$$f(t) = \frac{1}{C_{\psi_1, \psi_2}} \int_0^{+\infty} \int_{-\infty}^{+\infty} \langle \psi_{1_{a,b}}, f \rangle \psi_{2_{a,b}} \frac{da db}{a^2}. \quad (2.12)$$

2.4.2 Transformée en Ondelettes Discrète

La formule de construction des ondelettes est doublement continue. D'une part, les variables a et b sont réelles et d'autre part l'intégrale est continue. Dans un premier temps, la discrétisation porte sur a et b . De manière générale, on pose $a = a_0^m$ avec $m \in \mathbb{Z}$ et $a_0 \neq 1$. On pourrait poser de même pour b la valeur nb_0 avec $n \in \mathbb{Z}$ mais le choix d'un facteur d'échelle discret impose des conditions sur b pour couvrir l'axe temporel de manière complète. Il vient alors la discrétisation

$$a = a_0^m, b = nb_0 a_0^m \text{ où } m, n \in \mathbb{Z}, a_0 > 1, b_0 > 0$$

et l'ondelette qui lui est associée

$$\psi_{m,n}(t) = a_0^{-\frac{m}{2}} \psi(a_0^{-m}t - nb_0).$$

Le problème de la discrétisation est alors reporté sur la transformation de l'intégrale en une somme double sur les indices m et n . Dans le cas général, rien n'assure que le signal sera reconstruit parfaitement. Pour discrétiser l'intégrale la condition nécessaire et suffisante est la convergence de la somme discrète pour toute fonction.

$$A\|f\|^2 \leq \sum_{m,n} |\langle \psi_{m,n}, f \rangle|^2 \leq B\|f\|^2 \quad (2.13)$$

Pour la reconstruction du signal, une condition suffisante est d'avoir l'égalité des bornes A et B . Dans ce cas, la valeur des bornes est un indice de redondance lié à l'échantillonnage.

Le cas dyadique

Le cas dyadique est le plus utilisé dans le cadre algorithmique car il permet l'implémentation de transformées en ondelettes rapides (**TOR**). Il est obtenu pour $a_0 = 2$ et $b_0 = 1$ et conduit à des bases d'ondelettes orthonormales. Dans

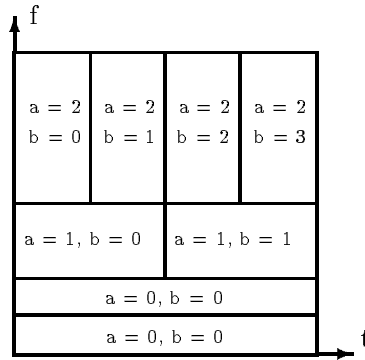


FIG. 2.2: Exemple de pavage dyadique du plan temps-fréquence

une telle base, les ondelettes ont pour norme 1 et la valeur $A = 1$ indique que l'on a la condition d'échantillonnage critique. Dès lors il n'y a aucune redondance entre les coefficients d'ondelette et la reconstruction est parfaite. Du point de vue du plan temps-fréquence, une telle discrétisation conduit à un pavage dyadique. La décomposition est aussi appelée en bande d'octave car la largeur des bandes de fréquence double à chaque itération sur a tandis que la largeur des bandes temporelles est divisée par deux. Ainsi, dans ce type de pavage $\frac{\Delta(f)}{f}$ reste constant. La figure 2.2 illustre un pavage dyadique pour a allant de 0 à 2. a est aussi appelé profondeur de la décomposition. Le lien avec les bancs de filtres est présenté à la section suivante.

2.5 Bancs de filtres

Le principe des bancs de filtres consiste en séparer le signal en différentes sous-bandes fréquentielles. Le signal est alors sous-échantillonné d'un facteur égal au nombre de filtres pour ne pas d'avoir d'expansion de l'information (échantillonnage critique). Étant donné que les filtres n'ont pas une coupure parfaite, ils induisent inévitablement des distortions dues au recouvrement. Toutefois, les filtres de synthèse permettent de compenser ces artefacts pour obtenir la reconstruction parfaite, ou tout du moins une approximation du signal aussi fine que voulue.

Du point de vue mathématique, la famille de la base envisagée pour décrire le banc de filtre est de taille infinie. Aussi, elle est obtenue avec des filtres prototypes dont les séquences sont décalées dans le temps.

Le problème qui se pose est la réalisation de filtres à temps discret. Il en existe deux types suivant que leur réponse impulsionnelle est finie (filtres **RIF**) ou infinie (filtres **RII**). Nous préférons souvent les filtres RIF, aussi appelés non récursifs car ils dépendent uniquement de l'entrée et leur réponse impulsionnelle (**RI**) a un comportement de phase linéaire.

La réalisation des filtres à temps discret peut se faire de différentes manières.

- Soit par discrétisation de filtres analogiques, en échantillonnant la réponse

impulsionnelle du filtre, ce qui peut engendrer des instabilités, ou en convertissant la transformée de Laplace en transformée en z de la fonction de transfert du filtre.

- Soit par des méthodes de conception numérique de filtres qui présentent l'avantage, pour un ordre de filtre donné, d'aboutir à des filtres plus stables et présentant de meilleures caractéristiques (coupure, ondulation ...)

2.5.1 Bancs de filtres à deux voies

Dans la construction d'un banc de filtres, la reconstruction parfaite du signal est obtenue quand les filtres d'analyse et de synthèse sont biorthogonaux entre eux. Nous présentons ici le formalisme temporel lié aux bancs de filtres. Par rapport au point de vue mathématique, il permet d'exprimer la décomposition des signaux sur des bases orthogonales ou biorthogonales en terme de convolution, donc de filtrage. Ce formalisme présente l'avantage d'être vraiment pratique et de permettre l'implémentation de nombreuses ondelettes dans des systèmes physiques. Les filtres viennent en effet du domaine de l'ingénierie dans lequel ils sont sans cesse utilisés.

