Modèles granulaires pour les signaux sonores : contributions théoriques et expérimentales

Lorcan Mc Donagh

▶ **To cite this version:**

Lorcan Mc Donagh. Modèles granulaires pour les signaux sonores : contributions théoriques et expérimentales. Traitement du signal et de l'image [eess.SP]. Rennes 1, 2005. Français. NNT : 2005REN1S047 . tel-01363446

HAL Id: tel-01363446
https://theses.hal.science/tel-01363446

Submitted on 9 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre: 42

THÈSE

Présentée devant

devant l'université de Rennes 1

pour obtenir

le grade de : Docteur de l'université de Rennes 1
Mention Traitement du Signal

par

Lorcan Mc Donagh

Équipe d'accueil : METISS
École doctorale : MATISSE
Composante universitaire : IRISA/INRIA

Titre de la thèse :

Modèles granulaires pour les signaux sonores : contributions théoriques et expérimentales

soutenue le 12/04/2005 devant la commission d'examen

M. :	Delyon	Bernard	Président
MM. :	Régine	André-Obrecht	Rapporteurs
	Yves	Grenier	
MM. :	Geoffroy	Peeters	Examinateurs
	Pierrick	Philippe	
M. :	Frédéric	Bimbot	Directeur de thèse

Citer les mots de quelqu'un, c'est mettre sous verre
une collection de beaux papillons
qui ont perdu leur lumière et leur éclat.

Oscar Wilde

A mon père, Christopher.

Remerciements

Je tiens en premier lieu à remercier Frédéric Bimbot, Chargé de Recherches et Responsable Scientifique de l'équipe METISS à l'IRISA, pour avoir encadré mes travaux pendant la durée de cette thèse. Ses cours de traitement de la parole à l'ENSEA de Cergy-Pontoise, toujours très vivants et originaux, ont profondément modifié ma conception des sciences, et surtout m'ont donné le goût de la recherche et l'envie d'entreprendre ces travaux. Je lui suis reconnaissant de m'avoir confié un rôle d'« éclaireur » pour explorer la thématique de l'analyse/synthèse sonore, à l'époque naissante dans le projet METISS, et de m'avoir donné tant d' autonomie et de liberté dans l'organisation de mon travail.

Je voudrais exprimer ma sincère gratitude à Rémi Gribonval, qui m'a également encadré lors de ce travail de thèse. Son enthousiasme à partager ses connaissances et sa vision du traitement du signal sonore, sa disponibilité et sa rigueur scientifique ont contribué pour une grande part à ce que cette thèse se déroule dans les meilleures conditions.

J'ai été très honoré que Régine André-Obrecht, Professeur à l'Université de Toulouse III et Yves Grenier, Directeur du Département TSI à l'Ecole Nationale Supérieure des Télécommunications, acceptent d'être les rapporteurs de cette thèse. J'ai été particulièrement sensible à l'attention qu'ils ont porté à mes travaux ainsi qu'à la qualité et à la précision de leurs remarques, qui furent très riches en enseignements.

J'ai beaucoup apprécié le soin avec lequel Geoffroy Peeters, Chargé de Recherches à l'Institut de Recherche et Coordination Acoustique/Musique, a examiné ce travail ; son regard et ses conseils avisés de spécialiste de l'analyse/synthèse musicale ont été particulièrement précieux.

Je remercie vivement Pierrick Philippe, Ingénieur à France Télécom Recherche et Développement de Cesson-Sévigné, entre autres, d'avoir apporté son expertise pour évaluer l'intérêt pratique de ces travaux dans le domaine du codage et de la compression.

Enfin, j'ai été très flatté que Bernard Delyon, Professeur à l'Université de Rennes I, ait accepté de présider le jury de soutenance de cette thèse.

Je tiens également à témoigner toute mon amitié et mon estime à Laurent Benaroya, et le remercie chaleureusement pour les innombrables discussions scientifiques ou non, que nous avons eu ensemble

Mon séjour au sein de METISS n'aurait certainement pas été aussi agréable et enrichissant, à tous points de vue, sans la présence dans l'équipe de Mathieu Ben, Mickaël Betser, Raphaël Blouet, Marie-Noëlle Georgeault, Guillaume Gravier, Ewa Kijak, Sylvain Lesage, Alexei Ozerov et Fabienne Porée, entre autres. Je remercie également l'ensemble des membres de l'équipe système pour leur efficacité, leur promptitude et leur bonne humeur inébranlable, en dépit de mes requêtes techniques qui ont parfois pu confiner au

harcèlement, ainsi que le personnel de la cafétéria de l'IRISA, dont le café et les plaisanteries m'ont maintes fois maintenu éveillé.

Merci à Anne, Claire, Estelle, Fabrice, Gérald, Grégory, Jean-Damien, Thomas, Mathieu, Romain et les autres qui m'ont relu, écouté, parfois subi et toujours soutenu, par la musique de leurs instruments et de leurs mots ...

A mes enchanteurs, Karine, Louis, Sabine et Christopher qui, quand le chemin est rocailleux, le bordez d'arbres en fleurs - sinueux, me montrez comment voir par delà l'horizon, merci de m'avoir appris à savoir ne pas compter l'inestimable.

Résumé

Les techniques de synthèse sonore actuelles permettent de reproduire une grande variété de sons à partir de quelques paramètres. Du point de vue l'analyse, les signaux audio sont généralement représentés par une combinaison d'objets de forme sinusoïdale, en grand nombre, auquel on adjoint éventuellement une partie transitoire et/ou bruit. Qu'advient-il si remplace ces objets dérivés d'une famille de fonctions typiquement sinusoïdale, par des objets obtenus à partir d'une fonction de synthèse sonore ?

Cette thèse est consacrée à l'étude d'un modèle dit granulaire, et apporte un élément de réponse à la question ci-dessus. Les caractéristiques générales de ce modèle sont présentées et discutées, et on propose un exemple de méthode de calcul pratique du modèle. La première classe de modèles étudiée est dérivée de la synthèse par Table d'Ondes et les objets y sont sélectionnés parmi un dictionnaire de formes d'ondes complexes puis soumis à plusieurs déformations. Le second type de modèle, appelé TS, exploite un dictionnaire de profils de Densités Spectrales de Puissance de référence couplé à une fonction paramétrique de déformation du spectre de phase.

Les travaux présentés apportent une contribution au niveau du formalisme et des algorithmes de classification, et proposent une approche originale au problème de la représentation efficace de signaux, dont la compression audio est une application naturelle.

Les détails techniques de l'implémentation et les résultats des évaluations, menées sur un ensemble de signaux sonore réels, figurent également dans ce document, ceci afin de faciliter l'évaluation des performances du modèle.

Ce travail explore donc les possibilités d'incorporer des éléments issus du domaine de la synthèse sonore dans la modélisation du signal, et nous espérons que cette tentative de rapprocher les deux domaines sera accompagnée par de nouveaux développements aussi bien théoriques que pratiques.

Table des matières

Table des figures

Algorithmes

Introduction

L'objectif de ce travail est d'étudier un modèle dit *granulaire* du signal sonore, dans lequel le signal est représenté par un nombre restreint de formes d'ondes déformables. Cette étude s'inscrit dans la problématique plus large de la représentation efficace de signaux sonores, qui vise à dépasser certaines limitations propres aux représentations linéaires.

Une façon d'obtenir une représentation efficace est de décomposer le signal sur un ensemble d'objets choisi parmi une famille plus grand que la dimension du signal. On parle dans ce cas de famille surdéterminée ou de dictionnaire, car cette famille contient alors un nombre d'éléments supérieur à celui qui serait nécessaire pour représenter le signal avec une famille qui forme une base. C'est le caractère surdéterminé du dictionnaire qui conduit à l'efficace de la représentation, celle-ci n'incorporant alors que les objets pertinents par rapport au signal considéré.

Dans la plupart des approches, comme le Matching Pursuit (MP), méthode des Frames, sélection/adaptation de bases, etc. les objets du dictionnaire sont définis par le biais d'une famille de fonctions. Les fonctions utilisées pour les signaux audio sont essentiellement des fragments de sinusoïdes modifiées (atomes de Gabor, chirplets, etc.), qui sont bien adaptées pour la représentation efficace du contenu harmonique.

En pratique, étant donné la complexité des signaux rencontrés, un seul objet de ce type ne suffit généralement pas à représenter correctement la forme locale du signal. En supprimant la contrainte que le dictionnaire soit défini par une famille de fonctions, il devrait être possible d'utiliser des objets plus pertinents, c'est-à-dire dont la forme approche au plus près (localement) celle du signal considéré. Ce travail a consisté à vérifier cette hypothèse dans le cadre particulier d'un modèle semi-paramétrique dit "granulaire".

Avec le modèle granulaire, le signal est décomposé sous forme d'une combinaison linéaire d'*objets*, chaque objet étant engendré par déformation d'un élément du dictionnaire, appelé *grain* (ou *prototype*) . La fonction dite de *déformation* ou de *synthèse,* qui effectue la correspondance entre grains et objets, permet de construire plusieurs versions d'un objet à partir d'un seul et unique grain. L'utilisation combinée d'un grain-prototype et de la déformation permet de construire une collection d'objets de formes différentes, que l'on

peut voir comme l'ensemble des réalisations ou des variantes possibles de la forme de base définie par le prototype.

Le travail présenté dans ce document de thèse a consisté à mettre en place un cadre formel à partir de l'idée de modèle granulaire, définir la problématique associée et y rechercher des solutions qui permettent en pratique d'analyser un signal suivant ce modèle, pour enfin vérifier la validité du modèle à partir d'expériences.

Organisation du document

Ce document s'articule autour de quatre parties :

1. Présentation du sujet et état de l'art
2. Le modèle granulaire
3. Expérimentation et évaluation
4. Conclusions et perspectives

Présentation

La première partie présente un certain nombre de concepts spécifiques aux signaux audio. Le premier chapitre contient une description générale des méthodes de production d'un signal audio musical. On effectue notamment un parallèle entre les notions musicales de partition, d'orchestre et d'instrument d'une part, et les descriptions type MIDI ou SA issues de l'informatique musicale. On s'intéresse plus particulièrement à l'aspect de la synthèse musicale sans rentrer dans les détails de la composition. L'état de l'art est ainsi consacré aux méthodes de synthèse musicale les plus courantes.

Le second chapitre est dévolu à la modélisation de signaux, plus particulièrement à la modélisation par la méthode connue sous le nom de l'analyse par la synthèse. A cette occasion, on donne un certain nombre d'exemples d'applications reposant sur un schéma type analyse-(re-)synthèse.

Le modèle granulaire

Dans cette partie, nous exposons le modèle granulaire, qui est la contribution principale de ce travail, et proposons une méthode pratique permettant de calculer le modèle à partir d'un signal. Le premier chapitre de cette partie débute par une présentation du modèle sous sa forme générale, suivie par une discussion consacrée au possibilités concernant l'analyse d'un signal suivant ce modèle. On donne un exemple simple de modèle granulaire inspiré de la synthèse par Table d'Ondes (TO), et on réexamine une méthode classique, le codage CELP, vue sous l'angle granulaire.

Le deuxième chapitre expose la méthode proposée pour calculer un dictionnaire contenant les grains pertinents pour le signal analysé, grains qui sont alors qualifiés de *prototypes*. Le problème principal consiste à calculer pour un signal donné un dictionnaire qui minimise conjointement la qualité du signal re-synthétisé à partir du dictionnaire d'une part, et la complexité du dictionnaire d'autre part.

Notre choix s'est porté sur une approche par classification non-supervisée (clustering), pour laquelle on peut faire appel à des algorithmes classiques comme le K-Médians (variante des K-Moyennes). Nous proposons également d'autres algorithmes élaborés et expérimentés dans le cadre de ce travail, notamment un algorithme qui ne nécessite pas de spécifier le nombre de classes au préalable. En effet, le dictionnaire est ici construit de manière incrémentale, et ce jusqu'à ce que l'erreur de reconstruction descende en deçà d'un seuil fixé à l'avance.

Le troisième et dernier chapitre de cette partie est dévolu au choix d'un modèle de synthèse approprié pour les signaux audio. On part d'un modèle de base dénommé TO, qui est en fait équivalent à une représentation linéaire des trames sur une famille libre et non-génératrice de l'espace signal des trames. Les modèles suivants sont dérivés de ce modèle de base, la différence principale se situant au niveau de la fonction de synthèse d'un objet à partir du prototype qui n'est plus linéaire.

Le modèle TS utilise quant à lui un dictionnaire constitué de profils de densités spectrales choisies pour être les plus représentatives de l'ensemble des trames. Ce modèle pose la question de la modélisation du spectre de phase d'objets dont la DSP est connue, un problème difficile qui a semble-t-il été assez peu étudié par la communauté du traitement du signal audio et de parole, aussi on avance quelques idées pour tenter d'y répondre.

Expérimentation et évaluations

Le protocole expérimental est rapidement présenté dans le premier chapitre, puis dans un deuxième temps, on présente les résultats commentés des expériences menées au cours de cette thèse. On y rapporte les résultats comparés obtenus à partir des différentes modèles proposés, selon les différents choix possibles pour l'apprentissage du dictionnaire. Est également étudiée l'influence de paramètres « secondaires » lorsque ceci a été jugé nécessaire.

Conclusions et perspectives

Un chapitre présentant les conclusions sur le travail mené figure en clôture de ce document. Ce travail est loin d'apporter une solution définitive au vaste ensemble des problèmes posés ; il constitue une contribution prospective qui appelle un travail de recherche sup-

plémentaire, avant d'atteindre le stade applicatif. En conclusion, nous évoquons donc plusieurs perspectives concernant ces recherches additionnelles, ainsi que un certain nombre d'applications envisageables.

Première partie

Présentation

Cette partie constitue une introduction à la modélisation de signaux musicaux.

Le premier chapitre expose les spécificités de ce type de signaux musicaux, que l'on attribue à la présence d'une structure musicale préexistante au signal lui-même. Les normes MIDI et MPEG4-Structured Audio, qui sont deux exemples d'applications courantes exploitant cette structure pour aboutir à une représentation efficace d'un signal musical, sont décrites.

On dresse ensuite un état de l'art simplifié de la synthèse sonore. Par synthèse sonore, on comprend l'ensemble des moyens techniques permettant de produire un son "isolé", "élémentaire" comme par exemple une note de musique, un phonème ou un diphone, un timbre, que l'on appelle ici "objets sonores".

Le deuxième chapitre s'intéresse à la dualité existant entre les processus d'analyse et de synthèse, illustrée à travers un certain nombre d'exemples à la fois techniques et pratiques. On présente la technique générale de modélisation connue sous le nom d'*analyse par la synthèse*, puis quelques une de ses applications à l'audio.

Chapitre 1

Structure d'un signal musical

Les signaux dont il est ici question appartiennent à la « classe » des signaux audio musicaux. Le terme de classe est ici mis entre guillemets pour insister sur le fait que le caractère musical ou non d'un son n'est pas régi par des critères strictement mathématiques. Tenter de donner une définition de la *musicalité* d'un signal relève en effet plutôt du domaine de l'Esthétique, ce qui dépasse le cadre de ce travail.

Le présent chapitre introduit et motive l'idée qu'un signal musical peut être vu comme une collection d'*« objets »* sonores superposés dans l'*espace-signal*. Nous commençons par justifier l'idée d'une représentation structurée d'un signal musical à base d'*objets* sonores (1.1).

On s'intéresse ensuite aux différentes manières dont ces objets sonores peuvent être « construits », à travers un aperçu des méthodes de *synthèse* musicale existantes (1.2).

1.1 Le signal musical : une superposition d'*objets sonores*

L'utilisation du terme « objet » est relativement courante en traitement des images, elle l'est peut-être moins en traitement du signal audio. Un objet physique a la propriété de transformer la lumière qui l'atteint suivant différents modes (absorption, réflection, diffraction, etc.) et peut également émettre de la lumière.

A un objet physique on peut donc associer une ou plusieurs images, par le biais d'un capteur d'image.

Les images naturelles, par opposition aux images de synthèse, comportent fréquemment des objets superposés. Le mode de superposition de ces objets peut être complexe, par exemple si un objet est translucide, mais il est souvent réduit à une occlusion, c'est-à-dire qu'un objet masque les objets situés derrière lui, par rapport au point de vue d'observation.

Un objet physique a également la propriété de transformer une onde acoustique incidente, et il peut également produire un son, et donc être une *source sonore*.

FIG. 1.1 – Structure de production (simplifiée) d'un signal musical

Il est donc également possible d'associer un ou plusieurs sons à un objet physique. Notons que les effets de masquage existent également dans le domaine sonore, bien qu'il soit rare qu'un son en masque intégralement un autre. On considère ici exclusivement les signaux sonores musicaux, on ne s'intéressera donc exclusivement qu'aux sons produits par des instruments, acoustiques ou « électroniques ».

Un enregistrement d'un morceau de musique est la résultante d'une chaîne de production sonore, qui comporte notamment un *orchestre* et une *partition*, ces deux termes devant ici s'entendre dans un sens large. La partition désigne un document qui définit la façon dont le morceau est joué par les couples instrumentistes-instruments, ces derniers constituant l'orchestre. Le terme d'orchestre désigne ici un ensemble d'indications plus ou moins précises sur la manière de produire un son à partir de la partition. Cette chaîne de production sonore est schématisée sur la figure 1.1.

Les notions « traditionnelles » de partition et d'orchestre sont introduites en 1.1.1, puis on explique comment ces notions sont d'ores et déjà exploitées dans le contexte de l'informatique musicale (MIDI, §1.1.2) et du codage audio (MPEG4-Structured Audio, §1.1.3).

1.1.1 Structure générale d'un signal musical

La partition

La partition est un document où sont regroupées des indications de jeu destinées aux instrumentistes, indications auxquelles ces derniers se réfèrent pour exécuter une œuvre musicale. Ces indications sont exprimées à l'aide d'un système de notation codifié afin d'éviter de possibles ambiguïtés d'interprétation, mais également suffisamment concis pour permettre une lecture rapide et aisée.

Les informations de la partition sont écrites sur une ou plusieurs *portées*, où figurent notamment des symboles définissant les caractéristiques des notes de musique à jouer. Ces caractéristiques incluent l'instant auquel la note doit être jouée, sa durée, sa position sur une échelle de hauteur, etc. Les caractéristiques temporelles du son sont définies en subdivisons d'unités élémentaires de temps, la valeur de ces unités est quant à elle définie par le tempo.

Sans pour autant rentrer plus dans les détails, il est important de préciser que la partition ne définit pas le résultat sonore de façon univoque. En effet, le musicien dispose d'une certaine marge d'interprétation autour des indications portées sur la partition.

Cependant, un morceau de musique n'est pas non plus nécessairement le fruit de l'interprétation d'une partition par des musiciens jouant sur des instruments. Par exemple, dans la musique improvisée, on s'affranchit plus ou moins totalement de la partition, tandis que la musique dite « électronique » est produite sur des instruments électroniques tels que des séquenceurs, synthétiseurs, échantillonneurs ...

Il est toutefois possible d'établir une partition *a posteriori*, a partir d'un enregistrement audio par exemple, auquel cas on parle de relevé musical.

De manière générale, on désignera par « partition » un ensemble d'indications nécessaires à une reproduction plus ou moins fidèle et reproductible d'un morceau de musique. La norme MIDI, analogue informatique de la partition papier, constitue un exemple concret de système de représentation musicale largement répandu.

Par la suite, on qualifiera de « partition » toute représentation symbolique du morceau, et ferons abstraction du système particulier de notation utilisé : document papier, fichier MIDI, SASL (MPEG4), CSound ou autre.

L'orchestre

Au sens usuel, un orchestre désigne un groupe de musiciens accompagnés de leurs instruments, chacun de ces musiciens interprétant une partie musicale sur un instrument, acoustique, électrique ou électronique.

Le rôle de l'interprétation est parfois confié à une machine, appelée *séquenceur*, chargée de piloter un s*ynthétiseur* à l'aide de directives informatiques, *p.ex.* MIDI. Le séquenceur joue alors en quelque sorte le rôle de chef d'orchestre, et le synthétiseur le rôle combiné du musicien et de son instrument. Pour l'anecdote, il existe aussi des systèmes permettant de piloter un instrument acoustique à l'aide d'un séquenceur.

En réalité, on est amené à rencontrer à peu près toutes les combinaisons entre homme, instrument et machine possibles. C'est pourquoi on désignera par *orchestre,* l'ensemble des moyens mis à contribution dans la production du son, que ceux-ci soient « réels » (musiciens, instruments acoustiques et électriques) ou « virtuels » (synthétiseurs, dispositifs informatique), et quel que soit le mode de contrôle de ces moyens (gestuel, électronique ou informatique ...).

La description sous forme informatique d'un orchestre est rendue possible par des langages spécialisés comme CSound ou SAOL, sous-ensemble de MPEG-4. Ces langages

permettent de décrire les instruments de l'orchestre de manière formelle et univoque, et donc de spécifier exactement le processus de conversion de la partition en un signal audible.

1.1.2 La norme MIDI

MIDI (Musical Instruments Digital Interface) propose un format spécifiant de manière quantitative un certain nombre d'aspects de la chaîne de production du son. MIDI fournit un protocole standard de communication d'informations de nature musicale entre différents matériels compatibles. Un flux MIDI ne véhicule pas de signal audio proprement dit, mais une succession de notes, de paramètres de jeu, d'*étiquettes* spécifiant quel type instrument doit être utilisé.

Un contexte typique d'utilisation est celui où un séquenceur, relit une « partition » enregistrée sous forme de fichier informatique pour la transmettre en MIDI à un synthétiseur, qui restitue les sons correspondants.

Un flux MIDI est composé d'un certain nombre de messages qui sont envoyés au rythme d'une horloge, les uns à la suite des autres sur un câble d'interconnexion (protocole dit *sériel*). Chaque port MIDI peut véhiculer les informations correspondant à 16 canaux différents, chacun de ces canaux pouvant par exemple contrôler 16 instruments différents. Un message MIDI contient une en-tête contenant un numéro de canal ainsi qu'une action, à faire effectuer par le récepteur, parmi lesquelles les plus importantes sont détaillées ci-dessous :

NOTE
indique au récepteur de jouer une note (*p.ex.* Do4) spécifiée (message *Note On*) ou de mettre fin à une note (message *Note Off*), avec une intensité donnée (paramètre de *vélocité* dans la terminologie MIDI).

PROGRAM CHANGE
indique au récepteur (*i.e.* synthétiseur) de changer le n° du type instrument à la valeur spécifiée. Par exemple, passer de l'instrument 1 (Piano) à 5 (Guitare).

CONTRÔLEUR CONTINU
indique au récepteur de porter la valeur existante du paramètre spécifié à la valeur spécifiée. On peut ainsi contrôler le volume global de l'instrument, sa position dans l'espace stéréo (*Pan*), la fréquence de coupure d'un filtre (*Cutoff*) ou n'importe quel autre paramètre reconnu par le récepteur.

Un inconvénient majeur du MIDI est qu'il ne garantit en aucune manière que le résultat sonore soit identique d'un type de récepteur à un autre. En effet, tous les appareils n'implémentent pas les mêmes fonctions, les mêmes méthodes de synthèse, notamment pour des raisons de coût. Cette tolérance vis-à-vis du résultat sur le poste client est cependant contrebalancée par une large compatibilité et une certaine simplicité de mise en œuvre.

Le volet *Structured Audio* de la norme MPEG4 remédie à cela en incluant une couche additionnelle au flux de type MIDI, couche qui ajoute une description précise des instruments de l'orchestre. Les concepts principaux contenus dans la norme MPEG4-Structure Audio [II02] sont présentés brièvement ci-après.

1.1.3 MPEG4 - Structured Audio

Une description MPEG4-SA ou SA repose sur deux éléments centraux : le ***score*** (partition) et l'***orchestra*** (orchestre). Les instruments de l'*orchestra* contrôlés par le *score* génèrent des signaux audio, ces signaux pouvant être mélangé en sortie en stéréo, quadriphonie ou *surround* [1] selon la configuration de hauts-parleurs disponible sur le poste client.

Le point sur lequel SA diffère notablement de MIDI est donc que tous les aspects de la production du son sont spécifiés, via la description des instruments fournie par le *score*. Techniquement, la partie *orchestra* d'un flux SA intègre une description de tous les algorithmes de synthèse nécessaires, et seulement ceux nécessaires, à la reproduction du morceau.

Le *SCORE*, la partition dans le contexte de SA

Le *score* décrit l'ensemble des paramètres de jeu tels que hauteurs, instants de début et de fin des notes, paramètres de vibrato, ainsi que les effets éventuels. Il s'agit d'une représentation analogue au MIDI, à la différence qu'un *score* contient des informations spécifiquement liées au types d'instruments utilisés. Le *score* définit donc exactement un ensemble d'objets sonores, pourvu que les méthodes de synthèse correspondantes soient connues par ailleurs.

Dans SA, un « objet sonore » est constitué de l'ensemble des données suivantes :

- ▶ la dénomination de l'instrument qui doit jouer ou produire le son
- ▶ informations indépendantes du type d'instrument [2]
  - ▷ la note musicale du son joué : instant, durée ou longueur, hauteur ...

[1] Restitution sur au moins trois haut-parleurs simulant un champ sonore variable dans l'espace
[2] Informations type MIDI

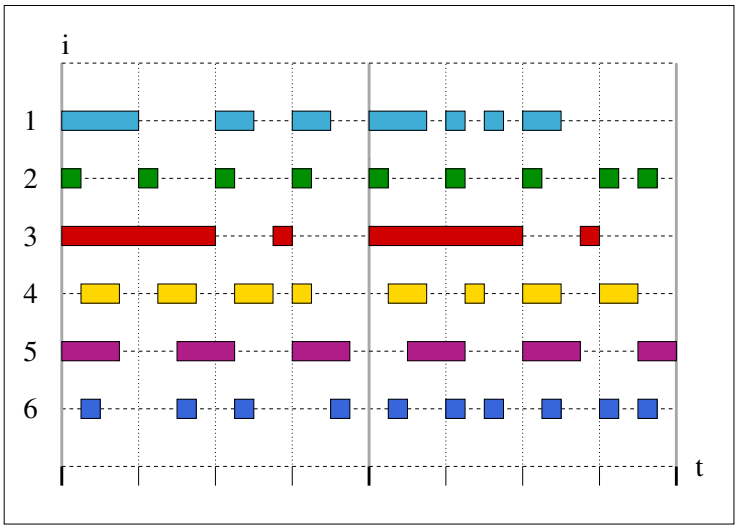


FIG. 1.2 – Représentation *temps-instrument* d'une partition
En ordonnées figure l'index k de l'instrument. Un bloc à la hauteur k indique que l'instrument k est actif entre les deux instants matérialisés par les bords du bloc.

- ▷ autres informations : volume de la note, présence de vibrato ou de trémolo
- ▶ informations propres à un instrument en particulier[3] : la façon de lier deux notes successives, etc.

Les séquenceurs informatiques utilisent généralement un schéma de représentation graphique simplifiée d'un *score* appelé *Piano Roll*, dont la figure 1.2 fournit un exemple. On parlera pour notre part de représentation TI pour *temps-instrument*, par analogie avec les représentations temps-fréquence. L'axe des ordonnées comporte l'index de l'instrument utilisé et/ou le numéro de note ; l'axe des abcisses fournit un repère temporel, souvent gradué en unités rythmiques et non en secondes. Un objet sonore est matérialisé par un bloc horizontal : les bords gauche et droite de ce dernier correspondent respectivement aux instants de début et de fin de l'objet.

La représentation TI peut également intégrer des données auxiliaires comme l'évolution dans le temps des paramètres de synthèse du son(intensité, fréquence de coupure d'un filtre, etc.), représentés sous forme de courbes temporelles. Cette représentation graphique de la partition est adaptée à tous les types d'instruments.

L'*ORCHESTRA*, ou l'orchestre dans le contexte de SA

L'*orchestra* ou *orchestre* spécifie des méthodes de production du son, ou synthèse sonores, qui sont dénommées *instruments* dans la terminologie MPEG4. Ces méthodes de

[3]Informations spécifiques Structured Audio

synthèse sonore sont décrites sans ambiguïté, à l'aide d'un langage spécialisé (SAOL), ce qui permet à un auteur de contenu MPEG4 de contrôler exactement le rendu sonore, sans se préoccuper des spécificités techniques du poste client. La syntaxe du langage utilisé est très proche de celle de CSound [Ver90]. Les algorithmes sont au final interprétés ou compilés à l'exécution, en instructions machines adaptées au processeur du client.

La standardisation du résultat sonore n'est pas le seul apport de SA par rapport au MIDI. Par exemple, pour restituer correctement toutes les nuances de jeu d'un violon solo, le créateur du contenu SA peut choisir d'utiliser un modèle de synthèse spécifique et de haute qualité. Le modèle sera plus complexe et donc plus exigeant, en termes de ressources de calcul, qu'un modèle généraliste comme la synthèse FM. Pour les sections de violons où la qualité de restitution de chacun des instruments isolé n'est pas primordiale, le créateur de contenu utilisera un modèle plus grossier ou même un modèle global pour l'ensemble de la section de violons.

SA autorise également la spécification d'un certain nombre de traitements audio ou *effets sonores*, appliquées au stade du mixage. De tels effets sonores ont notamment pour but de simuler les caractéristiques d'un lieu d'écoute à l'aide d'écho, de réverbération ou de procédés de spatialisation, d'améliorer la qualité du signal perçu par l'auditeur, de modifier le son d'un instrument dans un but esthétique (chorus, phaseur ...), etc.

En conclusion, on peut retenir que la norme MPEG4-SA est un mode de description exhaustive et structurée d'un signal audio, reposant sur des objets sonores obtenus par synthèse. On va maintenant exposer quelques notions de base et principes généraux de synthèse musicale. Bien que le modèle granulaire introduit aux chapitres suivants n'incorpore pas toutes ces connaissances, cet exposé pourrait servir de point de départ à un développement ultérieur du modèle.

1.2 Méthodes de synthèse musicale

Les synthétiseurs musicaux utilisent un ou plusieurs grands principes de génération du son, ou méthodes de synthèse. Les premiers synthétiseurs à avoir été construits sont dits « analogiques » car ceux-ci utilisaient des circuits à base de composants électroniques analogiques, dans lesquels les valeurs de tension et de courant sont continument variables. L'apparition du microprocesseur a permis de mettre en œuvre de nouvelles méthodes de synthèse, comme la synthèse à base d'échantillons, qui nécessite l'utilisation d'une mémoire.

On omettra de parler ici des instruments acoustiques, électro-acoustiques ou électromécaniques. Bien que ceux-ci puissent comporter certaines similitudes de principe, il ne s'agit pas à proprement parler de synthétiseurs.

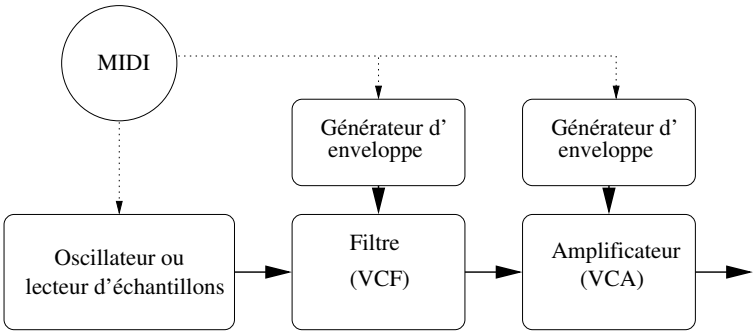


FIG. 1.3 – Schéma-type d'un synthétiseur musical

1.2.1 Concepts de base

Un synthétiseur comprend typiquement les éléments suivants (Fig.1.3) : un générateur de formes d'ondes, un filtre et un amplificateur. Chacun de ces éléments possède des entrées dites de *modulation* qui permettent de modifier les paramètres dans le temps, soit de façon interne au synthétiseur, soit externe, via MIDI par exemple.

Un certain nombre d'éléments de base reviennent fréquemment dans les schémas de fonctionnement des synthétiseurs, ce qui nous amène à présenter quelques « briques de construction » ci-dessous. Les termes et acronymes cités correspondent soit à des modules fonctionnels électroniques, soit à des portions de code d'un programme informatique, ou encore à des caractéristiques de fonctionnement.

Enveloppe - Courbe d'évolution de l'énergie moyenne du signal

Un *générateur d'enveloppe* produit un signal variant en général, lentement dans le temps, qui module un autre signal, la plupart du temps stationnaire. En modulant l'amplitude du signal, on peut contrôler les variations énergétiques du signal. Un son percussif sera obtenu en modulant avec un signal d'enveloppe présentant une croissance et une décroissance rapides, tandis qu'un son de type "nappe" (section de cordes ..) aura une enveloppe plus lente.

Il est courant de moduler un filtre avec un générateur d'enveloppe, ce qui permet d'agir sur l'évolution du timbre.

D'un point de vue mathématique, le générateur d'enveloppe est une fonction paramétrique. Le générateur de type ***ADSR*** **- A**ttack, **D**ecay, **S**ustain, **R**elease (Attaque, Décroissance, Tenue, Relâchement) est utilisé de façon quasi-universelle, car il permet de générer une grande variété de profils de courbes avec peu de paramètres. Les courbes obtenues permettent notamment d'imiter l'enveloppe énergétique des instruments existants.

LFO - Low Frequency Oscillator - Oscillateur Basse Fréquence

Oscillateur utilisé pour faire varier les paramètres de contrôle des filtres, des oscillateurs. Par exemple, en modulant le paramètre de hauteur d'un oscillateur avec un LFO, on peut simuler l'effet musical connu sous le nom de *vibrato*.

Polyphonique

Capacité d'un instrument à jouer plusieurs notes simultanément . Certains instruments acoustiques, les premiers synthétiseurs, la voix humaine (hors chant tibétain !) en sont techniquement incapables ; ces derniers sont alors qualifiés de *monophonique*s. La monophonie n'est pas obligatoirement la conséquence d'une limitation technique, cela peut être au contraire une caractéristique souhaitable, par exemple dans le cas d'un jeu en notes liées ou *legato*.

PWM - Pulse Width Modulation - Modulation de largeur d'impulsion

Un signal PWM est un signal périodique à valeurs binaires dont on module le *rapport cyclique*, c'est-à-dire le rapport entre les durées des deux états.

Ring modulator - Modulateur en anneau.

Le nom provient du circuit électronique analogique qui permettait, avant l'apparition des microprocesseurs, de réaliser la multiplication de deux signaux analogiques occupant la bande du spectre audible. Le signal résultant de cette opération est perceptivement très éloigné des deux signaux d'entrée. En effet, le spectre de ce signal est le convolué des spectres des signaux d'entrée, qui sont *a priori* quelconques et dé-corrélés.

Sampler - Échantillonneur

Instrument électronique permettant d'enregistrer, charger, rejouer et modifier en temps réel des échantillons sonores. Un sampler dispose également d'un générateur d'enveloppe et d'un VCF pour modifier les caractéristiques dynamiques du son, ce qui distingue le sampler d'un simple lecteur de séquences sonores pré-enregistrées.

Séquenceur

Dispositif permettant de stocker puis de restituer l'équivalent d'une partition et des indications de jeu sous forme informatique, qui seront alors interprétées par un synthétiseur.

SYNC*hronisation*

Dispositif permettant de réinitialiser la phase d'un oscillateur au rythme d'un second oscillateur calé sur une fréquence différente du premier.

VCF - V*oltage Controlled Filter* - Filtre commandé en tension.

Filtre, d'ordre généralement peu élevé (2,3 ou 4) dont la fréquence de coupure et le facteur de qualité peuvent être ajustés par un signal de commande. Le signal de commande provient d'un LFO ou d'un générateur d'enveloppe. Un VCF permet par exemple de simuler une déperdition d'énergie dépendante de la fréquence et du temps, caractéristique de la majorité des milieux acoustiques.

VCO - Voltage Controlled Oscillator - Oscillateur commandé en tension.

Circuit électronique générant un signal périodique, généralement une forme d'onde définie par une fonction mathématique simple : sinus, triangle, carré, dent de scie ... Un VCO dispose de paramètres contrôle tels que la fréquence d'oscillation ou la forme d'onde du signal généré.

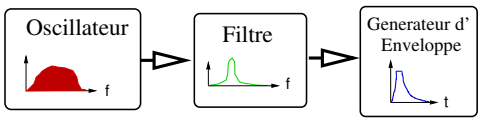
WaveTable - Table d'ondes.

Collection de formes d'ondes, ou signaux de courte durée sur quelques périodes, stockées dans une mémoire informatique.

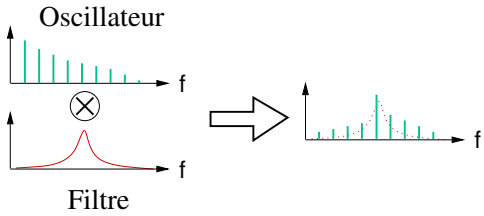
1.2.2 Synthèses soustractive, additive et dérivées

Synthèse soustractive

La synthèse soustractive [Roa96] utilise un oscillateur suivi d'un filtre puis d'un générateur d'enveloppe. Le terme « soustractive » vient de ce que l'on utilise un filtre pour *soustraire* de l'énergie à certaines fréquences du spectre de l'oscillateur. Le signal produit par l'oscillateur doit donc générer un spectre suffisamment étalé dans le domaine fréquentiel, tel qu'un spectre de raies en série harmonique. Les oscillateurs générant des signaux triangulaire, carré, PWM ou pseudo-aléatoire remplissent cette condition et sont en outre



(a) Principe de la synthèse soustractive



(b) Filtrage de l'oscillateur, du point de vue des DSP

FIG. 1.4 – Principe de la synthèse soustractive

faciles à implémenter, c'est pourquoi ces types sont donc les plus couramment employés pour la synthèse soustractive.

Sachant que les portions du spectre de l'oscillateur autour des zéros du filtre sont supprimées, et en jouant sur toutes les combinaisons possibles de type d'oscillateur et de paramètres de filtre, il est possible de générer une grande variété de profils de densités spectrales. D'autre part, en modulant les paramètres du filtre par un LFO et un générateur d'enveloppe, on obtient des variations temporelles du spectre et donc du timbre.

Synthèse additive, SMS ...

La synthèse additive procède par superposition de n formes d'ondes de base b_k,chacune étant modulée en amplitude par son propre signal d'enveloppe e_k.

$$s(t) = \sum_{k=1}^{n} e_k(t) \cdot b_k(t)$$

L'*addition* des composantes b_k se fait à la fois dans le domaine temporel et dans le domaine fréquentiel. L'idée est de parvenir à recréer des profils de DSP complexes à partir d'un grand nombre d'éléments b_k aux profils de DSP simples, tels que des sinusoïdes [MQ86] dont la DSP est un dirac à la fréquence correspondante.

Les fréquences fondamentales des signaux b_k sont la plupart du temps en rapport harmonique, la valeur de la fréquence de base étant déterminée par la fréquence correspondant à la note musicale désirée. Cette méthode de synthèse permet de générer des formes arbitraires de DSP dès lors que l'on utilise un nombre n suffisant d'oscillateurs. La contrepartie est inévitablement une certaine complexité de mise en œuvre, notamment au niveau de la

gestion des enveloppes.

Un type particulier de synthèse additive est connue sous le nom de Spectral Modelling Synthesis (SMS) [Sa97, Ser97]. Cette méthode a recours à un très grand nombre d'oscillateurs dont les variations temporelles d'amplitude e_k sont estimées de manière à re-synthétiser fidèlement le timbre de l'instrument modélisé. La synthèse SMS permet notamment de reproduire fidèlement le comportement non-stationnaire du signal pendant la phase transitoire de l'attaque.

1.2.3 Synthèse granulaire et ses variantes

Le son que l'on entend pendant une pluie battante est la combinaison des sons produits par chacune des innombrables gouttes d'eau lors de leur contact avec le sol. On pourrait imaginer de reproduire artificiellement le son de la pluie en additionnant une multitude de sons synthétiques générés par un modèle du son de gouttes d'eau. Il s'agit là d'un exemple illustrant les nombreuses possibilités offertes par le concept de synthèse granulaire [Roa78, Tru88]

De façon générale, la synthèse granulaire repose sur un modèle de synthèse de *fragments*, *corpuscules* ou encore *grains* sonores utilisé pour produire un grand nombre de ces sons élémentaires, avec des paramètres de synthèse différents à chaque fois. Les fragments de signaux obtenus sont ensuite additionnés entre eux pour donner un signal complet et complexe.

Il existe une grande variété de techniques granulaires, différant par la méthode de synthèse des grains atomes, par le mode de contrôle, déterministes ou stochastique, des paramètres de synthèse , ainsi que par l'ordre de grandeur du débit de grains par seconde.

Gabor [Gab46, Gab47] a proposé l'utilisation d'*atomes* sonores, qui portent son nom (essentiellement des sinusoïdes de durée limitée et de fréquence variable) pour représenter toute type de son. L'algorithme Matching Pursuit [MZ93] fournit de plus une méthode pratique pour décomposer un son quelconque en atomes de Gabor, avec une précision arbitraire.

La représentation obtenue est un exemple de modèle granulaire. Il n'existe toutefois pas à notre connaissance de méthode générique pour extraire d'un signal quelconque une représentation à base de grains sonores *de forme arbitraire*.

Synthèse FOF

L'ensemble des mécanismes physiques intervenant dans la production de la voix humaine est souvent modélisé par un système excitateur-résonateur, ou source-canal. A chaque configuration physique de l'appareil vocal sont associés un phonème et une configuration

de filtre correspondants. Les caractéristiques de l'excitation et de du résonateur varient à la fois en fonction du phonème prononcé, voyelle ou consonne, du locuteur, et du temps. La courbe de réponse de ces filtres exhibe un certain nombre de pics énergétiques caractéristiques, appelés *formant*s. L'excitation est périodique pour les sons voisés.

Pour synthétiser la parole, on peut injecter un signal d'excitation dans le résonateur, en travaillant échantillon par échantillon. Une telle approche présente cependant l'inconvénient de nécessiter une masse conséquente de calculs.

La méthode FOF [RPB85] développée par Xavier Rodet repose sur l'hypothèse que le filtre varie suffisamment lentement et donc que celui-ci peut être considéré comme constant à l'intérieur d'une fenêtre temporelle. L'excitation pour les sons voisés est quant à elle modélisée par un train d'impulsions.

La réponse impulsionnelle à une excitation dirac du filtre est appelée forme d'onde formantique (FOF), car celle-ci correspond à une configuration particulière de formants. Pour finir, les FOF sont tronquées à la longueur d'une période et concaténées au rythme de l'excitation périodique.

Les FOF sont un type particulier de *grains* sonores, que l'on combine selon des règles spécifiques, liées à la prosodie, pour synthétiser de la parole ou la voix chantée. De façon plus générale, on désigne par PSOLA [CS86, Pee98] l'ensemble des techniques permettant de combiner des grains de façon synchrone avec les périodes fondamentales du signal.

1.2.4 Autres méthodes

Synthèse FM

La synthèse FM, pour *Frequency Modulation* ou modulation de fréquence, utilise un nombre restreint, typiquement 6, d'éléments appelés *opérateurs,* interconnectés entre eux pour créer des timbres particulièrement complexes et évolutifs. Cette technique a été mise au point par John Chowning au CCRMA dans les années 70 [Cho73].

Un opérateur FM intègre un oscillateur sinusoïdal doté de deux entrées contrôlant la fréquence d'oscillation (Fig. 1.5a) : une entrée *pitch* déterminant la fréquence de base et une entrée *modulation,* permettant comme son nom l'indique de moduler la fréquence de base l'oscillateur via un autre signal audio. Un opérateur FM intègre en outre un générateur d'enveloppe type ADSR.

La sortie d'un opérateur m peut être reliée à l'entrée d'un autre opérateur c et ainsi moduler la fréquence de ce dernier. Dans ce cas, l'opérateur m est appelé le *modulateur* et le second, c, la *porteuse* (« carrier »). Les signaux en sortie des opérateurs m et c sont :

$$\begin{cases} m[t] & \propto & \sin[\omega_m \cdot t] \\ c[t] & \propto & \sin[\omega_c \cdot t + \alpha \cdot m[t]] \end{cases} \tag{1.1}$$

On relie plusieurs opérateurs en série ou en parallèle pour obtenir encore plus de variations timbrales (Fig. 1.5b), selon un certain nombre de configuration prédéterminées, appelées *algorithmes* (Fig. 1.5c).

Synthèse PD

Le synthèse PD, pour distortion de phase, [Dun80, Arf80] est assez similaire dans son principe à la synthèse FM. Ici la phase instantanée ωt d'un oscillateur sinusoïdal est « distordue » par une fonction f non-linéaire, *i.e.* :

$$s(t) = \sin(f(\omega \cdot t \mod 2\pi))$$

Si f est à valeurs dans $[0 \dots 2\pi]$, on peut vérifier que la fonction f ne modifie pas la période $\tau = 1/\omega$ du signal, mais seulement la forme du signal au cours d'une période. On obtient ainsi une grande variété de sons de hauteur constante mais de timbres différents en changeant la forme de f.

Synthèse à *Tables d'ondes* (WaveTable)

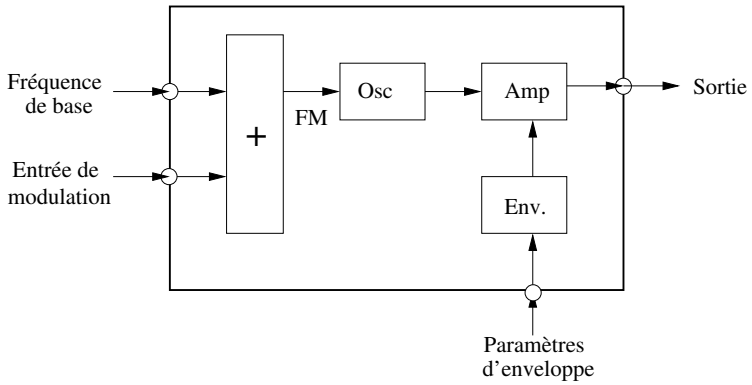
Le principe est de stocker des fragments d'enregistrements, d'instruments réels par exemple, dans une mémoire appelée pour l'occasion *table d'ondes*. Alternativement, la table d'ondes peut être constituée de descriptions de formes spectrales [NH02].

