Méthodes d'identification pour le contrôle de l'utilisation de documents audio

Jérome Lebossé

Département de formation doctorale en informatique **École doctorale SIMEM Normandie**
UFR Sciences

Méthodes d'identification pour le contrôle de l'utilisation de documents audio

THÈSE

présentée et soutenue publiquement le 7 Mai 2009

pour l'obtention du

Doctorat de l'Université de Caen

(spécialité informatique)

par

Jérôme Lebossé

Composition du jury

Rapporteurs :	Myriam Desainte Catherine	Professeur, Labri, Université de Bordeaux, France
	François Pachet	HDR, Sony CSL, Paris, France
Directeur :	Luc Brun	Professeur, Greyc Image, ENSICAEN, France
Examinateurs :	Thierry Lecroq	Professeur, LITIS, Université de Rouen, France
	Jean-Claude Paillès	Ingénieur, Orange Labs R&D, Caen, France
	Marinette Revenu	Professeur, GREYC Image, ENSICAEN, France

Laboratoire Greyc — UMR 6072

Mis en page avec la classe thloria.

Remerciements

Je souhaite tout d'abord remercier avec insistance Luc Brun. Grâce à ses conseils avisés et à son expérience, cette thèse a su s'orienter vers des chemins pertinents. Son aide, tant d'un point de vue recherche que morale dans les périodes de doute, a été plus que précieuse pour mener à terme ce travail de recherche et je l'en remercie encore.

Je suis redevable envers Jean-Claude Pailles pour avoir pris en charge l'encadrement de cette thèse à France Télécom ainsi qu'Yvan Rafflé pour avoir tout fait pour que cette thèse soit acceptée dans les hautes sphère de France Télécom et sans qui rien n'aurait été possible.

Myriam Desainte Catherine et François Pachet ont accepté d'être rapporteurs de cette thèse et c'est pour moi un très grand honneur et un motif de motivation supplémentaire.

Je suis reconnaissant envers Marinette Revenu qui, la première, a cru en moi au sein du laboratoire Greyc Image de Caen et a eu un rôle essentiel dans ma candidature pour cette thèse.

Je remercie aussi Thierry Lecroq et Christian Doncarli d'être jury de ce travail.

Je remercie mes collègues de France Télécom pour la très bonne ambiance quotidienne qui y régnait et plus particulièrement Marie, Julien et Vincent qui ont partagé mon bureau pendant tout ce temps.

Enfin, je souhaite remercier mes parents pour leur soutien de toujours et pour m'avoir permis de faire des études aussi longues.

Et je finirai par remercier ma femme, Céline, qui m'a apporté son aide totale et sans qui les périodes de démobilisation d'après thèse auraient surement eu raison de ce mémoire. C'est grace à toi que j'ai trouvé la motivation pour aller au bout de ce travail.

La vraie musique suggère des idées analogues dans des cerveaux différents.

L'Art romantique (1852)

Charles Baudelaire

Résumé

L'objectif de ces travaux de recherche est de proposer une méthode fiable et robuste d'identification de documents audio et plus particulièrement musicaux. Les contraintes de cette méthode sont nombreuses puisque nous désirons une méthode avec un fort pouvoir discriminant qui soit capable d'identifier un document audio parallèlement à sa lecture, qui requière de faibles capacités de stockage et soit robuste vis à vis de certaines altérations du signal.

Nous avons donc conçu une méthode d'identification de signaux audio basée sur l'extraction d'une empreinte. Cette empreinte permet de reconnaître un signal parmi un ensemble de signaux caractérisés par leurs empreintes. Pour cela, l'empreinte est calculée à partir de certaines propriétés du signal. L'originalité de notre méthode vient du fait que la plupart des méthodes existantes se basent sur une analyse des fréquences. Or notre méthode se base uniquement sur une analyse temporelle du signal et l'extraction de positions remarquables (onsets) à l'intérieur de celui-ci. Les mesures de similarité que nous proposons utilisent les spécificités de nos empreintes pour identifier de façon précise des documents tout en conservant de faibles temps de calculs malgré la taille et le nombre de nos empreintes.

Ce mémoire décrira les deux étapes conduisant à l'identification d'un extrait audio inconnu, à savoir une première phase de calcul d'empreinte et une seconde de comparaison avec un ensemble d'empreintes précalculées afin d'identifier l'extrait. L'efficacité de chacune de ces étapes sera démontrée à travers différents essais et comparée avec la référence en matière d'empreintes audio. Nous conclurons sur l'intérêt de nos travaux et les perspectives ouvertes par ceux-ci.

Mots clés: empreinte audio, segmentation audio, identification.

Abstract

This thesis aims at defining a reliable and robust identification method for audio documents and more particularly for musical ones. Our method has to satisfy many constraints : It must be able to discriminate between close signals and to identify an audio document during its reading by a player. It must also require low computational and storage costs and must finaly be robust against some common signal's alterations.

We have based our identification method of audio signals on the computation of a small hash of the signal called its fingerprint. This fingerprint captures essential properties of the signal. It characterizes it and allows to identify a signal among a set. The originality of our method comes from the fact that most of existing methods are based on an analysis of the signal's frequencies while our fingerprint is solely based on a temporal analysis of the signal and on the detection of particular positions (called onsets) along it. The similarity measures that we propose between fingerprints use the specific properties of our fingerprints to identify precisely a document while keeping low computational time.

This thesis describes the two steps leading to the identification of an audio file : The computation of the fingerprint and the comparison of an unknown fingerprint with a database of fingerprints corresponding to known audio files. The efficiency of each of these steps is evaluated by experiments and compared with the most known methods in this field. We conclude this thesis by the insight of our work and the perspectives that it opens.

Mots clés: audio fingerprint, onsets, identification.

Table des matières

Table des figures

1

Introduction

Ce travail de recherche traite de l'identification de documents audio numériques. La méthode proposée est basée sur des empreintes de documents audio. Cette méthode calcule une signature qui résume le signal audio et permet de le reconnaître parmi une base de données de signatures préalablement calculées. Un objectif additionnel de ce travail de recherche est de proposer un scénario de gestion de contenus audio s'articulant autour de notre méthode d'identification d'empreintes. Ce scénario pourrait être mis en œuvre pour contrôler l'utilisation des documents audio et faire respecter les droit d'auteurs. Ce travail de recherche s'est déroulé dans le cadre d'une bourse CIFRE co-encadrée par Jean Claude Paillès et Luc Brun appartenant respectivement aux laboratoires Orange Labs Caen et GREYC (Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (UMR 6072)).

Dans ce chapitre, nous présenterons le contexte sociologique et économique qui a motivé cette thèse. Nous introduirons ensuite la notion d'empreinte audio. Les applications potentielles de ce type de techniques et plus particulièrement la DRM analogique seront exposées. Nous décrirons aussi les critères permettant de mesurer l'efficacité de nos travaux ainsi que nos contributions dans ce domaine. L'organisation de ce mémoire conclura ce chapitre.

Sommaire

1.1 Contexte

La gestion de contenus numériques à caractère musical est à l'heure actuelle devenue un enjeu social, culturel et économique majeur. Le fait que la musique suscite un intérêt si important est principalement dû à la quantité de documents audio devenus facilement accessibles aux utilisateurs. Cette omniprésence a été grandement favorisée par les avancées technologiques récentes en matière de capacités de stockage, de vitesses d'accès internet et de diversité des types de lecteurs présents sur le marché. Cette prolifération de musique numérique provoque dès lors de nouveaux enjeux et nécessite de nouvelles techniques d'organisation et de gestion de quantités importantes de documents. Les techniques usuelles de recherche de musique sont généralement basées sur les metadonnées associées à un document (artiste, style, ...). Cependant, les metadonnées sont des informations qui peuvent être manquantes ou erronées lorsqu'on télécharge de la musique ou lorsqu'on encode un CD en mp3. Ces informations ne sont donc pas fiables et ne peuvent pas être utilisées pour décrire efficacement un document. On peut alors décrire un document audio à partir de son contenu, c'est à dire du signal, afin de le caractériser de manière efficace et pertinente. C'est pourquoi de nombreux chercheurs ont été attirés par la problématique générale de caractérisation du signal audio musical.

1.2 Description d'un document audio

Une information associée à un contenu musical est appelée métadonnée. Cependant, il existe une ambiguïté entre les métadonnées éditoriales, dépendantes du contexte de création du document (ex : artiste, album, maison de disque), les métadonnées culturelles liées à la perception des auditeurs (ex : « genre »fourni aux moteurs de recherche), et les métadonnées psychoacoustiques extraites à partir du signal (ex : tempo, rythme, énergie[47]). C'est pourquoi la description d'un document audio concerne un large panel de caractéristiques allant de l'acoustique au culturel en passant par la psychoacoustique. Si on se penche sur les caractéristiques de type acoustique extraites à partir du signal, on distingue trois types de descripteurs :

1. Les descripteurs de haut niveau font appel à la sémantique et ont, par conséquent, une signification compréhensible pour l'utilisateur (ex : émotion). Cela nécessite une modélisation de l'analyse du signal par un utilisateur. Les recherches sur ce sujet font appel aux sciences cognitives et psychologiques.
2. Les descripteurs de niveau intermédiaire analysent un lot important de données pour en déduire des groupes ou des généralisations (ex : styles de musique). Ces descripteurs conservent un sens pour l'utilisateur et font appel à l'analyse statistique et à l'apprentissage.
3. Enfin, les descripteurs de bas niveau caractérisent des propriétés calculées directement à partir du signal mais n'ayant pas forcément d'interprétation évidente pour l'utilisateur (ex : énergie, spectre). Cela permet d'extraire une information pertinente et propre à un contenu.