Le banc de filtres à deux voies conduit à une séparation dyadique du signal, au même titre que les ondelettes.

Banc de filtres d'analyse

Le filtre prototype H_0 est le filtre passe-bas pour lequel la réponse en fréquence peut être obtenue par les méthodes classiques de conception des filtres. Le passe haut est noté communément H_1 et sa réponse en fréquence doit assurer la couverture de la bande de fréquence complémentaire à celle de H_0 . Ceci est obtenu en construisant avec H_0 le filtre miroir en quadrature de manière à avoir une paire de filtres dits QMF (filtres miroir en quadrature). La figure 2.3 illustre les notations du banc de filtres à deux voies. Par la suite, dans les schémas, les noms des filtres ne seront plus précisés, le passe bas se situant toujours en bas et le passe haut en haut.

H_0 et H_1 sont supposés d'ordre L . Leurs réponses impulsionnelles sont notées $h_0[n]$ et $h_1[n]$, pour $n \in [0, L - 1]$. Alors sous forme matricielle, en notant T_a la matrice du banc de filtre, \mathbf{x} la séquence analysée d'un signal x et \mathbf{y} la séquence transformée qui contient les deux voies filtrées. Alors

$$\begin{pmatrix} \vdots \\ y_0[0] \\ y_1[0] \\ y_0[1] \\ y_1[1] \\ \vdots \end{pmatrix} = T_a \cdot \begin{pmatrix} \vdots \\ x[0] \\ x[1] \\ x[2] \\ x[3] \\ \vdots \end{pmatrix} \quad (2.14)$$

finies que désirées. L'intérêt de ces décompositions réside alors dans la possibilité d'un **choix de la meilleure base**.

Le principe des transformées temps-fréquence consiste à décomposer le signal $s(t)$ en une combinaison linéaire d'atome temps-fréquence r_j ³. Alors, en notant S_{r_j} la transformée de s sur r_j ,

$$s(t) = \sum_{j=0}^{\infty} \alpha_j S_{r_j}.$$

L'idée maîtresse de la recherche de la meilleure base est de trouver la combinaison linéaire comprenant le moins de termes possibles. L'analyse consiste, à l'aide critère entropique, à choisir la synthèse la plus efficace donnant la compression optimale. Il est donc important de noter que le critère de compression est dans ce cas là plus algorithmique que physique.

2.6.1 Structure dyadiques

Les structures dyadiques sont basées sur les transformations scindant en deux le signal à chaque transformée. Elles utilisent donc

- pour les ondelettes, un banc de filtres QMF à deux voies conduisant à deux sous-bandes de largeur égales,
- pour les ondelettes de Malvar, c'est la partie stationnaire de la fenêtre qui est réduite pour obtenir deux fenêtres dont les transitoires gardent une forme fixe. Il est à noter que les parties d'attaque et d'extinction de la fenêtre utilisée doivent remplir entre elles les conditions 2.3.

On aboutit en fonction du nombre d'itérations, aussi appelé profondeur de la décomposition, et des séquences itérées (on peut choisir de réitérer la transformation sur une ou les deux séquences obtenues) à différents pavages du plan temps-fréquence. La figure 2.6.1 montre trois exemples de pavages obtenus par transformées en ondelettes dyadique, les filtres qui leurs sont associés ainsi que la réponse en fréquence des filtres utilisés. Dans le cas des ondelettes de Malvar, on peut reprendre exactement les figures a) et c) en inversant les axes temporel et fréquentiel. La figure c) donnerait dans ce cas l'allure des fenêtres successives utilisées.

Ces structures sont très intéressantes car elles permettent l'implémentation d'algorithmes rapides et l'étape de synthèse est souvent assez simple. Un premier type d'algorithme de recherche de la meilleure base est associé à ce type de décomposition. C'est l'algorithme **split and merge** qui consiste à comparer l'entropie d'une séquence de coefficients avec l'entropie des deux séquences obtenues à la profondeur supérieure. L'entropie la plus faible décidera de garder soit la fusion des deux séquences (*merge*) soit la séparation de la séquence mère (*split*). Cet algorithme permet aussi de mesurer le gain entropique apporté par la