Pour un synthétiseur imitant le son d'un piano, on extrait une ou plusieurs périodes d'un signal à partir d'un enregistrement d'un Do4 de cet instrument et on stocke ce fragment n°1 dans la table d'onde. On procède de même pour les autres octaves Do5, Do6 etc., voire pour toutes les notes possibles pour cet instrument. Si l'on veut reproduire un Do5, on répète le fragment n°2 autant de fois que nécessaire, ce qui correspond à une opération de périodisation. Le moteur de synthèse intègre généralement un module de transposition, de dilatation temporelle, etc.

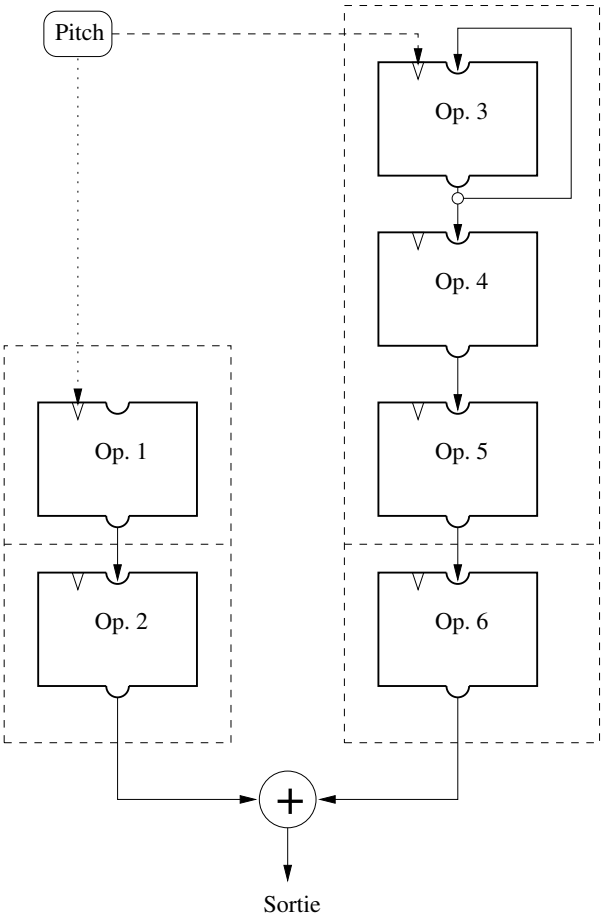
Pour plus de détails sur les différents raffinements et implémentations possibles, le lecteur est invité à consulter [Roa96].

Synthèse par modélisation

La synthèse par modélisation a pour objet la simulation informatique du comportement d'un instrument acoustique, électro-acoustique ou électronique. Cela suppose donc de par-



(a) Structure interne d'un opérateur FM



(b) Un exemple d'algorithme FM. Les opérateurs 2 et 6 sont ici connectés en porteuse, les op. 1, 3, 4 et 5 sont à la fois porteuse et modulateur. L'opérateur 3 est dans une configuration particulière de type *contre-réaction, i.e.* sa sortie module son entrée.

FIG. 1.5 – Synthèse FM

venir à analyser avec la finesse nécessaire l'ensemble des phénomènes physiques ou autres, qui régissent le fonctionnement de l'instrument, pour en dériver un modèle mathématique.

Les ressources de calcul étant limitées, il est presque toujours nécessaire de simplifier le modèle de manière sélective, dans la mesure où les approximations faites n'affectent que peu le résultat sonore.

La présence d'un grand nombre de constantes numériques qu'il faut fixer ou mesurer incite également à procéder à ces simplifications. En effet, l'interface présentée à l'utilisateur doit être suffisamment ergonomique pour permettre un jeu aisé. Si l'on rend trop de paramètres accessibles à l'utilisateur, l'utilisation en situation réelle risque de devenir problématique, tandis qu'un choix opposé conduira à limitent l'étendue des possibilités sonores.

Un synthétiseur par modélisation présente donc des capacités sonores qui ne sont pas limitées par des contraintes techniques habituelles, telle que la résistance des matériaux employés, les dimension de l'instrument. Quand tel ou tel instrument est très recherché pour son identité sonore unique, il souffre malheureusement bien souvent de problèmes de fiabilité dûs à son ancienneté, de reproductibilité d'un exemplaire à un autre, d'encombrement excessif , de coûts de fabrication et d'entretien dissuasifs, de fragilité, etc. Une modélisation précise et en temps-réel d'un tel instrument, pourrait alors servir à s'affranchir d'un certain nombre de ces contraintes.

Modélisation analogique

La *modélisation analogique virtuelle* s'attache à simuler le comportement des synthétiseurs analogiques. Après un examen du schéma électronique, on établit un modèle pour chaque composant ou bloc de composants, puis on cherche une manière d'implémenter ce modèle efficacement sur un microprocesseur. Il faut noter que la discrétisation d'un tel système à temps continu en préservant les propriétés essentielles de l'original pose des problèmes non-triviaux. Parmi ces derniers, mentionnons le passage d'un filtre analogique variable dans le temps à une version numérique, la préservation de la stabilité du système, la maîtrise des phénomènes de repliement de spectre.

Modélisation physique

La *modélisation acoustique virtuelle*, comme son nom l'indique, s'attache pour sa part à simuler le fonctionnement d'un instrument acoustique. Les modèles obtenus, des systèmes différentiels souvent non-linéaires, ne peuvent être résolus explicitement. Pour référence [SI92, Bil01], on emploie à cet effet des techniques de modélisation comme les guides d'ondes (propagation de l'excitation) ou les différences finies.

Autres méthodes

La synthèse dite *WaveShaping Synthesis* [Arf80] emploie une fonction non-linéaire paramétrée pour générer des formes d'ondes complexes à partir d'un signal de base simple comme une sinusoïde.

1.3 Conclusion

En supposant que l'on dispose d'un modèle *suffisamment réaliste* de chaque instrument et d'une partition spéciale contenant un *ensemble suffisant* de directives de jeu transmises aux instruments, il serait théoriquement possible de restituer un signal *suffisamment proche* de la performance originale.

Les principales difficultés résident dans l'interprétation que l'on doit faire des termes *suffisamment réaliste*, *ensemble suffisant,* etc. pour mettre cette idée en œuvre. Bien que la qualité de restitution typique du MPEG4-SA et *a fortiori* du MIDI, soit nettement inférieure à celle associée aux codeurs psychoacoustiques, ces formats de représentation symbolique offrent en contrepartie un certain nombre d'avantages significatifs.

En sus de conduire à des débits ou des tailles de fichiers largement inférieures, typiquement quelques kilo-octets pour MIDI contre plusieurs mega-octets pour mp3, une description du type SA ou MIDI se prête naturellement à l'extraction de descripteurs du contenu musical utilisables pour une recherche ultérieure dans une base de données par exemple, à des modifications de tempo, de hauteur, de mixage ... Les mêmes opérations sont également réalisables en travaillant directement à partir du fichier encodé ou sur le signal décodé, mais le plus souvent au prix de difficultés supplémentaires.

Malheureusement, il n'existe pas de procédé systématique et automatique permettant d'obtenir la *partition* et l'*orchestre* (au sens de SA) à partir d'un signal quelconque. Un certain nombre de techniques permettent d'extraire automatiquement une partition à partir d'un signal, mais celles-ci ne sont généralement applicables dans le cas d'un instrument isolé et lorsqu'on dispose de connaissances *a priori* sur l'instrument en question. Dans l'approche que nous avons choisi, on ne cherche pas explicitement à extraire de partition ni à obtenir de modèle de l'orchestre.

Chapitre 2

Analyse et synthèse de signaux musicaux

2.1 L'approche analyse-synthèse

2.1.1 Deux opérations duales

On a vu au chapitre précédent qu'un signal audio peut être vu comme la résultante d'un processus de combinaison d'objets sonores, eux-mêmes issus de différents processus de synthèse. On s'intéresse maintenant à l'opération inverse, à savoir une forme de modélisation du signal qui permettrait de retrouver l'ensemble de ces processus intervenant dans la chaîne de production..

Le processus « inverse » de la synthèse est dénommé *analyse*, sous réserve que cette opération soit réalisable. De façon schématique, l'analyse d'un signal permet d'accéder à un certain nombre de ses caractéristiques, tandis que la synthèse permet de construire un signal à partir de ces mêmes caractéristiques.

L'espace utilisé pour décrire les caractéristiques d'un signal est appelé *espace d'analyse* ou de *représentation.* L'ensemble des caractéristiques quantitatives extraites par l'analyse, ou *coefficients* détermine une *représentation* du signal.

Disposer d'une méthode unifiée, combinant analyse et synthèse, au sens ou l'on puisse passer de l'espace signal à l'espace d'analyse, dans l'une ou l'autre direction, présente cependant un certain nombre d'avantages. Il est ainsi envisageable d'appliquer des transformations, conversions, altérations ... en analysant le signal, modifiant la représentation obtenue, et en définitive re-synthétisant un autre signal, aux caractéristiques modifiées

Dans l'optique d'un codage efficace du signal, ou *compression*, on veut réduire la quantité de données nécessaires au stockage ou à la transmission du signal. Un moyen d'y

parvenir est de rechercher une représentation de ce signal qui puisse être décrite de façon *économique*, c'est-à-dire qui fasse intervenir un nombre restreint de paramètres ou de coefficients.

La section 2.2 présente un certain nombre d'autres applications incorporant ces trois phases d'*analyse-(modification)-synthèse*, parmi lesquelles la reconnaissance de partitions, la constitution de bases de données structurées (par rapport au contenu musical), l'assistance au diagnostic pour la lutherie ou la fabrication de haut-parleurs.

Dans beaucoup de cas, on se contente d'une représentation approchée du signal, c'est-à-dire que l'on ne requiert pas que le signal resynthétisé soit exactement identique au signal de départ, et ceci pour plusieurs raisons telles que :

1. notion de différence perçue
2. distinction entre *signal utile* et *bruit*
3. optimisation des capacités de stockage/bande passante
4. simplicité de mise en œuvre

Prendre en compte la notion de différence perçue La première raison est directement liée au contexte d'application : quand le signal est uniquement destiné à être reproduit par des haut-parleurs, le signal peut alors tout à fait subir une dégradation, à condition que celle-ci ne soit pas, ou peu perceptible pour l'auditeur. Le niveau de dégradation tolérable est ensuite déterminé par rapport à l'application spécifique considérée.

Coder uniquement le « signal utile » La deuxième raison est plus théorique. Dans le cas général d'un signal bruité, le signal se décompose en un signal utile et un bruit. Le signal utile est la partie contenant le « message musical », le bruit ayant pour sa part des origines diverses : bruit intrinsèque des microphones et du préamplificateur, bruit de quantification, etc. Deux réalisation différentes d'un bruit de statistiques équivalentes seront généralement perçues comme identiques par l'auditeur, dans le cas d'un bruit gaussien i.i.d par exemple.

On peut donc substituer le bruit présent dans le signal par une réalisation différente, générée à la reconstruction du signal à l'aide d'un générateur aléatoire de même statistiques que le bruit originel. On peut aussi tout simplement omettre de reconstruire ce bruit au moment de la décompression, sachant que le bruit de fond est perçu comme une dégradation sonore.

En définitive, si l'on ne supprime pas ce bruit, opération qualifiée de *débruitage*, on va chercher une représentation prenant en compte ce bruit, bien que seule la représentation de l'information utile nous intéresse. L'intérêt d'un débruitage préalable provient surtout

du fait que l'addition de bruit a généralement pour effet de dégrader les performances, en termes de taux de compression.

Notons cependant que le débruitage n'est pas une opération triviale, qui nécessite notamment de savoir estimer précisément les statistiques du bruit, sous peine d'engendrer des artefacts audibles plus gênants que le bruit le même. Les types de bruits rencontrés sont très divers, et tous ne peuvent être modélisés par des statistiques simples (*p.ex.* souffle produit par un ventilateur, bruit ambiant, vent ...).

Intérêt de la compression La troisième raison invoquée est d'ordre technique : les capacités physiques des appareils de transmission et, dans une moindre mesure, de stockage, sont limitées. En dépit des progrès constamment réalisés dans ces domaines, le coût de stockage ou de transmission d'un signal sera toujours plus ou moins proportionnel aux poids des données nécessaires à la représentation de ce signal. Les contraintes d'encombrement limitent également la capacité de stockage d'un appareil embarqué, la miniaturisation des composants ayant certaines limites.

Si l'on accepte une certaine erreur entre le signal original et le signal reconstruit, il est possible de diminuer de beaucoup le poids des données. Les algorithmes de codage psycho-acoustiques tel que mp3 permettent de diviser ce poids par un facteur 10 environ, au prix d'une dégradation considérée comme acceptable par la plupart des auditeurs.

Modéliser la structure du signal ?

Revenons à présent au problème de l'analyse de signaux sonores. On s'interroge sur les possibilités de remonter à une hypothétique structure « intrinsèque » (cf. Ch.1) du signal, *i.e.* remonter jusqu'à la partition et l'orchestre et donc d'obtenir un modèle de synthèse du signal. L'idée nous paraît aussi séduisante que difficile à mettre en pratique, pour un certain nombre de raisons brièvement énumérées ci-dessous :

- Superposition des objets

  On a accès uniquement à la somme des objets sonores, ce qui implique de savoir extraire chacun des objets du signal. Il s'agit là d'un problème dit de séparation de sources, problème difficile en soi et qui est l'objet d'une intense activité de recherche.

- Diversité des procédés de génération du son
  - Sélection de modèles concurrents
  - Élaboration d'un modèle pour chaque type d'instrument
  - Estimation des paramètres

En supposant que l'on ait identifié les objet sonores, il faut ensuite remonter aux paramètres utilisés à la synthèse. Ceci suppose d'une part de savoir sélectionner un modèle

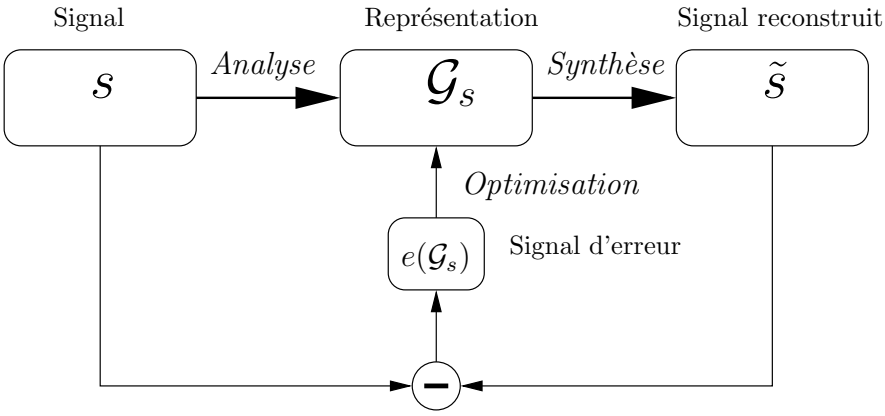


FIG. 2.1 – Analyse par la synthèse

de génération du son correspondant à l'instrument. D'autre part, il faut connaître ou à défaut, concevoir, un modèle du fonctionnement de chaque instrument rencontré, sachant que la conception d'un tel modèle est une tâche souvent longue et laborieuse.

De plus, l'estimation des paramètres d'un modèle nécessite dans beaucoup de cas une procédure spécifique au modèle. En outre, il existe des modèles, d'une utilité certaine (*p.ex.* synthèse physique, FM), qui ont la particularité d'être fortement non-linéaires, d'exhiber des comportements chaotiques ou de comporter un grand nombre de paramètres de contrôle. Il en résulte que les techniques d'estimation des paramètres pour de tels modèles sont à la fois compliquées et exigeantes en termes de ressources de calcul.

La modélisation du signal par une représentation type MIDI+modèle de synthèse des instruments est donc écartée.

2.1.2 L'analyse par la synthèse

Le principe de l'analyse par la synthèse est, en supposant que l'on ait postulé un modèle de synthèse, de construire un signal $\tilde{s}$ par synthèse à partir d'un modèle ou d'une représentation G_s du signal analysé s, en modifiant le modèle jusqu'à ce que la différence entre $\tilde{s}$ et s soit suffisamment petite. La figure 2.1 illustre ce principe. Il s'agit là d'un principe général très puissant car il permet de conduire une analyse du signal en l'absence de méthode directe de calcul du modèle. Pour le mettre en œuvre, on doit mettre en place une procédure d'optimisation progressive de la représentation, qui garantit si possible que l'erreur, ou l'erreur moyenne, diminue à chaque étape. La conception de la procédure d'optimisation dépend largement du choix du modèle de synthèse considéré. On présente rapidement ci-après deux exemples de schémas d'analyse par la synthèse : le Matching Pursuit et le codage CELP.

Alg. 1 Algorithme *Matching Pursuit*

Soit $B = \{b_k\}_{k=1\ldots K}$ une famille de vecteurs normés de dimension N. L'algorithme MP calcule une représentation approchée du signal s sur la famille B. A chaque étape m, l'algorithme fournit une approximation à m termes du signal s, notée $\widetilde{s}_m \triangleq \sum_{i=1}^{m} \alpha_m \cdot b_{k_m}$ et un *résidu* $r_m \triangleq s - \widetilde{s}_m$.
Étape 0 : Poser $\widetilde{s}_0 = 0$ et $r_0 = s$. Pour $m = 1 \ldots M$, itérer les opérations suivantes :
Étape m :

1. Chercher $k_m = \arg\max_k |\langle r_{m-1}, b_k \rangle|$.
2. Faire $\widetilde{s}_m = \widetilde{s}_{m-1} + \langle \widetilde{s}_m, b_{k_m} \rangle \cdot b_{k_m}$
   $r_m = r_{m-1} - \langle \widetilde{s}_m, b_{k_m} \rangle \cdot b_{k_m}$

2.1.3 Le Matching Pursuit

Les atomes de Gabor sont des sinusoïdes d'amplitude, de fréquence variables fenêtrées par une enveloppe gaussienne, de durée également variable. L'ensemble des atomes obtenus en faisant varier ces paramètres formant une famille sur-déterminée de l'espace signal, on ne peut pas directement projeter le signal sur cette famille de vecteurs. L'algorithme Matching Pursuit [MZ93, Gri99] est une méthode de calcul itératif par approximations succesives du signal par une combinaison d'atomes de Gabor. A chaque étape, le MP sélectionne l'atome conduisant à une diminution maximale de l'erreur de reconstruction. Le fonctionnement de l'algorithme est détaillé à l'encart 1.

On peut envisager d'étendre l'utilisation de cet algorithme à une famille de vecteurs autres qu'un dictionnaire de Gabor, en sachant que par construction, l'algorithme MP garantit une diminution de l'erreur à chaque itération

2.1.4 Codeurs CELP

Le codage par prédiction linéaire LPC [AH71, Spa94] repose sur une modélisation auto-régressive (AR) des trames du signal. A l'intérieur d'une trame s_n, le signal, supposé stationnaire, est décrit par le couple formé par l'excitation e_n - et le filtre AR A_n, *i.e.*

$$A_n \star s_n(t) = \sum_{p=0}^{P} \underbrace{a_{n,p}}_{\text{filtre AR}} \cdot s_n(t-p) = \underbrace{e_n(t)}_{\text{erreur}}$$

Le coefficient d'ordre p du filtre associé à la trame n est noté $a_{n,p}$, avec $a_{n,0} \triangleq 1$ (par convention), et le signal d'erreur e_n.

Les coefficients $a_{n,p}$ du filtre A_n sont estimés de manière à minimiser l'erreur quadratique $\|e_n\|^2$, par une méthode telle que celle proposée par Levinson et Durbin [RS78].

Le signal peut ensuite être reconstruit exactement à partir du filtre et de l'excitation par filtrage inverse, autrement dit $s_n(t) = A_n^{-1} \star e_n(t)$, mais pour les besoins du codage,

on substitue à l'excitation réelle une excitation synthétique, qui peut par exemple être un bruit ou un train d'impulsions.

Dans les codeurs type LPC et CELP, les coefficients $\mathbf{a}_n$ subissent une quantification vectorielle afin de diminuer la longueur de leur description. On doit prendre soin de transformer ces coefficients dans un espace approprié au préalable, afin d'éviter au maximum les éventuels effets indésirables provoqués par la quantification du filtre inverse A^{-1} , qui est à réponse impulsionnelle infinie. Ces coefficients quantifiés sont ici notés $\tilde{\mathbf{a}}_n$.

Pour le codeur CELP, on substitue également à l'erreur de prédiction e_n un signal $\tilde{e}_n$ choisi parmi un ensemble de signaux appelé *codebook*, ou dictionnaire, connu à la fois du codeur et du décodeur. Le signal approché $\tilde{s}$ est resynthétisé par filtrage inverse de l'élément du codebook par le filtre $\tilde{A}^{-1}$:

$$\tilde{s}_n(t) = -\sum_{p=1}^{P} \tilde{a}_{n,p} \cdot \tilde{s}_n(t-p) + \tilde{e}_n(t)$$

La recherche de l'élément optimal du codebook s'effectue suivant une procédure d'*analyse par la synthèse,* c'est-à-dire qu'on compare les valeurs des énergies des signaux d'erreur obtenues avec les différents éléments du codebook, pour ensuite sélectionner celui conduisant à la valeur minimale. La mesure de l'erreur utilisée prend en compte les variations de la sensibilité de l'oreille selon la fréquence, afin de minimiser la distortion perçue par l'auditeur.

2.2 Applications de l'analyse-synthèse

2.2.1 Le Vocodeur

Le vocodeur, de l'anglais *voice-coder* [Dud22], a été conçu au départ pour coder les signaux de parole dans le cadre des télécommunications . Le principe général simplifié en est le suivant :

1. Émetteur : le signal entrant est soumis à une analyse par un banc de filtres de type passe-bande ainsi qu'un détecteur voisé/non-voisé
2. Les valeurs des énergies mesurées en sortie du banc de filtre et l'état du détecteur de voisement sont transmis sur le canal.
3. Les données contenues dans le flux entrant alimentent un synthétiseur de parole qui reproduit une version approchée du signal original.

Les variantes autour de ce schéma de base sont nombreuses : le *vocodeur de phase* [Puc95] implémente le filtre sous la forme d'une transformée de Fourier discrète, le vocodeur à

prédiction linéaire LPC [AH71, Fla72] utilise une paramétrisation des coefficients de filtres auto-régressifs, le vocodeur à formants, etc.

En dehors des télécommunications, le vocodeur est notamment utilisé comme effet musical, pour imprimer la structure formantique d'un signal dit *modulateur* à un autre dit *porteuse*. En utilisant par exemple, une guitare électrique comme porteuse et de la parole ou du chant pour le modulateur, on obtient un effet de « guitare parlante ».

Dilatation temporelle

Le principe du vocodeur est également utilisé pour modifier la durée d'un signal, autrement dit le rythme de défilement des objets sonores. Cette opération aussi dénommée *time-stretching,* se doit de préserver le plus possible les autres caractéristiques propres au signal telles que hauteur, mais également le timbre des sons à structure de pitch, la perception des transitoires, etc.

Cette application trouve son utilité dans des domaines tels que le son à l'image, la création musicale à partir d'échantillons, et de façon générale dès qu'on a besoin de caler un signal par rapport à une référence temporelle précise et imposée.

Avec un signal généré par un séquenceur MIDI, il est très aisé de modifier le tempo, mais l'opération devient plus compliquée dès lors qu'on n'a accès qu'au signal lui-même, ce qui est généralement le cas. Un certain nombre de techniques de time-stretching [RPB85, Pee98, SSI90] adoptent un schéma analyse-modification-synthèse. La phase d'analyse fournit une représentation du signal où ses caratéristiques temporelles, rythmiques sont « découplées » des caractéristiques timbrales. On peut dès lors modifier les caractéristiques temporelles du signal *via* sa représentation sans altérer les autres, pour enfin re-synthétiser un signal à partir de la représentation modifiée.

Transposition

Aussi appelée pitch-shifting, la transposition a pour but de changer la hauteur globale d'un signal en conservant intactes sa structure formantique et la perception des transitoires. L'intérêt d'une telle opération est de pouvoir adapter la hauteur d'un accompagnement préenregistré à une performance musicale, de créer des harmonies artificielles ou de corriger la hauteur d'un instrument désaccordé. Couplée à un procédé de dilatation temporelle, la transposition permet de travailler la structure temps-fréquence d'un morceau enregistré très librement.

2.2.2 PSOLA

Les techniques PSOLA (***P**itch **S**yncronous **O**verlap-**A**dd*, [CS86]) et dérivées, ont été développées à l'origine pour la synthèse de parole. Le principe est de concaténer des formes d'ondes élémentaires, chacune correspondant à un phonème ou un diphone du texte à énoncer. PSOLA est également utilisée pour la transposition/dilatation de signaux sonores [RV93, Pee98, PR99], ce que nous expliquons ici brièvement.

Phase d'analyse

Le signal est découpé en trames synchronisées par rapport à la période fondamentale locale du signal. Ces trames s_m de durée variable sont obtenues par fenêtrage du signal à un certains nombre d'instants t_m particuliers appelés *marqueurs de lecture* :

$$s_m(t) = s(t) \cdot W_m(t - t_m)$$

Phase de re-synthèse

On resynthétise un signal $\tilde{s}$ en additionnant les trames $s_m(t)$ décalées d'une certaine durée Δ_m, choisie en fonction du facteur de dilatation temporelle désiré :

$$\tilde{s}(t) = \sum_m s_m(t - \Delta_m)$$

La figure 2.2 illustre le fonctionnement de PSOLA à partir d'un exemple.

Il existe plusieurs méthodes pour déterminer le placement des marqueurs de lecture et le choix des valeurs Δ_m , que nous ne détaillerons pas ici. Le fait que la forme d'onde et les relations de phase locales soient préservées est un des avantages notables de la méthode PSOLA.

2.2.3 Codage et Compression

Un certain nombre de procédés de compression du signal, généralement avec pertes, s'articulent sur les trois étapes d'analyse-modification-synthèse. Les codeurs en transformée, dont l'exemple le plus connu est probablement le MPEG 1 Layer 3 [ISO93], souvent abrégé en *mp3*, adoptent selon ce procédé général.

Le système auditif humain est sujet à des effets dits de *masquage fréquentiel*, qui apparaissent quand un signal est la superposition d'un certain nombre composante sinusoïdales de fréquences voisines. L'oreille humaine présente alors une sensibilité aux variations énergétiques maximale pour la composante dominante.

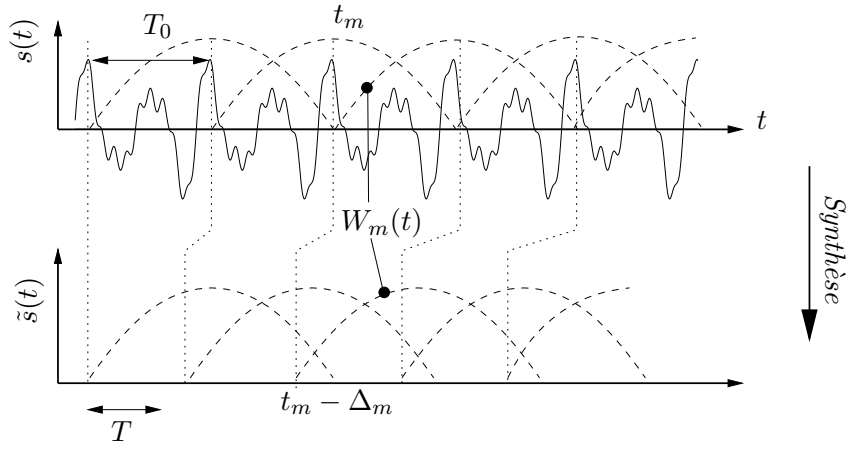


FIG. 2.2 – Correspondance entre marqueurs d'entrée et de sortie (d'après [Pee98]). T_0 est la période fondamentale (locale) du signal analysé, T celle souhaitée pour le signal resynthétisé.

Les codeurs perceptifs [PS97], dont le mp3 exploitent cette propriété en codant l'énergie de chaque composante avec une précision variable, déterminée par rapport à un modèle de l'effet de masquage, aussi appelé *modèle psycho-acoustique*. Ce modèle psycho-acoustique permet de déterminer la précision minimale sur les coefficients énergétiques dans une bande de fréquence donnée, à partir de laquelle l' »individu lambda » entendra une différence perceptible.

Le principe schématique d'un encodeur mp3 est illustré sur la figure 2.3. Le signal est décomposé en sous-bandes à l'aide d'un banc de filtres polyphase et d'une variante de transformée discrète en cosinus (MDCT). A l'intérieur d'une sous-bande, les coefficients de la DCT sont représentés sous une forme mantisse-exposant.

Plusieurs algorithmes itératifs se chargent alors de déterminer la précision requise pour quantifier ces coefficients, en tenant compte à la fois du modèle psycho-acoustique et du débit en bits disponible. En dernier lieu, un codage entropique vient éliminer les redondances encore présentes dans les données, suivi d'un ajout de données de contrôle et éventuellement d'informations décrivant le contenu du fichier.

Bien que les performances des codeurs existants soient déjà très satisfaisantes, la recherche dans le domaine de la compression audio continue. Celle-ci porte par exemple sur l'adaption du flux au capacités de la ligne de transmission et aux ressources matérielles disponibles côté client, à l'augmentation du taux de compression, ou l'étude d'approches nouvelles, par exemple en incorporant la théorie des fractales [WV97, BBP97].

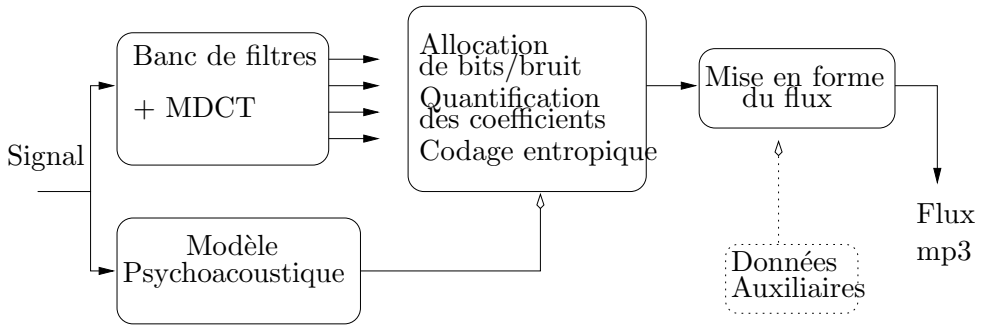


FIG. 2.3 – Principe général d'un encodeur mp3

2.2.4 Séparation de sources

La séparation de sources est l'opération qui consiste à séparer un signal dit de *mélange* en plusieurs signaux correspondant à autant de *sources* individuelles. Un cas typique est celui ou le mélange est un enregistrement mono ou stéréo d'un orchestre, les sources recherchés sont alors les signaux que l'on aurait obtenu en enregistrant isolément chaque instrument de l'orchestre. Le débruitage, qui consiste à supprimer le bruit parasite dans un enregistrement, peut être vu comme un cas particulier de séparation de sources.

La séparation de sources dite *aveugle* suppose que l'on ne dispose pas d'informations *a priori* sur les sources à part le mélange lui-même.

Une approche possible du problème est de rechercher une représentation du signal de mélange dans laquelle les sources sont naturellement bien séparées. Par exemple, lorsque on peut faire l'hypothèse que les sources évoluent dans deux parties disjointes du plan temps fréquence, l'utilisation d'une transformée du type Fourier à court-terme (TFCT) ou ondelettes facilite grandement la séparation des sources, en supposant que l'on ait accès à une certaine information sur le spectre de chaque source ...

Parmi les techniques de sépération existantes, citons le filtrage de Wiener (filtrage du spectre du mélange) et ses variantes [Ben03, BMDBG03], la projection sur des bases représentatives des sources [Jan02], la décomposition du signal en un mélange de processus AR [BJRwn] , *etc.*

2.2.5 Traitements sonores

Les traitements sonores, aussi appelés « effets », sont de plus en plus utilisés dans la production musicale. Il s'agit de modifier certaines caractéristiques du son, soit dans un but purement *esthétique*, par exemple pour étendre les possibilités sonores d'un instrument, soit pour conformer le son à certaines exigences techniques, etc. A l'ère de l'électronique analogique, la plupart des traitements s'opéraient sur le signal temporel, mais aujour-

d'hui les techniques numériques permet de modifier directement une représentation du signal puis de resynthétiser le signal traité. L'opération de *transposition*, c'est-à-dire le changement de la hauteur du son, s'exprime par exemple assez simplement dans l'espace temps fréquence, bien que celle-ci soit également réalisable directement dans le domaine temporel. De manière plus générale, pour un certain nombre de traitements, le fait de travailler sur une représentation du signal facilite souvent la modification de certaines de ses caractéristiques tout en laissant les autres inchangées.

2.3 Conclusion

Pour beaucoup d'applications audio, on a vu qu'il est nécessaire de pouvoir analyser un signal afin d'en dériver les paramètres d'un modèle, puis d'être capable de reconstruire un signal de synthèse « équivalent » au signal de départ, éventuellement en ayant modifié certains des paramètres du modèle.

En s'appuyant sur le paradigme de l'analyse par la synthèse, nous avons cherché à concevoir une méthode d'analyse à partir d'un modèle de synthèse sonore librement inspiré de la synthèse granulaire. La suite est consacrée à la présentation et à l'étude de ce modèle, qui constitue la matière principale de ce travail de thèse.

Deuxième partie

Modélisation granulaire du signal sonore

Cette deuxième partie, consacrée au modèle granulaire, contient les principales contributions de ce travail.

Le premier chapitre introduit le modèle granulaire, qui s'inspire de l'idée émise par Gabor, que les signaux peuvent être décomposés sous forme d'une somme d'*atomes* sonores d'une part, et du principe de la *synthèse granulaire* [1] *[Roa78, Roa96]* d'autre part. On discute ensuite du problème de l'analyse d'un signal suivant ce modèle granulaire et des différentes approches envisagées.

Le second chapitre présente une méthode de calcul du modèle granulaire par classification non-supervisée ou *clustering*. On rappelle le fonctionnement d'une variante des K-Moyennes, aussi appelé *nuées dynamiques*, parmi d'autres algorithmes classiques, en indiquant leurs avantages et inconvénients potentiels dans le présent contexte. On propose également les algorithmes de clustering conçus dans le cadre de ce travail.

Le dernier chapitre propose une étude successive de plusieurs types fonctions de synthèse ; ainsi que des perspectives pour des développements ultérieurs du modèle.

[1] Toute connection éventuelle avec le *Granular Computing* [Yao00], une branche récente de l'informatique théorique, serait purement fortuite.

Chapitre 3

Introduction du modèle

3.1 Présentation du modèle

3.1.1 Le signal sonore : une superposition d'objets élémentaires

L'objectif est de décomposer le signal en un certain nombre de sous-signaux « élémentaires » , que l'on qualifie d*'objets sonores,* ou plus simplement d'objets. L'idée de décomposer un tout en plusieurs parties plus simples, ou même élémentaires, est très présente en mathématiques (p.ex : décomposition sur une base vectorielle), en physique (théorie moléculaire, atomique, etc. de la matière) et également en traitement du signal, où on utilise couramment la décomposition du signal sur une famille de sinusoïdes (analyse de Fourier), d'ondelettes [Mal98] ou encore d'atomes [Gab47, GV97] de Gabor, d'un processus stochastique par rapport à un ensemble de gaussiennes ...

Au sens où on l'entend, le terme d'objet recouvre une grande variété de définitions possibles, selon que les objets sont définis a priori ou a posteriori de la connaissance du signal, selon que ceux-ci sont définis par rapport à une expression mathématique, un algorithme ...

Rien n'interdit *a priori* de rechercher des objets de forme quelconque, si ce n'est que le calcul de la décomposition est généralement facilité quand les objets sont connus à l'avance.

Quand les objets sont connus à l'avance et font partie d'une famille de vecteurs qui forme une base orthonormée de l'espace signal, on peut calculer directement et facilement la décomposition par projection du signal sur chacun des éléments de la base. Si de plus la base est définie par le biais d'une famille de fonctions, une partie des calculs est effectuée de manière symbolique.

Le cas échéant, ces calculs doivent être effectués numériquement, et on est amené à prendre en considération des questions supplémentaires telles que la précision et le temps de calcul. Ainsi, on peut concevoir d'utiliser des objets issus du « monde réel », par exemple

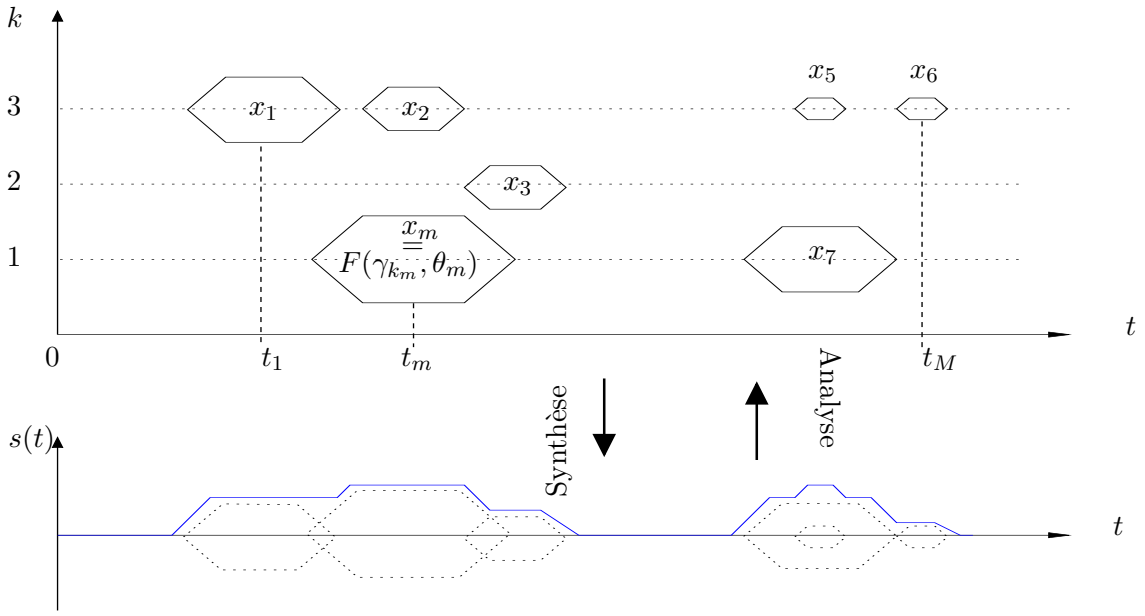


FIG. 3.1 – Une décomposition additive du signal sous forme d'objets

L'opération de *synthèse* consiste à additionner les signaux des M objets x_m, localisés autour des instants t_m , on obtient le signal s. L'opération « inverse » ou *analyse*, qui consiste à décomposer le signal en plusieurs objets est plus ardue, car on ne dispose pas d'une connaissance exhaustive des objets, telle que leur forme exacte, leur position dans le temps, leur nombre, etc.
Les objets sont classés verticalement selon leur forme globale, indexée par k. Dans le cas d'une représentation temps-fréquence, les objets sont des fragments de sinusoïdes, et à l'index k on peut faire faire correspondre la fréquence de la sinusoïde. Pour le modèle granulaire, k est l'index du prototype de l'objet.

une base de données de signaux issus d'enregistrements de différents instruments, de chanteurs, etc.

Enfin, on peut décider de ne pas spécifier *a priori* la forme des objets de façon explicite. Dans ce cas, il faut un moyen de calculer ceux-ci à partir du signal, après avoir introduit des critères qui font office de contraintes. Des techniques telles que l'ACI[1] ou l'adaptation/apprentissage de bases sont des exemples de ce type d'approche [Goo97].

C'est la direction générale que nous avons choisi, l'idée étant d'utiliser des objets *localement représentatifs* du signal.

La figure 3.1 montre une représentation générique de la correspondance entre un signal et une décomposition additive sous forme d'une combinaison objets.

3.1.2 Principe général d'un modèle granulaire

Plutôt que de spécifier explicitement et *a priori* la forme des objets, on suppose (implicitement) que plusieurs versions d'un même objet sont présentes à différents endroits du signal, moyennant certaines variations d'amplitude ou de phase, par exemple. Cette supposition semble se justifier au moins dans le cas des signaux musicaux. En effet, comme on l'a expliqué au chapitre 1, on retrouve généralement plusieurs instances d'une même note de musique à différents instants d'un morceau, et on s'attend à ce que ceci se traduise au niveau du signal par une forme de redondance temporelle (non-uniformément répartie).

Dans ce cas, plusieurs occurrences d'une même note de musique dans le signal ou objets sonores, devraient pouvoir être représentées par un même modèle. On verra plus loin que le modèle propose, pour des objets de même forme, d'utiliser un élément commun qualifié de *prototype* des objets.

Pour extraire les objets du signal, on parcourt le signal dans le temps pour rechercher des segments similaires et mesurer la ressemblance entre couples de segments. Si la ressemblance est élevée, on en déduit que ces deux segments contiennent des objets similaires. On peut ensuite représenter un ensemble d'objets suffisamment similaires à l'aide d'un objet de référence pour ce groupe, que l'on appellera prototype de l'ensemble. En faisant de même pour l'ensemble des objets, on constitue un ensemble de prototypes qui permet de décrire intégralement l'ensemble des objets constitutifs du signal, et donc le signal lui-même. Bien sûr, ceci n'est qu'un aperçu simplifié de la méthode de calcul.

Dans sa philosophie, le modèle granulaire peut être rapproché de la modélisation non-paramétrique de données et de l'extraction automatique de règles ou de structures, qui sont utilisées pour la reconnaissance de motifs[Elm90, HH02, HRH99], par exemple dans

[1] L'ACI, pour Analyse en Composantes Indépendantes, a pour but de séparer un signal en plusieurs signaux qui soient statistiquement les plus indépendants possibles.

les structures ADN ou les séries financières, en segmentation [Foo00, FC03], indexation et base de données [RKIHSK95, CW99, Yan02], « data mining » [DGM97, DLM+98], *etc.*

Nous allons maintenant présenter le formalisme qui sera utilisé dans toute la suite.

3.1.3 Le modèle granulaire en détail

Le signal s est vu ici comme résultant de la combinaison linéaire de M sous-signaux x_m appelés **objets sonores**, localisés temporellement autour d'instants t_m. Ces objets peuvent selon le contexte, être interprétés comme autant de notes ou fragments de notes, de sons, ou des diphones dans le cas ou l'on a affaire à un signal de parole. Cependant, il n'est absolument pas requis que les objets aient une quelconque signification « musicale » particulière. En résumé, le rôle d'un objet x_m est de décrire une partie du signal, localisée autour de l'instant t_m . Dans le cas non-bruité, ceci s'écrit :

$$s(t) = \sum_{m=1}^{M} x_m (t - t_m) \tag{3.1}$$

D'autre part, on se donne une fonction F qui décrit une méthode de **synthèse** pour l'objet sonore x :

$$\begin{array}{rcl} \mathbb{R}^\delta \times \mathbb{R}^q & \rightarrow & \mathbb{R}^d \\ (\gamma, \theta) & \longmapsto & x = F(\gamma, \theta) \end{array}$$

La fonction F prend deux variables vectorielles en argument : une variable contenant la donnée γ, appelée ***prototype***, et une autre θ, appelée ***paramètre***. F prend ses valeurs sur $\mathbb{R}^d$, l'ensemble des signaux réels à temps discret de durée d, dénommé espace des objets sonores dans le contexte présent.

En faisant varier le paramètre θ avec un prototype fixé γ, F permet ainsi de générer différentes « versions » d'objets sonores, qui ont toutes pour origine le même prototype γ, modulo une « déformation » paramétrée par θ.

Le *prototype* est ainsi nommé parce qu'il est la base qui sert à construire différentes versions d'objets, θ pouvant être vu comme un paramètre de contingence. Le but affiché est de parvenir à reconstruire autant d'objets que possible à partir d'un seul prototype.

Cas non bruité

Pour un objet donné x, on appelle ***grain*** un prototype γ vérifiant

$$\exists \theta_0 \in \mathbb{R}^q,\ x = F(\gamma, \theta_0)$$

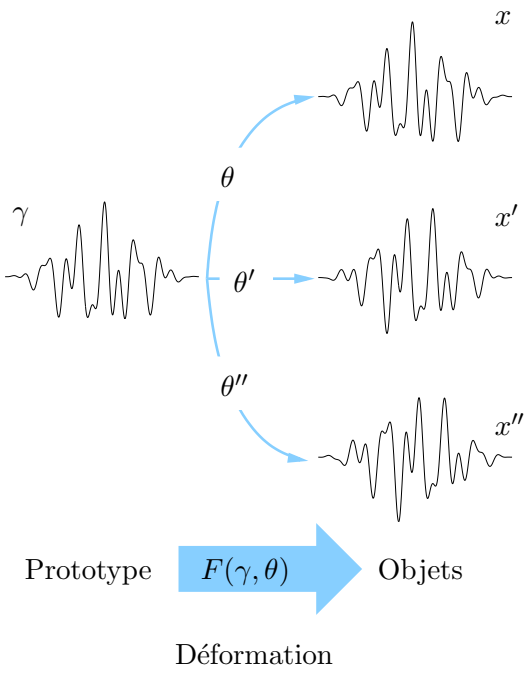


FIG. 3.2 – Des objets obtenus par déformation d'un prototype

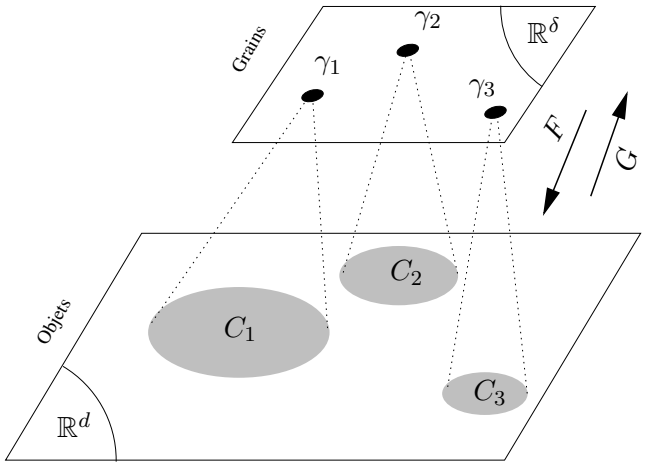


FIG. 3.3 – Principe schématique du modèle granulaire
Le nuage de points C_k représente l'ensemble des objets que l'on peut engendrer à partir du prototype correspondant γ_k, en lui appliquant la fonction F et en faisant varier θ sur l'ensemble des valeurs possibles.