1.3 La reconnaissance d'empreinte audio

La reconnaissance d'empreinte est une méthode qui attribue à chaque document audio, une courte signature (l'empreinte) le résumant. Cette technique extrait des caractéristiques acoustiques d'un contenu audio pour les stocker dans une base de données. Ces caractéristiques sont généralement des descripteurs de bas niveau. Quand un extrait audio inconnu est

présenté à l'algorithme, celui-ci calcule ses caractéristiques acoustiques et les compare avec celles de la base de données (Figure 1.1). Par l'utilisation d'une méthode de comparaison appropriée, l'empreinte d'un document dégradé (ex : par compression) peut tout de même être identifié comme étant une version dite « co-dérivée » [31] du document original dont la signature est stockée dans la base de donnée. Deux empreintes sont dites co-dérivées si elles ont été calculées à partir d'un même contenu ayant pu subir quelques altérations(bruit, compression, coupures, ...). Notons que deux chansons d'un même auteur ne sont pas co-dérivées. De même, une reprise d'une chanson n'est généralement pas un co-dérivé de l'original. La comparaison d'empreinte s'effectue généralement à partir de quelques secondes de signal extraites à n'importe quel moment. On calcule alors, pour cet extrait, une suite de valeurs appelées « sous-empreintes ». Si une suite de sous-empreintes stockée dans la base est suffisamment similaire à celle de l'extrait, l'extrait est identifié.

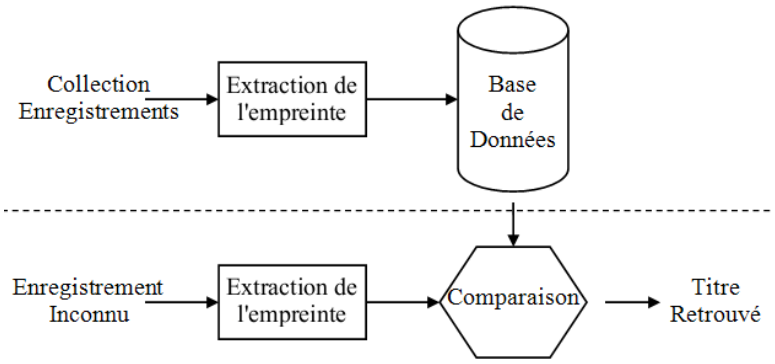


FIG. 1.1 – Identification par empreinte

1.4 Les applications potentielles

La reconnaissance audio basée sur l'empreinte a la particularité d'identifier un contenu si la signature de celui-ci est stockée dans la base de données, et ce, même après altération du contenu. Ce genre de technique trouve sa place au sein de nombreuses applications dont voici une liste non exhaustive.

Vérification de la qualité : Dans de nombreuses applications, l'intégrité d'un contenu audio doit être vérifiée avant d'utiliser le document. Par exemple, les distributeurs de

musique en ligne se doivent de vendre des documents numériques de bonne qualité et, par conséquent, d'adapter le taux de compression en fonction de la qualité du document compressé. En effet, la similarité de deux empreintes issues d'un même contenu caractérise la qualité du signal [19].

Management des réseaux de distribution : Les distributeurs tels que radios ou télévision doivent s'acquitter des droits d'auteur sur la plupart des documents. La technique d'identification par empreinte peut aider à identifier de manière automatique tout contenu musical diffusé à la radio où à la télévision afin de calculer ou vérifier l'acquittement des droits d'auteurs à la SACEM.

Surveillance des réseaux internet non protégés : Les réseaux d'échanges entre utilisateurs ont amené de nombreux documents à être exploités illégalement. L'industrie du disque a donc commencé à mettre en place des techniques de filtrage basées sur le nom du document. Mais ce genre de mesure s'est vite trouvé limité. La recherche d'empreintes audio peut alors analyser les documents transitant sur ce type de réseaux non protégés afin d'identifier les documents normalement soumis à des droits d'auteurs.

Gestion de droits d'auteurs : A la fin d'un échange sur réseau non protégé, l'utilisateur dispose de documents audio totalement libres d'utilisation. L'idée proposée dans ce travail est d'intervenir au sein de chaque appareil afin d'interdire la lecture de contenus audio acquis illégalement.

1.5 La DRM analogique

En effet, de nos jours, le téléchargement de fichiers est devenu un acte courant. Cependant, la majeure partie des téléchargements de fichiers multimédia concerne des documents normalement soumis aux droits d'auteurs. Il est donc normal que les ayant droits de ces œuvres (auteurs, éditeurs, producteurs) se préoccupent de la protection de ces documents. Aussi, progressivement, des dispositions ont été créées afin de protéger l'ensemble de ces contenus. Parmi elles, les DRM[1] comprennent un ensemble de technologies permettant de protéger les droits d'auteurs en chiffrant les contenus et en n'autorisant qu'un accès et une

[1] Digital Rights Management

utilisation spécifique du document en fonction des droits associés, en limitant, par exemple, le nombre de copies ou la lecture sur n'importe quel appareil.

Cependant, les techniques de DRM actuelles doivent faire face à des pirates ingénieux trouvant sans cesse des moyens de contournement des protections. Ces techniques de contournement peuvent être assez élaborées, par décryptage numérique des protections du contenu, ou plus simplement par capture analogique et ré-encodage numérique de l'œuvre, c'est l'« analog hole ». Ces contournements permettent donc de s'affranchir des protections DRM associées à un contenu. Par conséquent, les DRM n'empêchent en rien l'apparition d'œuvres soumises aux droits d'auteurs sur les réseaux d'échanges.

Enfin même si les techniques de DRM étaient parfaitement efficaces, elles ne pourraient pas protéger les œuvres produites avant la mise en place de ces techniques et donc déjà échangées et copiées.

En 1999, les principaux éditeurs phonographiques regroupant plus de 180 professionnels de la musique, parmi lesquels figurent les cinq grands groupes de l'industrie du disque (Universal Music, BMG, Sony Music, Warner Music et EMI) se sont regroupés pour créer un consortium appelé la Secure Digital Music Initiative (SDMI [12], initiative de sécurisation des musiques au format numérique). Le but de cette association était de définir des standards et mettre au point un système de protection de la musique au format numérique pour la distribution de contenus protégés sur Internet dans le respect des droits d'auteurs associés. La principale production du consortium a été un système DRM, les inconvénients de tels systèmes ayant été exposés. Ce système DRM a été mis en œuvre pour la première fois dans le cadre du format musical de Microsoft, WMF (Windows Media Format).

La SDMI a aussi proposé une solution de protection à base de watermarking. Cette technique impose l'ajout d'une marque digitale (watermark) au contenu audio, sans altération significativement audible de sa qualité et qui permette de l'identifier. Cependant cette stratégie pose plusieurs problèmes. Le premier concerne le respect de la vie privée et des informations personnelles car cette technique permet d'associer un utilisateur aux musiques qu'il achète (tout comme la DRM). De plus, la marque ajoutée au contenu peut être, par des techniques très simples de traitement du signal, modifiée sans altérer significativement la qualité du signal. Modifier ou supprimer la marque engendre alors la non-reconnaissance du document

et permet sa lecture malgré les droits d'auteur. Cette solution censée prévenir le piratage a été mise à l'épreuve en 2001 par un concours de piratage à l'initiative de SDMI et a été prise en défaut. Cette expérience a donc abouti à un échec, au moins partiel.

Actuellement, pour lutter contre le piratage, la méthode qui prime est la dissuasion. Cette méthode n'est pas technique. Elle consiste à condamner les personnes qui téléchargent des contenus normalement soumis à des droits par de lourdes amendes, voir des peines de prison avec sursis ou une coupure de la connexion ADSL. Même si les sites de vente de musique en ligne permettent d'avoir accès à des contenus audio à moindre prix par rapport aux CD, le piratage par échange sur réseaux Peer To Peer reste important. Il est donc nécessaire d'apporter une nouvelle approche pour répondre à ce problème.

Nous proposons, dans cette thèse, une solution alternative ou complémentaire aux DRM afin de contrôler l'utilisation de documents audio et faire respecter les droits d'auteur : l'ADRM (Analogic Digital Rights Management). Comme son nom l'indique, il ne s'agit plus de gérer les droits de manière numérique mais analogique, par identification d'un document audio à partir de ses caractéristiques perceptuelles. Cette donnée ne peut donc pas être piratée ou modifiée sous peine de devoir détériorer le signal au point qu'il devienne inécoutable.