³ces atomes sont toujours des rectangles de Heisenberg

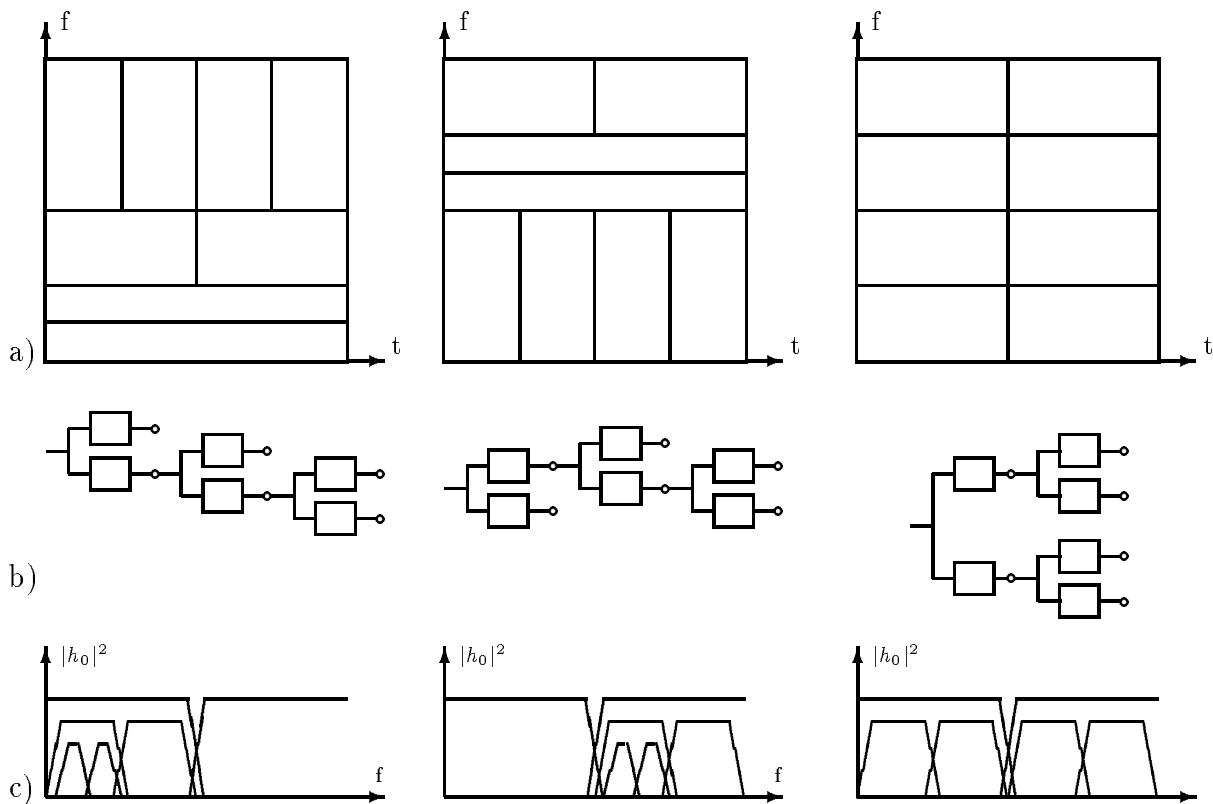


FIG. 2.4: a) Différents pavages du plan temps-fréquence b) les filtres QMF associés c) réponse en fréquence du banc de filtres

décomposition d'un segment si bien que l'on peut fixer une résolution entropique à la décomposition en paquets.

2.6.2 Structure s-dyadiques

Imad Abdallah montre dans [1] que les algorithmes *split and merge* introduisent des ruptures dans le choix de la meilleure base qui sont dues à la nature algorithmique de la décomposition et non à la nature du signal. En effet, la fusion porte toujours sur deux bandes provenant de la même séquence mère et rien ne garantit que cette fusion ne serait pas possible avec une séquence adjacente, provenant d'une autre mère.

Il propose donc une méthode élégante pour pallier ce problème. En partant de la profondeur de décomposition la plus élevée que l'on s'est fixée, il n'est plus nécessaire d'effectuer la séparation (*split*) et l'on s'intéresse à la fusion mais cette fois ci sur tous les séquences adjacentes à une même profondeur sans se soucier de leur *mère algorithmique*.

Une telle décomposition conduit pour un bloc analysé à une segmentation temporelle ou fréquentielle vraiment adaptée au bloc de signal considéré. On peut de la sorte envisager l'extension de la comparaison à deux blocs adjacents. Une idée pour réaliser cela serait par exemple d'intégrer à l'analyse d'un bloc la dernière fenêtre du bloc précédent.

Chapitre 3

Intégration de critères perceptuels

Devant les impératifs de compression du codage, un certain nombre de critères perceptuels auditifs permettent de supprimer une partie non négligeable de l'information sans dégradation subjective de la qualité. Ces résultats de la psychoacoustique applicables au codage sont exposés dans [2]. Le phénomène de masquage vient des propriétés mécaniques de l'oreille :

Un son fait vibrer la membrane basilaire. La position du maximum d'amplitude engendre une vibration plus importante des cellules ciliées situées à proximité. Cette position contribue à l'information de fréquence transmise au cerveau. Ainsi, d'autres vibrations moins importantes à proximité du maximum peuvent être considérées comme une contribution du même son. Ceci est à l'origine du phénomène de *masquage fréquentiel*.

De plus, le temps de réponse de la membrane peut entraîner que des sons survenants ultérieurement ou antérieurement soit masqués. Ceci est à l'origine du phénomène de *masquage temporel*. Le masquage temporel est difficilement exploitable de manière précise dans le cadre du codage. Qualitativement, lorsque l'on émet deux sons à quelques millisecondes d'intervalle, on retient qu'il y a premièrement un *pré-masquage* de l'ordre de la *ms*, c'est à dire qu'un son peut être masqué fréquentiellement même s'il est émit quelques *ms* avant le son le masquant, deuxièmement un masquage simultané de l'ordre de *10ms*, c'est à dire que l'on distingue les sons comme émis simultanément alors qu'ils sont émis jusqu'à *10ms* décart, et troisièmement un *post-masquage* de l'ordre de 5 à *10ms*. Ces grandeurs permettent de choisir les longueurs des fenêtres d'analyse suivant la nature du signal. Pour les phases transitoires des signaux, par exemple, des fenêtres de longueur 2 à *5ms* permettent lors de la synthèse de bons rendus perceptifs.

Dans le cadre du codage audio, le phénomène le plus exploité est le masquage fréquentiel car le but est de quantifier le spectre en lui allouant les bits de telle

sorte que le bruit de quantification¹ soit inaudible. Il faut donc estimer pour un spectre quelconque (aussi appelé complexe) le **seuil d'audition masqué** c'est-à-dire le niveau de signal masquant pour lequel le bruit est masqué. Les expériences difficilement réalisables pour l'étude de ce phénomène ont consisté à décomposer le problème en deux cas plus simples qui sont les cas d'un *bruit masquant un bruit* et d'une *tonale masquant un bruit* puis en réunissant de manière cohérente les résultats trouvés. Les expériences ont montré que le calcul des courbes de masquage passait par la connaissance de deux valeurs empiriques qui sont l'excitation et le taux de masquage.