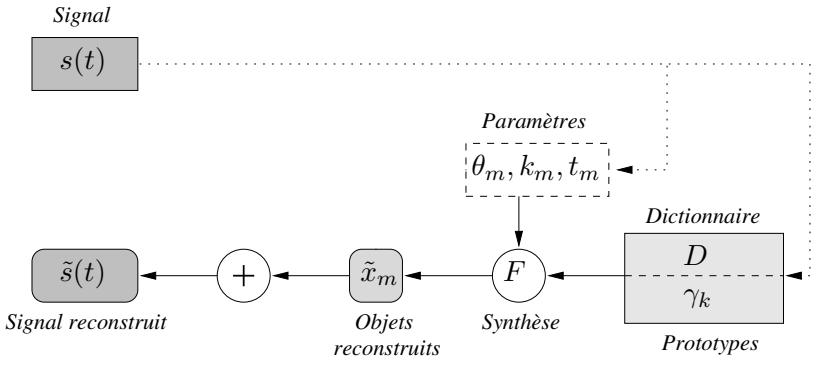


FIG. 3.4 – Reconstruction du signal par sommation de M objets $\tilde{x}_m$, versions déformées d'un prototype γ_{k_m}.

Dans ce cas, le grain est noté $g(x)$ ou plus simplement g quand aucune ambiguïté n'est à craindre ($g = \gamma$ dans l'équation ci-dessus).

On note G une fonction qui à un objet x associe un grain g, sous réserve que cette fonction existe et soit définie :

$$\begin{aligned} G: \quad \mathbb{R}^d &\to \mathbb{R}^\delta \\ x &\longmapsto g = G(x) \text{ t.q. } \exists \theta_0 \in \mathbb{R}^q,\ x = F(g, \theta_0) \end{aligned}$$

Dans la suite, on choisit la fonction F de manière à ce que G existe et soit définie $\forall x \in \mathbb{R}^d$.

Cas bruité

On conçoit aisément que le spectre des variations rencontrées au sein d'un ensemble d'objets réels ne pourra pas être intégralement modélisé par la fonction de déformation F. C'est pourquoi on introduit un terme d'erreur ou de bruit, additif, noté e, l'existence de ce terme permettant de tolérer une certaine imprécision dans la modélisation de l'objet.

Dans le cas général, l'objet x est donc une version bruitée d'une déformation d'un prototype γ :

$$e = e(x, \gamma, \theta) \triangleq x - F(\gamma, \theta) \tag{3.2}$$

e représente alors l'erreur de modélisation de l'objet x par le modèle F avec le prototype γ et le paramètre θ.

3.1.4 Définitions et Notations

Signal

$s(t)$	signal d'analyse
s_n	trame d'indice n
N	nombre de trames

Modèle granulaire

x_m	objet sonore d'indice m ($x_m \in \mathbb{R}^d$) [2]
γ	prototype d'objet sonore ($\gamma \in \mathbb{R}^\delta$)
g_m	grain d'indice m ($g_m \in \mathbb{R}^\delta$)
$F(\cdot,\cdot)$	fonction de synthèse [3]

$$\begin{array}{rcl} \mathbb{R}^\delta \times \mathbb{R}^q & \rightarrow & \mathbb{R}^d \\ (\gamma, \theta) & \longmapsto & \tilde{x} = F(\gamma, \theta) \end{array}$$

$\tilde{x}$	objet reconstruit à partir d'un prototype γ quelconque ($\gamma \neq G(x)$)
M	nombre d'objets sonores, nombre de grains
d	durée d'un objet sonore, en échantillons
δ	taille d'un grain
q	nombre de paramètres de la méthode de synthèse

L'index k_m désigne le prototype utilisé pour générer l'objet $\widetilde{x}_m$, tandis que m_k désigne l'objet utilisé pour obtenir le prototype γ_k.

Relations importantes

- Signal de référence - Trames

$$s(t) = \sum_{n=1}^{N} s_n(t - t_n) \tag{3.3}$$

[2] On utilise indifféremment les notation x_n et $x[n]$.
Dans ce document, seul le cas $x_m = s_m$ et $M = N$ est effectivement considéré.

[3] Le symbole tilde permet de différencier un objet $\tilde{x}$ obtenu par synthèse d'un objet préexistant ou « réel », noté $x \in \mathbb{R}^d$.
F n'est pas nécessairement une fonction bijective.

▶ Signal reconstruit - Objets reconstruits

$$\tilde{s}(t) \triangleq \sum_{m=1}^{M} \tilde{x}_m (t - t_m) \tag{3.4}$$

▶ Grain - objet

$$\begin{aligned} \exists \theta_0 \in \mathbb{R}^q,\ x_m &= F(g_m, \theta_0) \\ g_m &= G(x_m) \end{aligned} \tag{3.5}$$

3.1.5 Interprétation probabiliste

Étant donné un objet x existant, un élément quelconque $\gamma \in \mathbb{R}^\delta$ et une valeur donnée de paramètre θ, on suppose que l'erreur de modélisation $e(x, \gamma, \theta)$ est un bruit i.i.d. suivant une loi normale de variance inconnue $\sigma(\theta)$:

$$x = F(\gamma, \theta) + e(x, \gamma, \theta), e \sim \mathcal{N}_{\sigma(\theta)}$$

On peut alors chercher le paramètre θ le plus vraisemblablement probable, connaissant γ et x. L'estimateur de θ au sens du maximum de vraisemblance s'écrit :

$$\begin{aligned} \widehat{\theta}_{ML} &= \arg\max_\theta p(\theta / x, \gamma) \\ &= \arg\min_\theta \|x - F(\gamma, \theta)\|_2 \\ &= \arg\max_\theta \frac{\langle x, F(\gamma, \theta) \rangle}{\|x\|_2 \cdot \|F(\gamma, \theta)\|_2} \end{aligned}$$

Le bruit suivant une loi normale, l'estimateur au sens du maximum de vraisemblance est équivalent à l'estimateur au sens des moindres carrés. Dans une évolution du modèle, on pourrait introduire un *a priori* non gaussien sur l'erreur, par exemple pour prendre en compte des éléments perceptifs, auquel cas cette équivalence ne serait plus valide.

Remarques

La quantité $\frac{\langle x, F(\gamma,\theta) \rangle}{\|x\|_2 \cdot \|F(\gamma,\theta)\|_2}$ est toujours comprise entre -1 et 1.

Touts les modèles utilisés (cf. Ch.5) incluent un paramètre scalaire d'amplitude α dans le jeu de paramètres θ, tel que F soit linéaire par rapport à α :

$$F(\gamma, [\alpha, \theta_2, \ldots, \theta_q]) = \alpha \cdot F(\gamma, [1, \theta_2, \ldots, \theta_q])$$

On pose $\alpha \triangleq \theta(1)$ par convention.

On montre que la quantité

$$\max_{\theta} \frac{|\langle x, F(\gamma,\theta)\rangle|}{\|x\|_2 \cdot \|F(\gamma,\theta)\|_2}$$

est toujours comprise dans l'intervalle $[0,1]$.

La norme de l'erreur est :

$$\begin{aligned} \|e[x,\gamma,\theta]\|_2 &= \|x - F(\gamma,\theta)\|_2 \\ &= \|x\|_2 \cdot \sqrt{1 - \frac{\langle x, F(\gamma,\theta)\rangle^2}{\|x\|^2 \|F(\gamma,\theta)\|^2}} \end{aligned}$$

3.1.6 Mesure de similarité

La quantité

$$\Gamma(x,\gamma) \triangleq \max_{\theta} \frac{|\langle x, F(\gamma,\theta)\rangle|}{\|x\|_2 \cdot \|F(\gamma,\theta)\|_2} \tag{3.6}$$

définit une ***mesure de similarité***.

Γ est une mesure quantitative qui caractérise l'adéquation aux données de l'objet sonore x d'un modèle F avec le prototype γ . En effet, la valeur de $\Gamma(x,\gamma)$ est une fonction décroissante de l'erreur minimale relative $\frac{\left\|e\left[x,\gamma,\widehat{\theta}\right]\right\|_2}{\|x\|_2}$ à x et γ fixés.

Étant donné un évènement x et deux éléments γ, γ', on peut donc choisir le modèle le plus adapté à l'objet x en comparant les valeurs de $\Gamma(x,\gamma)$ et de $\Gamma(x,\gamma')$. On remarque que la spécification de la fonction F est implicite dans la définition de Γ.

La matrice $\Gamma(x_m, g_{m'})_{1\leq m,m'\leq M}$ est dénommée ***matrice de similarité***, et celle-ci fournit un moyen de visualiser facilement les ressemblances locales entre différentes portions du signal.

Cette cartographie bi-dimensionnelle des ressemblances du signal a été introduite en traitement du signal indépendamment par J. Foote[Foo01], et introduite de façon intérieure en bio-informatique, pour la visualisation des ressemblances entre des sous-châines de séquences génétiques.

3.1.7 Exemple de modèle : Synthèse à Table d'ondes

Les grains g sont ici des signaux choisis parmi une *table d'ondes* $\{\gamma_k\}_{k=1\ldots K}$, qui est simplement une collection de signaux échantillonnés de durées identiques $\delta = d$.

Les objets sonores x_m sont obtenus en fenêtrant par une fonction W de support $[1,d]$, le signal γ_{k_m} , élément d'indice $k = k_m$ de la table d'ondes . Le facteur variable α détermine l'amplitude de l'objet.

$$x_m(t) = \alpha \cdot W(t) \cdot \gamma_{k_m}(t) \tag{3.7}$$

α paramètre d'amplitude

W fonction de fenêtrage (ou signal d'enveloppe)

γ_k forme d'onde n° k de la table d'ondes

Ce modèle sera présenté en détail au Ch.5, et il sert de base à l'élaboration de modèles plus sophistiqués.

3.2 Discussion autour du modèle

3.2.1 Caractéristiques des objets sonores

On a tenté de définir ce qu'on entend par *objet sonore* au chapitre précédent. De façon schématique, chaque objet sonore est un événement résultant du contrôle d'une méthode de production sonore (instrument acoustique, synthétiseur ...) par une directive (message MIDI, action physique de l'instrumentiste ...). Cependant, un tel modèle reflétant fidèlement la structure de production du signal n'est pas absolument nécessaire, et surtout il serait extrêmement difficile à mettre en œuvre.

Supposons que l'on dispose de connaissances suffisantes pour modéliser l'ensemble des méthodes de production sonores utilisées dans une performance, par le biais d'une fonction mathématique ou d'un algorithme F. Supposons également que nous sachions décrire de façon quantitative et exhaustive l'ensemble des directives de contrôle utilisées, celles-ci correspondent aux variables t_m et θ_m. Le prototype γ_k représenterait alors un ensemble des données, programme et mémoire brute, fournissant une description exhaustive du modèle $F(\gamma_k, \theta)$ du $k^{\text{ième}}$ instrument.

Cependant, comme on l'a expliqué précédemment (*cf.* 2.1.1), la diversité des méthodes de production sonore et la difficulté d'avoir accès aux paramètres de contrôle interdisent de mettre cette idée en pratique. Si on se place dans le cas où on ne dispose d'aucune information *a priori*, issue de mesures ou tirée de connaissances particulières éventuelles sur le signal, il faudrait alors de plus commencer par dériver ces informations à partir du signal.

Une telle approche, que l'on pourrait qualifier de modélisation « *structurelle* » ou « *explicative* » du signal, peut être opposée à un modèle qualifié quant à lui de « *phénoméniste* », au sens où on se contente de rendre compte du phénomène observé, c'est-à-dire du signal de référence.

D'un point de vue strictement mathématique, les deux approches sont identiques ; elles

se distinguent cependant par le sens que l'on donne à la fonction F. Dans l'approche dite structurelle, F représenterait une méthode de synthèse sonore, et aurait donc un sens physique (par exemple), alors que dans l'autre cas, on ne lui impose pas de posséder un sens autre qu'un sens mathématique.

L'approche que nous avons choisie est essentiellement « phénoméniste » : la fonction F n'a pas pour but de modéliser la façon dont l'objet a réellement été produit mais seulement de décrire l'objet le plus fidèlement possible. Ce choix a notamment l'avantage de permettre d'utiliser la même fonction F pour tous les objets et types de signaux.

3.2.2 Localisation temporelle des objets

A priori, rien ne permet de justifier de contraindre l'emplacement des support temporels des objets à des positions particulières. En pratique, le fait de prendre des objets de durée constante et égale, situés à des positions arbitraires, présente un certain nombre d'avantages. En plus de faciliter le calcul de distances entre deux objets et de simplifier la mise en œuvre de la partie informatique, on évite ainsi d'avoir recours à une étape préalable d'identification des frontières temporelles des objets. Il existe différentes approches pour effectuer une segmentation d'un signal, par rapport aux variations du profil énergétique ou aux variations d'une mesure d'innovation [Foo00], par détection de ruptures [BN93, BD00], approches bayésiennes [CG98] ou encore par modèle de Markov cachés ou HMM [AOJP97, BC00, Rap99], *etc.*

On a choisi de fixer la position des objets sur une grille régulière. Les objets x_m sont issus d'un découpage s_n en trames du signal s, et dans un premier temps tout du moins, on prend $x_m = s_m$, ce qui correspond au cas où les objets ne se recoupent pas. Le découpage en trames est détaillé au chapitre suivant (4.1.1).

Le cas où plusieurs objets $\{x_m\}_{m \in I_n}$ sont superposés est notablement plus compliqué. On a en effet directement accès à la somme algébrique des objets $s_n = \sum_{m \in I_n} x_m$ et non individuellement aux objets x_m.

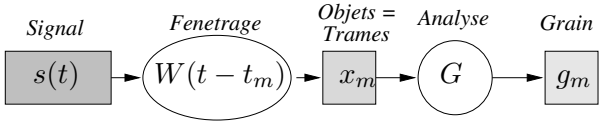


FIG. 3.5 – Des objets obtenus par découpage du signal en trames, suivi d'un fenêtrage.

3.2.3 Différents types de redondances

De manière générale, un message est qualifié de redondant lorsque celui contient une quantité d'information superflue. Dans le cadre du modèle granulaire, nous distinguons deux types de redondances :

- *inter-objets*, ou à long-terme, quand l'information contenue dans deux objets distincts se recoupe
- *intra-objet*, ou à court terme, quand la quantité d'information contenue dans un objet donné est inférieure à sa taille

Illustrons la redondance *intra-objet* à l'aide d'un exemple trivial : une trame de silence, constituée uniquement de zéros. L'information contenue dans cette trame est minimale, l'instruction « générer d valeurs égales à zéro » suffisant en effet à reconstituer le signal.

La quantité d'information peut être mesurée de plusieurs manières, selon que l'on adopte un point de vue statistique (*p.ex.* entropie de Shannon, information mutuelle) ou un point de vue déterministe et algorithmique (*p.ex.* complexité de Kolmogorov, Minimum Description Length).

La question de l'exploitation de la redondance *intra-objet* dans le modèle n'est pas abordée ici, sachant qu'il s'agirait essentiellement d'appliquer des algorithmes de compression existants sur les vecteurs-prototypes.

A contrario, la prise en compte des redondances *inter-objets* est une des caractéristiques principales du modèle granulaire. L'idée est qu'en décrivant un ensemble de deux ou plus trames « redondantes » avec un seul et même prototype, on peut réduire la longueur du message à transmettre. L'information transmise ne sera bien sûr pas exactement identique, en raison des imperfections du modèle. L'objectif est alors de minimiser les différences entre le signal d'origine et le signal reconstruit à partir du modèle.

En réalité, on n'utilise pas de mesure de redondance mais une mesure de similarité entre objets, car cette dernière se relie facilement à la mesure d'erreur. Cette mesure de similarité s'apparente à une forme spéciale d'inter-corrélation.

3.2.4 Mesurer la similarité entre objets

Définition par rapport à la déformation F

$\mathbb{E}$ désigne ici l'espace dans lequel évoluent les objets sonores, *i.e.* $\mathbb{E} = \mathbb{R}^d$. On note également $\mathbb{E}^* = \mathbb{R}^d \setminus \mathbf{0}^d$ Soit alors une fonction Γ telle que :

$$\begin{array}{rccc} \Gamma : & \mathbb{E} \times \mathbb{E} & \rightarrow & [0,1] \\ & x, y & \mapsto & \Gamma(x, y) \end{array} \tag{3.8}$$

On dit que est Γ une mesure de similarité selon le modèle γ si elle vérifie les conditions suivantes :

$$\begin{array}{lcll} \Gamma(x,x) & = & 1 & \forall x \in \mathbb{E} \\ \Gamma(x,y) & = & \Gamma(y,x) & \forall (x,y) \in \mathbb{E}^* \times \mathbb{E}^* \\ \Gamma(x,\mathbf{0}) & = & 0 & \forall x \in \mathbb{E}^* \\ \Gamma(x,y) & = & 1 & \Leftrightarrow \exists (g,\theta_0,\theta_1) \ / x = F(g,\theta_0), y = F(g,\theta_1) \end{array} \tag{3.9}$$

La première condition, triviale, impose qu'un grain soit considéré comme exactement similaire, donc identique à lui-même. La seconde impose la symétrie de la mesure par rapport à une permutation des objets. La quatrième condition impose l'invariance de la mesure par rapport aux variations des paramètres du modèle.

Dans le reste du document, on utilisera aussi la notation équivalente $\Gamma(x,\gamma)$

Autres définitions possibles de la similarité

Il est possible de définir une mesure quantitative du degré de ressemblance entre objets sans pour autant spécifier un modèle de transformation F correspondant. On peut par exemple utiliser les mesures existantes, parmi lesquelles l'ensemble des distances spectrales. Bien que moins contraignante que la précédente, cette définition a un intérêt limité dans le cadre du modèle granulaire, sachant qu'il est nécessaire de pouvoir exhiber la fonction F si l'on veut pouvoir reconstruire un ou plusieurs objets à partir d'un autre.

Nous exhibons à présent deux exemples de mesures de similarité, et verrons comment celles-ci peuvent être reliées à une déformation F particulière.

3.2.5 Exemples de mesures

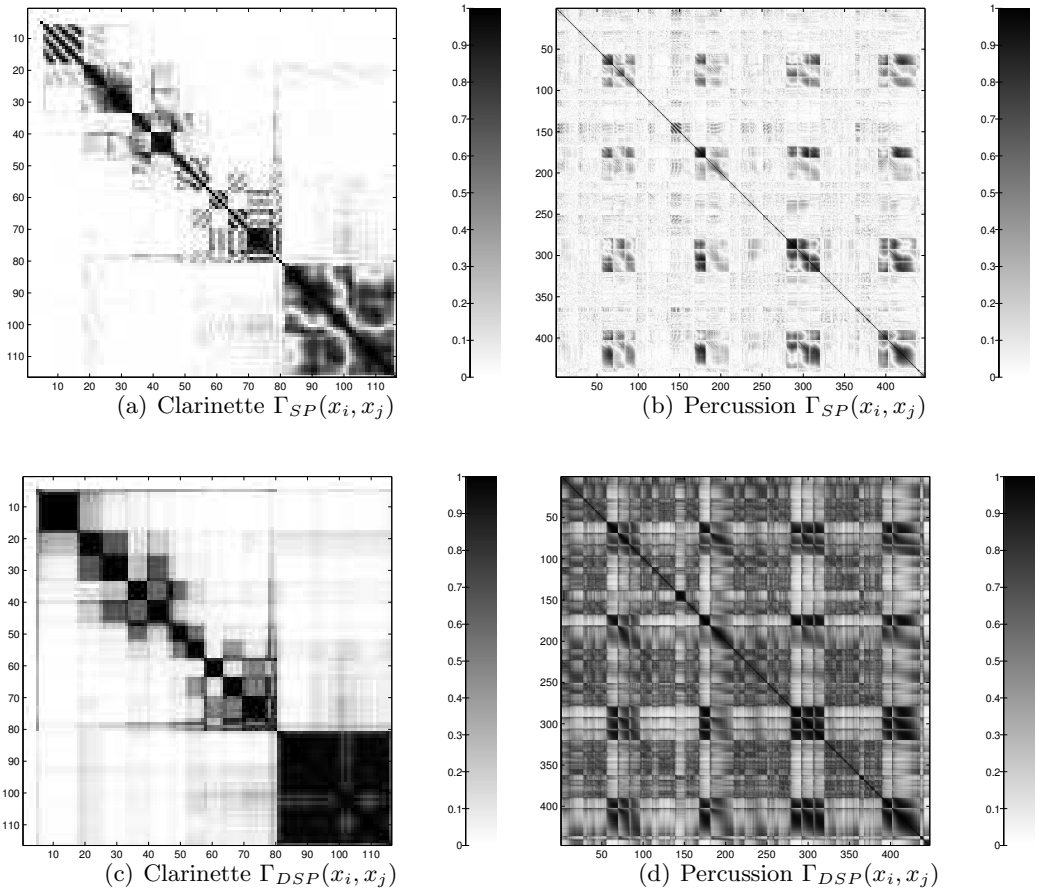
Produit scalaire normalisé

$$\begin{array}{lcll} \Gamma_{SP}(x,y) & = & \frac{\left|\langle x,y\rangle_2\right|}{\|x\|_2\|y\|_2} & \forall (x,y) \in \mathbb{E}^* \times \mathbb{E}^* \\ & = & 0 & \text{si } \|x\|_2 \|y\|_2 = 0 \end{array} \tag{3.10}$$

La mesure $\Gamma_{SP}(x,y)$est invariante à l'amplitude des grains, c'est-à-dire :

$$\Gamma_{SP}(\alpha \cdot x, y) = \Gamma_{SP}(x,y) \ \forall \alpha \neq 0, (x,y) \in \mathbb{E}^* \times \mathbb{E}^*$$

L'examen des équations 3.6 et 3.7 permet de constater que cette mesure est reliée au modèle Table d'Ondes présenté au paragraphe 3.1.7.



(a) Clarinette $\Gamma_{SP}(x_i, x_j)$ (b) Percussion $\Gamma_{SP}(x_i, x_j)$

(c) Clarinette $\Gamma_{DSP}(x_i, x_j)$ (d) Percussion $\Gamma_{DSP}(x_i, x_j)$

FIG. 3.6 – Exemples de matrices d'auto-similarité

L'intensité du pixel de coordonnées (i, j) dépend de la valeur de $\Gamma(x_i, x_j)$: un pixel noir correspond à $\Gamma = 1$, un pixel blanc à $\Gamma = 0$.
La matrice (a) a été obtenue en calculant les valeurs de la mesure de similarité entre 120×120 trames longues de 512 échantillons et se recoupant sur 256, à une fréquence d'échantillonnage 11kHz , calculées en utilisant le produit scalaire normalisé, à partir d'un enregistrement d'une mélodie de clarinette [Bou].
La mélodie est constituée de onze notes de hauteurs différentes. A première vue, on distingue une structure approximativement bloc-diagonale. Des trames correspondant à une même note se ressemblent vraisemblablement beaucoup (à un facteur d'échelle près), ce qui explique que les valeurs de similarité observées soient proches de 1.
La matrice (c) correspond au produit scalaire des DSP normalisées des trames pour le même signal. Les figures (b) et (d) ont été obtenues à partir d'un enregistrement de percussions (djembé).

Produit scalaire des DSP normalisées

$$\begin{array}{rcll} \Gamma_{DSP}(x,y) & = & \frac{\langle |Sx|,|Sy| \rangle}{\|x\|_2 \|y\|_2} & \forall (x,y) \in \mathbb{E}^* \times \mathbb{E}^* \\ & = & 0 & \text{si } \|x\|_2 \|y\|_2 = 0 \end{array} \tag{3.11}$$

Sx désigne la transformée de Fourier du grain x, $|Sx|$ est donc une estimation de la densité spectrale de puissance (DSP) de l'objet x.

La mesure $\Gamma_{DSP}(x,y)$ est par conséquent invariante à un changement d'amplitude et de spectre de phase des objets. Sachant qu'une transformation arbitraire du spectre de phase d'un objet de dimension d nécessite la connaissance des valeurs prises par $d/2$ variables, ce modèle apparaît comme étant très peu efficace, à moins de prendre des mesures pour réduire le nombre de ces variables. Cette question difficile sera abordée au chapitre 5, section 5.2.

La figure 3.6 présente plusieurs représentations graphiques des valeurs de la mesure de similarité, obtenues à partir de deux signaux de nature très différente. L'un est un signal de clarinette, de nature essentiellement harmonique, et l'autre un signal de percussions, contenant surtout des transitoires.

3.3 Représentation par un modèle granulaire

Conventions de notation

Pour plus de clarté, on utilise les notations vectorielles $\mathbf{t} = \{t_m\}_{m=1...M}$, $\mathbf{k} = \{k_m\}_{m=1...M}$ et $\boldsymbol{\theta} = \{\theta_m\}_{m=1...M}$, $\boldsymbol{\theta}$ étant la matrice de dimension $M \times q$ dont les colonnes sont constituées des vecteurs θ_m.

A chaque fois qu'un élément du modèle sera considéré comme connu et fixé, on ne fera pas apparaître celui-ci dans les équations.

3.3.1 Formulation générale du problème

Après avoir présenté le modèle granulaire dans ses grandes lignes, on s'intéresse au problème de l'analyse d'un signal pour obtenir une représentation selon ce modèle. Étant donné un signal quelconque s et une fonction de synthèse F, le problème général de l'analyse consiste à déterminer :

M	$\rightarrow$	le nombre d'objets
$\mathcal{D} \triangleq \{\gamma_k\}_{k=1\ldots K}$	$\rightarrow$	le dictionnaire
K	$\rightarrow$	la taille du dictionnaire
$\Theta \triangleq \{(t_m, k_m, \theta_m)\}_{m=1\ldots M}$	$\rightarrow$	l'ensemble des paramètres permettant de reconstruire les objets $\tilde{x}_m$
e_s	$\rightarrow$	l'erreur totale sur le signal

tels que

$$s(t) = \sum_{m=1}^{M} F(\gamma_{k_m}, \theta_m)(t - t_m) + e_s(t) \tag{3.12}$$

Les données de F, M,K, D et Θ définissent entièrement une représentation ou modèle granulaire du signal s, qui sera notée $\mathcal{G}$.

$$\mathcal{G} \triangleq \{F, M, K, D, \mathbf{t}, \mathbf{k}, \boldsymbol{\theta}\}$$

Approximation du signal

La représentation $\mathcal{G}$ permet de construire un signal dénommé approximation du signal s par $\mathcal{G}$. Ce signal noté $\tilde{s}(t)$, est défini par la relation suivante :

$$\tilde{s}(t) \triangleq \sum_{m=1}^{M} F(\gamma_{k_m}, \theta_m)(t - t_m) \tag{3.13}$$

La différence entre le signal original et son approximation, ou *erreur d'approximation,* est notée $e(s; F, K, D, M, \Theta)$, $e(s, \mathcal{G})$ ou encore $e_{\mathcal{G}}$, selon le contexte[4].

$$e_{\mathcal{G}} = e(s, \mathcal{G}) \triangleq s - \tilde{s} \tag{3.14}$$

Contraintes imposées

Un examen rapide de 3.12 permet de constater que l'égalité est vérifiée pour n'importe quelle représentation $\mathcal{G}$, pourvu que l'on choisisse $e_s = e_{\mathcal{G}} = s - \tilde{s}$. L'introduction de contraintes mathématiques appropriées va nous permettre de restreindre l'espace des solutions de 3.12.

On introduit à cet effet deux critères caractérisant deux propriétés différentes d'une représentation :

- la **qualité** de la représentation, qui mesure l'erreur $e_{\mathcal{G}}$ commise par rapport au signal d'origine s

[4]L'égalité $s = \tilde{s} + e_{\mathcal{G}}$ est par définition vérifiée quel que soit la représentation $\mathcal{G}$ adoptée.

- l'**économie** de la représentation, qui mesure la quantité d'information nécessaire à la description ou à la transmission des données de la représentation.

Notons que l'on cherche à obtenir une représentation à la fois de la meilleure qualité et de la plus grande économie possible. Pourtant, ces deux critères sont *a priori* antinomiques : on s'attend à ce que pour un signal donné, augmenter la qualité d'une représentation ait pour effet de diminuer l'économie correspondante, et inversement.

Ces deux notions sont semblables respectivement à celles de *distorsion* et *débit* utilisées en codage. La théorie débit-distorsion[GG92] s'appuie sur un cadre statistique et les valeurs du couple débit-distorsion sont alors des valeurs moyennes. Ces valeurs moyennes du couple débit-distorsion peuvent soit être calculées en faisant un certain nombre d'hypothèses sur la distribution du signal à représenter, soit estimées de manière empirique, à partir d'un ensemble d'exemples de signaux réels. Cette dernière façon de faire, bien que plus réaliste, présente l'inconvénient de conduire à des résultats différents selon les exemples choisis.

Les mesures de qualité et d'économie auquel on fait appel sont calculées à partir d'un seul « exemple » de signal d'analyse. En effet, on cherche a obtenir une représentation avec le meilleur couple économie-qualité pour un seul signal, et non pour un ensemble ou une classe de signaux.

3.3.2 Qualité de l'approximation

La qualité de l'approximation sera mesurée par la valeur de la norme L_2 de l'erreur d'approximation $e_{\mathcal{G}}(t)$, qui correspond physiquement à l'énergie du signal d'erreur. Cette mesure est choisie à la fois pour sa simplicité de calcul et pour son caractère largement répandu, ce qui facilite les comparaisons éventuelles avec d'autres méthodes. Selon l'application envisagée, il peut cependant s'avérer plus pertinent d'utiliser un critère dit *perceptif*, c'est-à-dire qui prend en compte les caractéristiques de l'oreille humaine comme la sensibilité variable selon la fréquence (courbe de Fletcher-Munson) ...

On définit par ailleurs le *Rapport Signal sur Bruit* ou RSB, noté ici μ, comme le rapport de puissance entre le signal et l'erreur d'approximation :

$$\mu \triangleq \frac{\|e_{\mathcal{G}}\|^2}{\|s\|^2} \triangleq \mu(\mathcal{G}) \tag{3.15}$$

Il s'agit d'une mesure de l'erreur relative, que l'on exprime le plus souvent en décibels, et que l'on note dans ce cas μ_{dB}

$$\mu_{\mathrm{dB}} = -20 \cdot \log \mu = -20 \cdot \log \frac{\|e_{\mathcal{G}}\|_2}{\|s\|_2} \tag{3.16}$$

s étant fixé, on peut choisir de minimiser $\|e_{\mathcal{G}}\|_2$ ou μ ou encore de maximiser μ_{dB}. L'op-

timisation de l'un de ces trois critères conduit donc à résoudre un problème d'optimisation au sens des moindres carrés.

3.3.3 Économie de la représentation

On examine les diverses possibilités de mesurer le coût correspondant à la description, noté χ, des données nécessaires à la représentation d'un signal par un modèle granulaire.

Cas des représentations linéaires

Dans le cas où la représentation est linéaire (bases, frames ...), la notion de **parcimonie** renvoie au nombre de composantes effectivement utilisées pour représenter le signal. La parcimonie est un indicateur de la dispersion des valeurs des coefficients d'une représentation d'un signal sur une famille vectorielle (base, dictionnaire ...). La norme L_0 des coefficients ou le rapport des normes L_1/L_2 sont communément utilisées.

Notons de plus que la parcimonie peut être reliée [KDRE$^+$99] à la capacité d'un signal à être codé de façon efficace. Par exemple dans le cas de la norme L_0, plus la représentation du signal comporte de coefficients nuls et plus petit sera le nombre de ces même coefficients à transmettre.

Cas de la représentation granulaire

Du fait que la fonction de synthèse F n'est en général pas linéaire par rapport aux paramètre θ, un critère de parcimonie n'est cependant pas suffisant pour mesurer le caractère économique d'une représentation granulaire. Une raison pour ne pas utiliser la parcimonie est que les différentes composantes $\theta_1 \dots \theta_d$ de θ ne sont ni de même nature, ni du même ordre de grandeur ; il n'y aurait donc aucun sens à additionner des puissances de celles-ci.

Une deuxième raison, plus profonde, est que la parcimonie n'est pas équivalente à l'économie, au sens ou la parcimonie ne mesure pas la régularité des coefficients. Les techniques telles que le codage entropique permettent de diminuer la quantité de donner nécessaires à la description quantitative des coefficients en exploitant les régularités dans la suite des coefficients. Autrement dit, il est aussi économique de coder un vecteur de coefficient α que le vecteur $\alpha + c$, où $c \neq 0$ est un vecteur constant, bien que $L_0(\alpha + c) > L_0(\alpha)$.

La troisième raison, et à nos yeux la plus importante, est que la connaissance de la base, du dictionnaire ou autre est, en plus des coefficients, nécessaire à la reconstruction du signal. Ainsi, quand on transmet une suite de coefficients de Fourier, la base est connue implicitement par l'inclusion du terme 'Fourier', et il est facile de construire une base de Fourier à partir de la seule donnée de la taille de la base. *A contrario*, le dictionnaire utilisé

pour la représentation granulaire dépend lui-même du signal, et il faut donc le transmettre tel quel (ensemble de vecteurs) ou codé sous une forme ou une autre.

Le critère d'*économie* ou de *complexité*

Dans l'idéal, le critère que l'on voudrait réellement mesurer est la *quantité d'information strictement nécessaire* à la description complète des différents éléments de la représentation, qui est formalisée par les notions de *Complexité de Kolmogorov* [GV99, LV93, VM02] et de *Minimum Description Length,* dans sa version dite *idéale*[5] [GV03]. La définition de la complexité proposée par Kolmogorov est particulièrement élégante, en ce sens que celle-ci ne requiert ni ne présuppose aucune autre connaissance que celle des observations elle-mêmes, et en particulier d'aucun *a priori* ou modèle des données observées, comme c'est par exemple le cas avec la théorie débit-distorsion.

Malheureusement, s'il a été démontré l'existence et l'universalité de cette mesure de complexité, il a également été démontré qu'on ne peut pas concevoir de programme pour la calculer de manière systématique. On peut toutefois employer certaines stratégies pour essayer en pratique de contourner le problème, comme de chercher des majorants de la complexité de Kolmogorov, typiquement en utilisant un algorithme de compression exploitant les redondances statistiques des données. Par exemple, le système de classification automatique de morceaux de musique par genres, présenté dans « *Algorithmic Clustering of Music* » [CVdW03], utilise une mesure d'information mutuelle dérivé de la complexité de Kolmogorov, qui est approximée en mesurant le taux de compression d'un algorithme standard type Lempel-Ziv. Ce système exploite en l'occurrence la représentation MIDI, représentation qui peut déjà elle-même être considérée comme une forme du signal très compressée (avec pertes).

Sachant que l'on travaille directement à partir de fragments de signaux et non à partir d'une représentation symbolique, qui est naturellement beaucoup plus compacte et structurée, l'approche consistant à relier la complexité de Kolmogorov du dictionnaire au taux de compression par un algorithme standard ne nous a pas semblé réaliste. Le problème est en effet que la mesure résultante serait excessivement dépendante du choix de l'algorithme de compression utilisé, par exemple selon que l'on utilise un codeur sans pertes (type zip) ou psychoacoustique (type mp3), auquel viennent se rajouter des problèmes purement calculatoires résultant de la masse conséquente de données à traiter.

Le choix a donc été fait de définir le coût χ simplement comme le *nombre de variables scalaires indépendantes nécessaires à la description de la représentation.*

Défini de cette manière, χ peut être qualifié de *coût structurel* de description, au sens

[5] Le terme MDL recouvre plusieurs définitions selon les différents auteurs et ouvrages considérés.

ou celui-ci n'intègre que les *relations entre les objets* et fait entièrement abstraction de la partie bas-niveau de la représentation, c'est-à-dire des données du dictionnaire. Avant de détailler les différents termes intervenant dans l'expression du calcul de χ, on insiste sur les points suivants :

- χ est un coût associé à *une* description de la représentation, ce n'est donc en aucune sorte un coût minimal.
- χ n'intègre pas le coût de description du signal d'erreur e_s.
- χ est rapporté au coût de description des données « brutes » du signal, *i.e.* N , où N est la longueur du signal en échantillons, conduisant ainsi à une quantité sans dimension. Cette normalisation par rapport à la taille du signal prendra tout son sens au moment de comparer la parcimonie de deux représentations pour deux signaux différents.

Expression détaillée de χ

Pour décrire une représentation d'un signal, il est *a priori* nécessaire de disposer des données correspondant aux valeurs prises par les variables ci-dessous.

Le coût de description d'une variable scalaire est choisi égal à 1. Ce choix permet d'éviter à ce stade le problème de la quantification des variables. Logiquement, on considère également qu'une variable vectorielle de dimension n a un coût n, équivalent à n scalaires. Les différents termes intervenant dans le calcul du coût total χ sont :

χ_0 données indépendantes du signal :
- données du programme de calcul numérique des valeurs prises par la fonction F
- données correspondant aux valeurs des constantes d, δ, q, K et M

χ_t données des instants t_m

χ_D données nécessaires à la description du dictionnaire $D = \{\gamma_{k_m}\}_{m=1\ldots M}$

χ_I données des index k_m des objets dans le dictionnaire

χ_θ données contenant les valeurs des paramètres θ_m

L'expression des coûts individuels est détaillée ci-dessous, puis simplifiée quand cela s'avère possible, :

χ_0 Les données de la fonction F, les dimensions d des objets et δ des prototypes sont fixées à l'avance. Les valeurs de M et de K peuvent en outre être déduites des données de k_m et de D respectivement : il suffit de compter leur nombre. Le coût χ_0 ne sera par conséquent pas pris en compte dans le coût global χ.

χ_t Les instants t_m étant fixés à l'avance, à moins que le contraire ne soit spécifié explicitement, ce coût est nul.

χ_D Le coût χ_D dépend de la nature des prototypes, il sera par conséquent explicité au moment de fournir des exemples concrets de modèles granulaires, au chapitre 5, p.101. Dans tous les cas, ce coût est inférieur ou égal à $K \times \delta$.

χ_I Ce coût est simplement le nombre d'objets, M.

χ_θ χ_θ est égal au nombre M d'objets multiplié par la dimension q du vecteur θ.

Le coût global peut donc d'ores et déjà être écrit sous la forme simplifiée :

$$\chi = \chi_D + M + M \cdot q$$

La quantité β est définie comme le coût total normalisé par rapport au coût N de description du signal temporel. Cette quantité est l'analogue d'un taux de compression, à ceci près qu'il est fait abstraction de la quantification des variables.

$$\beta \triangleq \frac{\chi}{N} \tag{3.17}$$

3.3.4 Le problème sous contraintes

Dans l'absolu, on peut vouloir optimiser conjointement la qualité μ et l'économie χ de la représentation. Théoriquement, on peut alors chercher à minimiser la norme L_2 une combinaison linéaire de μ et χ, *i.e.* résoudre :

$$\left\{\widehat{M}, \widehat{K}, \widehat{D}, \widehat{\Theta}\right\} = \arg \min_{\Theta, K, D} \left\{ \|e\left(s, F, K, D, M, \Theta\right)\|_2 + \lambda \cdot \chi\left(s, F, K, D, M, \Theta\right) \right\}$$

On abordera ici uniquement la question de la maximisation de la qualité μ_{dB} (équivalente à la minimisation de μ).

Dans ce qui suit, on maximisera la qualité μ de la représentation à taille de dictionnaire K fixée, ce qui est plus aisé que de maximiser à χ fixé. On s'intéressera aux relations entre μ et χ[6] au moment de présenter les résultats des expériences (Cha.7)

3.4 Conclusion

Le problème à résoudre peut donc se formuler de la façon suivante :

[6] $\beta = \chi / N$

$$\left\{\widehat{M}, \widehat{D} = \{\gamma_k\}_{k=1\ldots K}, \widehat{\Theta}\right\} = \arg\min_{\Theta, D} \|e\left(s, F, K, D, M, \Theta\right)\|$$

On ne sait pas résoudre directement le problème de l'optimisation simultanée de tous les éléments constitutifs de la représentation, aussi le décomposer en trois sous-problèmes plus simples :

1. Identification des objets sonores (Section 4.1)

Comment décomposer un signal sur un dictionnaire donné ?

Si l'on suppose les objets sonores et le dictionnaire connus, ce problème revient alors à trouver pour chacun de ces objets le prototype et les paramètres minimisant l'erreur d'approximation

2. Choix du dictionnaire (Section 4.2)

Comment choisir le dictionnaire optimal étant donné un signal s ?

Ce problème inclut les questions liées au

- ► calcul du dictionnaire proprement-dit, sa taille K étant connue et fixée.
- ► choix du nombre K de prototypes dans le dictionnaire

3. Spécification de la forme générale des objets (Chapitre 5)

Quelle forme adopter pour la fonction F ?

Les chapitres suivants sont consacrés à l'étude de ces problèmes, moyennant certaines restrictions : les objets sonores sont fixés comme étant les trames du signal après fenêtrage, le domaine de recherche du dictionnaire est restreint et la fonction F est la même pour tous les prototypes.

Chapitre 4

Calcul du modèle

Un synopsis du principe général du modèle est présenté sur la figure 4.1 . La section 4.1 apporte des précisions sur les caractéristiques communes et fixées à l'avance pour tous les objets , ainsi que sur la manière d'estimer les caractéristiques inconnues, à savoir la valeur des paramètres qui minimisent l'erreur individuelle sur chaque objet. La seconde section est consacrée au problème du calcul d'un dictionnaire maximisant la qualité de la représentation pour une valeur de complexité donnée.

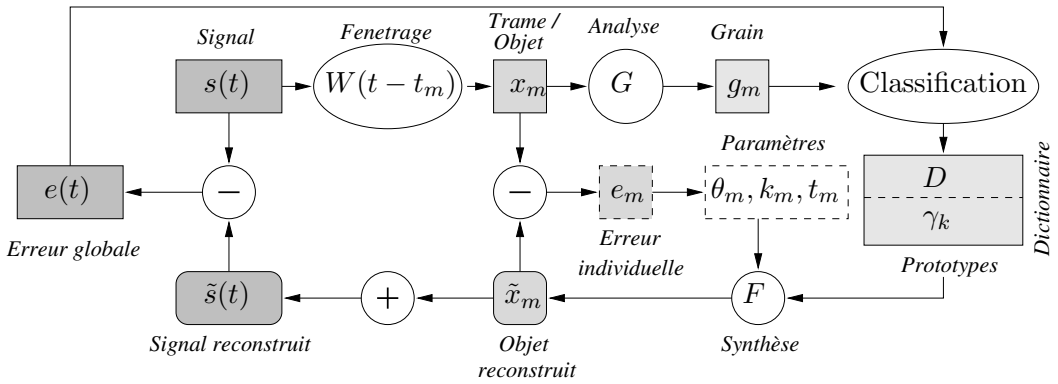


FIG. 4.1 – Principe général du calcul du modèle granulaire

4.1 Caractérisation des objets sonores

Rappel : dans cette section, le dictionnaire D et la spécification de la fonction de F sont supposés **connus**. Il s'ensuit que la complexité de la représentation est elle aussi connue et fixée. On s'intéresse au problème de la minimisation de e_s au sens des moindres carrés.

4.1.1 Localisation temporelle des objets

Ce paragraphe revient plus en détail sur le problème de l'identification d'un ou plusieurs objets sonores dans un signal.

D'un point de vue théorique, rien ne justifie de contraindre la forme des objets, et en particulier de restreindre leurs positions dans le temps. Cependant, si l'on n'impose aucune restriction, l'optimisation de la représentation devient un problème mal posé.

Le choix qui a été fait d'utiliser des objets de longueur fixe, à des positions temporelles situées sur une grille régulière, est classique. Les raisons avancées à ce choix sont que les méthodes existantes (*p.ex.* [BGC01, MC90]) permettant d'utiliser des objets de taille variable ne paraissaient pas facilement adaptables dans le contexte présent, d'une part.

D'autre part, les inconvénients liés à ce choix sont largement compensés par la possibilité d'exhiber des solutions au problème, d'autant que l'utilisation de certains types de modèles (cf. 5.1.2) permet de s'affranchir de certains de ces inconvénients.

Découpage du signal en trames

Une idée simple est de contourner le problème de l'identification des objets en fixant arbitrairement la forme des objets sonores, et c'est l'idée qui a été retenue en premier. Les objets sonores sont obtenus par un découpage en trames du signal, qui consiste en l'application d'une translation temporelle $-t_m$ suivie d'une multiplication, ou « fenêtrage » par une fonction W, comme suit :

$$x_m(t) \triangleq W(t) \cdot s(t - t_m)$$

Conditions sur la fenêtre W

Certains types particuliers de fonctions W sont particulièrement adaptés dans le contexte présent de l'analyse/synthèse, parce qu'elles possèdent des propriétés intéressantes. Dans l'esprit du modèle, les objets représentent des parties ou fragments du signal, qui ressemblent localement le plus au possible signal s, *i.e.* $x_m(t - t_m) \simeq s(t)$ autour de l'instant $t = t_m$ et $x_m(t - t_m) = 0$ ailleurs. Pour ces raisons, seules les fonctions W prenant leurs valeurs sur $[0, 1]$ seront retenues.