Notre technique d'identification par empreinte serait donc la pierre angulaire d'un scénario de contrôle de la lecture de contenus audio introduit sous forme de plug-in au sein de chaque machine. Ce plug-in reprend le principe de l'empreinte digitale à l'entrée d'un hall pour autoriser la lecture de documents audio uniquement si l'utilisateur a déjà apporté la preuve de la possession de l'original.

1.6 Les paramètres de reconnaissance

Les contraintes imposées à une méthode d'identification basée sur les empreintes dépendent principalement de l'utilisation qu'on en fait. Cependant, l'IFPI (International Federation of the Phonographic Industry) et la RIAA (Recording Industry Association of America) ont décrit un ensemble de propriétés qui font référence pour l'évaluation et la comparaison d'algorithmes d'empreintes audio [50] :

Efficacité. Cela correspond à l'ensemble des résultats d'identification à savoir, le nombre d'identifications correctes, de non-identification (faux négatifs), et de mauvaises identifications (faux positifs).

Robustesse et invariance. C'est la faculté à identifier un document audio, indépendamment du fait qu'il ait été fortement compressé ou qu'il ait subi d'autres altérations (décalage temporel, parties manquantes, ...). A noter que si l'identification d'un document donne accès à sa lecture, la non-reconnaissance d'un document dont l'empreinte est dans la base de données est équivalente à un refus de service. Cette propriété désigne aussi la faculté d'identifier une chanson à partir d'un court extrait de quelques secondes pris n'importe quand dans le signal original (granularity ou cropping). Ceci implique de devoir synchroniser le processus de calcul de l'empreinte.

Fragilité. Certaines applications nécessitent de ne pas reconnaître le contenu ayant subi certaines altérations. L'empreinte correspondant à un contenu co-dérivé doit être invariante aux transformations préservant le contenu. Cependant, elle ne doit pas permettre de l'identifier s'il a été soumis à de fortes distorsions. C'est donc l'inverse de la robustesse.

Complexité. C'est le coût de calcul requis pour l'extraction de l'empreinte, l'espace mémoire requis pour stocker une empreinte ainsi que la complexité de recherche de l'empreinte dans la base de données.

Sécurité. C'est la vulnérabilité de l'algorithme aux tentatives de cracking. En contraste avec la robustesse, ce sont les manipulations du contenu nécessaires pour berner l'algorithme d'identification d'empreinte.

Cependant, améliorer les performances vis à vis d'une contrainte peut parfois entraîner la chute de performance d'une autre. Par exemple, l'empreinte doit contenir suffisamment d'information pour être discriminante mais doit, à l'opposé, avoir un faible coût de stockage. En fonction de l'application, il est nécessaire de privilégier certaines propriétés, mais pas toujours au détriment des autres. Pour notre application par exemple, il sera important d'être robuste à la compression tout en étant capable de reconnaître une œuvre musicale à partir d'un extrait de courte durée.

1.7 Nos contributions

Ce travail de recherche apporte une contribution à l'état de l'art en matière de segmentation de signal audio, de construction de signature robuste et de détermination de score d'identification. En effet après avoir analysé les méthodes existantes en matière de caractérisation et d'identification audio, nous nous somme orienté vers une nouvelle approche pour la définition d'empreintes audio. Nous avons donc développé une méthode en adéquation avec les contraintes de stockage, de robustesse et d'efficacité imposées par notre application. Nous avons également étudié les méthodes de comparaison de chaînes pour nous orienter vers une technique qui soit adaptée à la variation de l'empreinte vis à vis des altérations de contenu. Nous avons enfin proposé un scénario d'utilisation de cette technique d'identification et développé un démonstrateur qui simule la gestion de documents audio dans le respect des droits d'auteurs.

1.8 Organisation de la thèse

Cet ouvrage est structuré de la manière suivante :

Chapitre 2 - Propriétés d'un signal audio. Cette thèse introduit tout d'abord ce qu'est un signal audio analogique ou numérique ainsi que les algorithmes les plus courants de compression audio. Dans ce même chapitre, j'exposerai un état de l'art des principaux travaux de recherche permettant de caractériser et de décrire un contenu musical. Suivant l'application, nous remarquerons qu'un contenu audio peut très bien être caractérisé de manière subjective et sémantique mais aussi de manière acoustique et perceptuelle.

Chapitre 3 - Méthodes d'identification existantes. Nous étudierons dans ce chapitre les principales heuristiques utilisées dans le cadre de l'identification de fichiers audio. Ainsi, les principales méthodes d'extraction d'empreinte seront tout d'abord exposées. Nous décrirons ensuite les techniques d'identification de documents audio basées sur la reconnaissance d'empreintes.

Chapitre 4 - Empreinte audio. Dans ce chapitre, j'expliquerai tout d'abord les pistes que nous avons étudié afin d'extraire une empreinte à partir d'un fichier audio. Puis j'exposerai notre proposition finale.

Chapitre 4 - Identification audio. L'algorithme utilisé afin de déterminer une signature potentiellement proche sera présenté en premier lieu. Puis nous expliciterons les critères utilisés pour savoir si cette empreinte correspond à un contenu audio connu du système.

Chapitre 6 - Analyse des résultats et discussion. Ce chapitre sera décomposé en deux étapes distinctes. Tout d'abors nous exposerons les expérimentations réalisées afin d'évaluer la méthode d'extraction d'empreinte et d'étudier ses propriétés. Nous évaluerons ensuite l'efficacité de notre méthode de reconnaissance au travers de différents tests. Ces résultats seront comparés avec les méthodes existantes dans ces domaines.

Chapitre 7 - Scénario de contrôle de l'utilisation. Le scénario de mise en œuvre de la technique d'identification permettant de contrôler la lecture de documents audio sera expliqué dans ce chapitre. Je commencerai par introduire le cadre applicatif et expliquerai ensuite les cas d'utilisation.

Chapitre 8 - Conclusion générale. Enfin, je discuterai des résultats obtenus en rapport avec l'objectif initial ainsi que des perspectives ouvertes par celui-ci.

2

Le Signal Audio

Comprendre les particularités de l'audition humaine, c'est mieux comprendre les réflexions qui ont mené aux différentes techniques de traitement du signal audio comme par exemple la numérisation et plus particulièrement la compression.

Sommaire

2.1 Structuration d'un signal audio

2.1.1 Perception du son

Le son que nous entendons est le fruit de vibrations se propageant dans l'air et interceptées par notre capteur naturel, l'oreille, à la manière d'une parabole. Cependant, notre appareil auditif ne perçoit les sons que s'ils sont compris dans une gamme de fréquences allant de $20Hz$ à $20KHz$ environ. Plus précisément, la bande fréquentielle que capte le mieux l'oreille humaine varie entre $2KHz$ et $6KHz$ puisque l'impression de l'intensité sonore diffère suivant la fréquence du signal sonore perçu. En effet, les niveaux de sensibilité (seuil d'audition minimal) et de douleur (seuil maximal) ne sont pas constant et varient en fonction de la fréquence (Figure 2.1).

De plus, dans la partie centrale du champ d'audition où elle est la plus sensible, l'appareil auditif humain arrive à déceler une infime variation de l'intensité du niveau sonore entre deux sons séparés de seulement $3Hz$. Cependant, la perception d'un signal audio de faible intensité sera modifiée par la présence d'un autre signal audio très intense. Ce son de volume plus élevé pourra même empêcher totalement la perception de sons de faible puissance sonore, c'est ce que l'on appelle le phénomène de masquage (Figure 2.2). Ce phénomène se manifeste dans

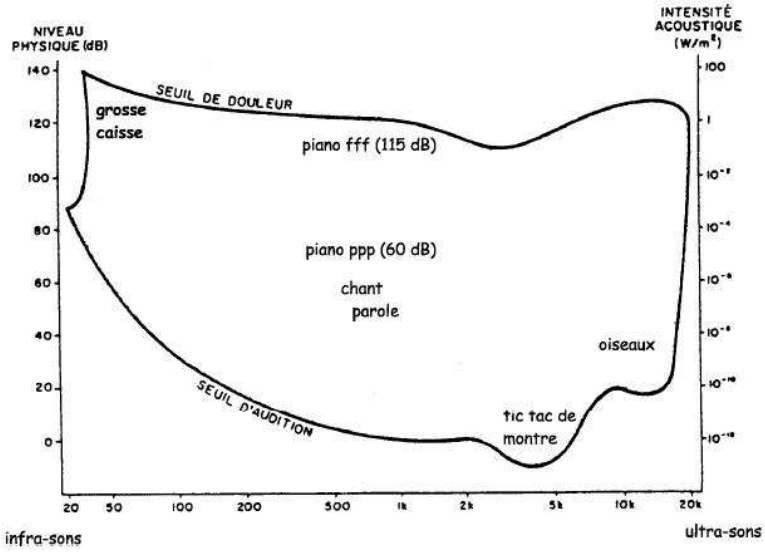


FIG. 2.1 – Courbe de Fletcher et Munson

une plage de fréquence autour du signal sonore intense (masquage fréquentiel) et pendant toute la durée de ce signal et même un peu au delà (masquage temporel). L'exemple le plus courant est celui d'un avion passant au dessus d'un nid d'oiseaux. Cet évènement de volume sonore très élevé empêche alors totalement la perception du chant d'oiseaux et se prolonge quelques instants après le passage de l'avion le temps que notre oreille se réadapte progressivement à l'ambiance sonore plus faible.