3.1 Excitation

C'est l'étalement de la puissance sur la membrane basilaire. L'oreille intègre la puissance d'un signal sur une certaine bande, appelée **bande critique**. Ainsi, un bruit de largeur Δf supérieur à une bande critique donnera la même perception d'intensité qu'un bruit de la largeur de la bande critique. La largeur des bandes critiques varie en fonction de la fréquence mais elle correspond à une longueur constante sur la membrane basilaire. C'est pourquoi une unité plus *physiologique* a été définie, le **Bark**, notée j qui correspond à la largeur d'une bande critique. La relation entre les Barks et les Hertz est donnée par la fonction B

$$j = B(f_0) = 13 \arctan(0,76f_0) + 3,5 \arctan\left(\frac{f_0}{7,5}\right)^2. \quad (3.1)$$

Il est aussi courant d'exprimer la largeur d'une bande critique en Hertz, pour des raisons d'implémentation, ce qui est fait en approximant la fonction B^{-1} . Dans cette approximation, les largeurs Δf des bandes critiques valent

$$\Delta f = 100Hz = C^{te} \text{ pour } f < 500Hz,$$

$$\Delta f(f_0) = 25 + 75(1 + 1,4f_0^2)^{0,69} \quad (\text{linéaire en } \log f) \text{ pour } f > 500Hz.$$

Ces résultats traduisent le comportement en banc de filtre à décalage continu de l'oreille. L'information transmise au système nerveux correspond pour chaque fréquence (f) au signal filtré par un filtre dit **cochléaire** centré à la fréquence (f). Pour calculer l'excitation, il faut donc passer le signal dans un banc de filtres cochléaires, puis calculer la puissance du signal filtré sur un temps durant lequel il est supposé stationnaire. Les filtres cochléaires ont une réponse fréquentielle quasi-triangulaire qui dépend toutefois de la fréquence considérée et de son amplitude. Ils ont une largeur de bande supérieure à une bande critique ce qui amène à s'intéresser au masquage entraîné par un son (ou un bruit) sur les bandes critiques voisines.

¹voir section 5

3.2 Taux de masquage

Le taux de masquage est la différence de pression acoustique entre un son masqué juste audible et un son masquant. Il est noté av et on parle aussi du niveau de taux de masquage L_{av} en dB correspondant.

Il est lié au taux de modulation en amplitude juste audible m qui traduit la résolution en amplitude de l'oreille. Celle-ci est limitée et dépend de la fréquence et du niveau. Ainsi, deux sons d'amplitudes respectives $1 + m$ et $1 - m$ procureront donc la même sensation d'intensité acoustique. Par exemple, le taux de modulation pour un son pur vaut 1% à 100dB et 10% à 20dB. D'un point de vue expérimental, il est plus facile de déterminer m que av . Soit x le son masquant et t le son masqué, l'idée est de comparer x et $x + t$ plutôt que x et t . L'écart entre x et $x + t$ doit être inférieur au taux de modulation juste audible. Les différentes grandeurs sont reliées simplement par $L_{av} = L_x - L_t$ et en considérant que L_{x+t} est le niveau maximum, (pression $1 + m$) et L_x est le niveau minimum (pression de $1 - m$).

- Dans le cas d'un son pur masquant un bruit, $L_{av} = -(15,5 + j)$
- Dans le cas d'un bruit masquant un bruit de largeur égale à une bande critique, L_{av} varie entre $-2dB$ à $200Hz$ et $-6dB$ à $20kHz$.

Dans le cadre du codage, ces résultats permettent de prendre en compte un certain nombre de considérations qui pourront être utiles au moment de la procédure d'allocation de bit, au niveau du bruit de codage.

- L_{av} augmente si le son masquant est bruité ou si le son masqué est tonal.
- L_{av} reste constant si les sons masquant et masqué sont de même nature.
- Si le bruit de codage n'a pas de composantes tonales, il faut plus de bits pour coder une tonale qu'un bruit de bande étroite.
- Inversement, pour coder une tonale, le niveau de bruit admissible augmente si la nature du bruit devient une tonale voisine.

3.3 Seuil de masquage

Le seuil de masquage, ou seuil d'audition masqué est l'extension de la notion de taux de masquage à des sons de nature quelconque. Ceci se fait assez intuitivement en étendant les résultats obtenus pour un son pur ou un bruit à bande étroite à l'excitation sur une bande critique. Le seuil de masquage est alors la borne supérieure de la puissance du signal masqué dans chaque sous-bande basilaire. En effet, si dans ces conditions le son test dépasse le seuil dans une sous-bande, cela signifie que le bruit de codage est audible.

Le lecteur intéressé par l'implémentation précise de est invité à se reporter à [2] où les différents modèles utilisés à ce jour sont exposés. En résumé le calcul du niveau de bruit audible va se dérouler de la manière suivante :

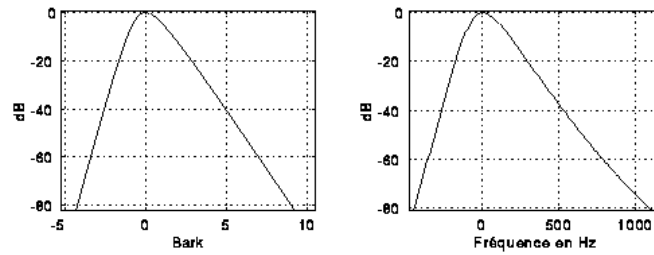


FIG. 3.1: Allure de la fonction d'étalement en bark et en Hertz

- **Discretisation des bandes critiques.** Souvent, les bandes critiques sont calées sur les valeurs entières des barks. Ainsi, il y a 25 bandes critiques, de largeur équivalente variable en Hz (la première va de 0 à 100 Hz et la dernière de 15500 à 20000 Hz). À l'aide de la fonction B les calculs peuvent se faire en Bark ou en Hertz.
- **Calcul de l'excitation**
 - Soit directement en calculant la puissance du signal convolué par les filtres cochléaires, ce qui représente une complexité de calcul très importante.
 - Soit à partir du spectre du signal par calcul de l'énergie du signal présente dans chaque sous-bande basilaire. Ceci donne le spectre basilaire qu'il faut alors convoluer avec la fonction d'étalement pour obtenir le niveau d'excitation. Il existe de nombreux modèles permettant d'implémenter cette fonction d'étalement à différentes complexités. Les plus courant sont les modèles linéaires par morceaux qui vont d'une approximation triangulaire constante à des approximations dépendant de l'amplitude et ayant des pentes différentes suivant les fréquences. La figure 3.1 montre les courbes de fonction d'étalement définies dans [14]. Elle permettent de se faire une idée de l'allure des filtres, leurs débordements sur les bandes critiques voisines et la raison de leur approximation par des fonctions linéaires par morceaux.
- **Choix du type de masquage**