En outre, les x_m vérifient l'équation 3.1, ce qui se traduit par les égalités suivantes :

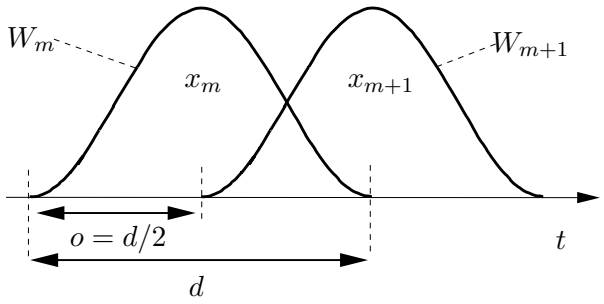


FIG. 4.2 – Recoupement temporel des fenêtres

$$\begin{aligned} s(t) &= \sum_{m=1}^{M} W(t - t_m) \cdot s(t) \ \forall t \\ &= \left(\sum_{m=1}^{M} W(t - t_m) \right) \cdot s(t) \ \forall t \\ \Leftrightarrow \sum_{m=1}^{M} W(t - t_m) &= 1 \ \forall t, s(t) \neq 0 \end{aligned}$$

Cette propriété de somme constante et unitaire des fenêtres garantit que l'on puisse reconstruire un signal identique au signal de référence, moyennant une reconstruction exacte des objets.

Enfin, la fenêtre W est choisie symétrique, *i.e.* $W(t) = W(-t)$ et les instants t_m sont alignés sur une grille régulière.

Le choix s'est porté sur l'utilisation d'une fenêtre de Hanning W_{Hann} d'une longueur d. La longueur d sera préférentiellement une puissance de 2, afin de pouvoir exploiter un algorithme de calcul de TF optimisé (FFT). D'autre part, les instants t_m sont distants de $d/2$ échantillons, ce qui correspond à un recouvrement des trames de 50%.

$$\begin{aligned} W_{\text{Hann}}(t) &\triangleq \begin{cases} \frac{1}{2}\left(1 - \cos\left(2\pi \frac{t}{d-1}\right)\right) \ \forall t \in [1 \dots d] \\ 0 \quad \text{sinon} \end{cases} \\ t_m &= \frac{m-1}{2} \cdot d + 1 \end{aligned} \tag{4.1}$$

4.1.2 Estimation des paramètres

Les localisations temporelles $\{t_m\}_{m=1\dots M}$ des objets étant désormais supposées connues, les événements sonores $\{x_m\}_{m=1\dots M}$ sont également connu. La notation abrégée $e(\mathbf{k}, \boldsymbol{\theta})$ désigne l'erreur d'approximation dans ces conditions.

Le problème consiste donc maintenant à déterminer l'ensemble des index de prototypes $\widehat{\mathbf{k}}$ ainsi que l'ensemble des paramètres $\widehat{\boldsymbol{\theta}}$ associés satisfaisant :

$$\left\{\widehat{\mathbf{k}}, \widehat{\boldsymbol{\theta}}\right\} = \arg\min_{\mathbf{k},\boldsymbol{\theta}} \left\| e\left(s; F, K, D, M, \mathbf{t}, \mathbf{k}, \boldsymbol{\theta}\right)\right\|_2$$

L'erreur minimale obtenue en faisant varier uniquement les index des prototypes et les paramètres des objets, le reste étant fixé, sera notée $e\left(s, F, K, D, M\right)$:

$$e\left(K, D\right) \triangleq e\left(K, D, \widehat{\mathbf{k}}, \widehat{\boldsymbol{\theta}}\right)$$

Estimation pour un objet sonore « isolé »

On note $e\left(k_m, \theta_m\right) = x_m - F\left(\gamma_{k_m}, \theta_m\right)$ l'erreur de modélisation d'un seul objet x_m.

$$\begin{aligned}\left(\hat{k}_m, \hat{\theta}_m\right) &= \arg\min_{k_m,\theta_m} \left\| e\left(k_m, \theta_m\right)\right\|_2 \\ &= \arg\max_{k_m,\theta_m} \Gamma\left(x_m, \gamma_{k_m}\right)\end{aligned}$$

Pour rechercher le meilleur prototype pour un objet donné, il faudrait *a priori* rechercher le maximum de $\Gamma\left(x, \gamma_k\right)$ sur tous les index k. Cependant, on peut dispose d'une expression analytique (eq. 4.2) de l'amplitude optimale , les autres paramètres étant libres :

$$\widehat{\alpha}_m = \frac{\left\langle x_m, F\left(\gamma_{k_m}, \underline{\theta}_m\right)\right\rangle}{\left\| F\left(\gamma_{k_m}, \underline{\theta}_m\right)\right\|^2} \text{ où } \underline{\theta}_m \triangleq \left[1, \theta_2, \ldots, \theta_q\right] \tag{4.2}$$

Estimation globale dans le cas où les objets sont orthogonaux

Le carré de la norme de l'erreur totale à dictionnaire fixé est

$$\begin{aligned}\left\| e\left(\mathbf{k}, \boldsymbol{\theta}\right)\right\|^2 &= \left\| \sum_{m=1}^{M} e\left(k_m, \theta_m\right)\left(t - t_m\right)\right\|^2 \qquad (4.3) \\ &= \sum_{m=1}^{M} \left\| e\left(k_m, \theta_m\right)\right\|^2 + \sum_{m \neq m'} \left\langle e\left(k_m, \theta_m\right)\left(t - t_m\right), e\left(k_{m'}, \theta_{m'}\right)\left(t - t_{m'}\right)\right\rangle\end{aligned}$$

Si les signaux $x_m\left(t - t_m\right)$ sont orthogonaux deux à deux, les signaux d'erreurs $e\left(k_m, \theta_m\right)\left(t - t_m\right)$ le sont aussi et le carré de l'erreur totale est la somme des carrés des erreurs individuelles des objets x_m. Dans ce cas, la minimisation de totale l'erreur revient à une minimisation individuelle sur chaque objet.

Dès que les fenêtres se recoupent, ce qui est le cas quand on choisit des fenêtres à

somme unitaire et de forme non-rectangulaire comme nous l'avons fait ici (cf. §4.1.1), les supports temporels des objets ne sont alors pas disjoints, et les objets ne peuvent pas par conséquent être systématiquement considérés comme étant orthogonaux.

Estimation globale dans le cas général

Lorsque les supports temporels de deux fenêtres adjacentes $W(t-t_m)$, $W(t-t_{m+1})$ se chevauchent, le produit scalaire $\langle e(k_m,\theta_m)(t-t_m), e(k_{m'},\theta_{m'})(t-t_{m'})\rangle$ est en général différent de zéro. Précisément, avec le choix de fenêtre retenu, le produit n'est en effet nul que pour des objets qui ne succèdent pas directement, *i.e.* pour un couple d'index (m,m') tel que $|m-m'|>1$. L'expression 4.3 peut alors s'écrire

$$\|e(\mathbf{k},\boldsymbol{\theta})\|^2 = \sum_{m=1}^{M}\|e(k_m,\theta_m)\|^2 + 2\cdot\sum_{m=1}^{M-1} w_m \tag{4.4}$$

où

$$w_m \triangleq \langle e(k_m,\theta_m)(t-t_m), e(k_{m+1},\theta_{m+1})(t-t_{m+1})\rangle$$

Un encadrement grossier de w_m est

$$|w_m| \leq \|e(k_m,\theta_m)\|_2 \cdot \|e(k_{m+1},\theta_{m+1})\|_2 \tag{4.5}$$

L'égalité est atteinte dans le cas où les signaux d'erreur correspondant à deux objets contigus sont colinéaires, et donc que leurs supports soient identiques, ce qui implique

$$\forall m \begin{cases} e(k_m,\theta_m)(t) & = 0,\ \forall t\in[0\ldots d/2] \\ e(k_{m+1},\theta_{m+1})(t) & = 0,\ \forall t\in[d/2\ldots d] \end{cases}$$

Ces deux égalités n'étant compatibles que dans le cas où $e(k_m,\theta_m)(t)=0,\ \forall t\in[0\ldots d/2]\,,\ \forall m$, on en conclut que l'inégalité 4.5 est une *inégalité stricte*, sauf quand l'erreur totale est nulle.

Pour aller plus loin, faisons l'hypothèse que l'énergie du signal d'erreur est uniformément répartie dans le temps, à savoir que

$$\sum_{t=0}^{d/2} e_m^2(t) = \sum_{t=d/2}^{d} e_m^2(t)$$

Sous cette hypothèse, cohérente avec la modélisation de l'erreur par un bruit identiquement distribué, l'encadrement de w_m devient

$$|w_m| \leq \frac{1}{2} \left\| e\left(k_m, \theta_m\right) \right\|_2 \cdot \left\| e\left(k_{m+1}, \theta_{m+1}\right) \right\|_2 \tag{4.6}$$

En voyant le terme $\sum_m^{M-1} \left\| e\left(k_m, \theta_m\right) \right\|_2 \cdot \left\| e\left(k_{m+1}, \theta_{m+1}\right) \right\|_2$ comme un terme d'autocorrélation du vecteur $\left[\left\| e\left(k_m, \theta_m\right) \right\|_2\right]_{m=1\ldots M}$, on majore $\sum_{m=1}^{M} |w_m|$ par $\sum_{m=1}^{M} \left\| e\left(k_m, \theta_m\right) \right\|^2$, et on aboutit en définitive à l'inégalité

$$\left\| e\left(\mathbf{k}, \boldsymbol{\theta}\right) \right\|^2 \leq 2 \cdot \sum_{m=1}^{M} \left\| e\left(k_m, \theta_m\right) \right\|^2 \tag{4.7}$$

Sachant les signaux d'erreurs $e\left(k_m, \theta_m\right)$ et $e\left(k_{m+1}, \theta_{m+1}\right)$ dépendent à la fois du signal, du dictionnaire et de la valeur des paramètres choisis, il nous paraît difficile d'aboutir à une caractérisation plus précise de leur corrélation sans faire d'hypothèses restrictives.

Conclusion

Bien que la minimisation de l'erreur totale $\left\| e\left(\mathbf{k}, \boldsymbol{\theta}\right) \right\|_2$ ne soit pas strictement équivalente à celle de l'ensemble des erreurs individuelles $\left\| e\left(k_m, \theta_m\right) \right\|_2$, l'inégalité 4.7 montre que si l'on se contente de minimiser individuellement chacun des M termes $\left\| e\left(k_m, \theta_m\right) \right\|_2$, on augmente *au maximum* l'erreur globale d'un facteur $\sqrt{2}$ soit 3 dB. On expose les conséquences pratiques de ce choix dans la partie expérimentale (§ 7.4).

Notons qu'il est possible de raffiner l'estimation des paramètres par un procédé itératif, par exemple en estimant dans un premier temps les paramètres des objets d'indices pairs et dans un second temps, en mettant à jour les objets d'indices impairs avec l'erreur commise sur les objets d'indices pairs , puis en estimant les paramètres correspondants, et ainsi de suite jusqu'à ce que le rythme de diminution de l'erreur globale atteigne un seuil proche de zéro.

On pourrait également exploiter le fait que le vecteur des amplitudes α_m minimisant l'erreur globale est solution d'un système linéaire dont la matrice est tridiagonale, et pour lequel il existe une méthode de résolution [PTVF92] rapide, bien qu'éventuellement instable.

Ces procédures nécessitant à chaque étape un nouveau calcul de matrice des similarités, elles n'ont pas été implémentées dans le programme, afin de ne pas l'alourdir encore plus.

4.1.3 Matching Pursuit et modèle granulaire

Le MP avec un dictionnaire connu ...

Dans le cas particulier où le dictionnaire D est connu à l'avance, on peut utiliser un des algorithmes existants pour identifier les objets sonores dans le signal. Par exemple, quand

on utilise un dictionnaire d'atomes de Gabor, l'algorithme Matching Pursuit décrit plus tôt dans ce document (*cf.* §2.1.3), fournit une solution au problème, bien que sous-optimale.

On peut adapter le MP au cas granulaire sans trop de modifications, sachant que cet algorithme a initialement été prévu pour le calcul d'une approximation d'un signal sur une famille de vecteurs. Il suffit de réécrire l'étape de maximisation du produit scalaire $\langle r_{m-1}, b_k \rangle$, précédemment notée « $m.1$ », qui est équivalente à une minimisation de la norme du résidu à l'étape suivante. A l'itération m, il faut maintenant maximiser le produit scalaire entre le résidu à l'étape courante r_m et l'objet reconstruit $F(\gamma_k, \theta)$, c'est-à-dire rechercher

$$\left(\theta_m^{(k)}, t_m^{(k)}\right) = \arg \max_{\theta^{(k)}, t^{(k)}} \left| \left\langle r_m(t), F\left(\gamma_k, \theta^{(k)}\right)\left(t - t^{(k)}\right) \right\rangle \right| \tag{4.8}$$

$$k_m = \arg \max_{k} \left| \left\langle r_m(t), F\left(\gamma_k, \theta_m^{(k)}\right)\left(t - t_m^{(k)}\right) \right\rangle \right| \tag{4.9}$$

La convergence de l'algorithme est assurée si le dictionnaire est complet [JS87], ou en dimension finie, comme c'est ici le cas, si la famille b_k engendre l'espace. Un problème se pose si la famille de vecteurs $\{F(\gamma_k, \theta)\}_{1 \leq k \leq K, \theta \in \mathbb{R}^q}$ peut éventuellement ne pas être génératrice de l'espace $\mathbb{R}^d$ des objets, particulièrement dans le cas où $K \ll d$.

Un MP sans dictionnaire *a priori ?*

Le véritable problème provient de ce que dans le cas qui nous intéresse, le dictionnaire n'est pas connu à l'avance et la simple adaptation du MP présentée plus haut n'a pas de sens. Pour contourner ce problème, on a envisagé de coupler le MP à une procédure de mise à jour du dictionnaire, sur un principe similaire à celui d'un algorithmes EM ou d'apprentissage de bases par renforcement. On examine à présent le schéma général de l'algorithme que l'on a imaginé pour ce « MP granulaire ».

Initialisation du dictionnaire On sélectionne soit les K trames du signal les plus énergétiques, soit les K premières composantes principales, ou encore les K premiers atomes de Gabor que l'on obtient avec un MP « classique » ... Une meilleure alternative à cette troisième option serait probablement d'utiliser le MP harmonique [GB03, Gri99], exploitant un dictionnaire de *molécules* harmoniques, qui sont des combinaisons d'atomes de Gabor dont les fréquences sont en rapport quasi-harmoniques, bien adaptées aux particularités des signaux audio.

Décomposition du signal sur le dictionnaire existant A partir du dictionnaire existant, on recherche successivement les M meilleurs objets avec le MP modifié avec le critère (4.8) approprié . La structure désormais tout à fait quelconque du dictionnaire interdit de

faire la moindre hypothèse par exemple sur la (quasi)-orthogonalité des objets, et l'absence d'expression analytique empêche toute simplification des calculs. Aussi certains procédés d'accélération (dictionnaires de maxima locaux, mise à jour rapide des produits scalaires, etc.) des calculs développés pour les dictionnaires d'atomes de Gabor ne sont plus exploitables, ce qui a un impact très négatif sur la complexité calculatoire. Toutefois, il est inutile de recalculer l'intégralité des produits scalaires $\langle r_m, F(\gamma_k, \theta)\rangle$ à chaque étape m, sachant que les résidus $r_m(t)$ et $r_{m+1}(t)$ ne diffèrent qu'à l'emplacement de l'objet sélectionné à l'étape m, autrement dit à l'intérieur de l'intervalle de temps $[t_m - d/2, t_m + d/2]$.

Rééstimation du dictionnaire On cherche un nouveau dictionnaire qui minimise l'erreur de reconstruction sur les M objets obtenus à l'étape précédente, en utilisant par exemple une méthode d'apprentissage par classification, méthode qui sera détaillée à la section suivante.

Un écueil potentiel serait qu'en ne prenant pas en compte le résidu r_M dans la mise à jour du dictionnaire, il est possible qu'une diminution de l'énergie de l'erreur de reconstruction sur les objets, accompagnée d'une augmentation de l'énergie du résidu r_{M+1} à l'étape suivante, débouche finalement sur une augmentation de l'erreur totale. La difficulté réside dans ce que l'on ne peut *par définition* pas calculer explicitement le dictionnaire qui minimise cette erreur totale, car dans le cas contraire, on n'aurait bien évidemment aucun besoin d'avoir recours à la procédure itérative décrite ici. Si l'on ne peut pas anticiper l'évolution relative de ces deux erreurs, on ne peut garantir la convergence théorique de l'algorithme, ce qui n'exclut heureusement pas que cette convergence soit observée expérimentalement.

Conclusion

La complexité calculatoire du MP avec des atomes de Gabor et dérivés est déjà élevée, et on s'attend à ce que celle de l'algorithme présenté ci-dessus lui soit encore nettement supérieure, d'autant que l'on ne peut pas ici utiliser les versions dites « rapides » du MP. En outre, l'implémentation de référence du MP par le package [GBA] présent au sein du logiciel LastWave [Bac], écrit en langage C, bien que plus rapide que le même algorithme programmé en Matlab, aurait été longue et difficile à modifier pour expérimenter un MP granulaire. Pour ces raisons, cette voie de recherche certainement très prometteuse, parce qu'elle permettrait notamment d'utiliser des objets de durée d variable et superposés[1], a été jugée trop ambitieuse et a donc été abandonnée.

[1] L'utilisation d'objets superposés dans le temps est indispensable pour la représentation de signaux polyphoniques.

Une solution plus réaliste compte tenu des capacités de calcul actuelles, serait d'utiliser un dictionnaire de prototypes de complexité supérieure à celle des molécules harmoniques, par exemple en rajoutant des degrés de liberté supplémentaires sur les paramètres de ses atomes constitutifs, puis de modéliser les variations des objets par rapport au dictionnaire.

On utiliserait alors une version du MP *guidée* part les ressemblances entre les différentes trames du signal pour accélérer la recherche de la localisation et des paramètres des objets. L'intérêt pressenti de ce guidage s'appuie sur l'intuition que plus la ressemblance (*p.ex* spectrale) entre deux trames différentes est forte, plus il est probable que la recherche des éléments du dictionnaire les plus corrélés avec le signal débouche sur des résultats similaires, et donc que l'on peut n'effectuer qu'une seule fois cette recherche.

4.2 Calcul du dictionnaire

On peut maintenant supposer que l'on sait estimer les paramètres des objets, et on va maintenant s'intéresser au problème de la constitution du dictionnaire.

Dans ce qui suit, les objets $\{x_m\}_{m=1\ldots M}$, leurs positions temporelles $\mathbf{t} \triangleq \{t_m\}_{m=1\ldots M}$, la fonction F et la taille du dictionnaire K sont supposés connus. Il s'agit donc désormais de constituer un dictionnaire D qui minimise l'erreur d'approximation du signal pour une taille de dictionnaire K donnée.

4.2.1 Position du problème

Le problème général à résoudre est :

$$\left\{\widehat{K}, \widehat{D} = \{\gamma_k\}_{k=1\ldots K}\right\} = \arg\min_{K,D} \|e\left(K, D\right)\|_2 \tag{4.10}$$

Pour rappel, $e\left(K, D\right)$ est l'erreur minimale obtenue avec un dictionnaire D et une fonction de synthèse F donnés. Le problème consiste ici à minimiser l'erreur quadratique par rapport au dictionnaire D.

En imaginant qu'on utilise une approche type « force brute », il faudrait pour chaque élément de D, parcourir un espace discret comportant $n_\gamma \sim \mathcal{O}\left(a^\delta\right)$ points, où a est une constante dépendant de la finesse du maillage de l'espace des grains, *i.e.* $\mathbb{R}^\delta$. Le nombre de points total à parcourir pour D, modulo les permutations sur les éléments de D serait donc en $C_{n_\gamma}^K \cdot n_\gamma$.

Pour avoir une idée des ordres de grandeurs considérés, les valeurs numériques utilisées pour K et δ sont typiquement de plusieurs dizaines de prototypes et de 512 échantillons respectivement. Il apparaît donc d'ores et déjà qu'une recherche exhaustive de $\widehat{D}$ sera hors de portée en raison des temps de calcul mis en jeu.

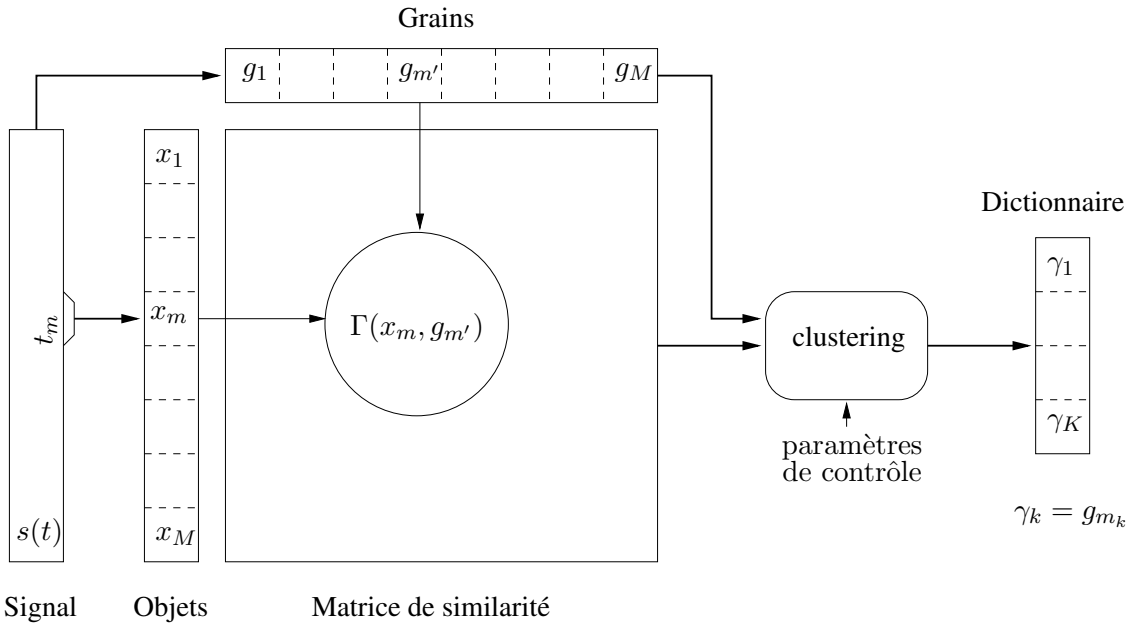


FIG. 4.3 – Calcul du dictionnaire par clustering

Choisir les prototypes parmi les grains, *i.e.* $D = \{g_{m_k}\}_{k \in 1 \ldots K}$

Une conséquence directe et triviale de ce choix est que les objets correspondants aux grains choisis comme prototypes sont modélisés avec une erreur nulle (pour rappel, on a $x_m = F(g, 0)$). L'espace de recherche de D est ainsi réduit à l'ensemble des parties non-ordonnées de $G = \{g_m\}_{m \in 1 \ldots M}$ à K éléments. Le cardinal de cet ensemble est C_M^K.

Une solution triviale : $\widehat{K} = M$

Le choix $K = M$ et $D = D_s = \{g_m\}_{m=1 \ldots M}$ conduit directement à une erreur de reconstruction nulle. Il suffit pour s'en rendre compte de prendre $k_m = m$ et $\theta_m = 0$, ce qui conduit à $F(\gamma_{k_m}, \theta_m = 0) = x_m$ et $\|e(s, K = M, D_s)\| = 0$. On peut également d'abord chercher une base b de $\overrightarrow{\text{vect}}\{x_m\}_{m=1 \ldots M}$ puis utiliser cette base pour construire un dictionnaire de taille $K = \text{rang}\{x_m\}_{m=1 \ldots M}$. On obtiendra alors $K \leq M$.

Bien que cet exemple de solution ne présente aucun intérêt pratique, il illustre la nécessité d'introduire une contrainte d'*économie de la représentation*. A présent, seules seront prises en compte les solutions pour lesquelles $\widehat{K} < M$.

4.2.2 Calcul du dictionnaire par apprentissage

L'optimisation exhaustive est hors de portée

Le choix qui a été fait ici consiste donc à rechercher le dictionnaire parmi l'ensemble des grains $G = \{g_m\}_{m=1 \ldots M}$. Autrement dit, le dictionnaire est contraint à être de la forme

$D = \{g_{m_k}\}_{k=1...K}$, m_k étant l'index du $k^{\text{ième}}$prototype dans l'ensemble G. La recherche exhaustive sur l'ensemble des parties non-ordonnées de G à K éléments impose de calculer et comparer C_M^K valeurs de $\|e(K, D)\|$correspondant aux C_M^K dictionnaires à parcourir. Pour des nombres de trames $M = 100$ et de prototypes $K = 10$, on atteint déjà un nombre de combinaisons à explorer de l'ordre de 10^{13}. Il est donc nécessaire de choisir une approche sous-optimale qui permette de réduire ce nombre de combinaisons de plusieurs ordres de grandeur.

Le principe de la classification automatique

Les termes de *classification non supervisée* ou *clustering* désignent une catégorie de techniques permettant de classifier automatiquement un ensemble de données. A l'issue du processus, chacune des données est donc affectée à une classe, sachant que les caractéristiques des classes ne sont bien évidemment pas connues à l'avance. Le but est d'aboutir à des classes contenant des données similaires, c'est-à-dire, par exemple que les distances entre éléments d'une classe soient minimisées.

L'utilisation d'un algorithme EM [DLD77] a également été considérée pour l'apprentissage du dictionnaire. Cette possibilité mériterait certainement d'être approfondie dans le futur, d'autant que B. Frey et consorts ont proposé l'utilisation d'une approche EM dans un contexte similaire [FJ03].

La suite présente un certain nombre de techniques de classification de données dans le contexte de l'apprentissage du dictionnaire. La classification est faite par rapport aux valeurs de la matrice $\Gamma(x_i, g_j)_{1 \leq i,j \leq M}$ des similarités. Traditionnellement, les algorithmes de classification travaillent sur les valeurs d'une mesure de la distance entre données. On établit la correspondance suivante entre une mesure de distance d et une mesure de similarité Γ :

$$d(x_i, x_j) = \frac{1}{\Gamma(x_i, g_j)} - 1$$

avec $g_j = F^{-1}(x_j, 0)$.

Terminologie et notations

On qualifiera d'***orphelins*** les objets qui, à une étape donnée, n'ont pas encore été affectés à une classe.

On parlera de grains ***candidats***[2] pour désigner des grains susceptibles d'êtres utilisés comme prototypes d'une classe, à une étape donnée du processus de classification.

[2] *N.B.* Un même grain peut être à la fois orphelin et candidat, par exemple dans le cas avec l'algorithme 5.

Les notations suivantes seront utilisées lorsque cela s'avérera nécessaire :

D	le dictionnaire constitué à l'étape courante ($D(i)$ à l'itération i)
K	le nombre de prototypes à l'étape courante ($K(i)$ à l'itération i)
$\mathcal{C}_k$	le cluster d'indice k à l'étape courante ($\mathcal{C}_k(i)$ à l'itération i)
γ_k	le prototype du cluster d'indice k à l'étape courante ($\gamma_k(i)$ à l'itération i)
$\sharp\mathcal{C}$	le nombre[3] de grains contenus dans le cluster $\mathcal{C}$
$\mathcal{O}$	ensemble des index des objets orphelins
$\mathcal{I}$	ensemble des index des grains candidats

4.2.3 Un exemple d'algorithme « classique », le K-Médians

Parmi l'ensemble des algorithmes de classification présentés dans la littérature, le *K-Moyennes* (K-Means) est un des plus couramment employés, et son fonctionnement est à la fois simple à comprendre et à mettre en œuvre. Lors des expériences, cet algorithme jouera le rôle de référence à laquelle on comparera les autres algorithmes. Une méthode différente telle que LBG, K-Moyennes Hiérarchique, Classification Ascendante Hiérarchique ou autres [Ber02, CH67, DEYG+02, FJ03, FK97, KR90] pourrait aussi bien être utilisée.

Principe de fonctionnement

Il est nécessaire de fixer au préalable une taille de dictionnaire K. L'algorithme 2 classe alors les M grains dans K clusters suivant le principe des plus proches voisins.

Étant donné un ensemble de prototypes $D = \{\gamma_k\}_{k=1\ldots K}$, le plus proche voisin $\gamma_{\overline{k}(x)}$ d'un objet x dans D est défini comme :

$$\gamma_{\overline{k}(x)} = \arg\max_{\gamma \in D} \left[\Gamma(x, \gamma)\right]$$

Une variante du K-Moyennes

On a expérimenté l'utilisation de l'algorithme *K-Médians* [JD98, KR90, NH94], ou *K-Medoids*, qui est une variante de l'algorithme K-Moyennes [JD98, Mac67], plutôt que ce dernier. La seule différence entre ces deux algorithmes réside au niveau de l'étape de mise à jour (2.1) : on utilise l'élément du cluster minimisant l'erreur sur le cluster, *i.e.* l'élément *médian*, et non le barycentre du cluster.

En effet, dans le cas présent, un prototype γ_k fait office de représentant de la classe $\mathcal{C}_k$, mais il appartient à un espace $\mathbb{R}^\delta$ *différent* de celui dans lequel évoluent les objets

[3] $\sharp$ désigne l'opérateur « cardinal de l'ensemble »

Alg. 2 Classification *K-Médians*

▶ `Choisir` K `index de grains` $\{m_1, \ldots, m_K\}$ `parmi` $\{1 \ldots M\}$
▶ `Construire le dictionnaire initial` $D(0) = \{g_{m_k}\}_{k=1\ldots K}$
▶ `Répéter ensuite les étapes` $i.1$ `et` $i.2$ `jusqu'à obtenir la convergence`

$i.1$ `Chercher le plus proche voisin` $\gamma_{\overline{k}(x_m)}$ `de chacun des objets` x_m `dans` $D(i)$
`Affecter l'objet` x_m `au cluster` $\mathcal{C}_{\overline{k}(x_m)}(i)$

$i.2$ `Chercher le grain médian` $\overline{g}_k$ `de chaque cluster` $\mathcal{C}_k(i)$

$$\overline{g}_k = \arg \max_{m' \in \mathcal{C}_k(i)} \sum_{m \in \mathcal{C}_k(i)} \Gamma^2(x_m, g_{m'})$$

`FAIRE` $\gamma_k(i) \leftarrow \overline{g}_k$ `et` $i \leftarrow i + 1$

$F(g, 0), g \in \mathcal{C}_k$, éléments de cette classe , à savoir $\mathbb{R}^d$. Dès lors, le barycentre

$$\overline{g}_k \triangleq \frac{1}{\sharp \mathcal{C}_k} \cdot \sum_{g \in \mathcal{C}_k} g$$

ne minimise l'erreur quadratique sur le cluster que lorsque la fonction F est linéaire en γ, ou plus précisément distributive par rapport à l'addition. En raison du caractère éventuellement non-linéaire de la correspondance entre un prototype et les objets, il est difficile de prévoir les effets de ce moyennage sur la valeur de l'erreur moyenne sur le cluster, comme cela est illustré sur la figure 4.4.

Dans l'absolu, le meilleur prototype est solution de la minimisation l'erreur quadratique moyenne sur le cluster, équivalente au problème

$$\min_{\gamma \in \mathbb{R}^\delta} \sum_{m \in \mathcal{C}_k} \min_{\theta_m} \|x_m - F(\gamma, \theta_m)\|^2 \tag{4.11}$$

La minimisation par rapport à θ_m est tractable, bien que relativement coûteuse, d'autant que cette minimisation devrait être effectuée un grand nombre de fois. La minimisation par rapport à γ étant de plus très difficile du fait de la non-linéarité de F, on peut considérer que ce problème n'est pas soluble en un temps raisonnable.

L'approche qui consisterait à prendre le barycentre des objets puis à calculer le grain correspondant en lui appliquant G, c'est-à-dire la transformation pseudo-inverse de F, n'est pas non plus justifiée, toujours du fait de la non-linéarité de F et donc de G.

Le médian fournit certes une solution en générale sous-optimale au problème 4.11, mais présente l'avantage majeur de ne pas entraîner de propagation de l'erreur au fur et à mesure des itérations de l'algorithme. Ceci découle directement directement de la définition

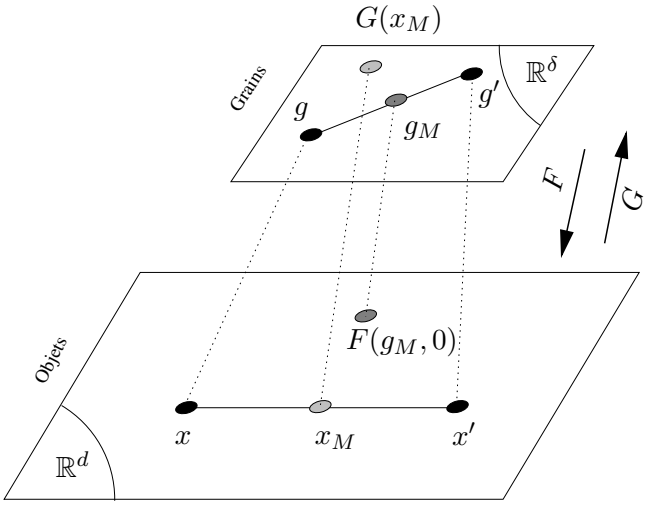


FIG. 4.4 – Exemple de fonction $F(\gamma, \theta)$ non linéaire par rapport à γ
x_M est la moyenne algébrique de x et x', g_M est la moyenne algébrique de g et g'. Quand F est non-linéaire par rapport g, on peut avoir $x_M \neq F(g_M, 0)$.

du médian, qui appartient nécessairement à l'ensemble des données observées.

Concluons en précisant qu'il pas exclu que, connaissant intégralement F, on puisse exploiter certaines de ses propriétés afin de mettre au point un algorithme efficace qui trouve des solutions meilleures que le médian.

4.2.4 Algorithmes à seuil(s)

Nous présentons ici deux algorithmes que nous avons mis au point en observant certaines particularités des matrices de similarité que nous avons étudiées. Ces algorithmes ont été conçus[4] pour exploiter une propriété que nous dénommons *quasi-transitivité*, qui a été vérifiée empiriquement sur les matrices de similarité des exemples de signaux étudiés.

L'idée est de constituer les classes progressivement, en comparant la valeur de la mesure de similarité à un seuil prédéfini.

L'erreur est (presque) transitive dans un espace métrique

Dans un espace métrique de dimension N muni d'une distance d, l'inégalité triangulaire est vérifiée, à savoir :

$$d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3) \ \forall x_1, x_2, x_3 \in \mathbb{R}^N$$

Ce résultat implique notamment que si on a $d(x_1, x_2) \leq \varepsilon$,$d(x_2, x_3) \leq \varepsilon$, on peut en

[4]Nous avons constaté *a posteriori* que ces algorithmes, de type *agglomératif*, ont un principe de fonctionnement proche de ceux présentés dans [AAPG92, ADPG92, APG94].

déduire que $d(x_1, x_3) \leq 2 \cdot \varepsilon$. De plus, ceci signifie que si l'on dispose d'un élément x_0 et d'un ensemble $\mathcal{C}$ de vecteurs de $\mathbb{R}^N$ tel que $d(x_0, x) \leq \varepsilon \; \forall x \in C$, on est assuré d'avoir $d(x, x') \leq 2 \cdot \varepsilon \; \forall x, x' \in \mathcal{C}$.

Ce dernier résultat est particulièrement intéressant dans l'optique du clustering. Ainsi, pour un ensemble de M vecteurs x_m, on peut construire un cluster tel que la distance entre deux éléments **quelconques** soit au plus $2 \cdot \varepsilon$, en effectuant seulement M comparaisons de distances.

A défaut d'exploiter cette propriété, il faudrait alors effectuer $M!$ opérations correspondant à la comparaison des valeurs de $d(x, x')$ avec ε pour toutes les paires non-ordonnées (x, x') possibles.

La question est maintenant de savoir si l'on peut obtenir une propriété analogue dans le cas granulaire.

Cas d'une mesure de similarité

On peut définir une propriété de *quasi-transitivité* par :

$$\exists \varepsilon_\Gamma \in \mathbb{R}^+ / \left|\Gamma(x_i, g_j) - \Gamma(x_1, g_2)\right| \leq \varepsilon_\Gamma \cdot \left|\Gamma(x_1, g_2) - \Gamma(x_3, g_2)\right| \\ \forall (x_i, g_j) \in \mathbb{R}^d \times \mathbb{R}^\delta, \{i, j\} \in \{1, 2, 3\} \quad (4.12)$$

La valeur de $\Gamma(x_1, g_2)$ permet de quantifier l'aptitude du modèle g_2 à décrire l'objet x_1, *i.e.* l'erreur de modélisation de x_1 par g_2. D'autre part, l'erreur de modélisation de x_2 par g_2 est par définition, nulle. Pour connaître l'erreur de modélisation par g_2 d'un troisième objet x_3, on calcule $\Gamma(x_3, g_2)$. La connaissance de $\Gamma(x_1, g_2)$ et $\Gamma(x_3, g_2)$ permet de caractériser l'erreur de modélisation de l'ensemble des objets par n'importe lequel des grains g_1, g_2 ou g_3, à un facteur multiplicatif ε_Γ près.

Au cours des expériences menées sur des signaux réels, on a pu observer que les matrices de similarité vérifient la propriété 4.12 avec $\varepsilon_\Gamma \simeq 1$. On a ensuite mis au point plusieurs variantes d'algorithmes exploitant cette propriété empirique de quasi-transitivité. On insiste à cette occasion sur le fait que les algorithmes présentés ci-dessous optimisent itérativement l'erreur relative sur un cluster, et non l'erreur absolue.

Algorithme à seuil simple 1S

Cet algorithme, présenté en détail à l'encart 3, nécessite de fixer au préalable la valeur du seuil de décision ε, comprise strictement entre 0 et 1

Le principe de base est assez simple : on choisit un grain de façon arbitraire, par exemple

Alg. 3 Algorithme à seuil simple 1S

- Initialiser l'ensemble des index des candidats $\mathcal{I} = \{1 \dots M\}$ et le nombre de clusters $K(0) = 0$
  L'ensemble $\mathcal{C}_k$ contient les index des objets affectés au prototype γ_k
- Itérer les étapes $i.1$,$i.2$ et $i.3$ tant que $\sharp\mathcal{I} > 1$

$i.1$ Sélectionner un candidat m_γ, *p.ex.* $m_\gamma = \mathcal{I}\{1\}$
Créer un nouveau cluster $\mathcal{C}_{K(i)+1} = \{m_\gamma\}$ avec $\gamma_{K(i)+1} = g_{m_\gamma}$

$i.2$ Pour $m \in \mathcal{I}$, $m \neq m_\gamma$, FAIRE
SI $\Gamma\left(x_m, g_{m_\gamma}\right) \geq \varepsilon$, FAIRE
$\mathcal{C}_{K(i)+1} \leftarrow \mathcal{C}_{K(i)+1} \cup \{m\}$ et $\mathcal{I} \leftarrow \mathcal{I} \setminus \{m\}$

$i.3$ FAIRE $\mathcal{I} \leftarrow \mathcal{I} \setminus \{m_\gamma\}$ et $K(i+1) = K(i) + 1$

le premier (g_1), comme prototype du premier cluster C_1, puis on parcourt l'ensemble des trames. Tous les objets x_m qui peuvent être décrits par le prototype $\gamma_1 = g_1$ avec une erreur relative inférieure ou égale à $\sqrt{1-\varepsilon^2}$ sont affectés au cluster C_1. Une fois la comparaison effectuée pour toutes les objets, on recommence avec les objets qui n'ont pas encore été assignés à un cluster, et on recommence jusqu'à ce que tous les objets soient classés.

Par construction, l'erreur **relative** de modélisation de n'importe quel objet est inférieure à $\sqrt{1-\varepsilon^2}$. Le seuil ε contrôle donc le « degré d'homogénéité » des clusters. Globalement, plus la valeur du seuil approche de 1, plus le nombre de clusters formés K est grand et plus l'erreur moyenne sur un cluster diminue.

Cet algorithme ne nécessite pas de fixer à l'avance la taille du dictionnaire K, et la valeur de K obtenue dépend uniquement de la valeur choisie pour ε.

Améliorations de l'algorithme à seuil : 2S

On peut apporter un certain nombre de raffinements par rapport à la version simple de l'algorithme, en introduisant :

- un seuil bas ε_0 , pour permettre un contrôle du degré d'homogénéité des clusters
- un seuil énergétique σ, pour éviter la sélection de grains correspondant à des trames faiblement énergétiques
- une taille minimale de cluster π_0 , pour éviter notamment la formation de clusters réduits à un singleton

Seuil bas ε_0

Comme on l'a vu au §4.2.4, dans le cas d'un espace d'un métrique, si l'on choisit un autre prototype que γ_k parmi $\mathcal{C}_k$, l'erreur moyenne sur ce cluster peut atteindre $2 \cdot \sqrt{1-\varepsilon^2}$, dans le pire des cas. L'introduction du seuil bas ε_0 ($0 < \varepsilon_0 < \varepsilon < 1$) permet un contrôle plus fin de la dispersion des erreurs de modélisation à l'intérieur d'un cluster. Pour ce faire, on vérifie pour chaque candidat m que tous les grains du cluster C_k déjà sélectionnés peuvent modéliser l'objet x_m avec une erreur relative inférieure à $\sqrt{1-\varepsilon_0^2}$.

L'utilisation de ce seuil doit permettre de réduire quelque peu la sensibilité de l'algorithme au choix des grains candidats, le caractère arbitraire de ce choix étant malheureusement imposé par le fonctionnement de l'algorithme.

Seuil énergétique σ

Il est également possible d'effectuer un pré-traitement au cours duquel on marque les objets x_m dont l'énergie est inférieure à un seuil σ, puis exclure les index correspondants de l'ensemble $\mathcal{I}$ des candidats.

En effet, en supposant que l'énergie du bruit de fond est constante dans le temps, il s'en suit que les objets de plus faible énergie sont également les plus bruités, et donc de mauvais candidats-prototypes.

D'autre part, ces objets sont vraisemblablement situés à proximité des frontières temporelles d'un évènement musical, *i.e.* une transition entre deux notes successives, et donc probablement dans une région non-stationnaire du signal.

L'utilisation de ce seuil constitue une tentative empirique d'améliorer le comportement de l'algorithme vis-à-vis de l'erreur absolue. Dans la même optique, il est envisageable de trier préalablement l'ensemble des candidats par ordre d'énergie décroissante, ce qui dispense du réglage d'un paramètre supplémentaire.

Taille minimale de cluster π_0

Le fait d'imposer une taille minimale aux clusters permet notamment d'éliminer les clusters singletons et de limiter la taille finale du dictionnaire K.

Avantages et inconvénients

Avec les algorithmes à seuils, le nombre de classes n'a pas besoin d'être connu à l'avance, contrairement au K-Médians par exemple. Le nombre de classes obtenu dépend alors des valeurs de seuils choisies, et naturellement, du signal traité.

Alg. 4 Algorithme à deux seuils 2S

- ▶ Initialiser l'ensemble ordonné des index des candidats $\mathcal{I} = \{1 \dots M\}$ et le nombre de clusters $K(0) = 1$
- ▶ *Pré-traitement :* Retirer de $\mathcal{I}$ les index m tels que $\|x_m\|_2 < \sigma$ et les marquer comme orphelins $\mathcal{O} = m \backslash \|x_m\|_2 < \sigma$
- ▶ Répéter les étapes $i.1$, $i.2$ et $i.3$ tant que $\sharp\mathcal{I} = 0$

  $i.1$ Sélectionner un candidat m_γ, *p.ex.* $m_\gamma = \mathcal{I}\{1\}$
  Créer le nouveau cluster $\mathcal{C}_{K(i)+1}(i) = \{m_\gamma\}$

  $i.2$ Pour $m \in I,\ m \neq m_\gamma$
  SI
  $$\max_{j \in \mathcal{C}_{K(i)+1}(i)} \Gamma(x_m, g_j) \geq \varepsilon \quad \text{ET} \quad \min_{j \in \mathcal{C}_{K(i)+1}(i)} \Gamma(x_m, g_j) \geq \varepsilon_0$$
  FAIRE
  $\mathcal{C}_{K(i)+1}(i) \leftarrow \mathcal{C}_{K(i)+1}(i) \cup \{m\}$

  $i.3$ SI $\sharp\mathcal{C}_{K(i)+1}(i) \geq \pi_{\min}$
  FAIRE
  $\mathcal{C}_{K(i)+1}(i+1) \leftarrow \mathcal{C}_{K(i)+1}(i)$, $K(i) \leftarrow K(i)+1$ et $\gamma_{K(i)} = g_{m_\gamma}$
  SINON
  Marquer m_γ comme orphelin : $\mathcal{O} \leftarrow \mathcal{O} \cup \{m_\gamma\}$ et $\mathcal{I} \leftarrow \mathcal{I} \setminus \{m_\gamma\}$

- ▶ *Post-traitement* : Affecter chacun des orphelins m au meilleur cluster $\mathcal{C}_{k_m}$, *i.e.* $\forall m \in \mathcal{O}, k_m = \arg\max_k \Gamma(x_m, \gamma_k)$

D'autre part, ces algorithmes peuvent fonctionner *en ligne* ou *au fil de l'eau*, ce qui est une caractéristique primordiale pour certaines applications telles que les télécommunications, où l'on ne peut se permettre d'attendre que l'intégralité du signal soit disponible avant de commencer le traitement. Pour que cela soit possible, il suffit de modifier l'algorithme pour qu'à un instant donné, les candidats ne soient choisis que parmi les grains effectivement disponibles à cet instant. L'algorithme doit attendre qu'un certain nombre M_0 de grains soient disponibles avant de commencer, sachant que le choix de la valeur M_0 effectue un compromis entre la latence[5] imputable à l'algorithme et la qualité du clustering. En effet, on peut tout à fait prendre $M_0 = 1$, avec pour conséquence que chaque nouveau grain sera sélectionné comme prototype supplémentaire, si sa similarité avec le dictionnaire déjà constitué est inférieure à ε, avec le risque de créer de nombreux clusters singletons. Le problème est donc de choisir une valeur de M_0 conduisant à la fois un temps de latence $M_0 \times \frac{d}{2 \cdot f_e}$ et à une dégradation acceptables des performances par rapport à la version hors-ligne de l'algorithme.