2.1.2 La numérisation

L'objectif de la numérisation d'un signal audio est de convertir ce signal en une séquence de nombres binaires, pouvant être traités par informatique. Cela s'effectue en mesurant l'amplitude de l'onde produite par le son à des intervalles de temps réguliers. On peut alors décomposer la numérisation en deux étapes :

Échantillonnage

Tout d'abord, l'échantillonnage est le fait de découper de manière régulière le signal analogique (Figure 2.3) et de prélever, en quelque sorte, une image instantanée du signal à chaque échantillon. Ainsi, une séquence d'échantillons successifs donne une représentation de la forme de l'onde de la même manière que les images d'un film

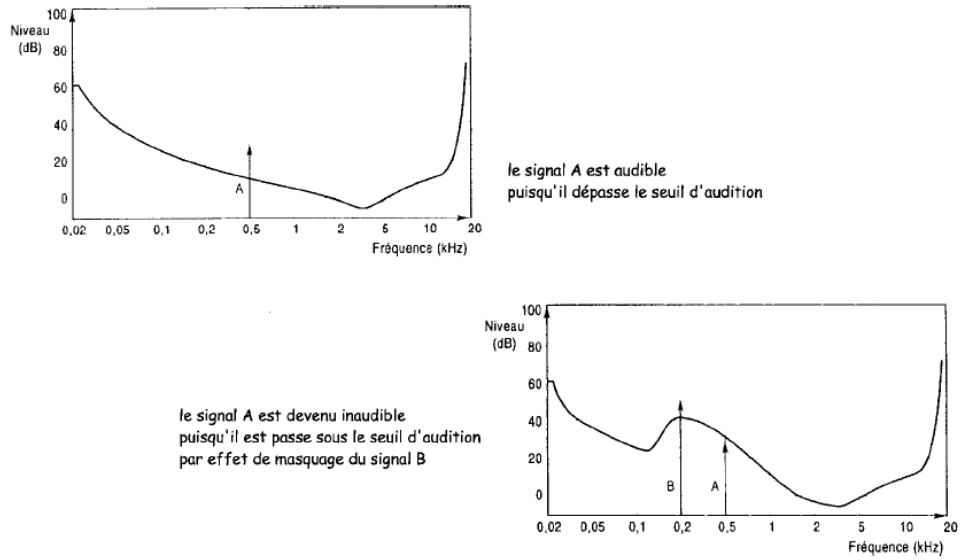


FIG. 2.2 – Masquage des sons perçus

projetées rapidement donnent l'illusion du mouvement.

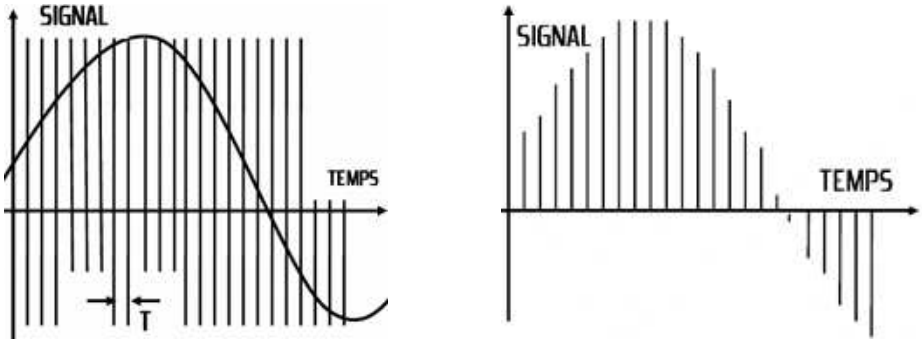


FIG. 2.3 – Technique d'échantillonnage

Afin de représenter fidèlement le signal, il est impératif de prélever un nombre suffisant d'échantillons à chaque seconde. Comme on peut le voir dans la Figure 2.4, si on prélève trop peu d'échantillons par rapport à la période du signal, l'allure de la forme d'onde ainsi reconstruite sera différente de la forme d'onde d'origine. Afin d'éviter ce phénomène d'aliasing, et selon le théorème de Shannon [52], la fréquence d'échantillonnage (nombre d'échantillons par seconde) doit être au moins égale à deux fois la fréquence maximum composant le signal à numériser. Il faut donc définir une

bonne période d'échantillonnage qui permette de restituer toutes les fréquences du signal. Or, la fréquence maximale que puisse entendre une oreille humaine est de $20KHz$. La fréquence d'échantillonnage des CD audio doit alors être supérieure à 40KHz et est en général fixée à $44,1KHz$.

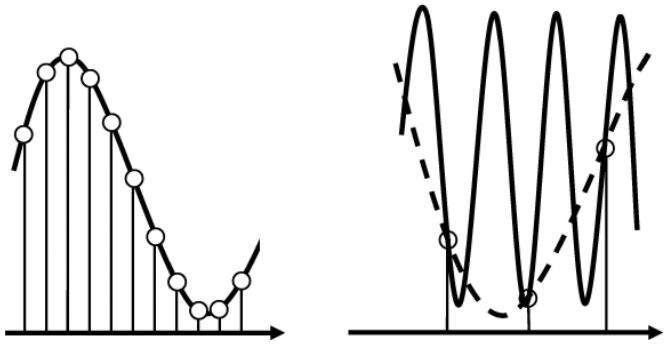


FIG. 2.4 – Principe du théorème de Shannon

Quantification

Ensuite, la chaîne d'impulsions est codée. La quantification consiste à évaluer l'amplitude de chacun des échantillons du signal et à placer ces amplitudes sur une échelle de valeurs à intervalles fixes (Figure 2.5). Cette échelle est définie suivant l'amplitude maximale et minimale possible et divise cet écart d'amplitudes en une série de paliers de hauteur égale. Ce procédé permet donc d'attribuer à chaque échantillon un mot binaire en fonction du palier auquel il correspond. En binaire, le nombre de pas de quantification sera alors égal à 2^n, avec n le nombre de bits utilisés pour représenter chaque échantillon. En ce qui concerne les CD audio, cette valeur est égale à 16 bits, soit $2^{16} = 65536$ paliers.

Par conséquent, le stockage d'une minute de signal audio stéréo, codé sur 2 octets par voie et échantillonné à $44,1KHz$ nécessitera : $60 * 2 * 2 * 44100 = 10,6Mo$

2.1.3 La représentation fréquentielle

Un phénomène physique dépendant du temps est décrit par un ou plusieurs signaux. Cependant, on ne peut interpréter ces signaux de façon simple. Le problème est donc de trouver une manière de décrire leur comportement. Plus particulièrement, le son est com-

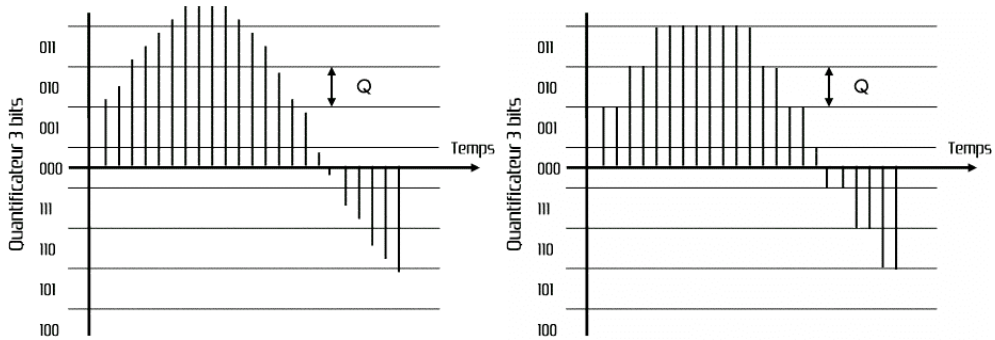


FIG. 2.5 – Technique de quantification

posé d'une somme de signaux de fréquences, amplitudes et phases différentes. L'analyse spectrale regroupe un ensemble de méthodes permettant d'analyser un signal dans le domaine fréquentiel. Elle nous permet notamment de déterminer la fréquence fondamentale ainsi que les fréquences dites harmoniques qui composent le signal sonore (Figure 2.6).