Il faut pour utiliser les résultats obtenus pour les sons simples, décider de la nature tonale ou bruitée du spectre. Il existe pour cela différents indices comme par exemple la "Mesure de Platitude du Spectre" (**SFM**), définie comme le rapport de la moyenne géométrique sur la moyenne arithmétique de la DSP du signal. Il est aussi possible de scinder le spectre en composantes tonales et bruitées pour appliquer les critères de masquage séparément et ensuite prendre le seuil le plus haut. Dans ce cas là, ce sont les résultats concernant un son pur masquant un bruit et un bruit masquant un bruit qui sont utilisés.

Chapitre 4

Allocation de bit

4.1 Principe

L'allocation propose une pré-optimisation de la quantification. C'est un problème de répartition de ressources. Le principe est de répartir les \mathbf{R} bits disponibles (fixés par le débit binaire du codeur) en satisfaisant un critère qui est dans le cadre du codage la minimisation de la distortion. Il faut donc définir un vecteur d'assignement de bits $b(k)_{k=0,\dots,M-1}$ (M est la taille d'un bloc de transformée) qui donne le nombre de bits dans la bande k . Alors,

$$\sum_{k=0}^{M-1} [b(k)] = R. \quad (4.1)$$

Après la transformation d'une séquence, l'erreur \mathbf{D} est la somme des erreurs de chaque échantillon. Deux approches du problème sont possibles. Soit on cherche à minimiser \mathbf{D} en respectant le budget binaire \mathbf{R} , soit à une qualité donnée \mathbf{D} on cherche à minimiser \mathbf{R} . Il est aussi possible de faire varier le nombre de bits disponibles d'un bloc à l'autre.

Dans cette phase d'allocation, il y a toujours un compromis sur le débit et la distortion si bien que pour diminuer la distortion il faut augmenter le débit et inversement. Il est souvent difficile d'évaluer la relation exacte entre le débit et la distortion et un problème de subjectivité peut aussi créer des différences entre la théorie et la mesure.

La procédure d'allocation la plus simple est la suivante :

- On suppose connues les distortions relatives à chaque coefficient en fonction du nombre de bits qui leur est alloué.
- $R_i[n]$ est le nombre de bits alloués au $i^{\text{ème}}$ coefficient à la $n^{\text{ème}}$ itération.
- A chaque itération, on alloue un bit au coefficient quantifié qui a la plus grande distortion (au moment de cette itération).

Dans un tel algorithme, la diminution de la distortion du point de vue mathématique ne va pas forcément améliorer la qualité subjective du signal. C'est pour cette raison que les procédures d'allocation de bits courantes en codage n'ont pas comme principe l'assignement de tous les bits et utilisent les modèles psychoacoustiques de masquage de certains bits pour ne pas les assigner.

D'autre part, la mise en forme des trames va se faire à cette étape, fixant par la même la flexibilité du codeur (voir section 1.2.5). C'est en effet l'allocation qui décide des coefficients à transmettre, et cette information doit aussi être transmise au décodeur. La trame est constituée du contenu du signal mais aussi de la façon dont il est contenu. Deux types d'adaptation du contenu de la trame sont possibles. Le tableau 4.1 résume les avantages et inconvénients des deux méthodes.

- Prédiction *avant* (forward adaptative allocation)
Le nombre de bits utilisés dans chaque bande est transmis avec le train binaire. Ceci permet de résoudre le problème du choix du nombre de bits par bande au niveau du codeur et laisse une grande flexibilité au décodeur. L'inconvénient de cette technique réside dans le fait que l'information du nombre de bits occupe beaucoup de place et entraîne une perte de qualité quand le codeur transmet à des faibles débits.
- Prédiction *arrière* (backward adaptative allocation)
 $b(k)$ est calculé par le codeur mais n'est pas transmis. Le décodeur reçoit uniquement l'information nécessaire pour recalculer la répartition. L'économie de bit est utilisée pour une meilleure quantification des coefficients. La qualité du signal est alors meilleure que dans le cas précédent car tous les bits ou presque sont consacrés à la définition des fréquences. L'inconvénient tient cette fois-ci au fait que le décodeur et le codeur doivent évoluer ensemble.

Type de prédiction	Avantage	Inconvénient
Forward	Flexibilité du codeur	Bruit de quantification plus élevé
Backward	Bonne qualité	Pas de flexibilité du codeur seul

TAB. 4.1: Avantages et inconvénients des méthodes d'allocation

4.2 Allocation paramétrique basée sur des critères perceptuels

Les premières procédures d'allocation de bits pour le codage de la parole ont été mise en place vers 1977 par Zelinski et Noll. Ils ont montré que la puissance du bruit de quantification est minimisée en allouant les bits proportionnellement à la puissance du signal d'entrée. Deux ans plus tard, en 1979, Tribolet et Crochiere

ont constaté que minimiser l'erreur quadratique donnait de moins bons résultats que de tenir compte des phénomènes de masquage. Depuis les modèles de courbes de masquage deviennent de plus en plus pertinents ce qui permet d'optimiser les procédures d'allocation de bits basées sur des critères perceptuels.

L'objectif de ce type de procédure est de répartir la distribution du bruit de quantification sur le spectre du signal pour le rendre le moins audible possible, voire inaudible. Pour chaque bande de fréquence dans un bloc de signal transformé, le seuil de masquage est calculé. Les bits sont alors alloués en suivant ces courbes qui représentent le niveau au dessous duquel le bruit est inaudible.