Dans sa version simple, l'algorithme construit des classes à l'intérieur desquelles l'erreur relative $\|e_m\|_2 / \|x_m\|_2$ est par construction inférieure à $\sqrt{1-\varepsilon^2}$, ε étant la valeur du seuil choisie. Le critère optimisé est donc l'erreur relative maximale sur un objet, et il n'est malheureusement pas possible de modifier l'algorithme afin de prendre en compte le critère que l'on souhaiterait véritablement optimiser, à savoir l'erreur globale sur le signal.

L'influence des différents paramètres de contrôle est par ailleurs difficile à expliciter, comme on a pu le constater au moment de réaliser les expériences. Nous n'avons ainsi pas été en mesure d'établir de relation simple et indépendante du signal traité entre les valeurs des paramètres de contrôle et les valeurs du couple erreur/complexité.

4.2.5 Algorithmes gloutons

Introduction

Les algorithmes cités jusqu'ici nécessitent de faire le choix préalable d'une taille de dictionnaire (K-Médians) ou de seuils. Les résultats obtenus dépendent également de l'initialisation du dictionnaire (K-Médians) ou de la procédure d'initialisation des clusters (algorithmes à seuils).

On souhaiterait pouvoir avoir un contrôle plus explicite sur la qualité de l'approximation, par exemple pouvoir spécifier la norme maximale de l'erreur.

Pour le K-Médians, il est possible de procéder à l'incrément de la taille K du dictionnaire jusqu'à l'obtention d'une norme de l'erreur inférieure à un seuil fixé à l'avance.

[5] délai de traitement imposé par l'algorithme, paramètre critique pour les applications de télécommunication, notamment dans le cas d'une transmission *full-duplex*, comme en téléphonie.

Il est également envisageable de lancer l'algorithme plusieurs fois avec une initialisation différente à chaque fois, puis garder le meilleur tirage.

L'influence de la valeur des seuils pour les algorithmes concernés est plus difficile à mettre en évidence. Théoriquement, on peut imaginer une procédure de recherche de(s) meilleur(s) seuil(s) en fonction d'une erreur de reconstruction maximale donnée. Ces procédures présentent cependant un inconvénient pratique, à savoir une augmentation significative du temps de calcul.

Nous avons alors tenté une autre approche du problème, à savoir la recherche de la valeur minimale de K permettant de reconstruire le signal avec une erreur maximale spécifiée. Nous allons maintenant présenter deux algorithmes itératifs que nous avons élaboré dans le but de répondre à cette question.

Ces algorithmes peuvent être qualifiés de *gloutons*, à savoir que l'algorithme sélectionne le meilleur candidat disponible à une étape donnée.

Algorithme glouton G1

Dans l'algorithme G1 détaillé à l'encart 5, les index des objets orphelins[6] sont également ceux des grains candidats, et on ajoute un prototype à chaque étape i ($K(i) = i$).

A chaque étape i, on choisit ($i.1$, $i.2$) comme représentant ou prototype, le grain qui minimise l'erreur moyenne sur l'ensemble des orphelins . On forme ensuite ($i.3$) la classe $\mathcal{C}_i$ en affectant les objets orphelins dont la similarité avec le prototype choisi précédemment est supérieure ou égale au seuil ε. On calcule enfin le rapport signal sur bruit (RSB) à l'étape courante et on vérifie que tous les grains n'ont pas encore été classés pour décider si l'on effectue ou non une itération supplémentaire.

En sortie de boucle, on affecte les grains éventuellement restés orphelins à leur plus proche voisin parmi l'ensemble des prototypes. Comme précédemment, on peut également choisir d'éliminer les clusters singletons, à l'intérieur ou à l'extérieur de la boucle.

Algorithme glouton G2

Cet algorithme est du type *divisif*, car chaque classe se voit divisée en deux nouvelles classes à chaque étape. A la différence de l'algorithme précédent, l'assignation $k(m)$ d'un objet x_m à un cluster $\mathcal{C}_k$ est réévaluée à chaque itération. Le dictionnaire est construit de manière incrémentale, c'est à dire qu'à chaque itération i, on rajoute un et un seul prototype γ_i au dictionnaire existant $D(i)$.

A chaque étape, on cherche un grain parmi l'ensemble des grains candidats. On compare les valeurs des erreurs[7] quadratiques moyennes $e^2(m, i)$, obtenues en effectuant à chaque

[6] *cf.* « Terminologie et notations », paragraphe 4.2.2.

[7] La minimisation porte sur $D(i) \cup g_m$, la sommation sur les index n des objets.

Alg. 5 Algorithme glouton G1

▶ `Initialiser l'ensemble des index des grains orphelins` $\mathcal{O} = \{1 \dots M\}$

▶ `FAIRE` $i = 1$ `puis aller en` $i.1$

$i.1$ `Calculer`

$$e^2(m, \mathcal{O}) \triangleq \sum_{n \in \mathcal{O}} \left[1 - \Gamma^2(x_n, g_m)\right] \forall m \in \mathcal{O}$$

$i.2$ `Rechercher l'index` $m_\gamma \triangleq \arg\min_{m \in \mathcal{O}} e^2(m, \mathcal{O})$ `du prototype` $\gamma_i \triangleq g_{m_\gamma}$ `du cluster` $\mathcal{C}_i$

$i.3$ `Former le cluster` $\mathcal{C}_i$

$$\mathcal{C}_i \triangleq m \in \mathcal{O} \quad \text{tq.} \quad \Gamma(x_m, \gamma_i) > \varepsilon$$

`Mettre à jour` $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{C}_i$

$i.4$ `Calculer le RSB` $\mu(i)$ `obtenu avec le dictionnaire` $D(i) = \{\gamma_k\}_{k=1\dots i}$
`SI` $\mu(i) < \mu_0$ `et` $\sharp\mathcal{O} > 0$ `FAIRE` $i \leftarrow i + 1$ `et retourner en` $i.1$
`SINON FIN`

▶ `Post-traitement : affecter chaque orphelin au meilleur cluster` $\mathcal{C}_{k_m}$, *i.e.*

$$k_m = \arg\max_k \Gamma(x_m, \gamma_k), \ \forall m \in \mathcal{O}$$

Alg. 6 Algorithme glouton G2

- ▶ `Initialiser l'ensemble` $\mathcal{I}$ `des index des grains candidats` $\mathcal{I} \triangleq \{1 \dots M\}$
- ▶ `Initialiser le dictionnaire` $D(0) = \emptyset$ `et faire` $K(0) = 0$

i.1 `Faire` $i \leftarrow i + 1$ `et calculer`

$$e^2(m, i) \triangleq \sum_{n=1}^{M} \min_{\gamma \in \{D(i) \cup g_m\}} \left[1 - \Gamma^2(x_n, \gamma)\right], \quad \forall m \in \mathcal{I}$$

i.2 `Chercher`

$$\widehat{m}(i) = \arg\min_{m \in \mathcal{I}} e^2(m, i)$$

i.3 `Le prototype du cluster` $\mathcal{C}_i$ `est` $\gamma_i = g_{\widehat{m}(i)}$

`Retirer l'index du prototype de l'ensemble` $\mathcal{I}$ `des index des grains candidats,` *i.e.* $\mathcal{I} = \mathcal{I} \setminus \{\widehat{m}(i)\}$

i.4 `Calculer le RSB` $\mu(i)$ `obtenu avec le dictionnaire` $D(i)$ `à l'étape` i

`SI` $\mu(i) < \mu_0$ ET $\sharp\mathcal{O} \neq 0$

`aller en` i.1

`SINON FIN`

fois la meilleure décomposition du signal sur le dictionnaire constitué de l'union

- ▶ du dictionnaire $D(i)$, à l'étape courante
- ▶ d'un des grains des candidats g_m

On sélectionne alors le grain minimisant cette erreur à l'étape donnée, pour l'ajouter au dictionnaire existant. On recommence ainsi jusqu'à ce que la condition d'arrêt soit remplie, si le RSB dépasse une valeur seuil ou le dictionnaire constitué atteint la taille maximale fixée.

A l'étape i, il faut calculer l'erreur obtenue avec chacun des $M - i$ candidats, ce qui représente environ $(M - i) \times M$ additions, et rechercher le minimum sur $M - i$ valeurs de l'erreur $e^2(m, i)$. Au total, la recherche d'un dictionnaire de taille K requiert donc environ $M \cdot \sum_{i=0}^{K-1} (M - i) \simeq M \cdot K \cdot (M - K)/2$ additions.

4.2.6 Considérations en rapport avec l'aspect calculatoire

Calcul approché de l'erreur

Au cours de l'exécution des algorithmes de clustering concernés, on doit calculer la valeur de la norme l'erreur quadratique moyenne

$$e_I(\gamma) \triangleq \sum_{m \in I} \left\| \left[x_m - F\left(\gamma, \widehat{\theta}_m\right) \right] \right\|^2$$

pour un ensemble d'index d'objets I et un prototype γ. En pratique, on calcule plutôt la valeur moyenne de la similarité au carré $\Gamma_I(\gamma) \triangleq \sum_{m \in I} \Gamma^2(x_m, g_{m_k})$, en effectuant une maximisation au lieu d'une minimisation. La somme correspondante étant évaluée de nombreuses fois, il peut être intéressant de réutiliser même partiellement les calculs déjà effectués. On peut pour cela utiliser la formule suivante :

$$\Gamma_J(\gamma) = \Gamma_I(\gamma) + \Gamma_{J \setminus I}(\gamma) - \Gamma_{I \setminus J}(\gamma)$$

Ceci n'a d'intérêt que dans le cas où $\mathrm{card}(I \cap J) \simeq \mathrm{card}(I)$, c'est-à-dire quand les ensembles I et J ont une majorité d'éléments en commun.

Calcul partiel de la matrice complète des similarités

Le calcul de la matrice des similarités est lourd, le nombre d'opérations requises croît en effet suivant N^2. Le temps de calcul devient prohibitif pour des valeurs de N supérieures à 1000, ce qui, pour une valeur de $f_e = 44100$ Hz typique, correspond à un signal d'une durée maximale d'une vingtaine de secondes.

Cette limitation diminue fortement l'intérêt pratique de la méthode proposée, à moins de trouver une solution pour diminuer le nombre de calculs requis. On a alors identifié deux voies d'amélioration possibles, à employer de préférence conjointement :

1. le calcul partiel de la matrice des similarités
2. la recherche de méthodes optimisées pour le calcul de la mesure de similarité.

La solution 1 consiste à partitionner la matrice en blocs puis à constituer un sous-dictionnaire pour chacun des blocs alignés sur la diagonale. On peut ensuite construire le dictionnaire complet en fusionnant les sous-dictionnaires, éventuellement en supprimant les « doublons », c'est à dire les prototypes qui sont très similaires.

Sur le même principe, on peut également envisager de réduire la dimension de l'espace de recherche dictionnaire en examinant d'abord les paires de trames consécutives. En effet, on observe expérimentalement que les trames consécutives sont fréquemment très

similaires. Sous réserve que le modèle choisi soit approprié, on s'attend à ce que toutes les paires parmi un ensemble de trames correspondant à une même note musicale conduisent à une similarité proche de 1.

La solution 2 consiste à rechercher les simplifications possibles de l'expression mathématique de Γ ainsi que les méthodes de calcul numérique approchées de cette même expression. L'efficacité de cette solution dépend bien évidemment du type particulier de modèle choisi, cet aspect sera donc évoqué au Cha.5.

En dernier lieu, mentionnons la possibilité d'éliminer les trames qui se situent au dessous d'un seuil de bruit prédéfini au moment de la constitution du dictionnaire.

4.3 Conclusion

La complexité du problème du calcul du modèle granulaire a obligé à un certain nombre de choix, parmi lesquels les plus marquants sont :

1) les objets sont obtenus à partir d'un découpage en trames du signal

2a) le dictionnaire est contraint à être une partition de l'ensemble des grains.

2b) les grains sont calculés à partir des objets par le biais de la fonction G

Ces choix peuvent apparaître comme des limitations, et on peut bien sûr imaginer d'autres possibilités. Par exemple, les point 2a et 2b impliquent que l'on dispose d'une fonction G qui joue le rôle d'inverse de F. Si l'on applique strictement le paradigme de l'analyse par la synthèse, ceci ne devrait pas être nécessaire. On pourrait par exemple tester un ensemble de grains situés sur une grille de $\mathbb{R}^\delta$ et conserver ceux qui minimisent l'erreur de reconstruction. Étant donné les dimensions mises en jeu ($\delta > 100$), ceci semble difficilement envisageable.

Nous avons consacré la suite de nos efforts à la recherche de fonctions de synthèse adaptées au problème considéré, qui sont présentés au chapitre suivant.

Chapitre 5

Étude spécifique de modèles

Ce chapitre est consacré à l'étude de plusieurs types de modèles granulaires. On présente les spécificités de chaque modèle, à savoir la forme choisie pour la fonction de synthèse ou de déformation F, et on expliquera la signification des q paramètres $\theta(1) \dots \theta(q)$ mis en jeu. On aborde également en détail l'estimation des paramètres et de l'apprentissage du dictionnaire.

Les modèles présentés ici font tous appel à un dictionnaire de « formes » spécialisées, chacune étant choisie pour représenter localement certaines parties du signal. Pour les modèles *Table d'ondes* (TO), les prototypes sont des formes d'ondes, c'est-à-dire des fragments de signaux de durée courte devant celle du signal analysé. Pour les modèles *Table de spectres*, les prototypes sont des profils de densité spectrales de puissance (DSP). A l'origine, les *Tables d'ondes* sont utilisées en synthèse musicale (*cf.* 1.2).

5.1 Modèles type Table d'ondes

5.1.1 Table d'ondes simple

Présentation du modèle

Les prototypes γ_k sont choisis parmi un dictionnaire de signaux appelé "table d'ondes" (TO), de même durée que les objets ($d = \delta$). Le prototype est fenêtré par W et mis à l'échelle par un facteur d'amplitude $\theta(1) = \alpha$.

L'expression générique du modèle $x_m = F(\gamma_{k_m}, \theta) + e_m$ devient alors :

$$x_m(t) = \alpha_m \cdot \gamma_{k_m}(t) + e_m \tag{5.1}$$

α_m amplitude de l'objet x_m

γ_k forme d'onde d'index k dans la table d'ondes

Estimation des paramètres

Pour un objet x isolé et un prototype donné $\gamma = w$, l'amplitude optimale $\widehat{\alpha}$ est simplement la longueur algébrique du projeté de x sur la droite engendrée par γ :

$$\widehat{\alpha}(x, \gamma) = \frac{\langle x, \gamma \rangle}{\|\gamma\|^2}$$

Une représentation d'un signal avec un modèle TO n'est ni plus ni moins qu'une variante particulière d'une décomposition d'un vecteur x_m sur une famille de vecteurs normés $D = \{\gamma_k\}_{k=1\ldots K}$, avec une contrainte sur la parcimonie des coefficients :

$$x_m = \sum_{k=1}^{K} a_{m,k} \cdot \gamma_k + e_m / \|a_m\|_0 = 1$$

Étant donné le modèle choisi, un objet x_m est représenté par sa projection sur un et un seul vecteur de base γ_{k_m}.

Représentation graphique de la matrice des similarités

La figure 5.1 est une représentation des valeurs de la matrice de similarité pour un signal de clarinette [Bou]. Chaque pixel de coordonnées (m, m') matérialise la valeur de $\Gamma(x_m, g_{m'})$, comprise entre 0 (blanc) et 1 (noir) . On appelle bloc un ensemble de couples d'index contigus

$$I_{m_1,m_2} \triangleq \left\{ \left(m, m'\right), m_1 \leq m, m' \leq m_2 \right\} / \Gamma(x_m, g_{m'}) \simeq 1$$

auquel correspond à un intervalle temporel (t_{m_1}, t_{m_2}) , à l'intérieur duquel les trames du signal sont toutes très similaires deux-à-deux.

Limitations du modèle

Une représentation avec le modèle TO correspond en réalité à une décomposition linéaire des trames du signal sur le dictionnaire D, avec une contrainte très forte sur la parcimonie des coefficients, à savoir qu'un seul coefficient peut être non nul.

Une deuxième limitation est que la fonction de synthèse F choisie ne permet de rendre compte que des variations d'amplitude des trames. Par exemple, pour un signal parfaitement sinusoïdal, de fréquence réduite f_0, un dictionnaire constitué d'un seul prototype ne pourra pas suffire à décrire le signal sans erreur. En effet, le découpage en trames induit, entre autres, un déphasage entre objets, dont la valeur dépend uniquement de la différence entre la longueur d'une période $\tau_0 = \frac{1}{2\pi \cdot f_0}$ du signal et de la position des instants de début

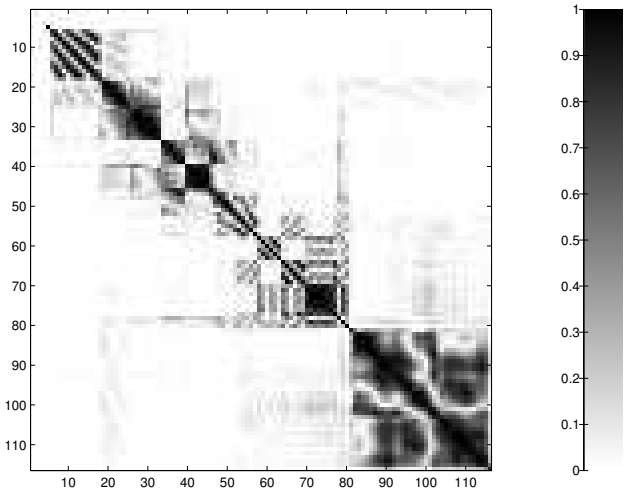


FIG. 5.1 – Exemple de matrice de similarité pour le modèle TO

Le signal à partir duquel est construite cette représentation est un enregistrement d'une mélodie constituée d'une dizaine de notes jouées successivement sur une clarinette. La matrice arbore globalement une structure par blocs. Le signal en question est en effet localement quasiment-périodique, sur des intervalles correspondant aux frontières, *i.e.* aux instants de début et de fin des notes. Les motifs en forme de vagues que l'on observe sur certains blocs sont vraisemblablement dûs à un phénomène d'interférences entre des signaux de fréquences proches mais de phases différentes. Ce phénomène est détaillé dans le texte.

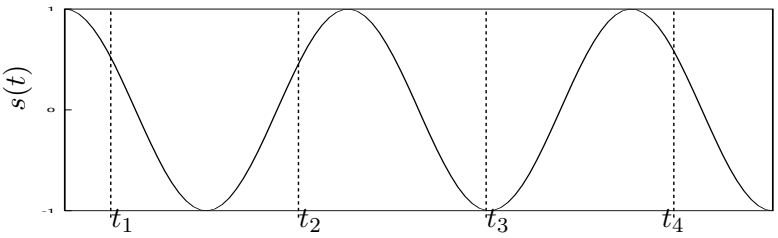


FIG. 5.2 – Découpage en trames non-synchrone par rapport à la période fondamentale du signal

des trames t_n. Ceci est illustré sur la figure 5.2.

Pour contourner ce problème, on pourrait envisager une approche type PSOLA[Pee98] en utilisant des fenêtres synchronisées par rapport à la période locale du signal. Cependant, d'autres problèmes se posent alors : comment élaborer une mesure cohérente pour comparer deux objets de longueur différente, comment reconstruire un objet à partir d'un autre de longueur plus courte ?

Nous avons choisi d'étendre le modèle TO, en conservant des objets de durées fixes et identiques. Le problème de la désynchronisation entre les frontières des objets et les périodes d'un signal périodique devra être résolu d'une autre manière que celle employée dans PSOLA.

5.1.2 Modèle table d'ondes avec décalage (TOΔ)

Présentation du modèle

Ce modèle reprend le précédent en ajoutant un facteur variable de décalage du prototype par rapport à la forme d'onde γ_k. L'idée est d'utiliser des prototypes de longueur supérieure à celle des objets et de permettre à la fenêtre utilisée à la synthèse de se déplacer autour d'une position de référence. Les paramètres utilisés sont $\theta(1) = \alpha$ et $\theta(2) = \Delta$.

$$x_m(t) = \alpha_m \cdot W(t) \cdot \gamma_{k_m}(t - \Delta_m) + e_m(t) \tag{5.2}$$

α_m amplitude de l'objet x_m

W fonction de fenêtrage (ou enveloppe) de support $[1 \dots d]$

γ_k forme d'onde d'index k dans la table d'ondes,
de support $[1 - \Delta_{max}, d + \Delta_{max}]$ avec $\Delta_{max} = \frac{\delta - d}{2}$

Δ décalage temporel du prototype par rapport à la forme d'onde γ_k
$-\Delta_{max} \leq \Delta \leq \Delta_{max}$

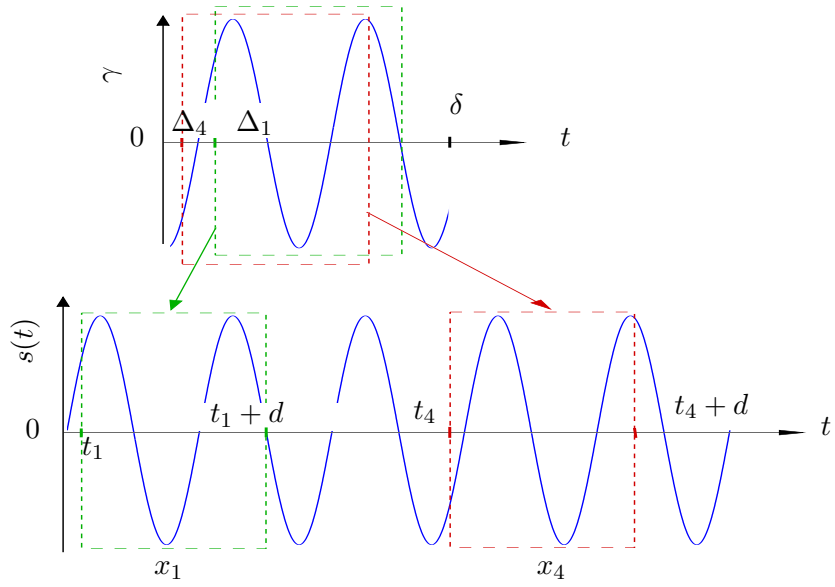


FIG. 5.3 – Recalage temporel du prototype par rapport à l'objet (TOΔ)
Pour reconstruire l'objet x_m, on multiplie le prototype γ par la fenêtre $W(t - \Delta_m)$, on translate le signal obtenu de t_m et enfin on multiplie par le facteur d'amplitude α_m.

Le paramètre Δ permet de choisir l'instant de début de la forme d'onde et correspond donc à une « phase initiale ».

Calcul des similarités

La mesure de similarité employée correspond au maximum du produit scalaire normalisé entre un objet/trame et l'ensemble des objets que l'on peut construire avec F et γ (en parcourant l'ensemble des décalages possibles),

Comme on l'a vu précédemment, le maximum par rapport au facteur d'amplitude α peut se calculer de de manière analytique comme précédemment :

$$\Gamma(x, \gamma) = \max_{\Delta} \frac{|\langle x(t), W(t) \cdot \gamma(t - \Delta) \rangle|}{\|x\| \, \|W(t) \cdot \gamma(t - \Delta)\|}$$

Le calcul de $\Gamma(x, \gamma)$ à l'aide de cette expression nécessite $2\Delta_{max}$ fois plus d'opérations que pour le modèle table d'ondes simple, ce qui motive la recherche d'un calcul optimisé. On remarque que la fonction

$$c(\Delta) \quad = \quad |\langle x(t), W(t) \cdot \gamma(t - \Delta) \rangle| \tag{5.3}$$

est également la convoluée circulaire des signaux $x(t) \cdot W(t)$ et $\gamma(-t)$, *i.e.*

$$c(\Delta) = |(x(t) \cdot W(t)) \star \gamma(-t)|$$

En effet, la somme sur t est limitée à l'intervalle $I_\delta = [-\Delta_{max}, d + \Delta_{max}]$, intervalle en dehors duquel les signaux $x(t) \cdot W(t)$ et $\gamma(t - \Delta)$ sont toujours nuls.

Réduction de la complexité calculatoire

Le terme $c(\Delta)$ peut donc être calculé à partir des transformées de Fourier[1] des signaux calculées sur l'intervalle I_δ, sachant que l'opération de retournement temporel se traduit par une conjugaison complexe dans le domaine de Fourier.

$$c(\Delta) = \left|\mathcal{S}^{-1} \cdot [\mathcal{S} \cdot (x(t) \cdot W(t)) \times \mathcal{S} \cdot \gamma(-t)]\right| \tag{5.4}$$

Le calcul de la matrice complète des similarités nécessite M^2 calculs de ce type, et en pratique, l'utilisation de la convolution permet de ramener les temps de calcul à des valeurs raisonnables.

Le calcul direct de $|\langle x(t), W(t) \cdot \gamma(t - \Delta)\rangle|$ fait intervenir $2\Delta_{max} \times d$ multiplications pour le calcul de $W(t) \cdot \gamma(t - \Delta)$, plus $2\Delta_{max} \times d$ multiplications-additions pour le calcul du produit scalaire pour l'ensemble des valeurs de Δ. Pour Mobjets, le nombre d'opérations est de l'ordre $4\Delta_{max} \times d \times M^2$.

Pour l'approche par convolution, on peut pré-calculer les TF $\mathcal{S}\cdot\ (x_m(t) \cdot W(t))$ et $\mathcal{S} \cdot \gamma_k(-t)$ en utilisant un algorithme de transformée de Fourier rapide, éventuellement en prolongeant avec des zéros pour atteindre une puissance de 2. La complexité algorithmique associée à ce calcul est en $\mathcal{O}(\delta' \log \delta' \times 2M)$ où δ' est la plus petite puissance de 2 supérieure à $\delta = d + 2\Delta_{max}$. Pour M objets, il faut d'abord multiplier M^2 TF (coût $\delta' \times M^2$) puis calculer les transformées inverses (coût $\delta' \log \delta' \times M^2$). Au total, la complexité par cette approche est en $\mathcal{O}\left(\delta' \log \delta' \times M^2\right)$; le gain en temps de calcul est donc d'autant plus important que Δ_{max} est grand.

Le calcul du dénominateur peut être effectué par l'entremise de la convolution. Mais on peut également choisir de négliger les variations de $\|W(t) \cdot \gamma(t - \Delta)\|_2$, ce qui revient à faire l'hypothèse que l'énergie est constante pour $\Delta \in [-\Delta_{max}, \Delta_{max}]$. En pratique, et en général, on constate effectivement que l'énergie varie lentement sur cet intervalle. Cependant, dans un souci de cohérence et afin de faciliter l'interprétation des résultats des expériences, on a implémenté le calcul exact dans nos programmes.

[1] $\mathcal{S}$ désigne l'opérateur transformée de Fourier discrète

Forme des grains

La solution la plus simple pour obtenir les grains g_m est de découper le signal en trames de longueur $\delta = d + 2\Delta_{max}$, calées sur les instants t_m qui déterminent la localisation des objets x_m. Dans ce cas, les grains sont définis par

$$g_m(t) = s(t + t_m),\ t \in [-\Delta_{max}, d + \Delta_{max}]$$

La valeur choisie pour Δ_{max} doit impérativement être strictement inférieure à $d/2$, pour la raison suivante : si $\Delta_{max} \geq d/2$, on a par construction $x_{m+1} = F(g_m, \theta \mid \Delta = d/2)$, autrement dit on modélise la donnée x_{m+1} par elle-même, ce qui n'offre aucun avantage par rapport à une représentation brute du signal.

Dans les expériences, nous avons fixé $\Delta_{max} = d/4$, et donc pour tous les modèles TOΔ :

$$\delta = 3 \cdot d/2 \tag{5.5}$$

A nombre de prototypes égal, le coût de description du dictionnaire est nécessairement plus élevé que pour le modèle TO simple. En contrepartie, on s'attend cependant à ce que l'erreur soit moindre grâce à la prise en compte du décalage. Les expériences menées montrent en effet qu'à valeur de complexité normalisée β constante et pour des signaux non-synthétiques, l'erreur est moindre avec le modèle TOΔ qu'avec TO.

Extrapolation des grains : modèle TO$\Delta^{\mathbf{x}}$

Une autre solution est d'extrapoler les bords gauche et droit de grains - de longueur $d + 2\Delta_{max}$ - à partir des trames - de longueur d. Nous présentons ici une méthode d'extrapolation des grains par prédiction linéaire bidirectionnelle que nous avons développée pour répondre à ce problème, indépendamment des travaux de I. Kauppinnen qui a proposé une approche identique [KR02, KK02] sur la plupart des points, pour la restauration d'enregistrements détériorés et l'augmentation de la résolution spectrale notamment. Naturellement, toute autre méthode telle que, par exemple [Cad79, CP91, JR81, Mah94, Pap75] devrait être également utilisable ici, bien que l'on ne les ait pas expérimentées, parce que la méthode proposée ici donnait des résultats jugés satisfaisants, et que son fonctionnement est à la fois simple à comprendre et à implémenter.

Principe A partir d'une trame s_m de durée d, on calcule deux filtres de prédiction d'ordre P : un prédicteur « avant » A_m^f pour le signal s_m direct et un prédicteur « arrière » A_m^b pour la retournée temporelle de s_m. Les prédicteurs optimaux avant et arrière sont

obtenus en minimisant [2] la norme L_2 des erreurs de prédiction correspondantes, définies respectivement par :

$$\left\{ \begin{array}{lcl} e_m^f(t) & = & \sum_{p=0}^{P-1} A_m^f(p)\, s_m(t-p) \\ e_m^b(t) & = & \sum_{p=0}^{P-1} A_m^b(p)\, s_m(p-t) \end{array} \right. \quad \forall t \in [1 \dots d/2]$$

Le rôle de l'extrapolation est ici de diminuer la complexité de la description du dictionnaire, et non pas de reconstruire des données manquantes. Précisément, on a bien accès aux données exactes de s_m sur les intervalles $[-\Delta_{max}+1 \dots 0]$ et $[d+1 \dots d+\Delta_{max}]$ dans la phase de construction du dictionnaire, mais on ne veut pas incorporer celles-ci au dictionnaire. Au moment de la reconstruction, on remplace donc les signaux e_m^f et e_m^b par les *excitations* de synthèse $\tilde{e}_m^f$ et $\tilde{e}_m^b$ respectivement. Pour l'extrapolation à droite du signal, *i.e.* $g_m(t), t \in [d+1 \dots d+\Delta_{max}]$, on applique le filtre inverse $\left(A_m^f\right)^{-1}$ à l'excitation « avant ». Pour le bord gauche, *i.e.* $g_m(t), t \in [-\Delta_{max}+1 \dots 0]$, on applique le filtre $\left(A_m^b\right)^{-1}$ à l'excitation arrière puis on effectue un second retournement du temps pour compenser le premier.

Les signal d'erreur exacts sont disponibles pour $t \in [1 \dots d]$, et peut utiliser simplement $\tilde{e}_m^f(t) = 0 \forall t \in [d+1 \dots d+\Delta_{max}]$ et $\tilde{e}_m^b(t) = 0 \quad \forall t \in [-\Delta_{max}+1 \dots 0]$. L'utilisation de réalisations de bruit blanc de variances identiques aux variance estimée sur les intervalles $[d+1 \dots d+\Delta_{max}]$ et $[-\Delta_{max}+1 \dots 0]$ donne expérimentalement une erreur d'extrapolation supérieure à celle obtenue avec l'excitation nulle. D'après les quelques tests que l'on a pu mener, il semblerait que les résultats obtenus avec cette méthode soient meilleurs qu'avec l'implémentation de la méthode proposée dans [KR02].

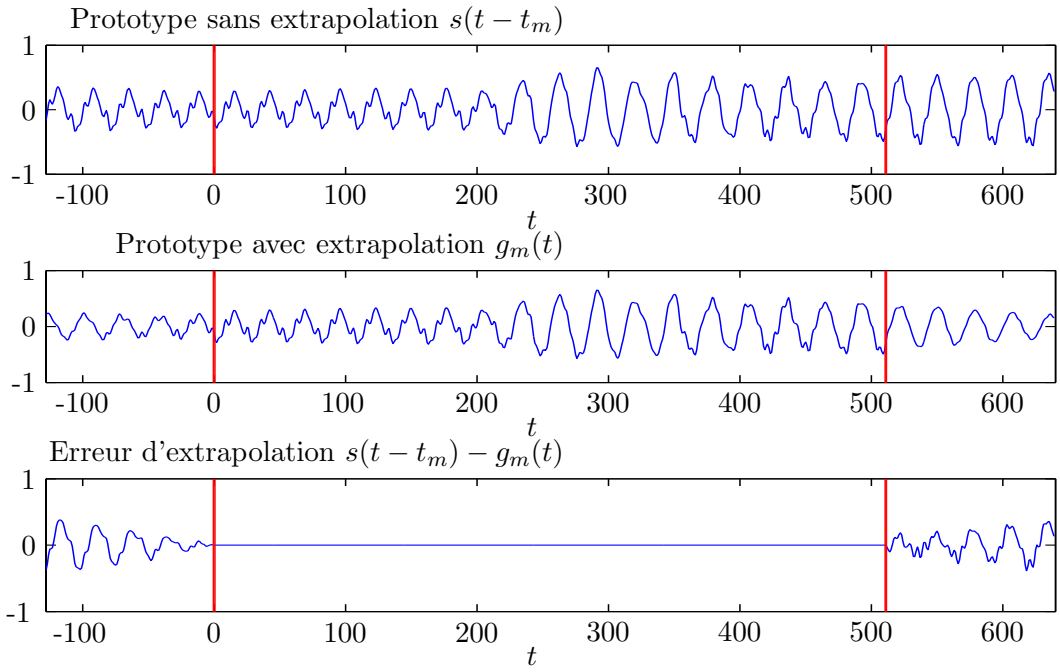
On peut voir sur la figure 5.4 deux exemples commentés de grains extrapolés qui correspondent à deux cas de figure assez typiques.

Recherche de Δ accélérée

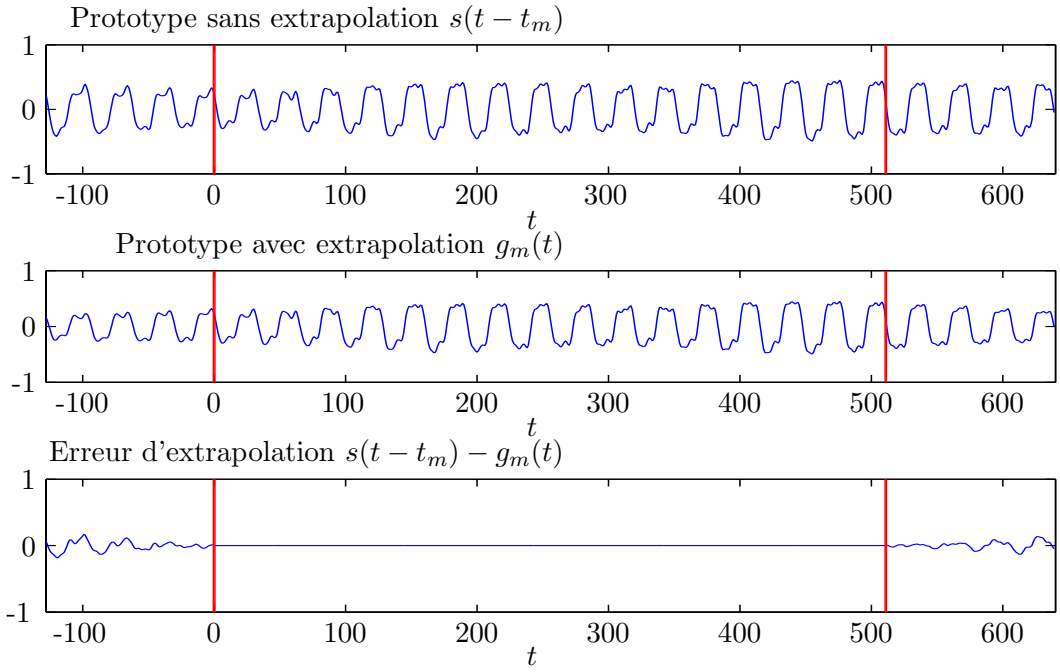
Dans le but d'alléger le calcul des similarités, il serait utile de disposer d'une méthode approchée pour estimer le décalage optimal au lieu d'effectuer une recherche exhaustive de $\widehat{\Delta}$ sur l'ensemble des entiers $[-\Delta_{max} \dots \Delta_{max}]$.

Par exemple, si l'on a déjà calculé i_0 valeurs de $c(\Delta)$ correspondant à i_0 décalages différents, on peut décider d'interrompre la recherche en fonction de la valeur du maximum partiel $c\left(\widehat{\Delta}_{i_0}\right)$ déjà obtenue. Pour une valeur du maximum partiel à $1-\epsilon$, l'erreur commise par rapport à une recherche exhaustive sur les valeur prises par $c(\Delta)$ est au plus de ϵ. Si ϵ est petit devant 1, on peut raisonnablement choisir de négliger cette erreur. A l'inverse, si le maximum est ϵ avec ϵ petit devant 1, on ne peut pas minorer la valeur de

[2] L'algorithme de Levinson-Durbin est utilisé pour calculer les prédicteurs optimaux.



(a)



(b)

FIG. 5.4 – Exemples de grains extrapolés, $d = 512$, $\Delta_{\max} = d/4$, $P = d$.
Les parties du signal extrapolées sont à gauche et à droite de la partie centrale, délimitée par les traits verticaux. Dans l'exemple (a), l'extrapolation est relativement mauvaise car la période fondamentale du signal change visiblement en $t \simeq 220$, alors qu'elle est correcte en (b) où la forme d'onde du signal paraît stable. On observe que la continuité de la forme d'onde est préservée par l'extrapolation, ce qui est très important d'un point de vue perceptif, toute discontinuité étant immédiatement décelable à l'oreille, sous la forme d'un « clic » ou d'un craquement particulièrement désagréable.

l'erreur sur Γ sans faire d'hypothèses supplémentaires sur x et γ.

En utilisant l'égalité de Parseval, on peut écrire l'équation 5.3 sous la forme d'un produit scalaire de spectres

$$c(\Delta) = |\langle \mathcal{S} \cdot x(t), \mathcal{S} \cdot (W(t) \cdot \gamma(t - \Delta)) \rangle|$$

Sachant que la translation circulaire se traduit par une modification du spectre de phase dans le domaine de Fourier (cf.Annexe), et en utilisant l'inégalité de Cauchy-Schwartz, on montre que

$$|c(\Delta)| \leq \langle |\mathcal{S} \cdot x(t)|, |\mathcal{S} \cdot (W(t) \cdot \gamma(t))| \rangle \, \forall \Delta \in [-\Delta_{max} \ldots \Delta_{max}] \tag{5.6}$$

On note $c_{\max}$ la borne supérieure de $c(\Delta)$ fournie par l'inégalité précédente. La valeur de $c_{\max}$ peut alors être utilisée comme critère de sélection d'une procédure de recherche du maximum de $c(\Delta)$.

Par exemple, si $c_{\max}$ est proche de 0, on peut décider de ne pas rechercher la valeur exacte de $\max_\Delta c(\Delta)$ et, d'utiliser la valeur $c(0)$. On peut également compléter ou remplacer ce critère d'arrêt par le suivant : arrêter la recherche dès qu'on a trouvé une valeur de Δ telle que $c(\Delta) \simeq c_{\max}$. En pratique, on fixerait un seuil ϵ pour d'interrompre la recherche dès que $|c(\Delta) - c_{\max}| \leq \epsilon$.

5.1.3 Coût de description du dictionnaire

Pour décrire un dictionnaire composé de K prototypes de longueur δ, on a au plus besoin de connaître $\chi_D = K \times \delta$ valeurs scalaires. En pratique, ces valeurs scalaires seraient nécessairement quantifiées, ce qui demanderait d'étudier les effets de la quantification sur la qualité de la représentation ainsi que les questions liées à la dynamique [3] des valeurs des vecteurs des prototypes. L'utilisation d'un codeur entropique permettrait éventuellement une réduction supplémentaire du coût de la description. Bien sûr, dans le cas où l'on chercherait à évaluer les performances du modèle en codage, on ne pourait se dispenser de cette étape. Ceci dit, l'utilisation d'un scalaire comme unité de base permet d'éviter d'aborder les questions liées à la quantification et rend les comparaisons entre différents modèles plus aisées.

[3] Le fait que les prototypes soient normalisés n'implique pas nécessairement que l'amplitude des valeurs des composantes des γ_k soit limitée.

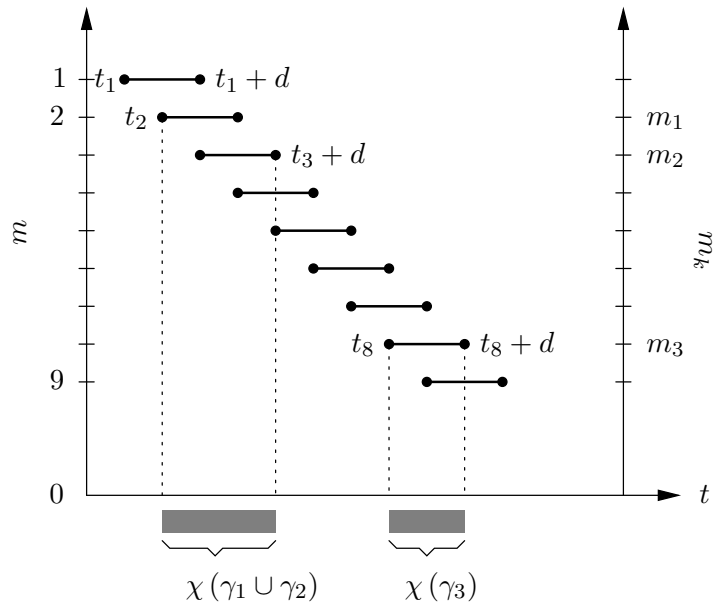


FIG. 5.5 – Segments du signal intervenant dans la construction du dictionnaire
Le dictionnaire est ici constitué de trois prototypes, dont deux sont contigus (γ_1 et γ_2).

Cas de grains contigus

Les données de deux prototypes γ_1, γ_2 contigus (c'est-à-dire tels que $\gamma_1 = g_{m_1}, \gamma_2 = g_{m_2}$ avec $m_2 = m_1 + 1$) , sont partiellement redondantes puisque celles-ci sont obtenus à partir de vecteurs qui ont en commun certains composantes. Au lieu de stocker les valeurs de γ_1 et γ_2 comme si celles-ci étaient indépendantes, il est donc plus efficace de stocker les valeurs du signal sur l'intervalle de temps correspondant, puis d'appliquer les transformations nécessaires pour obtenir les grains. La figure 5.5 montre un exemple de configuration de grains où ceci permet de diminuer l'espace occupé par le dictionnaire.

Pour le modèle TO simple, le coût de description de deux prototypes contigus est ainsi $\chi(\gamma_1, \gamma_2) = t_{m_2} - t_{m_1} = 3d/2$ et non $2 \cdot d$.

Pour tenir compte de ces redondances, le coût du dictionnaire est défini comme la longueur du support des grains-prototypes constitutifs de ce dictionnaire. Cela implique que le coût de description ne dépend pas uniquement du type de modèle et du nombre des prototypes, mais également de la localisation des grains dans le signal.

Coûts comparés des différents types de modèles TO

Le coût de description du dictionnaire avec le modèle TOΔ^x extrapolé est identique à celui du modèle TO simple, à nombre égal de prototypes. En supposant que les grains sélectionnés en tant que prototypes sont pris aux mêmes endroits, l'utilisation du modèle TO Δ^x conduit par construction à une erreur de reconstruction inférieure à celle obtenue

avec TO.

La comparaison des modèles TOΔ avec et sans extrapolation est plus difficile : l'extrapolation permet un gain en termes de complexité du dictionnaire, au prix d'une erreur aux bords. Les résultats expérimentaux semblent toutefois indiquer que le modèle TOΔ est plus performant avec des grains extrapolés, ce qui nous conduit à penser que l'erreur de prédiction aux bords est faible.

5.1.4 Conclusion : vers une généralisation du modèle TOΔ

On propose de généraliser les approches TO présentées précédemment, avec un modèle nommé TO+F pour Tables d'Ondes Filtrées.

$$x_m(t) = \alpha_m \cdot W(t) \cdot [B_m \star \gamma_{k_m}(t)] + e_m(t) \tag{5.7}$$

La déformation consiste en un filtrage convolutif (par exemple) par B_m suivi d'un fenêtrage par W.

B_m réponse impulsionelle du filtre convolutif, d'ordre $p = q - 1$, avec $B_m(1) = 1, \quad \theta_m(i) \triangleq B_m(i+1)$.