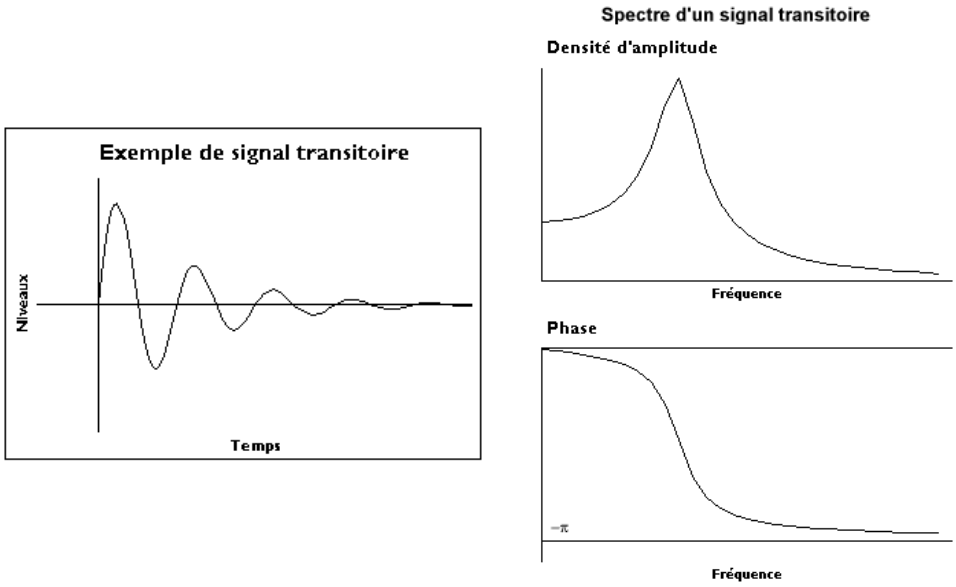


FIG. 2.6 – Représentation spectrale

L'outil mathématique généralement utilisé pour effectuer cette décomposition spectrale continue est la transformée de Fourier. Cette technique permet en effet de décrire la puissance des fréquences qui composent le signal audio.

Un signal audio peut donc être caractérisé par ses propriétés temporelles et fréquentielles.

2.2 Caractérisation d'un signal audio

Le nombre et la variété de contenus audio ont ouvert la voie à de nombreux domaines de recherche en rapport avec des applications à vocation industrielle. Ces applications se heurtent généralement aux mêmes problématiques de recherche : « comment caractériser un document audio ? » ou « quelle propriété ou caractéristique du signal va permettre de résoudre le problème posé par l'application ? ». Comme introduit en section 1.2, un signal audio peut être caractérisé à partir de propriétés appartenant à différents domaines d'abstraction, acoustiques ou culturels par exemple. Une propriété acoustique signifie que cette information est obtenue à partir de l'analyse du fichier audio sans référence à une information textuelle [48]. Par conséquent, il s'agit d'une information obtenue à partir du signal. Or, parmi les propriétés acoustiques du signal, il existe trois facteurs d'échelle permettant de définir la manière d'extraire ces descripteurs.

Descripteurs Globaux : Les descripteurs globaux regroupent un panel de propriétés qui décrivent un document audio dans sa totalité. Ce qui signifie que chacune de ses propriétés ne peut être extraite qu'à partir de l'étude de toute la durée du signal audio. Le genre, le rythme, ou encore l'humeur sont, par exemple, des descripteurs globaux. On remarque que ces descripteurs ont une réelle signification pour un utilisateur et ne nécessitent aucune connaissance spécifique. De plus, ces propriétés sont définies par des termes linguistiques et non par des valeurs. En effet le genre peut avoir comme définition "rock", le rythme "lent et l'humeur "mélancolique". Par conséquent, ce genre de descripteur est très utilisé dans les catalogues des distributeurs ou moteurs de recherche.

Descripteurs Intermédiaires : Les descripteurs de niveau intermédiaire regroupent des propriétés résultant de l'analyse de quelques secondes de signal audio. Cela permet en général de détecter certains phénomènes acoustiques à cette échelle. On peut donc segmenter un signal audio par détection de texture ou ruptures ce qui permet de séparer le signal en parties bien distinctes. Chaque partie ainsi extraite est classée dans une

des catégories, définies en fonction de l'application. Ce genre de descripteur est par exemple utilisé pour segmenter les émissions radio en trois parties (voix-jingle-musique) ou effectuer un résumé d'extrait musical (chant-instrumental et introduction-couplet-refrain).

Descripteurs Locaux : Les descripteurs locaux sont calculés à partir de quelques dixièmes voir millisecondes du signal et ne sont en général compréhensibles que pour des experts. On y retrouve le timbre, la percussivité, la hauteur, ... Ces propriétés sont généralement utilisées dans des applications nécessitant la gestion de bases de données de grande taille.

2.2.1 Propriétés acoustiques globales

Un exemple typique d'information acoustique est le tempo, c'est à dire le nombre de pulsations par seconde. L'extraction des pulsations et du tempo a longuement intéressé la communauté du traitement du signal et certains systèmes obtiennent à l'heure actuelle des performances intéressantes[51]. D'autres informations plus complexes peuvent également être extraites comme la structure du rythme. Derrière le rythme, d'autres perceptions virtuelles ont été sujettes à de nombreuses investigations comme la percussivité, la reconnaissance des instruments[29] , ou encore l'énergie perçue[61], voir même l'humeur[42]. Cependant, à notre connaissance, aucune application commerciale n'utilise encore ces descripteurs. Mais nul doute que l'efficacité de ceux-ci s'améliorera dans les prochaines années grâce à l'attention croissante dont ils font l'objet.

2.2.2 Propriétés acoustiques de niveau intermédiaire

Les descripteurs globaux sont décrits par une valeur ou un terme unique à propos de la totalité d'un titre musical. De plus, ils ne dépendent pas d'autres paramètres comme par exemple, l'instant auquel l'information est calculée. Inversement, les descripteurs de niveau intermédiaires sont calculés de façon régulière sur quelques secondes du signal. Ce genre de descripteurs évoluant au cours du signal sont très utilisés pour gérer de larges collections de titres. Le contour de l'enveloppe, ou l'extraction du pitch, peuvent par exemple être utilisés pour des applications de query-by-humming[5] qui consistent à retrouver un titre à partir du

chantonnement de l'utilisateur. A un degré supérieur, un document audio peut être analysé afin d'en extraire une structure et de trouver les répétitions de refrains ([49]) dans le but de déduire automatiquement un résumé du fichier audio.

La norme Mpeg-7 est un standard normalisant la représentation de ces descripteurs d'une manière syntaxique mais les résultats dépendent réellement de la méthode utilisée pour extraire ces caractéristiques.

2.2.3 Propriétés acoustiques de bas niveau

Le timbre est probablement la propriété de bas niveau la plus difficile à définir et à caractériser [30]. La définition du timbre est vague. Il s'agit en fait de toute caractéristique acoustique de bas niveau qui ne soit ni le pitch ni l'intensité. La perception du timbre est associée à la structure du spectre du signal et, par conséquent, à sa représentation dans le domaine fréquentiel. Pour cette raison, la transformée de Fourier est un des outils les plus utilisés dans l'analyse du timbre et de l'évolution temporelle d'un signal audio en général. Cependant, le spectre ou tout autre décomposition temps-fréquence ne peuvent pas être utilisés en tant que descripteurs à cause de leur dimensionnalité élevée. C'est pourquoi l'extraction du timbre est basée sur l'extraction de caractéristiques bas-niveau correspondant à des propriétés perceptuelles. Les études psychoacoustiques ont mis en évidence certains paramètres comme le taux de passages par zéro[2][26], le spectral centroïd[23], spectral loudness[33], roughness[44] qui sont considérés comme des descripteurs de timbre. Les descripteurs à base de MFCC[3] [43, 9, 35] ont été très largement utilisés dans le domaine de la recherche d'information musicale[4]. Ces descripteurs ont d'abord été utilisés dans la reconnaissance de genre et de voix et sont considérés comme des outils de base dans ces domaines.

2.3 Méthodes de compression

Dans la section 1.6, nous avons introduit les notions de robustesse et d'invariance d'une méthode d'identification comme la capacité à identifier un extrait audio, indépendamment

[2]Zero Crossing Rate
[3]Mel Frequency Cepstral Coefficients
[4]MIR : Music Information Retrieval

du fait qu'il ait été altéré ou dégradé. Or, ceci est l'un de nos objectifs principaux. En effet, notre application nous impose d'être capable de reconnaître un contenu musical dont on a acquis les droits d'auteurs afin de pouvoir l'écouter, et ce, malgré l'influence de certaines dégradations telles que la compression.

Tout acheteur de CD audio a le droit de compresser un CD légalement acquis afin d'en faire une copie de sauvegarde à faible espace de stockage. Il doit donc pouvoir, s'il le souhaite, écouter la version compressée d'un contenu original. Notre méthode doit donc être robuste à de forts taux de compression (pouvant aller de 320kbps[5] à 32kbps).