Aucune *règle* d'allocation n'est encore définie de manière optimale dans ce domaine là. Dans les différentes normes incluant une telle allocation, seul les impératifs de décodage sont fixés afin de laisser aux différents constructeurs de codeur les possibilités d'optimiser leur algorithme comme ils l'entendent et de pouvoir améliorer la qualité du codeur par la suite. Dans l'état actuel des choses, différentes stratégies sont utilisées suivant les codeurs existants.

Dans le cas du codeur Dolby AC-3 utilisé pour les DVD, la courbe de masquage est recalculée au niveau du décodeur ce qui permet d'allouer uniquement des bits aux fréquences non masquées.

Dans le cas du codeur ATRAC utilisé dans les minidisc, toutes les fréquences sont codées mais moins de bits sont alloués aux fréquences masquées. Le premier, s'il est plus efficace au niveau de la compression (environ $\frac{1}{12}$ pour AC-3 contre $\frac{1}{5}$ pour le ATRAC), ne permet pas l'enregistrement en direct (temps-réel), ce que permet le codage du Minidisc.

Chapitre 5

Quantification

Cette phase consiste à faire correspondre les valeurs quelconques (réelles) d'un signal à un nombre fini de valeurs discrètes contenues dans un dictionnaire. Une telle opération ne va pas sans entraîner sur le signal des distortions non linéaires appelées **bruit de codage**. Ce bruit de codage est donc l'erreur commise entre le signal et sa version quantifiée. La quantification vise donc à réduire ce bruit, vu comme une erreur au sens des moindres carrés. La performance d'un quantificateur est donc mesurée par cette erreur entre l'entrée et la sortie. La mesure de cette erreur peut s'effectuer en considérant le pire des cas, mais il est plus fréquent de calculer une distortion moyenne, définie par :

$$D = E(|x - \hat{x}|^2) = \sum_i \int_{x_{i-1}}^{x_i} (x - y_i)^2 f_x(x) dx, \quad (5.1)$$

où

- \hat{x} est le signal quantifié,
- y_i est la valeur décodée,
- $\Delta = x_{i-1} - x_i$ est l'intervalle ou pas de quantification,
- $f_x(x)$ est la densité de probabilité du signal x .

Il existe trois types de quantifications, qui sont :

1. Quantification scalaire : Chaque échantillon est quantifié indépendamment.
2. Quantification vectorielle : Elle porte sur un vecteur d'échantillons. Cette quantification, bien que plus efficace, est très lourde à mettre en œuvre.
3. Quantification prédictive : C'est la différence entre l'échantillon et une valeur prédite qui est quantifiée.

Avant de présenter les différents types de quantification, une présentation qualitative des niveaux de bruits s'impose. En prenant $0dB$ comme niveau de référence pour le plus petit son audible et $120dB$ pour le niveau le plus fort avant

la douleur. $25dB$ représente le niveau de bruit dans un studio d'enregistrement et $35dB$ est le niveau de bruit dans une maison calme.

5.1 Quantification scalaire

Le signal x est supposé réparti suivant une densité de probabilité f_x . La quantification scalaire suit cette loi de répartition pour quantifier plus finement les valeurs apparaissant plus souvent. C'est le cas par exemple des lois μ ou A utilisées pour le codage des signaux de parole qui tiennent compte du fait que les fortes amplitudes portent moins d'information que les zones d'attaques. Ainsi, les faibles valeurs sont quantifiées plus finement tandis que les valeurs importantes sont codées *grossièrement*.

Le cas de la densité de probabilité uniforme est très utilisé dans le monde du codage audio. Il conduit au codage PCM de la musique. Le signal est supposé prendre ses N valeurs uniformément entre deux bornes $[a, b]$. On dispose de R bits qui définissent 2^R pas de quantification. En nous plaçant dans le pire des cas, c'est à dire en considérant $y_i = \frac{x_i - x_{i-1}}{2}$, il est facile d'évaluer la distortion moyenne :

$$D = \sum_{i=1}^N \int_{x_{i-1}}^{x_i} (x - y_i)^2 \frac{1}{N\Delta} dx = \frac{\Delta^2}{12}. \quad (5.2)$$

Ceci permet de retrouver le résultat concernant l'ajout d'un bit à la quantification scalaire uniforme. L'ajout d'un bit réduit l'intervalle d'approximation car $\Delta = \frac{(b-a)}{N}$ où $N = 2^R$. Alors

$$D' = \frac{\Delta'^2}{12} = \frac{(b-a)^2}{12} \cdot \frac{1}{N^2} = C^{te} \cdot 2^{-2(R+1)} = \frac{D}{4} \quad (5.3)$$

L'ajout d'un bit diminue donc la distortion du signal de $6dB$, car $10 \log_{10} 4 = 6$.

Le calcul ne se fait pas toujours aussi bien et dans les lois non linéaires de quantification, l'ajout d'un bit n'aura rien d'aussi uniforme. Toutefois, quelque soit la loi de distribution du signal, deux principes minisant l'erreur quadratique sont à retenir :

- *Règle du plus proche voisin* : les entrées doivent être quantifiées à la valeur la plus proche dans le dictionnaire.
- *Condition du centre de gravité* : la valeur optimale au niveau du décodeur (phase de quantification inverse) est la moyenne de la distribution de l'entrée sur chaque intervalle.

5.2 Quantification prédictive

C'est une version *améliorée* de la quantification scalaire qui code chaque échantillon séparément, mais en tenant compte de l'évolution des échantillons

du premier en ordonnée et du second en abscisse va permettre de visualiser les concentrations sous forme de taches. La quantification consiste alors à partitionner le plan de manière optimale, c'est à dire en ayant le moins de pavés distincts avec la plus petite erreur possible.

Le travail de quantification vectorielle, possiblement réalisable à deux ou trois dimensions, devient vite d'une complexité énorme quand la taille des vecteurs augmente. De par sa nature, la taille du dictionnaire croît de manière exponentielle à chaque ajout d'un vecteur et le codage en devient donc proportionnellement complexe. De plus, le décodeur doit lui aussi disposer de ce dictionnaire qui sera de par sa complexité difficilement adaptatif.