α_m facteur d'amplitude

W fonction de fenêtrage de support $[1 \dots d]$

γ_k forme d'onde d'index k dans la table d'ondes

Le rôle du filtre est double : d'une part, compenser les décalages de forme d'ondes entre objet et prototype, et compenser les variations de timbre d'autre part. Le premier point a été traité avec le modèle TOΔ, la réponse impulsionelle du filtre correspondant étant simplement $\delta(t - \Delta_m)$. Pour prendre en compte le second point, nous proposons de décomposer le prototype en n_b sous-bandes, auxquelles on appliquerait individuellement une correction d'amplitude et un décalage temporel. Ce principe peut par exemple être implémenté, comme indiqué sur la figure 5.6, à l'aide d'un banc de filtres type QMF ou autre [Ngu95], suivi d'une décimation, puis application d'un délai par bande.

Ou plus simplement, et ceci fait le lien avec la section suivante, on peut modéliser les variations de la TF de l'objet par rapport à celle du prototype avec un paramètre d'amplitude et de décalage temporel pour chaque sous-bande :

$$x_m(t) = W(t) \cdot \mathcal{S}^{-1}[\alpha_m(f) \cdot \mathcal{S}\gamma_{k_m}(t) \star \Delta(f)] + e_m(t)$$

$\alpha_m(f)$ modèle des variations fréquentielles d'amplitude

$\Delta_m(f)$ modèle des variations fréquentielles du décalage

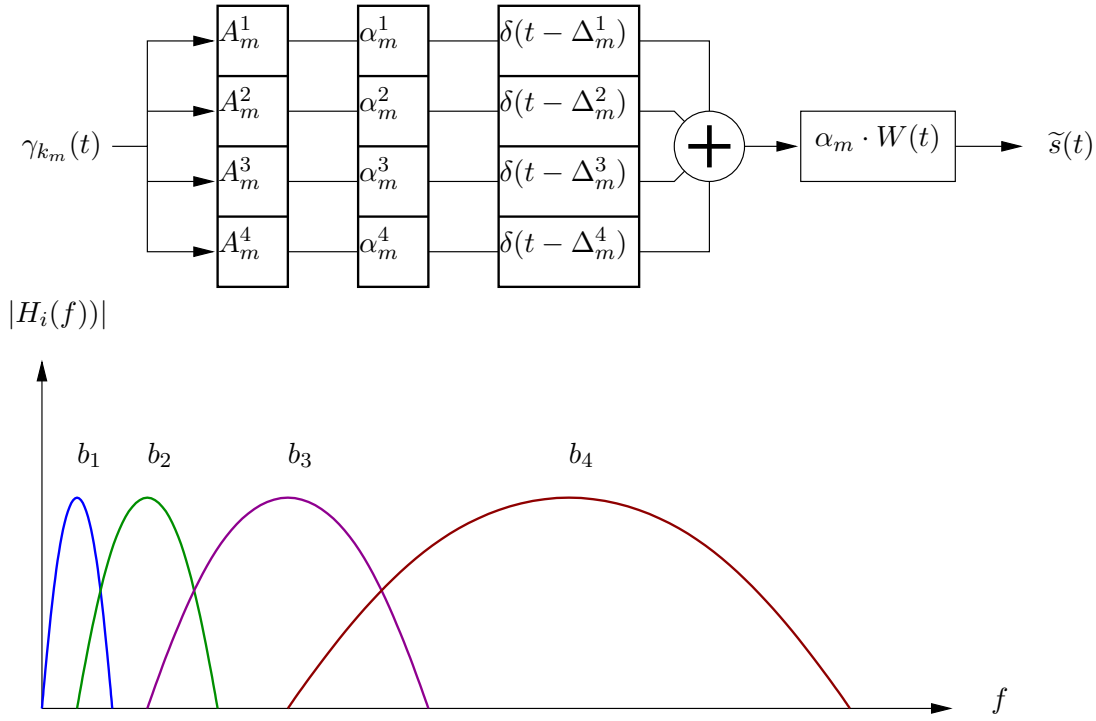


FIG. 5.6 – Implémentation du modèle TOF par banc de filtres (en octaves) et fonctions de transfert

Les fonctions précédentes peuvent être choisies parmi la classe des polynômes de f de degré égal au nombre de sous-bandes, ou plus simplement des fonctions constantes par morceaux sur les intervalles de fréquence du découpage en sous-bandes.

5.2 Modèles type tables de spectres

On s'intéresse maintenant à une autre classe de modèles granulaires, où le dictionnaire est une collection de profils de densités spectrales, et un objet est synthétisé à partir d'un prototype de DSP. Le principe général reste similaire à celui utilisé précédemment :

1. calcul des grains (DSP des objets)
2. constitution d'un dictionnaire de prototypes à partir des grains
3. re-synthèse des objets à partir du dictionnaire

Le calcul des grains ne pose pas de problème particulier : l'estimation de la DSP à partir d'un signal supposé stationnaire est une question largement couverte par la littérature. Le choix de la mesure de distance/similarité peut être fait *a priori* parmi la grande variété de mesures de distance spectrale existantes : euclidienne, logarithmique, cepstrale, Itakura-Saïto, rapport de vraisemblance, COSH, perceptives [RS78] ... Cependant, la seule donnée de la DSP d'un objet ne caractérise pas totalement celui-ci, que ce soit en termes

mathématiques ou perceptifs. L'influence du spectre de phase sur la perception auditive, bien que certaine, est difficile à caractériser et donc à modéliser.

Aussi l'étape de re-synthèse des objets est-elle problématique, car l'on voudrait éviter de spécifier intégralement le spectre de phase de l'objet en tant que paramètre, sachant que le coût associé au spectre de phase est $d/2$. Afin de diminuer ce coût, il est donc nécessaire de modéliser le spectre de phase, ou plus exactement les variations du spectre de phase de l'objet par rapport à celui de son prototype.

Les paramètres du modèle sont estimés de manière à minimiser l'erreur de reconstruction de l'objet. L'utilisation d'un critère d'erreur perceptive serait souhaitable, particulièrement dans le cadre d'une application au codage de signaux sonores. Par souci de simplicité, on utilisera ici l'erreur quadratique.

5.2.1 La hantise du Spectre de phase

On admet généralement que, lorsque le signal est stationnaire, la densité spectrale de puissance suffit à caractériser entièrement le son perçu par l'auditeur . Dans le cas non-stationnaire, le spectre de phase a au contraire une importance cruciale bien que relativement peu étudiée, tout du moins à notre connaissance [Pat87].

Voyons à partir d'un exemple, bien que quelque peu simpliste, pourquoi le spectre de phase ne peut être négligé dans le cas où l'on envisage le cas non-stationnaire. Une impulsion très brève, modélisée par un Dirac en temps discret, et la réalisation d'un bruit blanc i.i.d sont deux signaux possédant des DSP analogues, et ne se différencient que par la structure de leur spectre de phase.

Le premier est un signal dont l'énergie est concentrée sur un intervalle de temps très court, ce qui lui vaut le qualificatif de transitoire. Le spectre de phase d'un Dirac est une fonction linéaire de la fréquence, et il s'agit d'une structure déterministe. A l'inverse, le spectre de phase d'une réalisation d'un bruit stationnaire [4] est lui-même une réalisation d'un processus aléatoire.

La différence auditive entre ces deux signaux est pourtant patente, ce qui nous amène à penser que le spectre de phase a une importance certaine pour des signaux non-stationnaires.

Par ailleurs, si l'on effectue la transformée de Fourier à court-terme (TFCT) d'un signal sinusoïdal et qu'on effectue ensuite une TF inverse, il est primordial de restituer les relations de phase entre les spectres de deux trames consécutives, sous peine d'introduire des discontinuités artificielles dans le signal re-synthétisé.

Concluons en indiquant qu'un certain nombre d'études [JP78, J.C57, Moo77, PS69] montrent que les relations de phase entre les différentes composantes d'une structure har-

[4] Le spectre de phase d'un processus aléatoire n'est pas défini de cette manière, mais rien n'interdit de prendre la TF d'une réalisation de ce processus.

monique sont perceptivement peu significatives, bien que pas totalement négligeables pour autant.

5.2.2 Structure générale du modèle

Dans ce modèle, un prototype γ est un profil de spectre, ou plus simplement un vecteur à composantes positives ou nulles, supposé normé.

En plus d'un facteur d'amplitude globale $\theta_m(1) = \alpha_m$, les paramètres $\theta_m(2) \dots \theta_m(q)$ concernent le modèle $\Phi_m(f)$ du spectre de phase pour l'objet sonore x_m.

$$x_m(t) = \alpha_m \cdot S^{-1}\left[\gamma_k(f) \cdot e^{j\Phi_m(f)}\right](t) + e_m(t) \tag{5.8}$$

α_m paramètre (positif) d'amplitude de l'objet

$\gamma_k(f)$ forme spectrale d'index k, choisi parmi un « dictionnaire de formes spectrales »$(\gamma_k \in \mathbb{R}^{+\delta})$

$\Phi_m(f)$ modèle du spectre de phase pour l'objet $x[m]$, $\Phi_m(f) \in [0, 2\pi]$

Contraintes sur les spectres

Le signal d'analyse étant toujours réel, les objets sonores sont également réels, ce qui impose certaines contraintes sur leur spectre : positivité et symétrie par rapport à la fréquence nulle, anti-symétrie et nullité à la fréquence nulle du spectre de phase. En temps discret et pour une TFD sur un intervalle fini, ces contraintes s'expriment sous la forme :

$$\begin{aligned} \gamma(f) &\geq 0 \quad \forall f \in [0 \dots d-1] \\ \gamma(f) &= \gamma(d-f) \quad \forall f \in [0 \dots d-1] \end{aligned}$$

$$\Phi_m(f) = \begin{cases} -\Phi_m(d-f) & \forall f \in [1 \dots d/2] \\ \Phi(0) \equiv 0\,(\pi) \\ \Phi\left(\frac{d}{2}\right) \equiv 0\,(\pi) \end{cases}$$

En outre, afin de simplifier les calculs, on suppose la DSP normalisée *i.e.* :

$$2 \cdot \sum_{f=0}^{d/2-1} \gamma^2(f) + \gamma^2\left(\frac{d}{2}\right) \triangleq 1$$

Le cas échéant, il suffit de diviser la DSP par sa norme $\mathcal{L}_2$ pour se ramener à ce cas particulier.

5.2.3 Modélisation du spectre de phase

Introduction

Le coût de description d'un prototype avec le modèle TS est $d/2$, et la description des paramètres liés au modèle de phase Φ_m requiert alors $q-1$ valeurs par objet[5]. Si on laisse Φ_m libre, on a $q = d/2+1$, et $\beta \geq \frac{1}{2}$ quelle que soit la taille du dictionnaire[6], c'est à dire que le modèle est très peu efficace.

Il est donc souhaitable de réduire le coût de description associé au spectre de phase de chaque objet. On a envisagé deux possibilités : modéliser les dépendances entre les valeurs du spectre de phase à des fréquences différentes pour un objet donné d'une part, et modéliser les dépendances entre les phases de deux objets différents d'autre part. La première option est qualifiée de *verticale*, car on modélise le spectre à un instant donné, tandis que la seconde est dite *horizontale*, car on modélise les variations temporelles de la phase pour une fréquence donnée.

Pour un objet x_m donné, on veut minimiser la norme L_2 de l'erreur $e_m = x_m - F(\gamma_{k_m}, \theta_m)$

ce qui, d'après l'égalité de Parseval, équivaut à chercher les solutions [7] du problème :

$$\left(\widehat{\alpha}_m, \widehat{\underline{\theta}}_m\right) = \arg\min_{\alpha_m, \theta_m} \left\| \mathcal{S}x_m(f) - \alpha_m \gamma_k(f) \cdot e^{j\Phi_m(f)} \right\|^2 \tag{5.9}$$

Après avoir posé $X_m \triangleq |\mathcal{S}x_m|$ et $\phi_m \triangleq \arctan \mathcal{S}x_m$ l'erreur quadratique s'écrit :

$$\|e_m\|^2 = X_m^2 + \alpha_m^2 \cdot \|\gamma_{k_m}\|^2 - 2\alpha_m \cdot \sum_f X_m(f) \cdot \gamma_{k_m}(f) \cdot \cos\left(\phi_m(f) - \Phi_m(f)\right)$$

L'amplitude optimale $\widehat{\alpha}_m$ se calcule explicitement en annulant la dérivée partielle correspondante :

$$\widehat{\alpha}_m = \frac{\sum_f X_m(f) \cdot \gamma_{k_m}(f) \cdot \cos\left(\phi_m(f) - \Phi_m(f)\right)}{\|\gamma_{k_m}\|^2}$$

Le calcul des paramètres optimaux $\widehat{\underline{\theta}}_m$ du modèle de phase Φ_m est plus difficile du fait que l'erreur n'est pas linéaire en Φ_m d'une part, et que chaque composante du vecteur ϕ_m n'est connue qu'à 2π près.

---

[5] Pour rappel, q est le nombre total de paramètres du modèle, le premier étant l'amplitude α.

[6] $\beta = \frac{K}{N}$ est le gain en termes de complexité apporté par la modélisation (*cf.* 3.3.3)

[7] Pour rappel, $\underline{\theta} \triangleq [\theta(2), \ldots, \theta(q)]$ est le vecteur contenant les paramètres du modèle autres que l'amplitude.

Modélisation verticale

Dans un premier temps, on a envisagé de modéliser l'écart entre les spectres de phase ϕ_m de l'objet et ϕ_{k_m} du grain-prototype $\gamma_k \triangleq g_{m_k}$ des objets par une fonction linéaire de la fréquence. Ce modèle utilise un paramètre $\tau \triangleq \theta(2)$ dépendant explicitement de m et un spectre de phase de référence ϕ_{k_m} :

$$\begin{cases} \Phi_m(f) &= \phi_{k_m}(f) + 2\pi\frac{\tau}{d}\cdot f \quad \forall f \in \left[0 \ldots \frac{d}{2} - 1\right] \\ \Phi_m\left(\frac{d}{2}\right) &= \phi_{k_m}\left(\frac{d}{2}\right) = 0 \text{ ou } \pi \end{cases}$$

Pour estimer le décalage optimal τ, en supposant que $X_m \propto \gamma_{k_m}$, procéder comme suit :

1. Calcul de la phase de l'objet $\phi_m = \arctan\left[\mathrm{Im}\left(\mathcal{S}x_m\right)/\mathrm{Re}\left(\mathcal{S}x_m\right)\right]$, $\phi_m(f) \in [0 \ldots \pi]\, \forall f$ et de celle de son prototype $\phi_{k_m} = \arctan\left[\mathrm{Im}\left(\mathcal{S}x_{m_k}\right)/\mathrm{Re}\left(\mathcal{S}x_{m_k}\right)\right]$
2. Déroulement de la phase, *i.e.* chercher un vecteur $\mathbf{b} \in \mathbb{N}^d$ tel que $\phi_m - \phi_{k_m} + 2\pi \cdot \mathbf{b}$ présente une « régularité » maximale. [8]
3. Régression linéaire sur les valeurs du spectre de phase déroulé

A noter que l'on ne trouve pas ainsi une solution exacte du problème 5.9. En effet, la procédure d'estimation présentée ci-dessus suppose implicitement $X_m \propto X_{m_k}$ d'une part, et elle optimise la somme pondérée des $|\phi_m(f) - \Phi_m(f)|^2$ au lieu de $\cos(\phi_m(f) - \Phi_m(f))$.

Pour des valeurs entières de τ, la définition du modèle équivaut à supposer que x_m s'obtient à partir de x_{m_k} en effectuant un décalage de τ vers la droite, avec une permutation circulaire modulo d aux bords, *i.e.* $x_{m_k}(t) = x_m(t + \tau \equiv d)$. A moins que le signal étudié admette d pour période, la permutation circulaire introduit inévitablement une discontinuité en $t = \tau$ du fait que $x_m(\tau) \neq x_m(d + \tau \equiv d)$.

Par exemple, si s est une sinusoïde de fréquence f_0 échantillonnée à la fréquence f_e, notons $\Omega_0 = 2\pi f_0/f_e$ la pulsation réduite correspondante. L'amplitude de la discontinuité du signal modifié en $t = \tau$ est $|s(d) - s(0)| = |\sin(\Omega_0 d)|$, et elle est maximale quand f_0 est située à mi-chemin entre deux multiples de la résolution de Fourier pour la durée de fenêtre choisie, *i.e.* pour $f_0 = [0.5 + p] \cdot \frac{f_e}{2d}$, $p \in \mathbb{N}$. L'intérêt pratique de ce modèle est fortement limité par la présence d'une telle discontinuité dans les objets reconstruits.

[8] Par facilité, nous avons utilisé l'algorithme de déroulement de phase intégré dans Matlab^{TM} ; il serait intéressant d'expérimenter plusieurs méthodes alternatives telles que [Tri77].

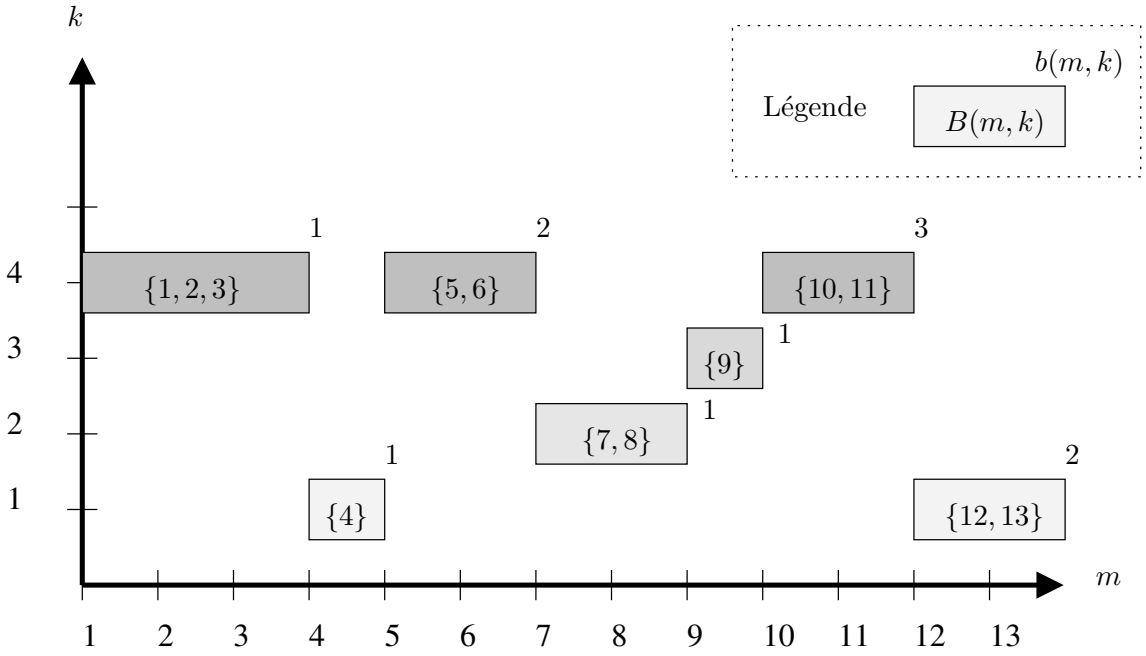


FIG. 5.7 – Découpage des clusters en blocs d'objets contigus
$b\,(m,k)$ désigne le numéro du bloc du cluster $\mathcal{C}_k$ auquel appartient l'objet x_m
$B\,(m,k)$ contient l'ensemble des index des objets du bloc $b\,(m,k)$

Modélisation horizontale

Dans ce cas, on modélise les dépendances du spectre de phase par rapport au temps, ou ce qui revient au même, à l'index m de l'objet. On fait l'hypothèse que l'évolution dans le temps de la phase à une fréquence donnée est relativement lente et prévisible, et ceci tant que les objets restent dans la même classe k_m.

Ceci implique que l'on change de modèle dès qu'on détecte un changement de classe, *i.e.* $k_{m+1} \neq k_m$. De plus, quand on détecte un retour à un état précédemment actif, on utilise un nouveau jeu de paramètres. La raison en est que les relations entre les spectres de phases respectifs de deux objets correspondant à un même état mais séparés dans le temps, sont *a priori* tout à fait arbitraires. Le modèle est alors défini comme suit :

$$\Phi_m\,(f) = \phi_{f,b(m,k)}\,(m)\,\forall f \in \left[0 \ldots \frac{d}{2} - 1\right] \tag{5.10}$$

Les paramètres du modèle sont la fréquence f et l'index de bloc $b\,(m,k)$, où $b\,(m,k)$ correspond au $b^{\text{ième}}$ groupe ou bloc d'objets du cluster C_k contigus dans le temps et $B\,(m,k)$ représente les index des objets dans ce bloc. Ce découpage des clusters en blocs à l'intérieur desquels les objets sont contigus est schématisé sur la figure 5.7.

Le calcul des paramètres optimaux modèle de phase nécessite de maximiser l'erreur

moyenne sur le bloc à la fréquence f :

$$\sum_{i \in B(m,k)} X_i(f) \cdot \gamma_{k_i}(f) \cdot \cos(\phi_i(f) - \Phi_i(f))$$

Pour les expériences de mise au point, nous avons utilisé un modèle polynomial d'ordre p, i.e.

$$\Phi_m(f) = \sum_{i=0}^{p} a_m(i,f) \cdot m^i$$

Conformément à ce que l'on a dit plus haut, on a $a_m(i,f) = a_{m'}(i,f)$, $\forall m' \in B(m,k)$. Les coefficients a_m sont calculés en effectuant une régression polynomiale sur le vecteur $\Phi_m(f), m \in B(m,k)$.

5.2.4 Discussion autour du problème de l'estimation de la phase

Le calcul des paramètres d'un modèle de phase nécessite de maximiser une quantité de la forme suivante

$$h(\theta) \triangleq \sum_{f \in \mathcal{F}, m \in \mathcal{M}} X_m(f) \cdot \gamma_{k_m}(f) \cdot \cos(\phi_m(f) - \Phi_m(f))$$

que l'on peut réécrire sous la forme équivalente

$$h(\theta) = \sum_{f \in \mathcal{F}, m \in \mathcal{M}} y_m(f) \cdot \cos(\varphi_m(f)) \tag{5.11}$$

après avoir posé $y_m \triangleq X_m(f) \cdot \gamma_{k_m}(f)$ et $\varphi_m \triangleq \phi_m - \Phi_m$. Le cas horizontal correspond à $\mathcal{F} = \{f_0\}$, $\mathcal{M} = B(m,k)$, le cas vertical à $\mathcal{F} = \{1 \dots d\}$, $\mathcal{M} = \{m_0\}$. La fonction h n'étant pas convexe, son optimisation de h par rapport à θ se révèle être singulièrement difficile, et ce même dans le cas où $\Phi_m(f)$ est linéaire, soit par rapport à m (horizontal), soit par rapport à f (vertical).

L'approche qui consiste à effectuer un déroulement de la phase suivi d'une régression n'est pas réellement satisfaisante. En effet, d'une part celle-ci suppose implicitement que $y_m(f)$ est constant, soit par rapport à f, soit par rapport à m, ce qui n'a aucune raison d'être vrai en pratique. On peut remédier partiellement à ce problème en effectuant une régression avec pondération par l'amplitude $X_m(f)$.

D'autre part, avec cette approche, on optimise l'erreur quadratique moyenne sur les *phases déroulées* $\sum(\phi_m(f) - \Phi_m(f))^2$ au lieu de l'erreur quadratique sur le signal lui-même. Comme le déroulement et la régression sont découplées, toute erreur sur le dérou-

lement de la phase affecte directement l'estimation des paramètres.

Une étude plus poussée des méthodes d'optimisation de 5.11 sera certainement nécessaire pour améliorer sensiblement les performances du modèle TS. Comme point de départ, nous avons examiné une partie de la littérature [ARF04, AS98, BLS+98, GA89, GMDM+03, LABF02, PR99, PK99] consacrée a la reconstruction de la phase à partir de l'amplitude du spectre, qui est un problème central en imagerie SAR notamment. Le problème qui nous intéresse est malgré tout quelque peu différent, à savoir que l'obtention d'un vecteur de phase déroulé n'est pas directement l'objectif visé. De plus, il ne nous semble pas possible de relier la phase à une quelconque vérité-terrain ni à des caractéristiques physiques ou géométriques sous-jacentes comme on peut le faire en imagerie.

Un estimateur empirique pour le modèle vertical

A partir de la variation de phase entre f et $f+1$, on peut calculer une valeur $\tau(f)$ du « décalage instantané à la fréquence f »

$$\tau(f) \triangleq \frac{d}{2\pi} \cdot \arctan\left[\mathrm{Im}\left(\frac{\mathcal{S}x_m(f+1)}{\mathcal{S}x_m(f)}\right) / \mathrm{Re}\left(\frac{\mathcal{S}x_m(f+1)}{\mathcal{S}x_m(f)}\right)\right]$$

puis calculer la valeur moyenne de ce décalage, pondérée par un rapport d'amplitudes $\varsigma(f)$

$$\begin{aligned} \tau_0 &\triangleq \sum_{f=1}^{d/2} \varsigma(f) \cdot \tau(f) \\ \varsigma(f) &\triangleq \sqrt{\frac{X_m(f) \cdot X_m(f+1)}{\gamma_{k_m}(f) \cdot \gamma_{k_m}(f+1)}} \cdot \left(\sum_{f=1}^{d/2} \sqrt{\frac{X_m(f) \cdot X_m(f+1)}{\gamma_{k_m}(f) \cdot \gamma_{k_m}(f+1)}}\right)^{-1} \end{aligned}$$

Cet estimateur empirique du décalage a pour avantage de ne pas nécessiter de déroulement de phase préalable, et de tenir compte de l'amplitude relative de chaque composante fréquentielle.

Évaluation de la qualité du modèle

Le critère général choisi reste l'erreur quadratique de reconstruction, mais on peut évaluer la qualité du modèle de phase indépendamment du reste du modèle, en comparant les erreurs obtenues avec la phase exacte ϕ_m et avec la phase modélisée $\Phi_m(f)$. La quantité

$$\frac{\sum_f X_m(f) \cdot \gamma_{k_m}(f) \cdot \cos(\phi_m(f) - \Phi_m(f))}{\sum_f X_m(f) \cdot \gamma_{k_m}}$$

comprise entre -1 et 1 permet de quantifier simplement la qualité du modèle de phase,

la valeur 1 correspondant à une modélisation « parfaite », c'est-à-dire avec une erreur de reconstruction du spectre de phase nulle.

5.2.5 Un embryon de modèle perceptif

Toute erreur de modélisation du spectre de phase n'est pas nécessairement nuisible : en définitive, l'erreur perçue par l'auditeur est le seul critère d'importance. Pour cette raison, il nous a semblé que l'élaboration d'un modèle de phase prenant en compte l'insensibilité de l'oreille humaine à la phase *dans certaines conditions,* serait une piste de recherche prometteuse. L'idée est de décomposer chaque objet en trois composantes, une partie tonale, une partie transitoire et une partie bruit, puis d'utiliser un type de modèle du spectre de phase différent, en fonction du mode de perception de la phase associé à chacune de ces trois composantes.

Ainsi, on pense que seul le caractère aléatoire du spectre phase des composantes correspondant au bruit est perceptivement significatif. Dans ce cas, après avoir identifié quelles composantes correspondent au bruit, on peut se contenter de modéliser les variations de la phase de ces composantes par un distribution simple (*i.e.* gaussienne), pour ensuite resynthétiser un vecteur de phase suivant cette même distribution.

On a conduit un certain nombre d'essais pour tester la validité de cette hypothèse. Le principe général est d'obtenir les grains g_m par seuillage des valeurs des amplitudes des TF X_m des objets, *i.e.*

$$g_m \triangleq T\left(|\mathcal{S} \cdot x_m|\right)$$

où T est un opérateur de seuillage

$$\begin{aligned} T(y(f)) &= y(f), y(f) \geq \varepsilon_m(f) \\ &= 0, y(f) < \varepsilon_m(f) \end{aligned}$$

Le seuil $\varepsilon_m(f)$ est soit absolu, *i.e.* $\varepsilon_m(f) = \varepsilon \ \forall f, m$, soit dépendant seulement de l'index m de l'objet, par exemple en le calculant par rapport à l'énergie de l'objet, ou encore dépendant à la fois de la fréquence et de m. Dans ce dernier cas, la courbe de seuillage est calculée à partir de l'histogramme des valeurs de X_m, par exemple pour prendre en compte des phénomènes de masquage.

On peut alors construire un dictionnaire à partir des similarités calculées sur les grains, donc à partir des ressemblances entre les spectres en amplitude seuillés des objets. L'étape suivante consiste à construire un modèle *horizontal* de phase par bloc et à chaque fréquence où l'amplitude $\gamma_k(f)$ de la composante correspondante du prototype n'est pas nulle.

Le problème de cette approche est qu'il arrive fréquemment que $\gamma_k(f)$ soit nulle par suite du seuillage bien que certaines composantes $X_m(f)$ des objets du bloc ne le soient pas. Les tests de reconstruction à partir d'un tel modèle sur des signaux réels ne se sont pas révélés très concluants. On pense pouvoir améliorer la qualité du signal reconstruit en remplaçant le seuillage des composantes du spectre d'amplitude des grains par une quantification sélective, établie à partir d'un modèle de masquage fréquentiel à la manière des codeurs psycho-acoustiques (Mp3, AAC ...).

5.2.6 Observations à partir d'expériences préliminaires

Les différentes approches pour la modélisation du spectre de phase on été implémentées dans le programme. Il n'a pas été conduit d'étude systématique des performances de ces modèles, en grande partie parce que l'on a rapidement constaté que le niveau de qualité perçu n'avait qu'un lointain rapport avec la valeur du rapport signal sur bruit. L'écoute des signaux reconstruits fait apparaître un grésillement prononcé que l'on attribue à la création de discontinuités au rythme de la fenêtre d'analyse, discontinutés qui découlent manifestement des erreurs de modélisation du spectre de phase. Les valeurs de complexité choisies étaient très élevées, de l'ordre de 1 : 100, à rapporter par exemple au 1 : 10 typique du mp3.

Il est fort probable que la proéminence des artefacts puisse être réduite par l'utilisation d'une fenêtre de synthèse non-rectangulaire, qui réalise un lissage ou « cross-fading » du signal dans la zone de recoupement entre deux fenêtres adjacentes. Mais la véritable solution du problème est à notre avis plutôt dans la recherche d'un meilleur algorithme d'estimation (§5.2.4) d'une part, et dans l'élaboration d'un critère d'erreur perceptif intégrant la sensibilité aux variations du spectre de phase, avec un algorithme d'optimisation correspondant d'autre part. Le travail que cela implique est certes conséquent, mais il devrait à terme permettre une augmentation sensible des taux de compression couramment associés aux codeurs perceptifs.

5.3 Conclusion

On a élaboré une série de modèles en modifiant la fonction de synthèse F pour à chaque fois élargir la classe des déformations possibles : mise à l'échelle (TO), décalage temporel (TOΔ), et déphasage par un filtre passe-tout paramétrique (TS). L'augmentation de la complexité de la fonction de synthèse a pour but d'améliorer la qualité de la représentation, et c'est ce qu'on observe dans les expériences. Mais il faut garder à l'esprit que cette complexification s'accompagne généralement d'une augmentation du temps de calcul. En

pratique, il sera donc indispensable de réaliser un compromis entre la complexité *calculatoire*[9] de la fonction de synthèse F, qui conditionne la qualité de la représentation obtenue pour une valeur de complexité β donnée, et les ressources disponibles en matière de temps de calcul, sachant que la nature de ce compromis dépendra pour beaucoup de l'application considérée.

Ceci étant dit, on peut en théorie utiliser n'importe quelle fonction de synthèse F, à condition de disposer d'une fonction G correspondante, pour extraire les grains à partir des objets. En conséquence, le procédé granulaire semble difficilement applicable aux méthodes FM ou par modélisation physique, pour lesquelles l'inversion de la synthèse est loin d'être une étape triviale, pour les raisons suivantes :

- FM : non-linéaire, interdépendance des paramètres, ne permet pas de synthétiser tous les types d'instruments de façon réaliste
- PM : non-linéaire, interdépendance des paramètres, souvent nombreux, modèle spécifique à un type d'instrument

En ce qui concerne les développements futurs, nous préconisons l'incorporation de transformations simples et relativement générales plutôt que l'utilisation de modèles de synthèse très précis, mais dédiés à un instrument ou type d'instrument en particulier.

On propose par exemple de modéliser les variations de l'enveloppe énergétique par une fonction d'enveloppe paramétrique type ADSR.

On pourrait également envisager de modéliser plusieurs notes d'un même instrument mais de hauteur différentes, en développant un modèle de transposition en fréquence. Dans tous les cas, il est souhaitable de pouvoir estimer les différents paramètres du modèle indépendamment les uns des autres afin de limiter le nombre de calculs.

[9] Il est ici question de la complexité calculatoire du calcul de la fonction F et non de la complexité β de la représentation auquel on fait référence ailleurs dans ce document. La complexité calculatoire est liée au nombre d'opérations arithmétiques mis en jeu lors de l'évaluation numérique de $F(\gamma, \theta)$ sur un calculateur.

Troisième partie

Expérimentation et évaluation

Cette partie traite de l'aspect expérimental de notre travail. Le modèle granulaire y est évalué par rapport à ses capacités en termes de représentation efficace des signaux audio.

Le premier chapitre expose les critères d'évaluations retenus, à savoir le rapport *erreur/complexité*, similaire au rapport *débit-distortion* classiquement utilisé pour mesurer les performances en codage. Le deuxième chapitre présente les résultats obtenus sur différents signaux de tests non-synthétiques et pour la plupart monophoniques. On compare à cette occasion les différents modèles et les différents algorithmes de classification, et on discute d'aspects spécifiques telles que les particularités de leur implémentation, lorsque cela a été jugé significatif.

Chapitre 6

Critères d'évaluation

6.1 Protocole expérimental

Le protocole expérimental spécifie le corpus des données sur lesquelles porteront l'évaluation ainsi qu'un certain nombre de critères de mesure. Conformément aux restrictions évoquées au chapitre 4, le corpus contient des signaux *monophoniques*. On a choisi d'utiliser exclusivement des enregistrements d'instruments acoustiques.

Pour chaque signal, après avoir choisi un modèle, un algorithme de classification et fixé les paramètres de réglage si nécessaires, on calcule une représentation granulaire, puis on reconstruit un signal approché $\tilde{s}$ à partir de celle-ci.

On peut alors calculer la valeur du couple erreur/complexité, qui permet de caractériser la qualité de la représentation.

6.2 Critères d'évaluation

Mesure de qualité de la représentation approchée

La mesure principale utilisée pour évaluer les performances est l'opposé de l'erreur quadratique relative globale exprimée en décibels (dB), dénommée Rapport Signal sur Bruit (RSB). On note cette quantité μ_{dB}, et on la définit comme :

$$\begin{aligned} \mu_{dB} &= -20 \cdot \log \frac{\|s - \widetilde{s}\|_2}{\|s\|_2} \\ &= -10 \cdot \log \frac{\|s - \widetilde{s}\|_2^2}{\|s\|_2^2} \end{aligned}$$

Pour certaines applications, il serait pertinent d'utiliser une mesure prenant en compte les effets liés à la perception auditive humaine. On peut par exemple pondérer l'erreur suivant la courbe de réponse en fréquence de l'oreille, prendre en compte les phénomènes de

masquage, etc. Pour rester cohérent, il conviendrait de modifier l'algorithme de clustering pour optimiser le critère choisi pour l'évaluation.

Complexité de la représentation

La mesure utilisée pour quantifier la complexité de la représentation a été détaillée précédemment (*cf.* 3.3.3) . La complexité est la proportion des coefficients nécessaires à la description granulaire par rapport à une représentation dans une base, qui nécessite en général autant de coefficients que d'échantillons du signal.

Comparaisons effectuées

Pour une valeur donnée de la complexité normalisée β[1], une représentation a est meilleure qu'une autre b si $\mu_a > \mu_b$. Les résultats sont présentés sous la forme de courbes illustrant la dépendance du RSB $\mu(\beta)$ par rapport au taux de compression, pour différents

- signaux
- types de modèles granulaires
  - choix de la fonction F
  - taille de fenêtre d
- algorithmes de classification
  - type d'algorithme
  - choix des valeurs des seuils, influence de l'initialisation ...

6.3 Conditions de mise en œuvre des expériences

6.3.1 A propos de la base de test

Les caractéristiques des signaux utilisés [2]dans les expériences sont résumées dans le tableau 6.1, en sachant que la fréquence d'échantillonnage f_s mentionnée est celle du fichier d'origine. Les signaux ont parfois été rééchantillonnées à une fréquence inférieure (22kHz) afin d'accélérer les calculs, en utilisant l'algorithme implémenté dans le logiciel Matlab. Ce rééchantillonage n'a que peu affecté la qualité perçue de ces signaux, car l'énergie est quasiment nulle au dessus de la fréquence de Nyquist correspondante (11kHz). Le nombre de signaux testés était en réalité plus grand, mais pour ne pas alourdir la présentation, nous avons sélectionné 6 signaux supposés suffisamment différents et représentatifs du type de signaux que l'on est susceptible de rencontrer en pratique.

[1] $\beta \triangleq \chi/N$

[2] La base de test est disponible en téléchargement aux adresses suivantes :

- `http://www.irisa.fr/metiss/lmcdonag`
- `http://lorcan.mcdonagh.free.fr`

Signal	Durée	f_s (kHz)	Commentaires
Batterie	5s.	44, 1	polyphonique (grosse caisse, charleston, caisse claire, shaker)
Clarinette	2s.	16	
Congas	7s	44, 1	présence de réverbération
Flûte	10s.	44, 1	respiration de l'instrumentiste audible
Guitare jazz	4s	44, 1	jeu en notes simples (pas d'accords)
Saxophone alto	8s.	44, 1	présence de vibrato, jeu legato (notes liées)

TAB. 6.1 – Caractéristiques des signaux de test

On pourrait aussi envisager d'utiliser une base de signaux de référence provenant d' organismes comme ISO (éditeur des standards MPEG) pour mener des évaluations à plus grande échelle. L'utilisation d'une base de référence commune permet en effet des comparaisons objectives de performances, à la fois par rapport aux techniques existantes, pour lesquelles des évaluations ont déjà été effectuées, mais aussi par rapport aux techniques futures, sans qu'il soit nécessaire de réévaluer l'ensemble des techniques comparées à chaque fois.

6.3.2 Observations à partir des matrices de similarité

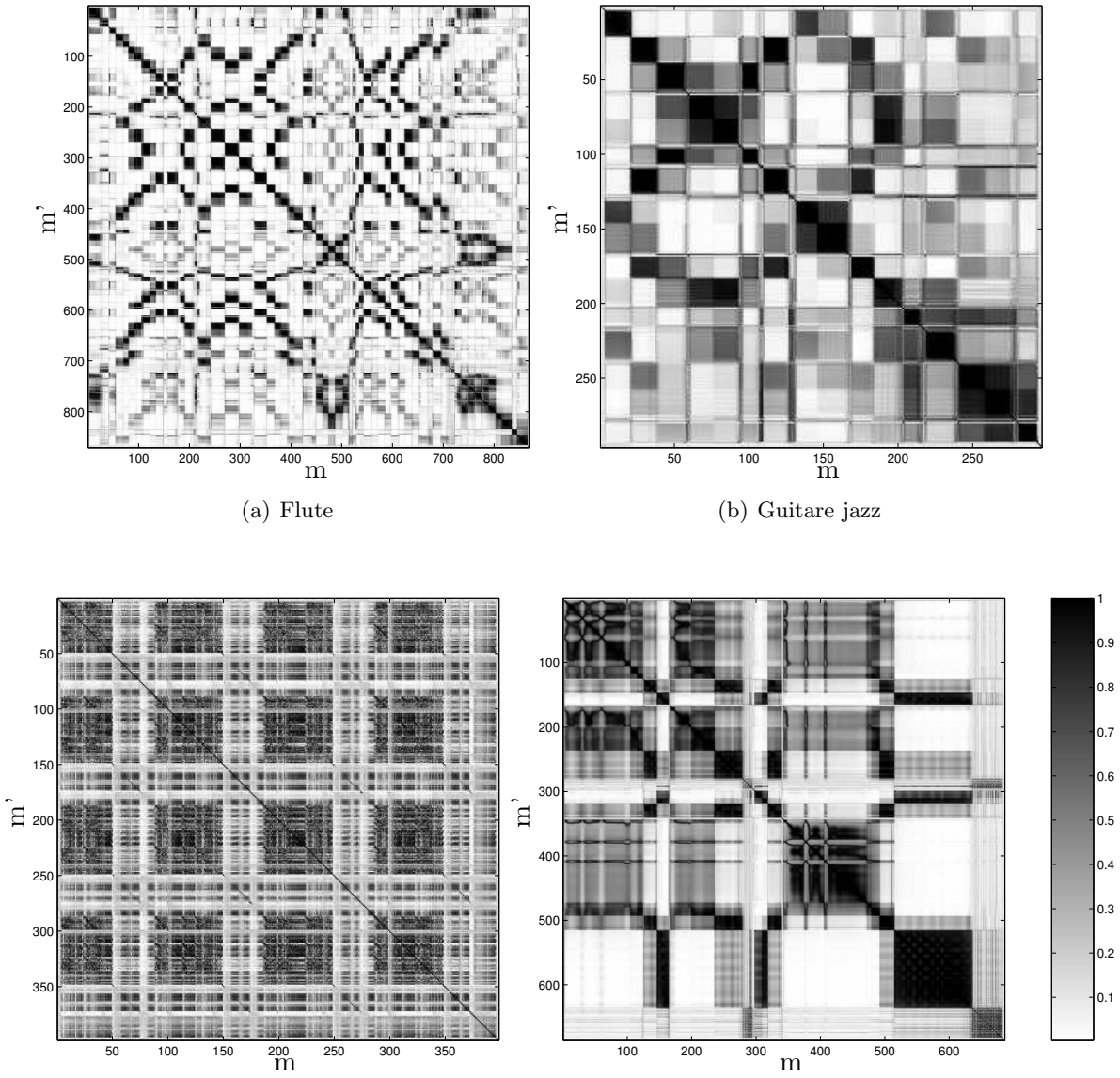
La figure 6.1 contient une représentation graphique des matrices de similarité calculées pour 4 signaux différents avec le même modèle, représentation qui a déjà introduite et commentée aux chapitres 3 (Fig. 3.6) et 5 (Fig. 5.1). On rappelle que plus un bloc est sombre, plus la similarité entre les objets qui le constituent est proche de 1, et plus l'erreur de modélisation par un prototype choisi à partir d'un de ces objets est faible. En conséquence, on s'attend à ce que chaque bloc sombre corresponde grossièrement à un cluster au plus, ce qui dépend bien sûr de l'efficacité de l'algorithme de classification utilisé.

Un bloc sombre situé sur la diagonale peut être vu comme des versions ou des occurrences consécutives d'un même prototype. De même, un bloc sombre situé en dehors de la diagonale correspond à plusieurs occurrences non-consécutives d'un même prototype.

Le premier cas correspond à celui où un « type de son » s'étend sur plusieurs trames, le second au cas où le même son est répété à deux instants du signal séparés de plusieurs trames. En écoutant isolément une portion du signal correspondant à un bloc sombre, on constate que ces portions correspondent presque toujours à l'emplacement et à la durée d'une note de musique dans le signal.

6.3.3 Un modèle de la distribution des valeurs de Γ

Dans ce qui suit, on abrège $\Gamma(x_m, \gamma_{m'})$ en $\Gamma(m, m')$. La distribution des valeurs des similarités est un indicateur qualitatif de l'aptitude du modèle à représenter correctement



(a) Flute (b) Guitare jazz

(c) Batterie (d) Saxophone alto

FIG. 6.1 – Matrices de similarité avec le modèle TOΔ^{x}

($d = 512 \simeq 23$ms, $f_e = 22$kHz)

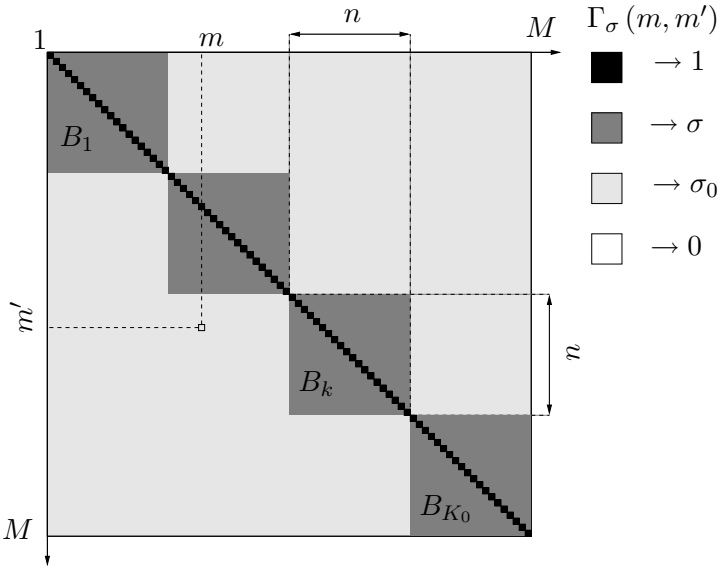


FIG. 6.2 – Un modèle simple de la distribution des valeurs de $\Gamma(m, m')$

le signal. En effet, la forme de cette distribution nous renseigne sur l'aptitude de la fonction de synthèse F à rendre compte de l'ensemble des variations des objets les uns par rapport aux autres. Une distribution binomiale des valeurs de $\Gamma(m, m')$ comportant deux pics aux extrémités ($\Gamma = 0$ et $\Gamma = 1$) et une vallée au centre indique que F permet de modéliser l'ensemble des déformations rencontrées. Une distribution avec un profil plat indique que l'erreur de modélisation d'un objet par un grain est quelconque.