On rappelle donc que les contraintes de restitution d'un signal analogique de bonne qualité pour l'oreille humaine (Section 2.1.1) ont permis de définir le format standard de stockage du son sur CD à savoir :

- Fréquence d'échantillonnage à 44,1 kHz
- Quantification des échantillons sur 16 bits
- Voie gauche et droite pour le son en stéréo

D'après ces informations, une minute de musique sera alors stockée sur plus de 10 Mo. Même avec les possibilités croissantes des médias de stockage, la compression reste inévitable que ce soit pour stocker un nombre plus important de contenus sur un média de style CD ou DVD ou pour un transfert/téléchargement plus rapide de ces contenus.

Parmi les techniques de compression de fichier audio, nous nous intéresserons ici aux techniques dites destructrices. Une compression destructrice est une compression réalisée en perdant de l'information. Cela signifie que si l'on décompresse le signal compressé à l'aide d'une telle technique, on ne retrouvera pas le signal de départ. Parmi les techniques de compressions destructrices, on a essentiellement des méthodes qui exploitent les propriétés de l'oreille humaine (Section 2.1.1) qui ne distingue que les sons entre 20Hz et 20kHz avec une sensibilité maximale entre 1kHz et 5kHz. Ainsi, la compression vise à analyser le signal afin de déterminer les sons inaudibles en vue de les supprimer, d'où la notion de compression destructrice. Les principaux formats de compression sont les suivants :

MP3

[5]Kilo Bits Par Seconde

La permière étape de la compression MP3 consiste à supprimer les fréquences inaudibles en fonction de la courbe 2.1. C'est à dire que suivant le taux de compression, les fréquences supérieures ou inférieures à une certaine fréquence seront supprimées (par ex : $F > 15kHz$). Puis, le phénomène de masquage (Section 2.1.1) fait que certaines fréquences sont inaudibles du fait de la présence de fréquences proches d'amplitudes plus intenses. De ce fait, ces fréquences inaudibles seront elles aussi supprimées. La compression MP3 utilise aussi une technique appelée "réservoir d'octets". Certains passages d'une musique ne peuvent pas être compressés sans diminuer la qualité d'écoute. Ainsi, un petit réservoir d'octets permet de ne pas compresser ces passages en utilisant les octets économisés lors de l'encodage de parties à un taux supérieur. De plus, la compression MP3 utilise le fait qu'en dessous d'une fréquence donnée, l'oreille humaine n'arrive pas à localiser la provenance du son. Ces fréquences seront alors enregistrées en mono accompagnées d'informations complémentaires permettant de restituer un minimum d'effets, c'est ce qu'on appelle le "joint stereo". La technique du Codage utilise le fait que l'oreille humaine ne puisse pas distinguer la différence de fréquence entre deux sons quasi identiques. Ainsi, si on a une suite de fréquences très proches, on ne codera plus chaque fréquence, mais seulement la valeur de référence ainsi que le nombre de répétitions. Enfin, le codage d'Huffman est un algorithme de codage agissant à la fin de la compression basé sur des codes de longueur variables et la probabilité d'apparition d'un événement. Ainsi, une compression classique réduira la taille du fichier audio à la hauteur de 1 : 12.

OGG

Ogg Vorbis est un format de compression avec perte. La différence principale avec le MP3 réside dans le fait que ce dernier utilise des séquences de bits fixes pour coder les passages audio. A l'inverse, Vorbis est un format à débit variable utilisant plus ou moins de bits suivant les passages audio à compresser. Par exemple, un passage sans son ou ayant uniquement un tempo de batterie nécessitera un nombre de bits moins important que des passages contenant de nombreuses fréquences aiguës. Ceci explique par conséquent un meilleur taux de compression pour Vorbis mais une complexité et un temps de traitement plus important.

Les autres formats

Il existe plusieurs normes de compression audio (MPEG1, MPEG2, ...). Ces normes sont établies par des organismes de normalisation qui établissent des formats de compression donnant parfois lieu aux dépôts de brevets. Parmi les autres formats audio compressés, on trouve le mp3PRO, le WMA qui constituent les formats les plus connus mais il existe aussi les formats AAC, VQF ...

Pour résumer, un encodage MP3 à 64kbps réduit de 25 fois la taille du fichier audio mais perd considérablement en qualité d'écoute. Or, ce type de traitement est communément utilisé dans l'échange et le stockage de documents audio. Lors de la conception de l'empreinte, il sera important de prendre en considération de telles dégradations afin de proposer une empreinte qui y soit peu sensible.

3

Identification de documents audio

Sommaire

3.1 Conception d'identifiants audio

La génération d'empreinte audio consiste à calculer une courte signature (l'empreinte) dérivée des caractéristiques perceptuelles d'un extrait audio et qui permette de l'identifier. Pour cela, les caractéristiques d'un ensemble de documents sont stockées dans une base de données. Quand un extrait inconnu est présenté, ses caractéristiques sont calculées et comparées avec celles présentes dans la base de données. Si la comparaison satisfait certaines conditions dépendantes de la méthode utilisée, alors le document est identifié comme étant dérivé du même contenu. Ce domaine de recherche implique donc le traitement du signal et l'analyse musicale d'un côté ainsi que l'indexation et la reconnaissance d'identifiants d'un autre côté.

Par conséquent, indépendamment des approches retenues pour calculer, rechercher, ou comparer la signature, l'architecture d'un tel système peut être divisée en deux modes opératoires :

1. Tout d'abord, le module de construction de la base de données extrait des propriétés acoustiques d'un ensemble de contenus et calcule, pour chacun d'eux, une signature compacte et unique le caractérisant. Cette représentation compacte est alors stockée dans la base de données et associée à un ensemble de metadonnées ou tags pour chaque document.
2. Ensuite, le processus d'identification calcule la signature d'un court extrait inconnu et la compare avec celles de la base de données afin de retrouver une signature qui soit

identique ou très similaire si le signal a été dégradé.

Toutes les implémentations actuelles suivent ce schéma de fonctionnement mais diffèrent au niveau des propriétés acoustiques utilisées pour caractériser le signal audio ainsi qu'au niveau des algorithmes d'indexation et de comparaison utilisés.

La première étape consiste donc à analyser le signal audio et à obtenir un ensemble de valeurs caractéristiques de l'extrait audio. Or, la plupart des méthodes se basent sur la représentation fréquentielle du signal pour calculer ces valeurs, et plus particulièrement sur la transformée de Fourier. Cependant, calculer une telle transformée sur le signal complet n'apporte pas d'information temporelle sur les instants où interviennent les fréquences. Il est donc nécessaire de diviser le signal audio en un ensemble d'intervalles temporels afin d'avoir une représentation fréquentielle sur chacun de ces intervalles. Notons qu'appliquer la transformée de Fourier discrète sur chaque intervalle revient à supposer que le signal est stationnaire sur celui-ci (ce qui n'est évidemment pas le cas). Pour chaque intervalle, les fréquences composant le signal seront analysées afin d'en déduire une valeur de sous-empreinte. L'empreinte finale sera alors composée de la concaténation par ordre chronologique des valeurs de sous-empreinte correspondant à chaque intervalle.

3.1.1 Séparation du signal en intervalles

La méthode couramment utilisée pour découper le signal audio en intervalles s'appelle le « fenêtrage ». La plupart des méthodes de calcul d'empreinte audio se basent sur la sélection d'intervalles par fenêtrage. L'idée est de diviser le signal en une succession de segments temporels de taille fixe à partir du début de l'extrait. Mathématiquement, cette opération de troncature revient à multiplier le signal x(t) par une fenêtre rectangulaire f(t) de durée T. Cependant, cette multiplication dans l'espace temporel correspond à un produit de convolution dans l'espace fréquentiel entre le spectre du signal et le spectre en sinus cardinal de la fenêtre rectangulaire. Il en résulte alors une déformation du spectre causée par les ondulations du sinus cardinal. Il est donc nécessaire de réaliser une troncature moins abrupte, afin d'éviter ces effets indésirables appelés « étalement spectral ». L'utilisation d'une fenêtre telle que la fenêtre de Hamming rend les transitions aux bords de l'intervalle sélectionné plus

douces. Cependant, sélectionner les intervalles à la suite engendre un nouveau problème. En effet, il y a une perte d'information sur les fréquences aux transitions de deux fenêtres. De plus, cela rend sensible le système aux altérations temporelles du contenu telles que la suppression d'un peu du signal en début de piste. En effet, les intervalles seront alors complètement décalés. Pour minimiser ces problèmes, les intervalles ne se succèdent plus mais se chevauchent en partie, c'est ce que l'on appelle le recouvrement ou « overlap ». La figure suivante résume le principe de la division du signal audio en fenêtres (Figure 3.1).