Bien que difficilement réalisable, ce type de quantification n'est pourtant pas à écarter. Les grands groupes de recherche dans le domaine du codage mettent aujourd'hui en place des méthodes de quantification vectorielle pour optimiser les différents codeurs existants. Ainsi, le codeur TwinVQ (Twin Vector Quantization) propose une amélioration du standard MPEG2-III incorporant une quantification vectorielle et figure dans la norme MPEG4.

Chapitre 6

Codage entropique

6.1 Quantité d'information et Entropie

Le codage entropique est une méthode de compression sans perte qui peut être appliquée dans tous les domaines cernés par la théorie de l'information. Cette théorie s'intéresse à la quantité d'information véhiculée par un signal. De façon intuitive, la quantité d'information est associée à la mesure du temps et du coût nécessaire à la transmission du message. La quantité d'information d'un message ou entropie de la source qui le génère mesure la redondance, ou le désordre dans le message à transmettre.

Définition 4 :

L'entropie d'une source x constituée de N mots apparaissant avec les probabilités $\{p_i\}_{i=1,\dots,N}$ est notée $H(x)$ est vaut

$$H(x) = \sum_{i=1}^N -p_i \log_2 p_i \quad \text{où} \quad \sum_{i=1}^N p_i = 1 \quad (6.1)$$

Il est facilement démontrable que l'entropie d'une source est maximale si le désordre est maximal donc si tous les mots apparaissent de manière équiprobable. A l'inverse, une source dans laquelle la distribution des mots est non uniforme pourra être codée avec un gain non négligeable par rapport à un code binaire par exemple.

L'efficacité η d'un code est mesurée en comparant la longueur moyenne de codage avec la longueur minimale définie par l'entropie. La longueur moyenne correspond au centre de gravité des mots de code pondérés par leur probabilités respectives. La longueur minimale est une borne inférieure pour le code fixée par l'entropie de la source.

Définition 5 :

La longueur moyenne d'une source codée en base q composée de N mots de longueur $\{l_i\}_{i=1,\dots,N}$ apparaissant avec les probabilités $\{p_i\}_{i=1,\dots,N}$ vaut

$$l_{moy} = \sum_{i=1}^N p_i l_i$$

Définition 6 :

L'efficacité η d'un code est le rapport

$$\eta = \frac{l_{min}}{l_{moy}} \quad \text{où} \quad l_{min} = \frac{H(x)}{\log_2(q)},$$

l_{min} est la longueur minimale des mots de code fixée par l'entropie.

Cette efficacité peut aussi être vue comme un rendement du codeur et la quantité $1 - \eta$ mesure le taux de redondance du code.

Le codage entropique consiste donc à tenir compte des critères entropiques pour trouver un code optimal. En d'autres termes, il faut redéfinir un dictionnaire où les séquences apparaissant le plus souvent sont codées par des mots courts tandis que celles apparaissant rarement vont être codées par des mots plus longs. Le but est de se rapprocher de la borne inférieure entropique fixée par l_{min} . Un exemple, connu de tous, de code entropique est le code morse qui prend en compte les probabilités d'apparition des lettres dans le langage pour coder les plus fréquentes par des séquences courtes.

Cette dernière étape n'est donc pas spécifique au codage audio mais peut présenter de bon taux de compression. Dans notre cas, elle n'a rien à voir avec le signal de départ et va traiter la trame binaire avant sa transmission.

Pour la trame binaire, il faut considérer les alphabets constitués du regroupement de mots binaires de longueurs différentes et voir pour quelle longueur de mots l'entropie est la plus petite. Cela revient pour chaque longueur de mots à connaître les probabilités d'apparition des différents mots. Une fois la longueur permettant la meilleure compression trouvée, il faut appliquer à l'alphabet un codage minimisant l'entropie moyenne de la séquence. Il existe différentes méthodes comme celles de Huffman ou de Fano permettant de réaliser ceci de manière assez simple.

6.2 Recherche du code optimal

Le code recherché doit vérifier deux propriétés qui sont l'unicité du code et la propriété de décodage instantané. Un code vérifiant ces deux propriétés est dit instantané.

Pour qu'un code soit instantané, il faut et il suffit qu'aucun mot du code ne soit le début, ou préfixe, d'un autre mot du code

Cette règle du préfixe impose que dans une arborescence de code, si un mot est retenu pour le code, aucun des mots générés par ce nœud ne seront utilisés. La figure 6.1 illustre dans le cas d'une arborescence binaire un choix possible des mots respectant la règle du préfixe.

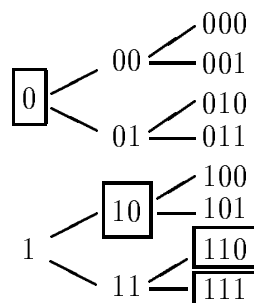


FIG. 6.1: Choix d'un dictionnaire respectant la règle du préfixe, les mots encadrés sont retenus

Ils existent plusieurs techniques simples qui permettent d'approcher la limite inférieure de l_{moy} . Les plus connues sont les méthodes de Fano et d'Huffman qui conduisent à un code instantané proche de la solution optimale.

Méthode de Fano

Cette méthode permet d'obtenir un code binaire instantané. La procédure d'obtention de ce code est la suivante.

- Classer les différents messages de la source dans l'ordre des probabilités décroissantes.
- Diviser l'ensemble de ces messages en deux sous ensembles de probabilités aussi proches que possible
 - On attribue 0, par exemple au premier,
 - On attribue alors 1 au second.
- Continuer les subdivisions jusqu'à obtenir un mot de code pour chaque message.

Méthode d'Huffman

Cette méthode permet d'obtenir un code de base quelconque mais nous ne présentons ici que le cas du code binaire.

- Classer les différents messages de la source dans l'ordre des probabilités décroissantes.
- Contracter les deux messages de probabilités les plus faibles en leur assignant 0 et 1, ce qui va constituer la fin du mot de code.
- Reclasser les différents messages de la source dans l'ordre des probabilités décroissantes et réitérer l'opération précédente.