Considérons maintenant la matrice de similarité Γ_σ « idéalisée » représentée sur la Fig. 6.2 : la similarité vaut σ indifféremment sur chacun des K_0 blocs, qui sont tous de même taille (n), à part sur la diagonale où, par définition $\Gamma_\sigma(m, m) = 1$. Le reste de la matrice a une valeur « résiduelle » constante $\sigma_0 < \sigma$. Le modèle s'écrit :

$$\Gamma_\sigma(m, m') \triangleq \begin{cases} 1, m = m' \\ \sigma, m \in B_k, m \neq m' \\ \sigma_0 \text{ ailleurs} \end{cases} \tag{6.1}$$

Dans ces conditions, la distribution des valeurs de $\Gamma_\sigma(m, m')$ est, proportionellement au nombre total d'éléments M^2 :

- sur la diagonale : $1/M$ valeurs à 1 (par définition)
- à l'intérieur des blocs B_k privés de la diagonale : $1/K_0 - 1/M$ valeurs à σ
- autour des blocs : $1 - 1/K$ valeurs à σ_0

Pour ce type de matrice, la similarité moyenne vaut

$$E(\Gamma_\sigma(m, m')) = \frac{1-\sigma}{M} + \frac{\sigma - \sigma_0}{K} + \sigma_0$$

En conditions typiques, K est petit devant M et le premier terme peut être négligé. L'expression de la variance de Γ_σ est plus lourde et n'est pas explicitée ici.

On définit l'erreur relative moyenne minimale $\bar{\mu}$ comme le minimum de l'erreur relative moyenne pour l'ensemble des dictionnaires possibles, les prototypes étant choisis parmi les grains ($\gamma_k \triangleq g_{m_k}$), *i.e.*

$$\bar{\mu} \triangleq \min_{\boldsymbol{m_k} \in \{1 \ldots M\}^{K_0}} \sum_{m=1}^{M} \sqrt{1 - \Gamma\left(x_m, g_{m_k}\right)^2}$$

Avec le modèle de matrice de similarité Γ_σ, K_0 étant le nombre de blocs et K la taille effectivement choisie pour le dictionnaire, on établit que :

- pour $K < K_0$,

$$\bar{\mu} = \left(\frac{1}{K_0} - \frac{1}{M}\right) \cdot K \cdot \sqrt{1 - \sigma^2} + \left(1 - \frac{K}{K_0}\right) \cdot \sqrt{1 - \sigma_0^2}$$

- avec exactement K classes/prototypes, $\bar{\mu} = (1 - K/M) \cdot \sqrt{1 - \sigma^2}$. Cette valeur est atteinte en prenant au moins un prototype γ_k par bloc B_k.
- quand on augmente la taille K du dictionnaire au delà de K_0, $\bar{\mu}$ diminue plus lentement.

Le modèle Γ_σ de la distribution des valeurs de $\Gamma\left(m, m'\right)$ est certes grossier : en pratique, les blocs ne sont bien sûr pas de taille uniforme, leurs frontières ne sont pas aussi nettement définies, et on trouve des blocs qui ne sont pas sur la diagonale du fait de l'existence de redondances à long-terme dans le signal.

En outre, le critère d'erreur qui nous intéresse est l'erreur globale de reconstruction et non directement l'erreur relative moyenne minimale, et la dépendance entre ces deux erreurs fait intervenir les valeurs de l'énergie de chacun des objets. La relation entre ces deux différentes mesures de l'erreur est complexe et difficilement prévisible.

Enfin, le deuxième critère d'intérêt, la complexité du dictionnaire, est lié de manière seulement indirecte à la taille du dictionnaire K , et qui plus est, selon une relation non-bijective.

Ces réserves étant apportées, on verra au chapitre suivant que les courbes d'erreur-complexité présentent bien une tendance générale bimodale : croissance rapide puis lente pour des valeurs de la complexité situées dans la région comprise entre 0 et 0.5 environ.

Au delà, pour $0.5 < \beta < 1$, l'allure de la courbe dévie de celle prévue par le modèle, sans que cela n'ait toutefois de conséquence en pratique, cette zone correspondant à un « taux de compression » proche de 1.

La figure 6.3 présente un exemple d'une courbe de $\bar{\mu}$ en fonction de la taille normalisée

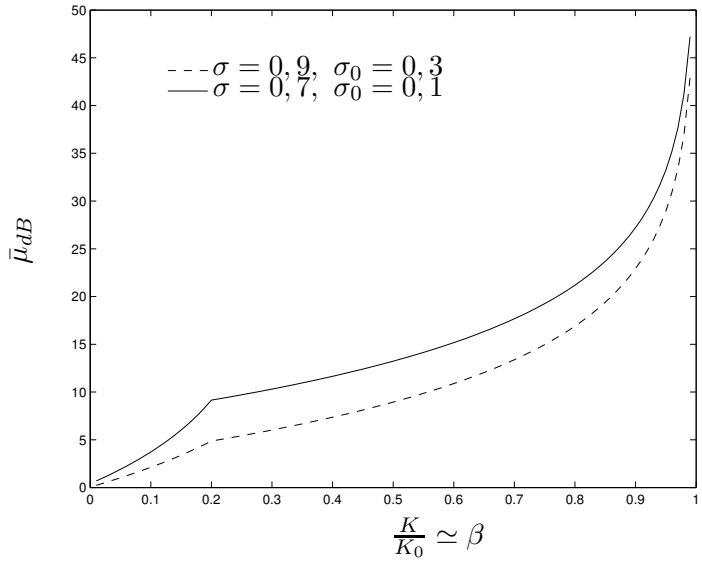


FIG. 6.3 – Erreur relative moyenne minimale en fonction de la taille normalisée du dictionnaire K/K_0, exprimée en décibels, avec $K_0 = M/10$

du dictionnaire K/M établie à partir du raisonnement ci-dessus, pour différentes valeurs de K/K_0 et avec des valeurs vraisemblables de σ, σ_0 et K_0/M.

6.3.4 Implémentation logicielle

Les algorithmes et des outils de mise au point et de diagnostic ont été programmés à l'aide du logiciel Matlab.

Avantages et inconvénients de Matlab L'utilisation du type *structure* permet de regrouper plusieurs variables de types différents au sein d'un même objet, qui apparaissent comme des champs nommés de la structure. Par exemple, dans notre programme, la structure *Cluster* contient les paramètres spécifiques de l'algorithme de classification, le dictionnaire (un tableau des vecteurs-prototypes), les index des objets dans le dictionnaire .. L'utilisation des structures a permis d'obtenir un programme clair et structuré, et donc de limiter les risques d'erreurs lors du développement. Une petite interface graphique pour le chargement et la manipulation de signaux a en outre pu être développée grâce aux outils intégrés à Matlab.

Au chapitre des limitations, signalons l'impossibilité d'avoir recours au passage par adresse, ce qui augmente inutilement l'occupation mémoire, déjà pénalisée par l'utilisation systématique de la représentation en flottants double précision (64 bits ou 4 octets), sachant que pour notre application des entiers 16bits voire des flottants 32bits auraient suffit dans la plupart des cas.

Matlab étant optimisé pour les traitements par blocs associés aux calculs matriciels, il faut limiter l'utilisation de boucles, qui ont un impact très négatif sur les performances. Cela n'est toutefois pas toujours possible, notamment en cas de branchement conditionnel ou de dépendance entre une instruction par rapport au résultat de la précédente. L'utilisation d'un certain nombre de boucles est malheureusement inévitable dans des portions cruciales de nos programmes (calcul de la matrice des similarités et du dictionnaire/clustering).

L'ensemble des calculs, comprenant le chargement du signal, le calcul des grains et de la matrice de similarité, la classification et la reconstruction du signal nécessite un temps égal à environ 10 fois la durée du signal, sur un système Linux équipé d'un processeur AMD Athlon 1.2 Ghz et de 512 Mo de RAM. La partie responsable du calcul de la matrice complète des similarités est de loin la plus lourde en termes de ressources processeur et mémoire. On propose donc des solutions pour diminuer ces ressources en conclusion, page 160.

Chapitre 7

Évaluations sur des signaux monophoniques

7.1 Comparaison des algorithmes de classification

On commence par comparer les performances des algorithmes de classification. Les figures 7.1 et 7.2 illustrent les performances obtenues avec trois algorithmes de classification (K-Médians, seuil 2S et G2), avec un modèle TOΔ. Les fenêtres utilisées ont une longueur d égale à une puissance de 2 et le décalage maximal Δ_{max} est de $\pm d/4$ échantillons.

Les spécificités de chaque algorithme

La classification obtenue avec le K-Médians (KM), à nombre de classes K fixé, est dépendante de l'initialisation (aléatoire). On peut tracer une courbe correspondant à la moyenne encadrée par deux courbes distantes de plus et moins un écart-type de la courbe de moyenne. Cependant, le taux de compression ne dépend pas seulement de la taille du dictionnaire K mais aussi de la configuration initiale du dictionnaire, et la présentation sous forme de courbes ne permet pas de rendre compte de l'échantillonnage irrégulier des valeurs sur l'axe des abscisses. On a donc choisi d'illustrer les performances de l'algorithme *K-Médians* par des nuages de points.

Pour chaque valeur de K, l'algorithme a été lancé successivement 30 fois, en effectuant un tirage aléatoire (différent à chaque fois) de la configuration initiale du dictionnaire. On observe 30 réalisations de la distribution des performances ; sachant que la densité des points dans une zone de l'espace donne une valeur indicative de cette distribution. D'autre part, l'algorithme KM a été itéré 15 fois, après avoir vérifié que l'erreur se stabilisait après 3 itérations en général.

L'algorithme 2S utilise deux seuils $\varepsilon, \varepsilon_0$, compris entre 0 et 1. On a donc lancé l'algo-

rithme pour un ensemble discret d'un millier de valeurs possibles de couples de seuils.

Enfin, l'algorithme G2 a été itéré jusqu'à atteindre la taille maximale du dictionnaire $K = N$, en mesurant l'erreur et la complexité à chaque étape.

Forme générale attendue pour les courbes

De manière générale, on peut s'attendre à ce que l'erreur diminue quand la taille du dictionnaire augmente. Dans le cas contraire, on serait tenté de suspecter un défaut dans l'algorithme de classification.

Pour 2S, on s'attend à ce que le nombre de classes obtenues soit une fonction croissante de la valeur du seuil haut ε, et relation inverse pour le seuil bas ε_0. Un examen empirique des courbes des valeurs de β en fonction de ε_0 à ε fixé, et inversement, montre que ces relations ne sont que très grossièrement vérifiées.

Par construction, l'algorithme G2 conduit à un RSB μ croissant avec β.

Performances comparées

Une observation grossière montre que globalement, l'erreur diminue bien quand β augmente. Le fait que l'on puisse rencontrer une situation où on ait $\beta_1 > \beta_2$ et $\mu_2 < \mu_1$ est notamment imputable au caractère sous-optimal de l'algorithme de classification d'une part.

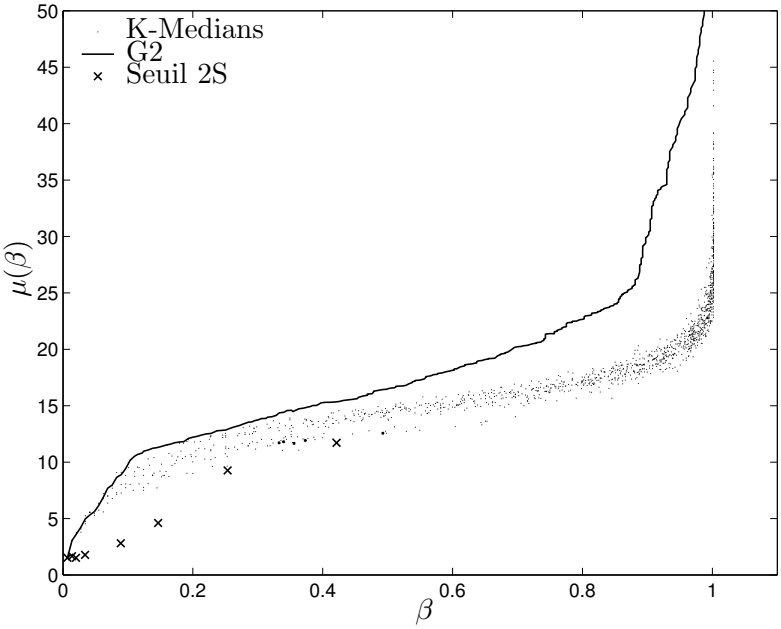
D'autre part, pour KM, les performances dépendent de la configuration initiale du dictionnaire, ce qui explique qu'on puisse avoir des cas où $\beta_1 > \beta_2$ et $\mu_2 < \mu_1$. Cependant, on vérifie que la courbe moyenne d'erreur-complexité est bien croissante.

Pour les signaux utilisés, on constate que l'algorithme G2 conduit aux meilleurs performances dans la majorité des cas de figure, c'est-à-dire que le RSB μ est maximal pour la quasi-totalité des valeur de β.

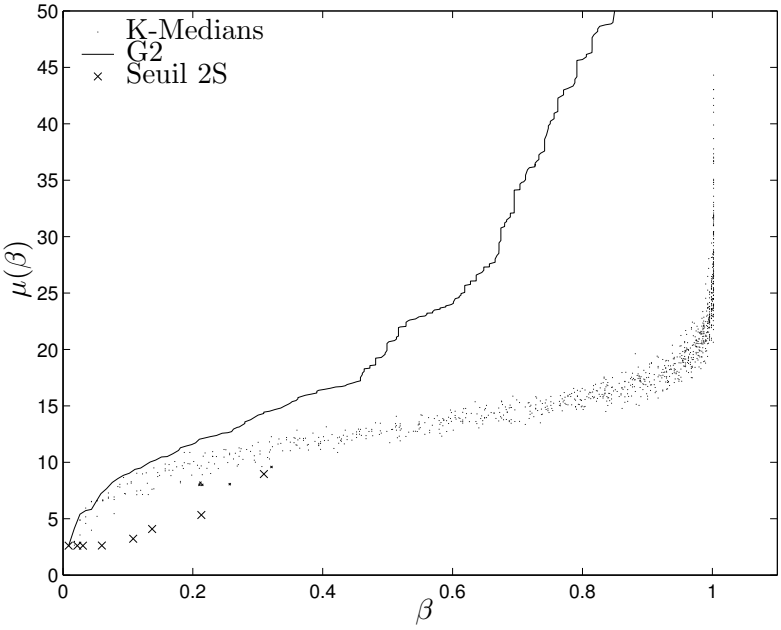
Observations diverses

Quel que soit le signal considéré, les courbes pour K-Médians et G2 arborent approximativement une même forme globale, constituée de trois tendances ou zones majeures : 1) croissance rapide puis 2) croissance lente puis 3), à nouveau rapide. Si l'on écarte l'influence de la classification (nécessairement imparfaite), on peut tenter de donner une interprétation de ce comportement en 3 phases :

1. Chaque nouveau prototype ajouté au dictionnaire contient une « quantité d'information » significative, car ce prototype supplémentaire permet de modéliser des portions de signal sur lesquelles l'erreur était très élevée.

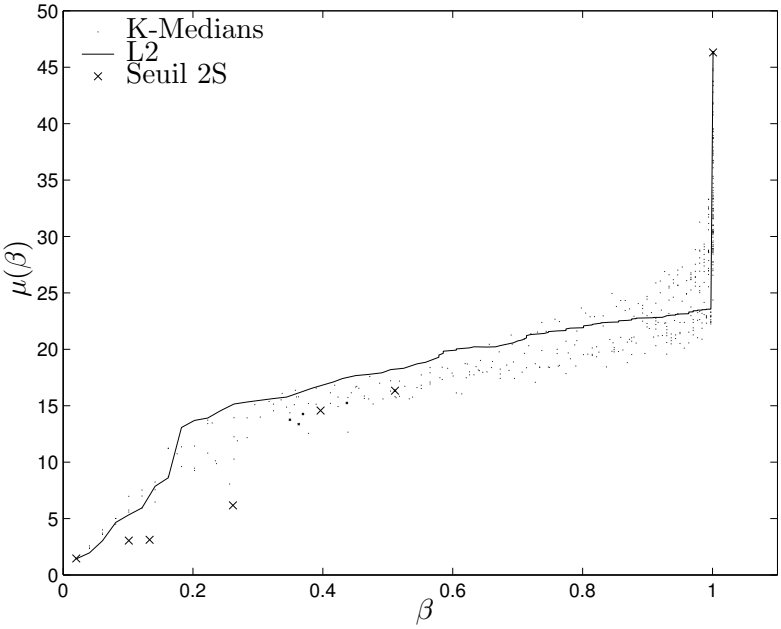


(a) Flute

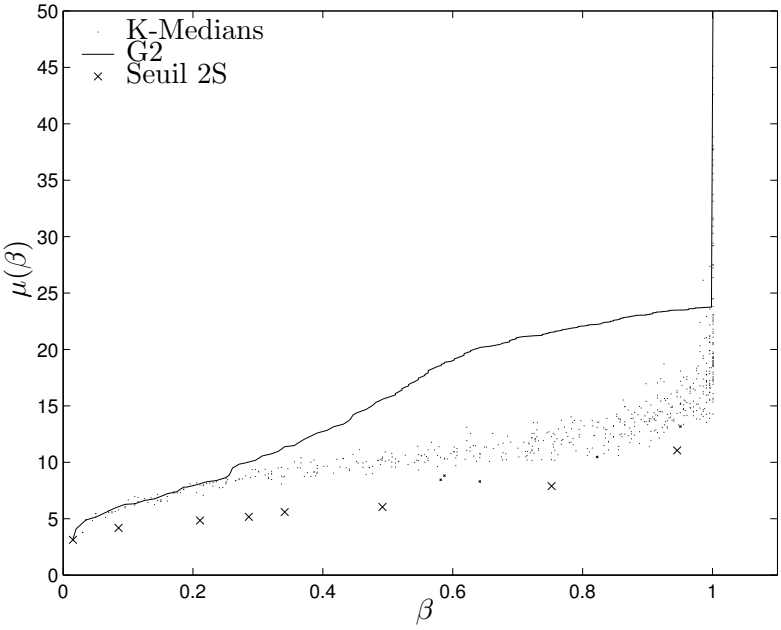


(b) Saxophone alto

FIG. 7.1 – Courbe d'erreur-complexité - $f_e = 22\text{kHz}$, $d = 1024\,(46\text{ms})$



(a) Guitare jazz



(b) Batterie

FIG. 7.2 – Courbes d'erreur-complexité - $f_e = 44\text{kHz}$, $d = 2048\,(46\text{ms})$

2. L'erreur relative pour chaque trame a déjà nettement diminué par rapport à sa valeur de départ (*i.e.* 100%). Le fait de scinder une classe en deux ne permet pas de diminuer autant l'erreur globale que pour la tendance 1.
   On peut penser que cela soit dû à la contrainte imposée au représentant d'une classe d'être lui-même un élément de la classe (choix du médian). Une autre explication possible est que les limites du modèle soient atteintes, c'est à dire que les variations autour du prototype ne sont plus correctement décrites par la transformation F. Dans les deux cas, le fait de rajouter un prototype à une classe déjà formée ne diminue alors l'erreur que pour un faible nombre d'objets de cette classe, l'erreur restant à sa valeur précédente pour la majorité des objets de la classe.

3. Le nombre de classes approche du nombre d'objets, l'erreur globale tend alors naturellement vers zéro. Sur certaines courbes, on constate cependant que l'erreur descend brusquement d'environ -30dB à au dessous de -100dB au voisinage de $\beta = 1$. En effet, pour que l'erreur soit strictement nulle, le dictionnaire doit [1] en général contenir l'ensemble des M grains et le coût de description β de la représentation est alors strictement supérieur à 1.

Conclusions

L'algorithme K-Médians présente deux inconvénients au regard de sa mise en œuvre pratique. D'une part, le nombre de classes K doit être spécifié à l'avance sans que connaître ni la valeur de β ni la valeur de μ associée. D'autre part, les performances sont en pratique grandement sensibles à l'initialisation aléatoire, pour K fixé.

L'algorithme 2S présente des inconvénients similaires au K-Médians. Premièrement, on ne peut dire avec précision quelles performances seront obtenues pour des valeurs données des seuils $\varepsilon, \varepsilon_0$. Deuxièmement, même si un jeu donné de valeurs de seuils conduit toujours à la même classification et donc aux mêmes performances, les performances sont très sensibles aux valeurs exactes des seuils. En raison de la diversité des signaux que l'on est susceptible de rencontrer, il nous paraît vain de chercher à établir une relation de dépendance entre ϵet les performances.

L'algorithme G2 ne présente aucun des inconvénients mentionnés plus haut et conduit dans la majorité des cas au meilleures performances. Il faut toutefois noter que cet algorithme demande un temps de calcul un peu plus long que les autres. Pour les expériences suivantes, on a donc choisi d'utiliser une classification par G2.

[1] Ceci est seulement vrai dans le cas général. L'erreur peut être rendue nulle avec $K < M$ quand certaines trames sont constituées de silence pur ou que deux trames sont exactement identiques, à une transformation par F près.

7.2 Comparaisons entre modèles

7.2.1 Modèles type table d'ondes

On a tracé le graphe des performances obtenues avec G2 et trois modèles granulaires différents, à savoir

1. Table d'ondes simple (TO, trait plein)
2. Tables d'ondes avec décalage (TOΔ, trait interrompu)
3. Tables d'ondes avec extrapolation des prototypes (TOΔ^{x}, pointillés)

La figure 7.3 présente les résultats obtenus sur deux signaux de nature très différente, à savoir une mélodie de flûte et un rythme de batterie.

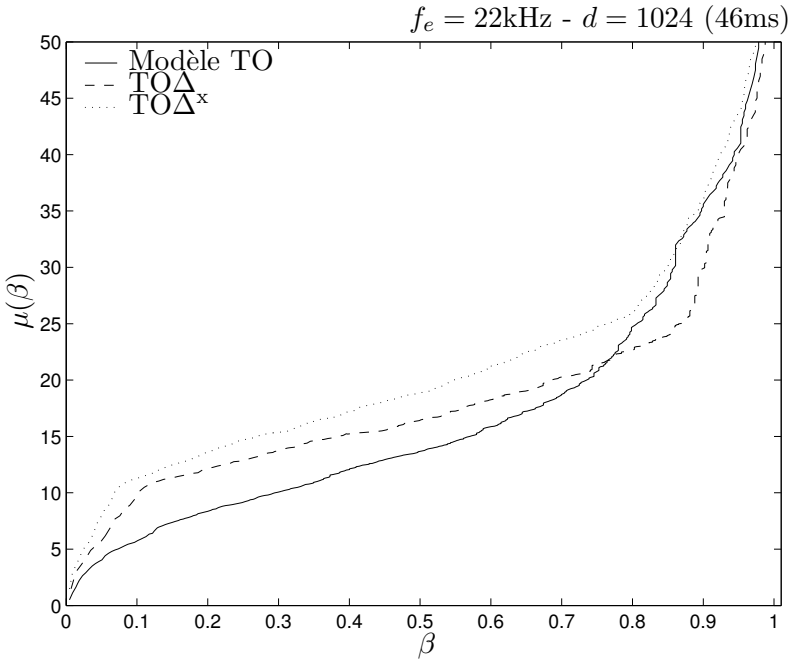
Pour la flûte [Fig. 7.3a], on constate que les raffinements apportés par chaque modèle apportent une amélioration des performances, sauf pour des gros dictionnaires. Autrement dit, en moyenne, un prototype permet de décrire un nombre d'objets d'autant plus élevé que le numéro du modèle augmente. On a constaté que l'ordre 1,2,3 des modèles par rapport à leurs performances était conservé pour des instruments « accordés » comme les vents, les cuivres, les cordes, les percussions accordées (*p.ex.* le vibraphone). Dans ce cas, l'introduction du paramètre supplémentaire Δ diminue l'erreur globale à complexité égale.

Pour le rythme de batterie [Fig. 7.3b], les courbes pour les trois modèles sont très proches, séparées par un écart maximal de 3 dB. La matrice des similarités correspondante 6.1 ne présente pas de structure par blocs bien apparente comme pour les autres signaux. Les sons produits par une batterie comportent une partie transitoire très marquée, ainsi qu'une décroissance énergétique rapide (grosse caisse, caisse claire, charleston) et/ou un spectre à structure proche d'un bruit blanc (charleston, cymbales ...) . La variabilité des objets associé à ce type de signaux est très importante, et le modèle TOΔ ne permet pas de décrire cette variabilité.

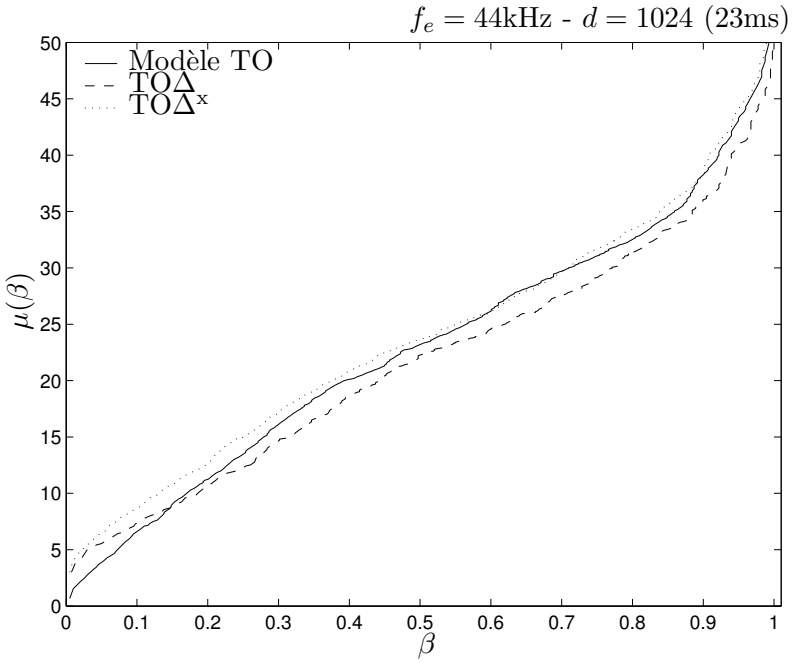
Si les prototypes sont sélectionnés à des instants identiques, l'erreur globale obtenue avec le modèle TOΔ^{x} est par construction, inférieure ou égale à celle obtenue avec TO. D'autre part, par rapport au modèle TOΔ, la diminution du RSB liée à l'extrapolation semble être compensée par le gain en termes de coût de description, tout du moins sur les signaux utilisés ici. Pour ces raisons, seul le modèle TOΔ^{x} est utilisé dans ce qui suit.

7.2.2 Influence de la taille de la fenêtre

Le choix de la longueur d des objets est une question délicate, en raison du compromis entre résolutions temporelle et fréquentielle imposé par l'inégalité de Heisenberg. En l'absence d'hypothèses précises sur la structure du signal, il paraît difficile de mener une étude approfondie sur l'influence de la valeur d pour le modèle granulaire.



(a) Flute



(b) Batterie

FIG. 7.3 – Performances pour différents modèles (G2)

On peut toutefois observer les effets de l'inégalité temps-fréquence sur la figure 7.5. Avec une fenêtre courte (8ms), la similarité entre deux objets correspondant à deux notes différentes est élevée alors qu'elle est le plus souvent proche de zéro pour des objets plus longs (32ms). Sachant qu'à des notes différentes correspondent des fréquences fondamentales différentes, cet effet est directement lié à l'inégalité temps-fréquence. Pour certaines applications comme l'extraction automatique de partitions, il est crucial de pouvoir distinguer deux objets correspondant à deux notes différentes, même séparées d'un demi-ton, ce qui imposera d'utiliser des fenêtres suffisamment longues.

Lors d'une pause, c'est à dire quand l'instrument ne joue pas, le signal est constitué de souffle (essentiellement un bruit gaussien coloré) et de grésillements (bruit laplacien) lié lié aux imperfections des appareils de prise de son. Le signal pendant une attaque est qualifié de transitoire, et la DSP du signal a alors également une structure de bruit, bien que les relations de phase entre composantes fréquentielles ne soient plus aléatoires. Dans ces deux cas, si l'on fait l'approximation que les objets de ces deux types sont des réalisations d'un bruit suivant une distribution normale, la similarité moyenne avec un autre objet est inversement proportionnelle à $\sqrt{d}$, d étant la durée d'un objet. Sur la matrice de similarité, les pauses ou les débuts de notes se matérialisent par de fines bandes horizontales et verticales localisées aux instants correspondant, et on constate bien que ces bandes sont plus sombres quand d diminue.

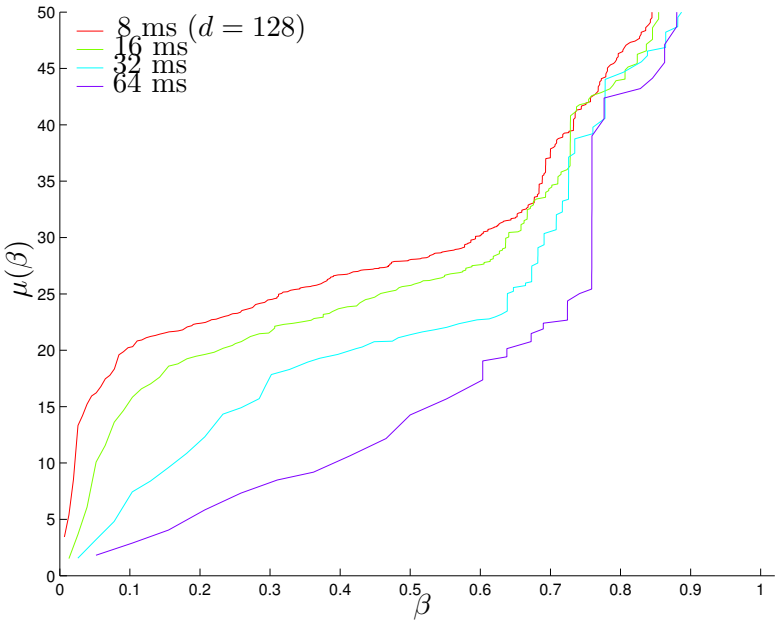
En contrepartie, la résolution temporelle est meilleure avec des fenêtres courtes. Une résolution temporelle élevée est indispensable lorsque que le tempo est élevé ou que les sons présents ont une enveloppe énergétique qui varie rapidement, comme le piano par exemple. Un autre avantage lié à l'utilisation d'une fenêtre courte est que pour une taille de dictionnaire K donnée, il y a plus d'objets x_m disponibles et donc plus de réalisations de $F(\gamma_k, \theta) + e$ par prototype γ_k, et donc en principe une estimation plus robuste de γ_k.

La figure 7.4 illustre les performances obtenues en faisant varier la taille de la fenêtre pour deux signaux de nature très différente, clarinette et conga.

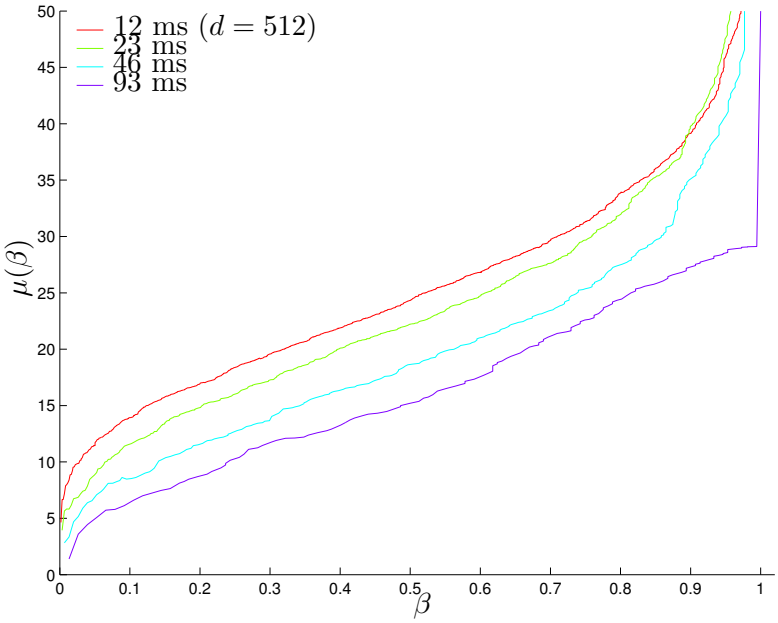
7.2.3 Similarité moyenne

La valeur moyenne des similarités, calculée comme

$$\overline{\Gamma}(s) = \frac{1}{M^2} \sum_{m,m'} \Gamma(x_m, g_{m'})$$

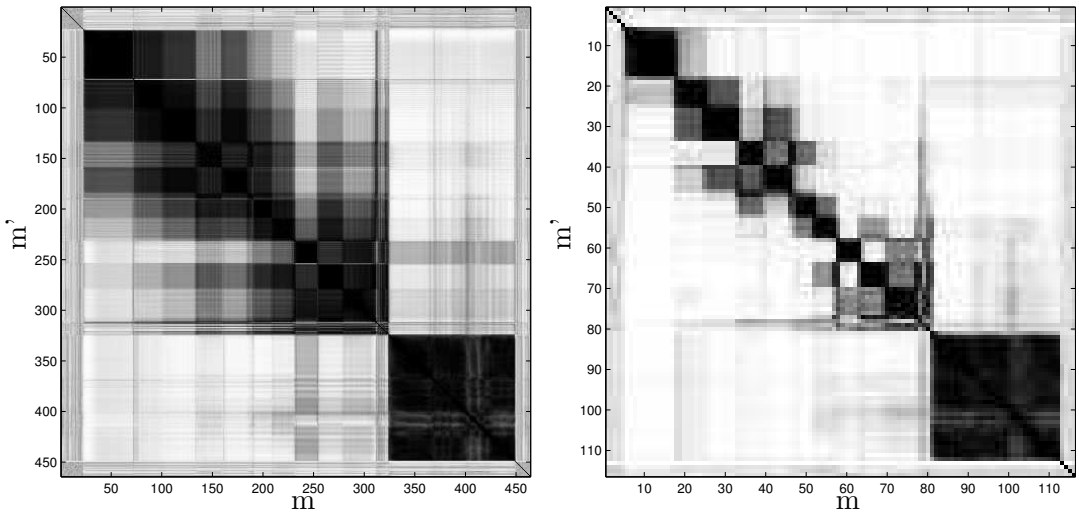


(a) Clarinette ($f_s = 16$kHz)



(b) Congas ($f_e = 22,05$kHz)

FIG. 7.4 – Performances pour différents tailles de fenêtre d (L 2, TOΔ)



(a) $d = 128$ - 8ms (b) $d = 512$ - 32ms

FIG. 7.5 – Influence de la taille de la fenêtre sur les similarités

Clarinette TOΔ^{x},$f_e = 16$kHz

$\overline{\Gamma}(s)$	8 ms	16 ms	32 ms	64 ms	128 ms
TO	0.243	0.164	0.118	0.097	0.095
TOΔ	0.407	0.278	0.207	0.173	0.168
TOΔ^{x}	0.409	0.280	0.208	0.173	0.168

TAB. 7.1 – Similarité moyenne (clarinette)

donne une indication du « taux de redondance » moyenne du signal. Cette valeur dépend du modèle utilisé et est indépendante de la configuration de dictionnaire choisie par la classification.

La figure 7.5 et la table 7.1 permettent de comparer les matrices de similarité obtenues avec une fenêtre courte (8 ms) et une fenêtre de durée moyenne (32 ms). Avec une fenêtre courte, on constate effectivement que la similarité moyenne est plus grande (image plus sombre), au détriment du pouvoir de résolution spectrale, matérialisé par des frontières de blocs moins bien définies.

7.3 Comparaisons entre différents signaux

La figure 7.6 résume les performances comparées obtenues avec 5 signaux différents, une classification G2 et le modèle TOΔ^{x}. Les performances obtenues pour la batterie sont

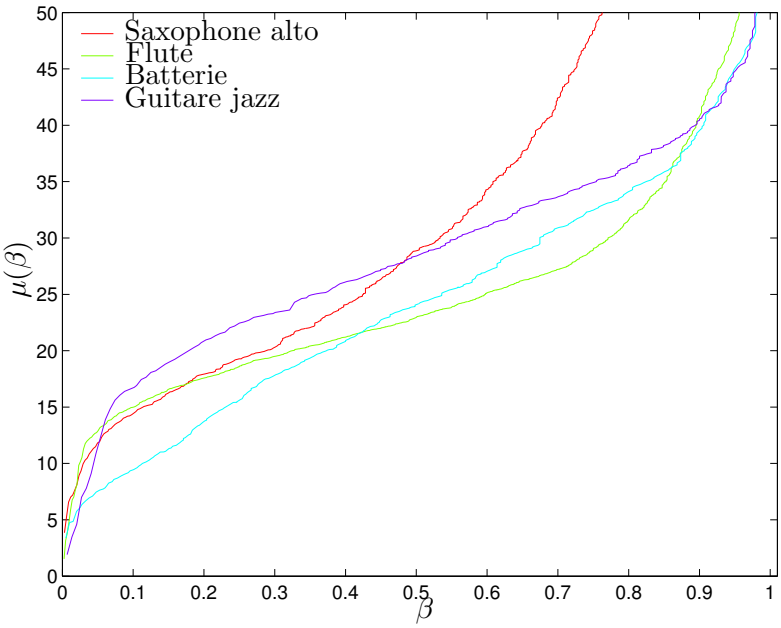


FIG. 7.6 – Variation des performances pour différents signaux

Classification G2, modèle TOΔ^{x}, $f_s = 22$kHz, $d = 512$ (23 ms)

clairement en deçà de la moyenne, ce qui est cohérent avec le fait que le modèle TOΔ^{x} convient surtout aux signaux « déterministes ».

7.4 A propos de l'erreur

7.4.1 Critère mesuré *vs.* critère optimisé

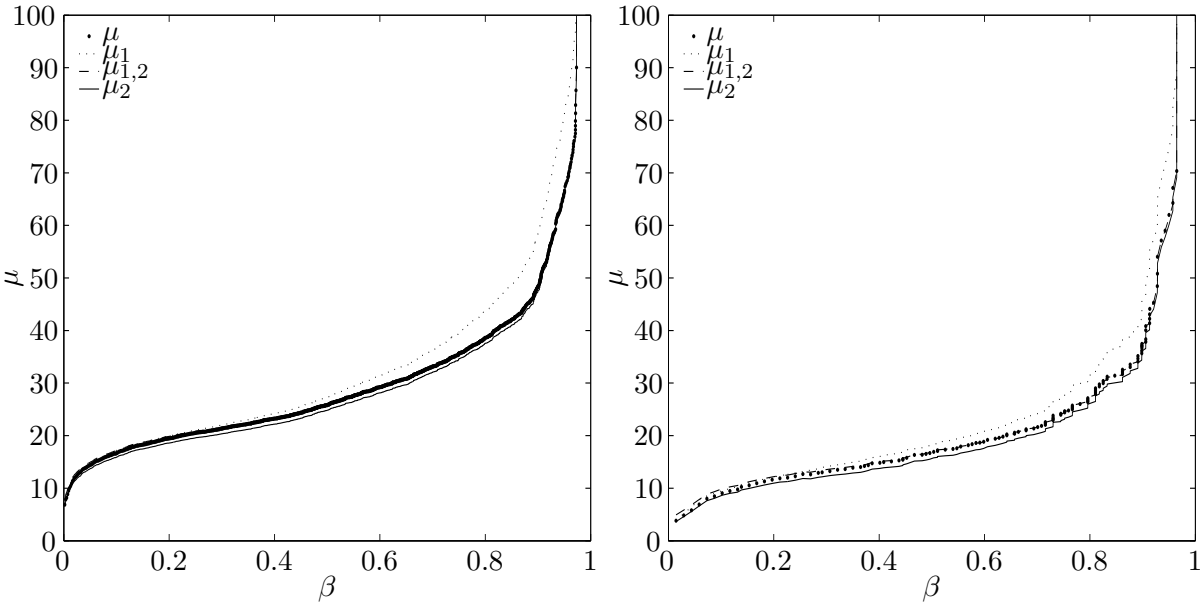
Le critère d'erreur mesuré est le rapport signal sur bruit

$$\mu \triangleq \frac{\|e(t)\|_2}{\|s(t)\|_2} = \frac{\left\|\sum_{m=1}^{M} e(k_m, \theta_m)(t - t_m)\right\|_2}{\left\|\sum_{m=1}^{M} x_m(t - t_m)\right\|_2} \tag{7.1}$$

Or les algorithmes présentés optimisent la moyenne quadratique des normes des erreurs individuelles, ce qui est équivalent à optimiser le rapport

$$\mu_2 \triangleq \frac{\left(\sum_{m=1}^{M} \|e(k_m, \theta_m)\|^2\right)^{\frac{1}{2}}}{\|s(t)\|_2} \tag{7.2}$$

ou encore le rapport



(a) $d = 64\,(6\text{ms})$ (b) $d = 512\,(46\text{ms})$

FIG. 7.7 – Comparaison de différentes mesures de rapport signal sur bruit

Caractéristiques de test : $f_e = 11\text{kHz}$, G2, Modèle TOΔ^{X}, saxophone alto.

$$\mu_{1,2} \triangleq \left(\frac{\sum_{m=1}^{M} \|e\,(k_m, \theta_m)\|^2}{\sum_{m=1}^{M} \|x_m\|^2} \right)^{\frac{1}{2}} \tag{7.3}$$

La comparaison des valeurs de ces différents rapports permet de vérifier les hypothèses faites au paragraphe 4.1.2. Les courbes présentées sur la figure 7.7 illustrent l'évolution de ces différents rapports en fonction de la complexité normalisée β du dictionnaire, pour deux signaux et deux tailles de fenêtres différentes. Nous avons également expérimenté, plus ou moins par hasard, l'utilisation d'une troisième quantité, définie comme le rapport des moyennes arithmétiques des normes des erreurs individuelles et des trames respectivement.

$$\mu_1 \triangleq \frac{\sum_{m=1}^{M} \|e\,(k_m, \theta_m)\|_2}{\sum_{m=1}^{M} \|x_m\|_2} \tag{7.4}$$

7.4.2 Commentaire

On constate que l'ensemble des courbes sont toutes très proches, quelle que soit la taille de la fenêtre choisie (6 ou 46ms).

La très grande proximité entre les courbes μ_2 et $\mu_{1,2}$ indique que la contribution des produits scalaires $v_m \triangleq \langle x_m\,(t - t_m)\,, x_{m+1}\,(t - t_{m+1})\rangle$ entre trames contigües à l'énergie

totale du signal est négligeable. Ceci est sans doute dû à la forme particulière de la fenêtre de Hanning, qui est proche de zéro aux bords. En exploitant la définition de la fenêtre de Hanning 4.1, on peut en effet écrire

$$\begin{aligned} v_m &= \langle W(t-t_m)\cdot s(t), W(t-t_{m+1})\cdot s(t)\rangle \\ &= \sum_{t=1}^{d/2} W(t)\cdot(1-W(t))\cdot s^2(t-t_{m+1}) \end{aligned}$$

Après avoir vérifié que $\sum_{t=1}^{d/2} W(t)\cdot(1-W(t)) = \frac{d^2}{16} = \frac{1}{6}\cdot\|W\|^2$ et $\|W\|^2 = \frac{3}{8}\cdot d^2$, on établit que $\mu_2 \simeq \mu_{1,2} + 1,3\,\text{dB}$ quand s est un signal *constant*. Les courbes permettent de vérifier que ces valeurs sont distantes d'au plus 2 dB environ.

On observe d'autre part que les courbes pour μ et μ_2 ne sont jamais distantes de plus de 2 dB environ, ce qui indique que la moyenne des produits scalaires w_m est effectivement négligeable par rapport à la moyenne des normes des erreurs individuelles sur les objets.

Pour rappel, au paragraphe 4.1.2, on avait estimé à 3 dB la différence maximale entre μ et μ_2 en supposant les énergies des signaux d'erreurs uniformément réparties à gauche et à droite de l'instant caractéristique t_m. En observant les signaux e_m obtenus avec les signaux de test, tel que l'exemple de la figure 7.8, on constate que cette condition est approximativement satisfaite.

L'enveloppe énergétique de l'erreur est nulle aux bords, ce qui s'explique aisément par le fait qu'aux bords l'erreur est la différence de deux signaux eux-mêmes nuls aux bords, suite au fenêtrage avec Hanning. L'enveloppe énergétique croît ensuite pour rapidement atteindre un plateau, et décroît assez brusquement juste avant le bord droit. La forme générale de cette enveloppe est donc celle d'une « Hanning aplatie ». Enfin, les signaux d'erreurs, bien qu'ayant une DSP blanchie par rapport à celle de l'objet, contiennent encore des composantes périodiques significatives.