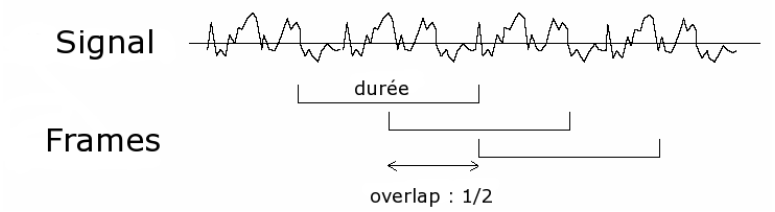


FIG. 3.1 – Technique de fenêtrage

Toutefois, l'utilisation d'un recouvrement entre les fenêtres diminue la sensibilité du système aux décalages temporels sans la supprimer totalement. Prenons par exemple un signal échantillonné à 44KHz, soit 44 échantillons par millisecondes, si on calcule l'approximation des fréquences par transformée de Fourier sur une fenêtre de 40ms, on se base donc sur $44x40 = 1760$ échantillons du signal pour effectuer cette approximation. Dans ce cas, un recouvrement d'un demi revient à sélectionner la nouvelle fenêtre avec 20ms de décalage, soit à la moitié de la précédente. Introduisons maintenant un décalage de 10ms au début du signal. Le calcul de la transformée de Fourier se basera alors toujours sur des fenêtres de 40ms mais dont seulement trois quart des échantillons seront identiques au calcul précédent. Ce qui veut dire que 1320 échantillons seront identiques mais 440 seront différents. Ce type de décalage induira des identifiant différents pour la majorité des méthodes d'identification basées sur le fenêtrage.

Une solution alternative, à priori moins sensible aux décalages, consiste à utiliser un ensemble d'intervalles sélectionnés à des positions particulières du signal. Ce type d'intervalles est par exemple utilisé lors de la détection du rythme des signaux audio. Cela se traduit concrètement par la recherche du début (aussi appelé onset) ainsi que la durée de chaque

partie composant le signal donnant alors une représentation du tempo. La plupart de ces méthodes sont basées sur la détection de changements significatifs dans une ou plusieurs propriétés calculées le long du signal audio. Ils reposent donc sur un vecteur de caractéristiques extrait grâce à l'analyse continue du document audio et résumant l'information rythmique du signal. Par exemple, si le signal audio est analysé dans le plan temps-fréquence, une augmentation de l'énergie de certaines bandes de fréquences indiquera l'apparition d'un changement brusque caractérisé par un onset. Si l'on considère la phase du signal comme un critère de détection d'onset, alors une opposition de phase ou la détection d'irrégularités de la phase indiquera elle aussi un onset. L'utilisation de l'énergie du signal a montré son efficacité dans le cadre de la détection d'onsets pour les signaux avec des changements de notes à forte consonance percussive comme la batterie, puisque l'énergie présente alors un fort gradient [24]. L'information de phase quant à elle permet de détecter les onsets dans des signaux aux sources mixtes et aux transitions moins franches ([21, 25, 39, 51, 57]).

Ansi Klapuri [38] fut un des tout premiers à utiliser l'amplitude de l'enveloppe du signal temporel afin d'y détecter des changements relatifs d'intensité. Hainsworth [28], plus tard, a introduit une technique similaire mais basée non plus sur l'enveloppe du signal temporel mais sur des mesures de distances entre l'énergie des bandes de fréquences résultantes de la Transformée de Fourier. Puis Juan Pablo Bello [4] incorpora à cette méthode la prise en compte de la phase de la Transformée de Fourier.

Alonso [2, 3] propose une méthode basée sur la détection de changements brusques du timbre et des harmoniques composant le signal. Pour cela, une transformée temps-fréquence est appliquée au signal. Le plan fréquentiel est ensuite envoyé vers un filtre à réponse impulsionnelle finie afin de diviser le plan fréquentiel en bandes. Les énergies des bandes fréquentielles sont alors calculées afin d'obtenir un vecteur de pulsations périodiques correspondant à la hauteur de l'énergie spectrale par bande. La détection des onsets est ensuite générée par analyse de l'autocorrélation de ce vecteur de pulsations.

L'algorithme développé par Dixon [17, 18] propose d'estimer le tempo et la durée du rythme musical. Pour cela, la première étape de l'algorithme recherche le début de changements brusques en trouvant des maxima locaux lors de l'analyse de l'amplitude de l'enveloppe du signal temporel. L'algorithme de déduction du tempo calcule alors l'intervalle inter-onset

entre deux évènements successifs et classifie ces intervalles en groupes de durées proches. Il ordonne ensuite ces classes en fonction du nombre d'intervalles qu'elle contiennent et de leur relations. Il obtient ainsi une liste ordonnée d'hypothèses de tempo du signal audio. Il utilise finalement un système multi-agents afin de tester les différentes hypothèses de tempo et trouver l'agent qui prédit au mieux le rythme par rapport au signal fourni.

Tzanetakis [56, 55] propose une méthode basée sur les ondelettes. Dans un premier temps, le signal est segmenté en fenêtres temporelles de 3 secondes avec un recouvrement de la moitié de la fenêtre. Chaque fenêtre est alors décomposée, par transformée en ondelettes, en 5 bandes de fréquences. L'énergie moyenne de chaque bande est calculée afin de déterminer l'autocorrélation de ces énergies. L'autocorrélation maximum est extraite pour chaque fenêtre afin de fournir une estimation du tempo. Le tempo final est défini par le résultat d'un filtre médian appliqué à cette estimation (Figure 3.2).

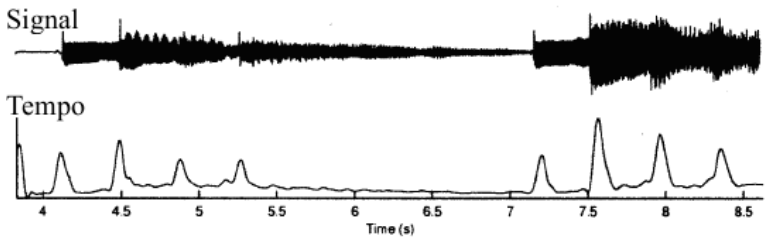


FIG. 3.2 – Détection d'onsets

La détection de rupture peut également être effectuée en analysant le signal sur deux intervalles situées respectivement avant et après un point que l'on suspecte d'être un point de rupture. Si le signal sur chacun des intervalles est décrit par un ensemble de vecteurs, la détection d'onsets revient à l'estimation de changements brusques parmi les ensembles de vecteurs décrivant le signal avant et après le point de rupture suspecté. Dans ce cadre, F. Desobry et al. [15] ont proposé une technique extrêmement novatrice basée sur l'utilisation de noyaux permettant de projeter l'ensemble des vecteurs sur la sphère unitaire d'un espace de grande dimension. F. Desobry, utilise également un SVM une classe pour estimer pour chaque ensemble de vecteurs le support de sa densité de probabilité (i.e. l'ensemble des points ou la densité est supérieure à un seuil). Tout l'intérêt de cette méthode est que l'estimation du support de la densité de probabilité ne requiert par une explicitation de la

densité de probabilité. On évite donc une tache souvent délicate et parfois un peu arbitraire. La distance entre deux ensembles de vecteurs est finalement estimée à partir de la distance entre leurs support de densité de probabilité sur la sphère. Notons que cette technique permet de détecter des changement entre ensembles de vecteurs et est donc applicable à bien d'autres domaines que la détection d'onsets.

3.1.2 Propriétés extraites

Une fois le signal audio séparé en intervalles de temps consécutifs ou sélectionnés à des instants particuliers, la construction d'empreinte se base sur l'extraction de propriétés acoustiques sur chaque intervalle afin d'en déduire une sous-empreinte. La plupart des méthodes dans ce domaine se basent sur des transformées temps-fréquence standard, telle la transformée de Fourier discrète (TFD) [20, 36] qui reste la plus utilisée. Pour autant, d'autres transformées ont été testées telles la transformée en cosinus discret (DCT), la transformée de Walsh-Hadamard [53], la transformée complexe modulée (MCLT) [8, 7], ou encore la transformée en ondelettes (DWT) [13, 41]. A partir de l'espace temps fréquence obtenu, l'idée est d'extraire des caractéristiques afin d'en déduire des sous-valeurs d'empreinte.

Dans ce but, de nombreuses méthodes ont été proposées. Une des plus répandue repose sur l'utilisation des MFCC[6] [43] pour analyser le spectre avec des bandes de fréquences correspondant à l'appareil auditif humain. Les mesures de Spectral Flatness ont été mises en oeuvre par Allamanche [1] car elles permettent d'estimer la qualité de la tonalité dans des bandes de fréquence du spectre. Il en résulte une détection des transitions du signal audio. Le signal est finalement caractérisé par une séquence de valeurs correspondant à la durée de chaque tonalité.

D'autres, comme Li et Hou [41] ont mis en place un dispositif basé sur un algorithme de transformée en ondelettes pyramidales. Leur algorithme calcule la transformée en ondelette à différentes résolutions (d'où la notion de pyramide). Comme le montre la Figure 3.3, un extrait audio inconnu est caractérisé par les coefficients de sa transformée en ondelettes à chaque résolution.