Le tableau 6.1 compare le résultat obtenu par un codage en binaire simple et par un codage utilisant les deux méthodes. Dans l'exemple proposé, et c'est souvent le cas, les méthodes de Fano et d'Huffman donnent le même code. Il est à noter que la méthode d'Huffman est surtout intéressante pour obtenir un code presque optimal dans une base quelconque.

Symbole à coder	Probabilité d'apparition	Codage binaire	Code de Fano	Code de Huffman
a	0.55	000	0	0
b	0.15	001	100	100
c	0.1	010	101	101
d	0.05	011	1100	1100
e	0.05	100	1101	1101
f	0.05	101	1110	1110
g	0.025	110	11110	11110
h	0.025	111	11111	11111
efficacité		71 %	99,1 %	99,1 %

TAB. 6.1: Comparaison de méthodes de codage

Chapitre 7

Objectifs de la thèse

La conception d'un codeur est un travail fastidieux qui nécessite outre des compétences diverses comme le traitement du signal, l'acoustique, l'algorithmique, une bonne connaissance des évolutions incessantes de ce domaine. Aussi, l'optimisation de toutes les parties du codeur n'est pas envisageable car chaque partie peut à elle seule susciter un travail de recherche.

Dans le cadre de cette thèse, les objectifs portent pour l'instant sur l'optimisation de la segmentation du signal conjointement en temps et en fréquence. La motivation pour ceci vient des résultats obtenus par Imad Abdallah [1] proposant des algorithmes performants de recherche de la meilleure base basés sur des critères entropiques. L'application au codage semble découler naturellement des impératifs d'élimination de la redondance de l'information qui fondent la compression.

Nous comptons premièrement utiliser la segmentation s-dyadique portant sur les ondelettes de Malvar pour obtenir des tranches de signal représentant des *unités élémentaires* sonores. Ensuite, l'utilisation des paquets d'ondelettes s-dyadiques sur chacun des segments temporels permet d'obtenir la meilleure décomposition en sous-bandes fréquentielles. Les coefficients découlant de ces segmentations imbriquées doivent alors fournir une information simple sur le signal, facilement codable. Parallèlement, nous comptons utiliser les courbes de masquage, pour alléger encore l'information contenue dans chaque segment de signal.

Enfin, ceci s'insère dans le cadre de la conception d'un codeur, dont les différentes parties implémenteront des solutions classiques utilisées dans les différents codeurs existants, l'optimisation portant donc sur la transformation du signal.

Bibliographie

- [1] Imad Abdallah. *Représentations Temps-Fréquence adaptatives de signaux acoustiques basées sur des critères entropiques*. PhD thesis, Université du Maine, December 1998.
- [2] Marcos Perreau Guimar aes. *Optimisation des ressources binaires et modélisation psychoacoustique pour le codage audio*. PhD thesis, Université Paris V, June 1998.
- [3] Albert Cohen. *Ondelettes et traitement numérique du signal*. Masson, 1992.
- [4] Grant A. Davidson, Louis D. Fielder, and Brian D. Link. Parametric bit allocation in a perceptual audio coder. In *Audio Engineering Society 97th Convention in San Francisco*, November 1994. Available at <http://www.dolby.com/tech/parametr.html>.
- [5] Ricardo L. de Queiroz and K. R. Rao. Time-varying lapped transforms and wavelet packets. *IEEE transactions on signal processing*, 41(12):3293–3305, December 1993.
- [6] Hervé Dia. *Codeur à débit variable*. PhD thesis, Université de Paris V, June 1992.
- [7] Barbara Burke Hubbard. *Ondes et Ondelettes, la saga d'un outil mathématique*. Pour la Science. Belin, 1995.
- [8] Henrique Malvar. Enhancing the performance of subband audio coders for speech signals. In *IEEE International Symposium on Circuits and Systems - Monterey*, June 1998. available at <http://www.research.microsoft.com/users/malvar/papers/index.htm>.
- [9] Henrique S. Malvar. Lapped biorthogonal transform for transform coding with reduced blocking and ringing artifacts. In *IEEE ICASSP Conference, Munich*, April 1997. available at <http://www.research.microsoft.com/users/malvar/papers/index.htm>.
- [10] Yves Meyer. *Les ondelettes, Algorithmes et Applications*. Armand Colin, 1992.
- [11] Yves Meyer. *Ondelettes et algorithmes concurrents*. Hermann, 1992.
- [12] Nicolas Moreau. *Techniques de compression des signaux*. Techniques et sciences des télécommunications. Masson, 1994.

- [13] Pierre Moulin and Mehmet Kivanç Mihçak. Theory of signal-adapted fir paraunitary filter banks. *IEEE transactions on signal processing*, 46(4):920–929, April 1998.
- [14] J. Picaut. Méthode de réduction de bruit basées sur des modèles psychoacoustique. Technical report, Laboratoire d’Acoustique de l’Université du Maine, 1994.
- [15] Seymour Shlien. The modulated lapped transform, its time-varying forms, and its applications to audio coding standards. *IEEE transactions on speech and audio processing*, 5(4):359–366, July 1997.
- [16] Pramila Srinivasan and Leah H. Jamieson. High-quality audio compression using an adaptive wavelet packet decomposition and psychoacoustic modeling. *IEEE transactions on signal processing*, 46(4):1085– 1093, April 1998.
- [17] Kyoya Tsustui, Hiroshi Suzuki, Osamu Shimoyoshi, Mito Sonohara, Kenzo Akagiri, and Robert M. Heddle. Atrac: Adaptive transform acoustic coding for minidisc. In *Audio Engineering Society Convention in San Francisco*, October 1992. Available at http://www.jyu.fi/minidisc/minidisc/aes_atrac.html.
- [18] Martin Vetterly and Jelena Kovačević. *Wavelets and subbands coding*. Prentice-Hall, 1995.