Plus étonnante est la relative proximité entre les courbes μ et μ_1, pour laquelle nous n'avons pas d'explication pertinente. L'intérêt du rapport μ_1 n'est cependant pas seulement anecdotique, mais surtout calculatoire. En effet, on peut remarquer que la modification d'un seul index k_m d'affectation d'un objet à un prototype impose normalement de réévaluer la somme intervenant dans la définition exacte du RSB μ. Si l'on utilise μ_1 à la place, la réévaluation du RSB ne demande que quelques opérations. Cette astuce calculatoire permet donc de réduire la complexité calculatoire de l'étape de clustering, au cours de laquelle le RSB doit être évalué pour chaque configuration $(\{\gamma_k\}_{k=1\ldots K}, \{k_m\}_{m=1\ldots M})$ testée, au prix d'une légère surévaluation par rapport à la valeur effective de μ. Cette surévaluation n'est toutefois pas génante en soi, puisqu'elle semble fluctuer peu et de manière

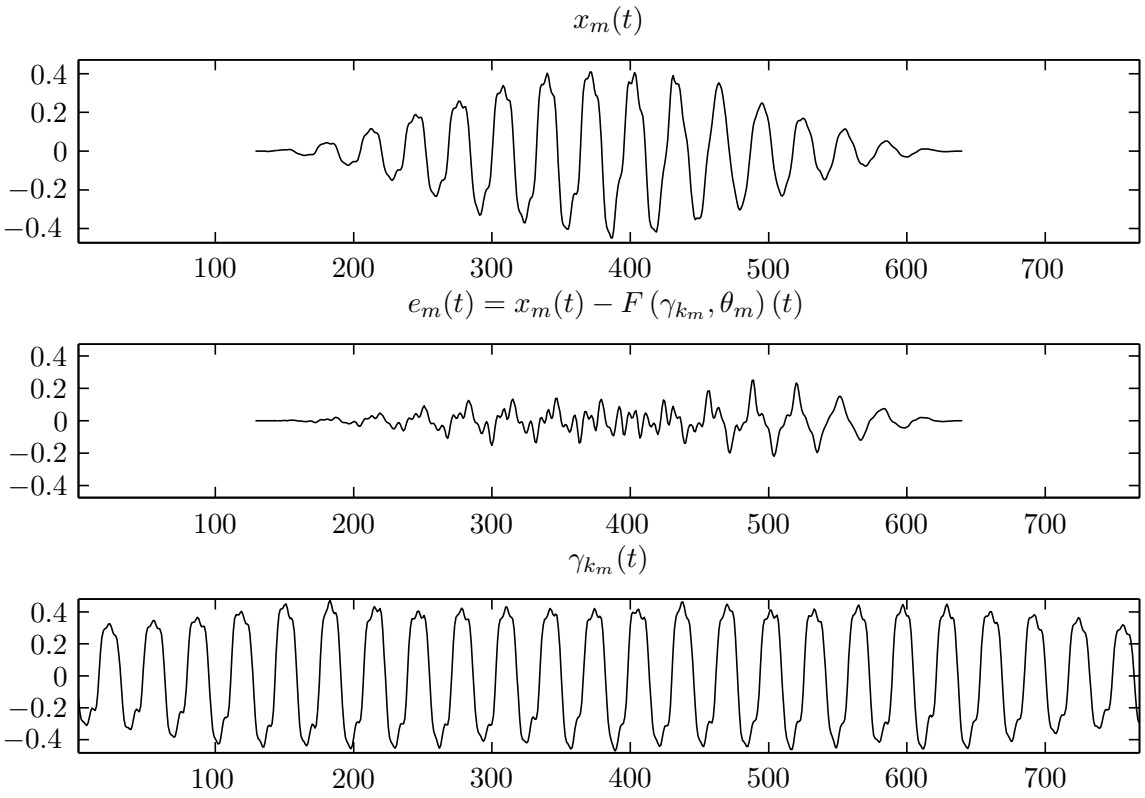


FIG. 7.8 – Exemple de signal d'erreur de reconstruction e_m

(Clarinette, TOΔ^{X}, G2, $\beta \simeq 0.1$, $d = 512$, $\Delta_{\max} = d/4$)

régulière, donc prévisible, et qu'elle est de toute façon de seulement 1 voire 2 dB pour des valeurs de complexité β typiques, c'est-à-dire environ 0.2 et moins.

7.5 Représentation *Temps-Prototype* et applications musicales

A partir du résultat de la classification, on établit une représentation temps-prototype, identique à celle de la figure 7.9, et analogue à la représentation temps-instrument ou « Piano-Roll » évoquée au §1.1.3, la seule différence étant que l'axe des ordonnées désigne des index de prototypes.

Pour obtenir une représentation temps-hauteur, il faut au prélable ré-ordonner les prototypes par fréquence fondamentale croissante, et ensuite éventuellement effectuer un traitement des blocs (*i.e.* les portions de clusters dont les objets constitutifs sont contigüs dans le temps, cf. §5.2.3) de très courte durée, qui sont probablement le résultat d'une erreur de classification. Un exemple d'une telle erreur apparaît sur la fig.7.9, où certains

objets autour de $m = 300$ sont affectés au cluster n°9 alors qu'il faudrait vraisemblablement les affecter au cluster n°10 pour maintenir une certaine continuité temporelle locale.

Pour une application à l'extraction automatique de partition (« audio-to-MIDI »), ce post-traitement peut être effectué soit entièrement automatiquement soit avec l'assistance de l'utilisateur. On peut pour ce faire incorporer directement dans l'algorithme de classification un critère de continuité temporelle, par exemple en optimisant une combinaison linéaire de la complexité [2] β et d'un « coût de rupture temporelle » χ_t, défini par exemple comme

$$\chi_t \triangleq \frac{1}{M} \cdot \sum_{m=1}^{M-1} \delta\left(k_{m+1} - k_m\right)$$

Selon cette définition, le coût est maximal et égal à 1 quand l'affectation k_m d'un objet à un prototype γ_{k_m} change à chaque instant, et il est nul quand aucun changement n'intervient sur toute la durée du signal. Une définition alternative serait

$$\chi_t \triangleq \frac{1}{K} \cdot \sum_{m=1}^{M-1} \delta\left(k_{m+1} - k_m\right)$$

En utilisant ainsi une pondération inverse par la taille du dictionnaire, on prend en compte le fait que le nombre de changements de clusters est au moins égal aux nombres de notes, qui idéalement serait égal à la taille du dictionnaire. A noter que pour l'application considérée, la reconstruction du signal n'est pas requise, on peut donc utiliser une mesure de similarité qui ne dérive pas d'une fonction de déformation efficace, comme par exemple une mesure de ressemblance spectrale, et s'affranchir des complications d'une modélisation les variations du spectre de phase.

Enfin, on peut exploiter cette représentation temps-prototype pour appliquer des traitements sonores sélectifs du signal, dont on va maintenant donner quelques exemples illustrant quelques possibilités d'utilisation du modèle granualaire dans le cadre de la musique assistée par ordinateur.

Transposition harmonique

Après avoir déterminé la hauteur, déterminée par la fréquence fondamentale ou *pitch* de chaque objet, puis déterminé le type de gamme (*p.ex.* mineure, majeure, *etc.*) impliqué par les relations de hauteur entre les notes, on applique un facteur de transposition ou *pitch-shift* différent pour chaque note. Par exemple, on peut de cette manière passer d'une mélodie en mode majeur à mineur et vice et versa, corriger les « fausses » notes ...

[2] Les algorithmes de clustering présentés jusqu'ici travaillant essentiellement à β fixé, il serait plus simple d'optimiser un critère type $\mu - \lambda \cdot \chi_t$.

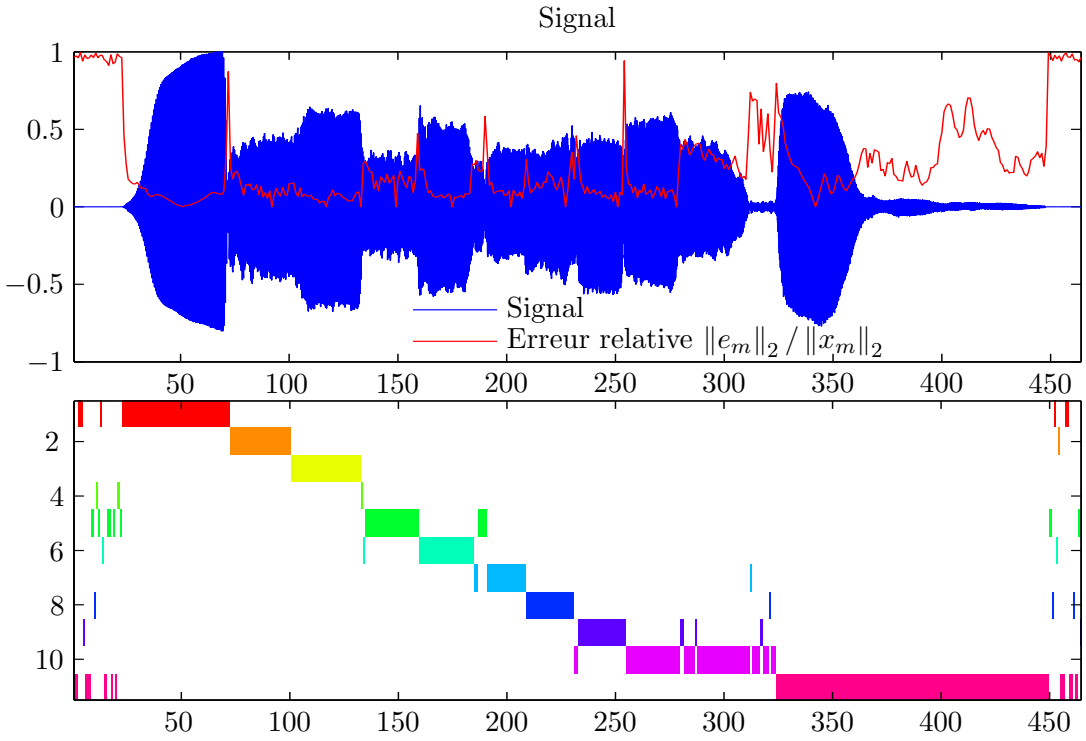


FIG. 7.9 – Représentation temps-prototype obtenue avec $K = 11$, $\beta \simeq 0.05$

Clarinette, G2, TOΔ^{X}, $d = 128$

Modification des paramètres d'expression

Après avoir déterminé les frontières temporelles des notes, on peut envisager par exemple de rajouter du *vibrato* (modulation de fréquence lente) à la fin des notes. On pourrait également passer d'un jeu *staccato* (notes piquées) à *legato* (jeu en notes liées), en prolongeant les notes à l'aide du prototype correspondant, c'est à dire en remplaçant la portion de silence suivant immédiatement une note piquée avec un signal de synthèse pertinent.

Il serait intéressant de modifier le *swing*[3] ou le *groove* du morceau, comme on le fait déjà sur des fichiers MIDI., ce qui est possible en déplacant les instants t_m des objets pour les faire correspondre à la nouvelle grille ryhtmique. Ceci nécessite d'analyser le signal pour déterminer sont tempo et le rythme d'origine d'une part, et de synchroniser les formes d'ondes des objets recalés dans le temps d'autre part, comme cela est réalisé avec les méthodes type PSOLA.

[3]Le swing consiste à anticiper ou retarder certaines notes par rapport à la grille rythmique stricte, telle que la jouerait un métronome.

7.6 Perspective d'application : compression avec pertes

De manière générale, la **compression** de données consiste à réduire le volume occupé par des données binaires [LH87]. On doit naturellement être capable de réaliser l'opération inverse, appelée la décompression, pour reconstituer les données d'origine. Le *taux de compression* est défini comme le rapport entre le poids des fichiers avant et après compression. Ce taux mesure l'efficacité d'un algorithme de compression, il dépend en général plus ou moins fortement de la nature des données que l'on souhaite compresser.

Pour un certain nombre d'applications, parmi lesquelles la transmission ou le stockage de signaux audio, on peut se contenter de reconstituer les données du signal d'origine seulement de façon approchée. On parle alors de compression *avec pertes*, ou *destructive*. L'objectif visé est d'augmenter le taux de compression tout en minimisant la différence perçue par rapport à l'original. Pour des signaux sonores de qualité « proche du CD », on peut atteindre un taux de l'ordre 10 pour 1 (avec un codeur psychoacoustique tel que Mp3 ou Ogg-Vorbis) contre 2 pour 1 pour les meilleurs algorithmes spécialisés dans la compression audio sans pertes (Shorten [Rob94], FLAC, etc.).

Une représentation granulaire pour la compression

A notre connaissance, les algorithmes de compression audio existant actuellement ne cherchent pas à exploiter explicitement l'existence de redondances inter-trames, contrairement aux algorithmes destinées à la compression de flux vidéo. L'approche proposée dans cette thèse apporte une contribution à cette question, qui semble-t-il a été assez peu étudiée, à part notamment, Y. Mahieux *et. al.* [MPC89] qui exploitent les corrélations entre blocs successifs d'une transformée du signal.

En utilisant un prototype pour représenter plusieurs trames, on aboutit à une forme de compression naturelle des données du signal. En effet, après avoir choisi un modèle puis constitué un dictionnaire à l'aide d'un des algorithmes présentés précédemment, on peut reconstruire une version approchée du signal, à partir des seules données du dictionnaire et des paramètres des objets.

Les objectifs de la compression

Cette application nécessite de concilier deux exigences *a priori* antagonistes : d'une part, on veut réduire au maximum le volume des données et de l'autre, on veut obtenir un signal qui s'éloigne le moins possible de la version originale avant compression. Ces deux notions seront évalués de manière quantitative à l'aide de deux mesures, respectivement, une mesure de type Minimum Description Length et le Rapport Signal sur Bruit. Les

quantités obtenues pour ces deux mesures seront mises en rapport pour différents point de fonctionnement du système, pour obtenir un rapport type Débit/Distortion.

Restrictions préalables

Nous n'avons employé aucune technique de codage particulière, telle que la quantification des coefficients ou le codage entropique des données de la représentation. L'usage de telles techniques eût certes indubitablement amélioré le taux de compression et aurait permis une comparaison objective avec des codeurs éprouvés (ADPCM, LPC, Codeurs psycho-acoustiques tel que MPEG 1 Layer III, Quantification vectorielle, etc.).

Cependant, il n'a pas été possible de consacrer le temps nécessaire à l'implémentation de ces techniques et à la mise en place, dans des conditions rigoureuses, d'un protocole de comparaison avec d'autres codeurs. Il convient donc de considérer les performance obtenues comme des bornes inférieures de celles qu'il serait possible d'obtenir en rajoutant une étape de quantification/codage de la représentation.

Toutes les données, paramètres et variables, à valeurs réelles ou entières, sont considérées comme étant codées sur 16 bits. Les données sont donc supposées être codées avec la même précision que le signal, c'est à dire typiquement des entiers relatifs sur 16 bits.

Toutes les questions relatives à l'influence de la quantification et de la dynamique des variables, comme l'influence du pas de quantification ou des bornes admissibles (*range*), de l'utilisation de flottants ou d'entiers sont donc éludées.

Ce choix n'introduit aucune erreur sur les prototypes, ceux-ci étant directement calculés à partir du signal, lui-même codé sur 16 bits. La quantification de l'amplitude α_m des objets introduit une erreur d'au plus ± 0.5 bit de poids faible (LSB) sur le signal reconstruit, dans le cas le plus défavorable. Les paramètres à valeurs entières tels que Δ_m ne subissent aucune quantification et leur dynamique est toujours largement inférieure à la dynamique admissible sur 16 bits signés (*i.e.* 32767).

Pour le modèle TS, on considère que les prototypes sont transmis ou stockés sous la forme de signaux temporels quantifiés, et que la transformée de Fourier est effectuée avec une précision suffisamment grande pour que l'erreur soit négligeable. En conditions réelles, il serait nettement plus avantageux de quantifier le spectre d'amplitude des prototypes, sur le modèle des codeurs psycho-acoustiques dont l'efficacité est largement admise.

7.7 Conclusion

En l'état actuel du modèle, on atteint au moins 15dB de rapport signal sur bruit avec une complexité de 0.1. Si on assimile la complexité au taux de compression, sachant

que comme on l'a expliqué ci-avant, ceci revient à se placer dans les conditions les plus défavorables, *i.e* aucune quantification du dictionnaire, ce chiffre peu sembler relativement modeste.

Toutefois, on sait que le RSB, ou l'erreur quadratique, sont des mesures qui ne rend que grossièrement compte de la dégradation réellement perçue par l'auditeur. L'oreille humaine est en effet sensible à une grande variété de paramètres, tel que la répartition spectrale du signal d'erreur, les variations temporelles de l'énergie, qui ne peuvent être synthétisés par une variable unidimensionnelle.

Sans prétendre bien sûr à une quelconque objectivité, la qualité perçue des signaux reconstruits nous a semblé plutôt bonne. Il se dégage après écoute une tendance notable : les artefacts audibles sont assez différents de ceux que l'on peut rencontrer habituellement avec d'autres méthodes. Les timbres des instruments sont bien respectés, on n'a pas entendu d'effet de type « phaseur » ni de pré-écho, et les détails présents dans la partie haute du spectre audible ne sont pas altérés.

En revanche, il arrive fréquemment que le signal reconstitué soit ponctuellement aberrant, quand une série d'objets est reconstituée à partir d'un prototype associé à un son totalement différent. Cela peut être une conséquence d'une erreur de classification, mais en général cela signifie plutôt que le dictionnaire ne comporte pas un nombre d'éléments suffisant pour décrire l'ensemble des sons présents. L'importance de ce défaut devra être évalué à la lumière des exigences spécifiques de l'application envisagée.

Enfin, les tests menés sur des signaux polyphoniques donnent de mauvais résultats, à la fois du point de vue des valeurs de RSB et du point de vue auditif. Ceci n'est pas véritablement surprenant étant donné les limitations du modèle, qui n'est pas adapté à ce type de signaux, tout du moins dans sa forme actuelle .

Quatrième partie

Conclusion et perspectives

Problèmes à résoudre

Prendre en compte les objets superposés

Jusqu'ici, on a supposé qu'à chaque trame correspondait un objet. Cette hypothèse paraît raisonnable dans le cas dit *monophonique*, où un seul instrument est « actif » à un instant donné. Dans le cas général *polyphonique*, c'est-à-dire quand plusieurs instruments sont actifs simultanément, il serait naturel de considérer que chaque trame est une superposition d'objets, chacun de ces objets correspondant à un des instruments actifs à cet instant. Il existe plusieurs modes de superposition possibles, le plus simple étant l'addition ou *mélange instantané* de signaux. Il est également possible d'envisager des modes plus compliqués comme le mélange convolutif, qui permet notamment de rendre compte du mode physique de mélange de signaux acoustiques dans une pièce réverbérante, ou les mélanges variables dans le temps, qui permettent de prendre en compte le cas de sources en déplacement.

Dans ces conditions, en notant $x_{m,i}$ l'objet correspondant à la source d'index i et à la trame s_n, et en supposant que cet objet suit le modèle granulaire

$$x_{m,i} = F\left(\gamma_{k_m,i}, \theta_{m,i}\right) + e_{m,i}$$

le modèle du signal s'écrit à présent

$$s_n = \sum_{i=1}^{p} x_{m,i} + e_m$$

où p est le nombre de sources et $e_m \triangleq \sum_{i=1}^{p} e_{m,i}$ est l'erreur totale sur la trame s_n. On a recours à p dictionnaires D_i, un par source.

L'estimation des paramètres optimaux peut vraisemblablement être dérivée à partir du cas monophonique, au prix bien sûr d'un accroissement de la complexité calculatoire. L'apprentissage du dictionnaire constitue le problème de fond, qui s'apparente à un cas particulier de séparation de sources dans le cas *sous-déterminé* (moins de capteurs que de sources) *aveugle* (pas d'exemples des sources).

Diminuer la complexité calculatoire

A une fréquence d'échantillonnage de $44,1$ kHz et avec des objets d'une durée de 2^{10} échantillons ou environ 50 ms, le débit est de de l'ordre 5000 objets par minute. L'approche proposée nécessite la connaissance de M^2 valeurs de similarité et ne peut être mise en oeuvre sur des signaux musicaux d'une durée typique, de l'ordre de 5 minutes. Avec les valeurs mentionnées ci-haut, il faudrait en effet calculer et stocker de l'ordre de 10^9 valeurs de similarités, ce qui est considérable en regard des capacités de calcul actuelles.

Pour remédier au problème, on propose deux approches utilisables conjointement : la réduction du nombre de calculs à effectuer par partitionnement de la matrice des similarités d'une part, et l'utilisation de méthodes de calcul approchées d'autre part.

Calcul partiel de la matrice des similarités

L'idée est de décomposer le signal en N_0 segments y_n de durée raisonnable, contenant chacun M_0 objets, puis de calculer une matrice $M_0 \times M_0$ de similarité par segment.

On calcule ensuite N_0 sous-dictionnaires D_n, un pour chaque segment y_n du signal, pour ensuite assembler un dictionnaire $D = \cup_{n=1}^{N_0} D_n$ pour la totalité du signal à partir des sous-dictionnaires déjà constitués. En définitive, on peut chercher à réduire la taille du dictionnaire D en supprimant les prototypes suffisamment similaires.

De cette façon, le nombre de calculs de similarité devient quasiment [4] proportionnel à la durée du signal.

Utilisation de méthodes de calcul approchées

Les valeurs de similarité sont utilisés à deux étapes de l'analyse, lors de la constitution du dictionnaire (clustering) et au moment de la reconstruction du signal à partir de ce même dictionnaire.

Pour la première étape, une certaine imprécision sur les calculs est acceptable tant qu'elle ne modifie pas (ou peu) la sélection des prototypes par rapport à un calcul exact. Supposons que l'on dispose d'une mesure approchée $\widetilde{\Gamma}$ de Γ telle que

$$\left|\widetilde{\Gamma}^2(x,g) - \Gamma^2(x,g)\right| \leq \varepsilon \; \forall (x,g) \in \mathbb{R}^d \times \mathbb{R}^\delta \tag{7.5}$$

Sans rentrer dans des détails spécifiques qui dépendent de la méthode de clustering particulière employée, indiquons que le fait de substituer $\widetilde{\Gamma}$ à Γ n'a d'influence que sur un sous-ensemble des grains susceptibles d'être choisis comme prototype. Supposons qu'un

[4] La dernière étape de réduction de la taille du dictionnaire nécessite de l'ordre de N_0^2 comparaisons, et N_0 est proportionnel à la durée du signal.

ensemble G de grains soient mis en concurrence, et que les objets x_i de la classe à décrire soient connus. Si on utilise la mesure exacte Γ, le grain sélectionné en tant que prototype γ de la classe est celui qui minimise l'erreur quadratique moyenne, définie comme $\|e(\Gamma, g)\|^2 \triangleq \sum_i \|x_j\|^2 \cdot \left(1 - \Gamma^2(x_j, g)\right)$

$$\gamma = \arg\min_{g \in G} \|e(\Gamma, g)\|^2$$

On commencer par calculer une matrice de similarité approchée $\widetilde{\Gamma}(x_m, g_{m'})_{m=1\ldots M, m'=1\ldots M}$, puis on recherche $\widetilde{\gamma} = \arg\min_{g \in G} \left\| e\left(\widetilde{\Gamma}, g\right) \right\|^2$.

D'après l'encadrement 7.5, on peut écrire

$$\|e(\Gamma, g)\|^2 \leq \left\| e\left(\widetilde{\Gamma}, g\right) \right\|^2 + \epsilon \quad \forall g \in \mathbb{R}^\delta$$

avec $\epsilon \triangleq \varepsilon \cdot \sum_i \|x_j\|^2$.

On a $\|e(\Gamma, \gamma)\|^2 \leq \left\| e\left(\widetilde{\Gamma}, g\right) \right\|^2 + \epsilon \quad \forall g$ et donc $\|e(\Gamma, \gamma)\|^2 \leq \left\| e\left(\widetilde{\Gamma}, \widetilde{\gamma}\right) \right\|^2 + \epsilon$. Ce résultat montre que l'on peut majorer l'erreur d'approximation additionnelle introduite par l'utilisation d'une méthode de calcul approchée de la mesure de similarité.

On peut également écrire

$$\gamma = \arg\min_{g \in G} \|e(\Gamma, g)\|^2 = \arg\min_{g \in G'} \left\| e\left(\widetilde{\Gamma}, g\right) \right\|^2$$

où

$$G' \triangleq \left\{ g \backslash \left\| e\left(\widetilde{\Gamma}, g\right) \right\|^2 \leq \left\| e\left(\widetilde{\Gamma}, \widetilde{\gamma}\right) \right\|^2 + \epsilon, g \in G \right\}$$

En d'autres termes, on peut commencer par calculer $\widetilde{\Gamma}(x_m, g_{m'})_{m=1\ldots M, m'=1\ldots M}$ puis rechercher le prototype « exact » γ en ne calculant que $\mathrm{card}^2(G')$ valeurs de Γ. Le gain en temps de calcul apporté par cette technique est bien sur dépendant à la fois de la précision de l'approximation (valeur de ε) et de la distribution des valeurs de $\Gamma(x_m, g_{m'})_{m=1\ldots M, m'=1\ldots M}$.

Conclusion

Ce document a présenté une possibilité d'étendre la modélisation par représentation d'un signal sur un ensemble de vecteurs, qualifié ici de dictionnaire. Il s'en dégage deux directions majeures : l'apprentissage d'un dictionnaire de vecteurs-prototypes directement à partir du signal considéré, et l'utilisation d'une fonction permettant d'appliquer plusieurs transformations à ces vecteurs. Dans les deux cas, on cherche à adapter au mieux la représentation au signal analysé.

Un des apports de ce travail est de montrer que la qualité de la modélisation n'est pas le seul critère d'importance, et que la complexité de la description est également à considérer. Insistons sur le fait que l'intérêt de minimiser la complexité (ou maximiser l'économie) n'est pas limité à la compression de signaux En analyse, l'interprétation du modèle par un humain se complique en effet dès lors que le nombre de variables augmente, et la synthèse musicale pose un problème analogue. La théorie de la complexité (*Minimum Description Length, Kolmogorov*) ne fournit malheureusement aucun moyen utilisable en pratique pour quantifier la complexité. Aussi a-t-on dû se contenter d'utiliser une mesure *ad hoc* de la complexité, adaptée au problème considéré, et donc renoncer au caractère universel de la mesure de complexité définie par Kolmogorov.

Le dictionnaire est calculé par classification automatique, et la complexité du modèle obtenu dépend essentiellement de la durée du signal et de la taille du dictionnaire. Une partie de notre travail a porté sur la recherche d'algorithmes de classification permettant un contrôle direct sur la qualité et de la complexité de la représentation obtenue. Le résultat de ces recherches constitue une contribution au domaine, sans pour autant apporter une réponse exhaustive au problème.

La fonction de déformation ou de synthèse F a une importance cruciale, son rôle étant de modéliser au mieux les variations observées à l'intérieur d'un cluster. Le travail effectué sur ce point a consisté à construire des exemples de fontions, en tenant compte des spécificités du signal musical, puis à évaluer les performances de ces différentes fonctions, en comparant les courbes d'erreur/complexité obtenues. Les résultats des expériences montrent notamment que l'utilisation d'un paramètre d'alignement temporel (modèle TOΔ) du

prototype par rapport à l'objet modélisé permet d'améliorer les performances de manière significative.

Ce travail est exploratoire et préliminaire, et il devra être complété par une étude comparant les performances du modèle granulaire avec l'état de l'art des méthodes de représentation du signal, sur un certain nombre de tâches courantes en traitement du signal.

La principale limitation du modèle, à notre avis, est liée à l'hypothèse de monophonie des signaux. En effet, lorsque plusieurs instruments jouent simultanément, le signal est une combinaison des signaux de chaque instrument. Les variations d'une trame à l'autre sont alors de nature tout à fait différente, et on imagine mal qu'un seul prototype puisse alors représenter l'ensemble de ces variations. Les quelques expériences menées sur des signaux polyphoniques montrent une dégradation sensible de la qualité de la représentation à complexité fixée. Le modèle peut être modifié pour que les trames du signal soient une combinaison linéaire de plusieurs objets, chacun de ces objets étant une version déformée d'un prototype différent. Le mode d'apprentissage du dictionnaire doit alors être totalement revu, et le problème s'apparente alors à celui de la séparation de sources, avec une difficulté supplémentaire due à la présence fonction de déformation.

Bibliographie

[AAPG92] N. Akrout, C. Allart, R. Prost, and R. Goutte. Application of a decision-directed clustering technique for codebook generation in vector quantization. In *IREE-ISSPA 92*, pages 303–306, Edgecliff Australie, August 1992.

[ADPG92] N. Akrout, C. Diab, R. Prost, and R. Goutte. A fast algorithm for vector quantization : application to codebook generation in image subband coding. In *Vol. Signal processing VI : Theories and applications*, pages 1227–1230. EUSIPCO-92 Brussels, August 1992.

[AH71] B. S. Atal and S. L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *J. Acoust. Soc. Amer.*, 50 :637–655, 1971.

[AOJP97] Régine André-Obrecht, Bruno Jacob, and Nathalie Parlangeau. Audio-Visual Speech Recognition and Segmental Master-Slave HMM. In *Workshop on Audio-Visual Speech Processing, Rhodes, Greece*, pages 49–52. Benoit, C. and Campbell, R., 26-27 septembre 1997. (Volume 1).

[APG94] N. Akrout, R. Prost, and R. Goutte. Image compression by vector quantization : a review focused on codebook generation. *Image and Vision Computing*, 12(10) :627–637, 1994.

[Arf80] Daniel Arfib. The musical use of non-linear distortion. In *Proceedings of the 1980 International Computer Music Conference*, pages 498–511. Computer Music Association, 1980.

[ARF04] Kannan Achan, Sam T. Roweis, and Brendan J. Frey. Probabilistic inference of speech signals from phaseless spectrograms. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[AS98] Sassan Ahmadi and Andreas S. Spanias. A new phase model for sinusoidal transform coding of speech. *IEEE Transactions on Speech and Audio Processing*, 6 :495–501, September 1998.

[Bac] Emmanuel Bacry. Lastwave - logiciel et documentation. WWW.

[BBP97] Kamel Belloulata, Atilla Baskurt, and Rémy Prost. Fast directional fractal coding of subbands using decision directed clustering for block classification. In *Proceedings ICASSP-97 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 3121–3124, Munich, Germany, 1997.

[BC00] Eloi Batlle and Pedro Cano. Automatic segmentation for music classification using competitive hidden markov models. In *Proceedings of International Symposium on Music Information Retrieval*, 2000.

[BD00] A. Benveniste and B. Delyon. Frequency domain local tests for change detection. In *12th Symposium on System Identification (SYSID)*, Santa Barbara, June 2000. Paper ThAM4-2.

[Ben03] Laurent Benaroya. *Séparation de plusieurs sources sonores avec un seul microphone*. PhD thesis, IFSIC/IRISA, 2003.

[Ber02] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[BGC01] Christophe Baverel, Philippe Gournay, and Gérard Chollet. Codage de la parole à très bas débit par indexation d'unités de taille variable. NATO IST Panel Symposium on Military Communications, Varsovie, Pologne, October 2001.

[Bil01] Stefan Bilbao. *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*. PhD thesis, Stanford University, June 2001.

[BJRwn] Radu Balan, Alexander Jourjine, and Justininan Rosca. AR processes and sources can be reconstructed from degenerate mixture. Siemens Corporation Research, (unknown).

[BLS+98] H. Banno, J. Lu, S.Nakamura, K. Shikano, and H. Kawahara. Efficient representation of short-time phase based on group delay. In *Proc. ICASSP*, page 861 ?864, 1998.

[BMDBG03] L. Benaroya, L. Mc Donagh, F. Bimbot, and R. Gribonval. Non negative sparse representation for wiener based source separation with a single sensor. In *Proc. IEEE Intl. Conf. Acoust. Speech Signal Processing*, 2003.

[BN93] M. Basseville and I. Nikiforov. *Detection of abrupt changes : theory and application*. Prentice Hall, Englewood Cliffs, NJ., 1993.

[Bou] Pierre Boulez. Dialogue de l'ombre double.

[Cad79] J. A. Cadzow. An extrapolation procedure for band-limited signals. *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-27 :4–12, 1979.

[CG98] S. Chen and P. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. DARPA speech recognition workshop, 1998.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13 :21–27, 1967.

[Cho73] John Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 21(7), 1973.

[CP91] S.D. Cabrera and T. W. Parks. Extrapolation and spectral estimation with iterative weighted norm modification. *IEEE Trans. on Signal Processing*, 39(4) :842–851, April 1991.

[CS86] F. Charpentier and M. Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Proceeding of the International Conference on Speech and Signal Processing*. ICASSP, 1986.

[CT93] Geoffrey J. Chappell and John G. Taylor. The temporal kohonen map. *Neural Networks*, 6(3) :441–445, 1993.

[CVdW03] Rudi Cilibrasi, Paul Vitanyi, and Ronald de Wolf. Algorithmic clustering of music. *New Scientist*, April 2003.

[CW99] Kelvin Kam Wing Chu and Man Hon Wong. Fast time-series searching with scaling and shifting. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania*, pages 237–248. ACM Press, 1999.

[DEYG+02] Shlomo Dubnov, Ran El-Yaniv, Yoram Gdalyahu, Elad Schneidman, Naftali Tishby, and Golan Yona. A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning*, 47(1) :35–61, 2002.

[DGM97] Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.

[DLD77] A. Dempster, N. Laird, and D.Rubin. Maximum Likelihood from incomplete data via the EM algorithm. *J. Royal Star. Soc.*, 39, 1977.

[DLM+98] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *Knowledge Discovery and Data Mining*, pages 16–22, 1998.

[Dud22] Homer Dudley. The vocoder. Bell Labs. Record., 17, p.122, 1922.

[Dun80] Shane Dunne. *A look at phase distortion synthesis as used in the Casio CZ series synthesizers*. Computer Music Association Publications, San Francisco, CA, 1980.

[Elm90] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2) :179–211, 1990.

[FC03] Jonathan Foote and Matt Cooper. Media segmentation using self-similarity decomposition. In *Proc. SPIE Storage and Retrieval for Multimedia Databases , Vol. 5021*, pages 167–75, January 2003.

[FJ03] Brendan J. Frey and Nebojsa Jojic. Transformation-invariant clustering using the em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 2003.

[FK97] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7) :1109–1120, 1997.

[Fla72] J.L. Flanagan. *Speech Analysis, Synthesis and Perception*. Springer-Verlag, New York, 2nd edition, 1972.

[Foo00] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo, vol. I*, pages 452–455, 2000.

[Foo01] Jonathan Foote. Visualizing musical structure and rhythm via self-similarity. In *Proc. International Conference on Computer Music (ICMC)*, September 2001.

[GA89] Dennis C. Ghiglia and Romero Louis A. Direct phase estimation from phase differences using fast elliptic partial differential equation solvers. *Optics Letters*, 14(20), October 1989.

[Gab46] D. Gabor. Theory of Communication. *J. IEEE*, November 1946.

[Gab47] Dennis Gabor. Acoustical quanta and the theory of hearing. *Nature*, 159(4044) :591–594, 1947.

[GB03] R. Gribonval and E. Bacry. Harmonic decomposition of audio signals with matching pursuit. *IEEE Trans. Signal Process.*, 51(1) :101–111, jan 2003.

[GBA] Rémi Gribonval, Emmanuel Bacry, and Javier Abadia. Mp - package et documentation. WWW.

[GG92] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Communications and Information Theory. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[GMDM+03] Laurent Girin, Sylvain Marchand, Joseph Di Martino, Axel Röbel, and Geoffroy Peeters. Comparing the order of a polynomial phase model for the synthesis of quasi-harmonic audio signals. In Workshop on Applications of Signal Processing to Audio and Acoustics - WASPAA'03, Mohonk Mountain House, New Paltz, New York, 2003.

[Goo97] Michael Goodwin. *Adaptive signal models : theory, algorithms, and audio applications*. PhD thesis, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 1997.

[GPO98] Robert M. Gray, Keren Perlmutter, and Richard A. Olshen. Quantization, classification, and density estimation for kohonen's gaussian mixture. In *Data Compression Conference*, pages 63–72, 1998.

[Gre83] Yves Grenier. Time-dependent arma modelling of nonstationary signals. *IEEE Transactions on Acoustics, Speech, and Signal processing*, ASSP-31(4), August 1983.

[Gri99] Rémi Gribonval. *Approximations non-linéaires pour l'analyse de signaux sonores*. PhD thesis, Université Paris IX Dauphine, 1999.

[GV97] Michael Goodwin and Martin Vetterli. Atomic decompositions of audio signals. In *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1997.

[GV99] Alexander Gammerman and Vladimir Vovk. Kolmogorov complexity : Sources, theory and applications. *Computer Journa*, 42(4) :252–255, 1999.

[GV03] Peter D. Grunwald and Paul M.B. Vitanyi. Kolmogorov complexity and information theory *with an interpretation in terms of questions and answers. Journal of Logic, Language and Information*, 2003.

[HH02] Pengyu Hong and Thomas S. Huang. Multimodal temporal pattern mining. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, page 30465. IEEE Computer Society, 2002.

[HRH99] Pengyu Hong, Sylvian R. Ray, and Thomas Huang. A new scheme for extracting multi-temporal sequence patterns. In *IEEE International Conference on Neural Networks (IJCNN'99)*, volume IV, pages 2643–2648, Washington DC, July 1999. IEEE.

[II02] ISO-IEC. Mpeg-4 structured audio. Technical Report ISO/IEC FCD 14496-3 Subpart 5, ISO, March 2002.

[ISO93] ISO/IEC. Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s - part 3 : Audio, 1993.

[Jan02] Gil-Jin et. al. Jang. Single channel signal seperation using time-domain basis functions. *IEEE Signal Processing Letters*, 2002.

[J.C57] J.C.R.Licklider. Effects of changes in the phase pattern upon the sound of a 16-harmonic tone. *J. Acoust. Soc. Amer.*, 29 :780, 1957.

[JD98] A.K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1998.

[JP78] J.Blauert and P.Laws. Group delay distortion in electroacoustical systems. *J. Acoust. Soc. Amer.*, 63(5) :1478–1483, 1978.

[JR81] A. K. Jain and S. Ranganath. Extrapolation algorithms for discrete signals with application in spectral estimation. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-29 :830–845, 1981.

[JS87] M. C. Jones and R. Sibson. What is projection pursuit ? *Journal of the Royal Statistical Society*, Series A(150) :1–38, 1987.

[KDRE+99] K. Kreutz-Delgado, B. Rao, K. Engan, T. Lee, and T. Sejnowski. Convex/schurconvex (csc) log-priors and sparse coding, 1999.

[KK02] Ismo Kauppinen and Jyrki Kauppinen. Reconstruction method for missing or damaged long portions in audio signal. *Journal of the AES*, 50(7/8) :594, July/August 2002.

[KR90] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data : an Introduction to Cluster Analysis*. John Wiley and Sons, 1990.

[KR02] Ismo Kaupinnen and Kari Roth. Audio signal extrapolation - theory and applications. In *Proceedings of DaFx conference*, pages 105–110, 2002.

[LABF02] C. Lacombe, G. Aubert, and L. Blanc-Féraud. Mathematical statement to one dimensional phase unwrapping : a variational approach. rapport de recherche 4521, Inria, jul 2002.

[LH87] Debra A. Lelewer and Daniel S. Hirschberg. Data compression. *ACM Computing Surveys (CSUR)*, 19(3) :261–296, 1987.

[LV93] Ming Li and Paul M. B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1993.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Symp. Math. Statist, Prob.*, pages 281–297, 1967.

[Mah94] Robert C. Maher. A method for extrapolation of missing digital audio data. *J. Audio Eng. Soc.*, 42(5) :350–357, 1994.

[Mal98] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[MC90] E. Moulines and F. Charpentier. Pitch-Synchronous Waveform Processing Techniques for Text-To-Speech Synthesis using Diphones. *Speech Comm.*, 9 :453–467, 1990.

[Moo77] B. C. J. Moore. Effects of relative phase of the components on the pitch of three-component complex tones. In E. F. Evans and J. P. Wilson, editors, *Directional Hearing*. Academic Press, 1977.

[MPC89] Y. Mahieux, J.P. Petit, and A. Charbonnier. Transform coding of audio signals using correlation between successive transform blocks. In *ICASSP-89, vol.3*, pages 2021–2024, 1989.

[MQ86] R.J. McAulay and Th.F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. on Acoust., Speech and Signal Proc.*, 34 :744–754, 1986.

[MZ93] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12) :3397–3415, 1993.

[Ngu95] Truong Q. Nguyen. A tutorial on filter banks and wavelets, 1995.

[NH94] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. of the 20th VLDB Conference*, pages 144–155, Santiago, Chili, 1994.

[NH02] Alan Ng and Andrew Horner. Iterative combinatorial basis spectra in wavetable matching. *Journal of the Audio Eng. Soc.*, 50(12) :1054–1063, December 2002.

[Pap75] A. Papolis. A new algorithm in spectral analysis and band-limited signal extrapolation. *IEEE Trans. on Circuits and Systems*, CAS-22 :735–742, Sept. 1975.

[Pat87] R. D. Patterson. A pulse ribbon model of monaural phase perception. *Journal of the Acoustical Society America*, 82(5) :1560–1586, November 1987.

[Pee98] Geoffroy Peeters. Analyse et synthÃ¨se des sons musicaux par la mÃ©thode psola. JIM98-Workshop, May 1998.

[PK99] Harald Pobloth and W. Bastiaan Kleijn. On phase perception in speech. In *Proc. ICASSP99*, 1999.

[PR99] Geoffroy Peeters and Xavier Rodet. Sinola : A new analysis/synthesis method using spectrum peak shape distortion, phase and reassigned spectrum. In *Proceedings of the ICMC*, Beijing, 1999. ICMC.

[PS69] R. Plomp and H. J. M. Steeneken. Effect of phase on the timbre of complex tones. *Journal of the Acoustical Society of America*, 46 :409–421, 1969.

[PS97] T. Painter and A. Spanias. A review of algorithms for perceptual coding of digital audio signals. In *Proc. of International Conference on Digital Signal Processing (DSP)*, pages 179–205, 1997.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipies in C*, chapter 2.4, pages 50–54. Cambridge University Press, 2nd edition, 1992.

[Puc95] Miller Puckette. Phase-locked vocoder. In *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, N.Y., 1995.

[Rap99] Christopher Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4) :360–370, 1999.

[RKIHSK95] Agrawal Rakesh, Lin King-Ip, Sawhney Harpreet S., and Shim Kyuseok. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, pages 490–501. Morgan Kaufmann, 1995.

[Roa78] Curtis Roads. Automated granular synthesis of sound. *Computer Music Journal - MIT Press*, 2(2) :61–62, 1978.

[Roa96] Curtis Roads. *The Computer Music Tutorial*. MIT Press, 1996.

[Rob94] Tony Robinson. Shorten : Simple lossless and near-lossless waveform compression. Technical Report CUED/F-INFENG/TR.156, Cambridge University Engineering Department, December 1994.

[RPB85] Xavier Rodet, Yves Potard, and Jean-Baptiste Barrière. Chant : de la synthèse de la voix chantée à la synthèse en général. Paris, France - Rapports de recherche IRCAM (35), 1985.

[RS78] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[RV93] Marc Roelands and Werner Verhelst. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech.

In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 554–557, Minneapolis, USA, 1993.

[Sa97] Xavier Serra and al. Integrating complementary spectral models in the design of a musical synthesizer. In *Proceedings of the International Computer Music Conference*, 1997.

[Ser97] Xavier Serra. *Musical Signal Processing*. Swets & Zeitlinger, 1997.

[SI92] Julius Orion Smith III. Physical modelling using digital waveguides. *Computer Music Journal*, 16(4) :74–91, Winter 1992. Special issue on Physical Modelling of Musical Instruments, Part I.

[Spa94] A. S. Spanias. Speech coding : A tutorial review. In *Proc. IEEE*, pages 1541–1582, 1994.

[SSI90] Xavier Serra and Julius Orion Smith III. Spectral modeling synthesis : A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 4(14) :12–24, 1990.

[Tri77] J. M. Tribolet. A new phase unwrapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal processing*, ASSP-25(2) :170–177, April 1977.

[Tru88] Barry Truax. Real-time granular synthesis with a digital signal processor. *Computer Music Journal - MIT Press*, 12(2) :14–26, 1988.

[Ver90] Barry L. Vercoe. *Csound : A manual for the audio processing system and supporting programs*. MIT Media Lab, Music and Cognition, Cambridge, Massachusetts, 1990.

[VM02] Paul Vitanyi and Li Ming. Simplicity, information, kolmogorov complexity, and prediction. In Arnold Zellner, Hugo A. Keuzenkamp, and Michael McAleer, editors, *Simplicity, Inference and Modelling*, pages 135–155. Cambridge University Press, 2001/2002.

[WV97] Robert A. Wannamaker and Edward R. Vrscay. Fractal wavelet compression of audio signals, 1997.

[Yan02] Cheng Yang. Efficient acoustic index for music retrieval with various degrees of similarity. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 584–591. ACM Press, 2002.

[Yao00] Y. Y. Yao. Granular computing : basic issues and possible solutions. In P.P. Wang, editor, *Proceedings of the 5th Joint Conference on Information Sciences, Vol. 1*, pages 186–189, Atlantic City, New Jersey, USA, March 2000. Association for Intelligent Machinery.

Résumé

Les techniques de synthèse sonore actuelles permettent de reproduire une grande variété de sons à partir de quelques paramètres. Du point de vue l'analyse, les signaux audio sont généralement représentés par une combinaison d'objets de forme sinusoïdale, en grand nombre, auquel on adjoint éventuellement une partie transitoire et/ou bruit. Qu'advient-il si remplace ces objets dérivés d'une famille de fonctions typiquement sinusoïdale, par des objets obtenus à partir d'une fonction de synthèse sonore ?

Cette thèse est consacrée à l'étude d'un modèle dit granulaire, et apporte un élément de réponse à la question ci-dessus. Les caractéristiques générales de ce modèle sont présentées et discutées, et on propose un exemple de méthode de calcul pratique du modèle. La première classe de modèles étudiée est dérivée de la synthèse par Table d'Ondes et les objets y sont sélectionnés parmi un dictionnaire de formes d'ondes complexes puis soumis à plusieurs déformations. Le second type de modèle, appelé TS, exploite un dictionnaire de profils de Densités Spectrales de Puissance de référence couplé à une fonction paramétrique de déformation du spectre de phase.

Les travaux présentés apportent une contribution au niveau du formalisme et des algorithmes de classification, et proposent une approche originale au problème de la représentation efficace de signaux, dont la compression audio est une application naturelle.

Les détails techniques de l'implémentation et les résultats des évaluations, menées sur un ensemble de signaux sonore réels, figurent également dans ce document, ceci afin de faciliter l'évaluation des performances du modèle.

Ce travail explore donc les possibilités d'incorporer des éléments issus du domaine de la synthèse sonore dans la modélisation du signal, et nous espérons que cette tentative de rapprocher les deux domaines sera accompagnée par de nouveaux développements aussi bien théoriques que pratiques.