[6] Mel-Frequency Cepstral Coefficients

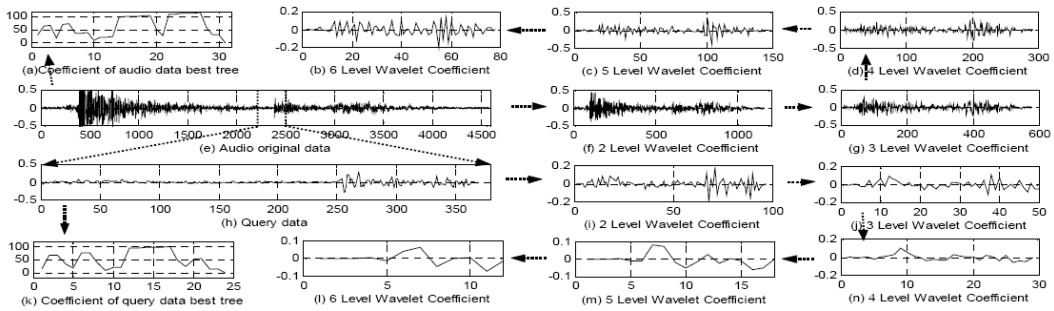


FIG. 3.3 – Caractérisation d'un signal par coefficients d'ondelettes

Avery Wang [58] utilise la méthode des fenêtres recouvrantes pour calculer et analyser le spectre de chaque intervalle temporel afin de trouver et marquer des maximum locaux dans le spectre. Ces marques sont déterminées, comme les onsets, en recherchant des changements brusques et définis par leur coordonnées temporelles et fréquentielles. Le spectre est alors divisé temporellement et fréquentiellement pour créer des zones composées d'un nombre fixe de marques. Ces zones, caractérisées par les positions et fréquences des marques qu'elles contiennent, seront alors comparées à celles pré-calculées pour trouver des zones identiques.

Dans une autre approche, Frank Kruth [40] décrit une technique de génération de très courtes signatures de signaux audio. Tout d'abord l'extrait est envoyé dans un filtre passe-bande linéaire dans le but de simuler grossièrement les effets d'une éventuelle distorsion. Ensuite, le signal temporel est décomposé en intervalles par utilisation de fenêtres recouvrantes. Pour chaque intervalle, l'énergie totale du signal est calculée afin de quantifier ce niveau d'énergie. Il compare ensuite ce niveau de quantification avec le précédent afin d'en déduire un bit par fenêtre traduisant le signe de la différence d'énergie à l'instant t avec l'énergie à l'instant t+1 : $S(k) = sign(x(k+1) - x(k))$ (Figure 3.4). Pour résumer, cette séquence de bits permet de coder la pente de la courbe dessinant la croissance ou la décroissance de l'énergie du signal.

Burges et al. [8] présentent un algorithme de réduction de la dimensionnalité d'un signal appelé Analyse Discriminante de Distorsions[7]. Ils proposent de se baser sur une transformé de Karhunen-Loeve modifiée appelée Oriented Principal Component Analysis[8]. Tout

[7]Distorsion Discriminant Analysis
[8]Oriented Principal Component Analysis

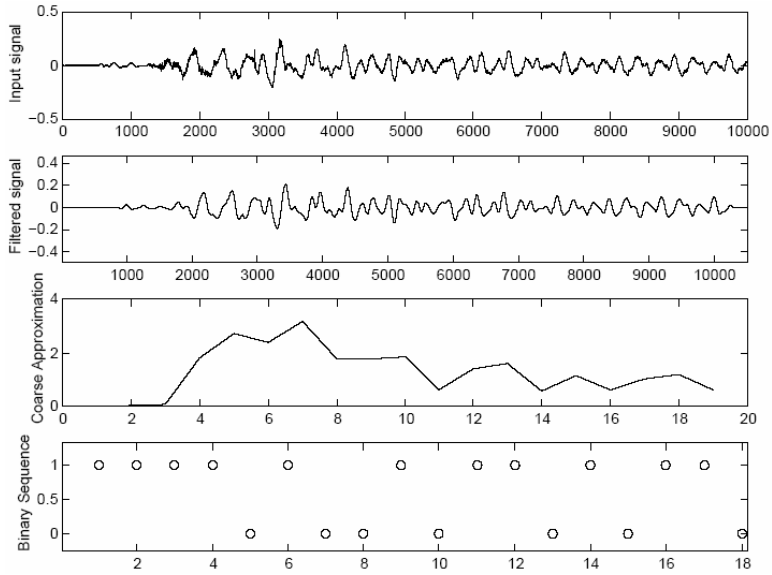


FIG. 3.4 – Empreinte par signe de la courbe d'énergie

d'abord, le signal audio est sous échantillonné à 11,025 KHz, converti en mono, et séparé en intervalles de 23,2ms avec recouvrement de $\frac{1}{2}$ sur lesquels ils appliquent une Transformée Complexe Modulée (MCLT). Il en résulte alors pour chaque intervalle un vecteur de 128 valeurs décrivant le spectre avec une échelle logarithmique. Ils utilisent ensuite l'OPCA pour réduire considérablement la dimensionnalité du signal audio et trouver un ensemble de projections du signal d'entré qui maximise le rapport signal sur bruit. Il est donc nécessaire de disposer d'un ensemble d'apprentissage pour chaque distorsion à prendre en compte. Ensuite, plusieurs niveaux d'Analyse en Composantes Principales Orientées sont combinées afin de créer un réseau de neurones qui extrait pour chaque intervalle un ensemble de caractéristiques robustes aux bruits appris. Après ces couches successives de traitement, on obtient un vecteur de 64 valeurs décrivant 20 secondes de signal. Ce vecteur est généré toutes les 243,6ms.

L'algorithme de Brück [6] applique une transformée de Fourier sur des fenêtres d'une durée de 40ms avec un recouvrement de 1/2. La représentation spectrale est envoyée vers un banc de 8 filtres passe-bandes entre 300 et 2000Hz. L'énergie de chaque bande est stockée sur 16bits, ce qui fait un total de 128 bits par intervalle.

Parmi les article dont s'est inspiré Brück figure celui de Haitsma et Kalker[36]. Dans cet

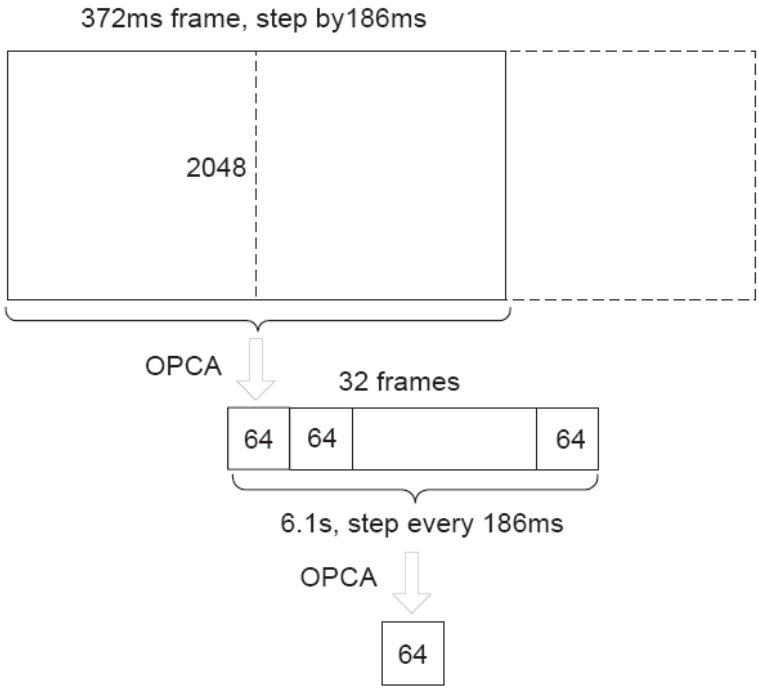


FIG. 3.5 – Analyse en Composantes Principales Orientées

article, les auteurs utilisent des intervalles de 0,37 secondes avec un recouvrement de 31/32 pour découper le signal audio. La sortie de la transformée de Fourier appliquée sur chaque intervalle est envoyée vers un banc de 33 filtres passe-bandes de fréquences de coupure fixées entre 300 et 2000Hz (Figure 3.6). L'énergie de chaque bande fréquentielle est calculée en sortie de chaque filtre. La valeur de sous-empreinte correspondant à chaque intervalle est définie sur 32 bits. Cette séquence de bits, est définie à partir du signe des différences d'énergie calculée entre deux bandes de fréquences consécutives d'un même intervalle ainsi qu'entre deux intervalles consécutifs. Plus précisément, définissons $EB(n,m)$ comme l'énergie de la m^{iene} bande de l'intervalle n et $\Delta EB(n,m) = EB(n,m) - EB(n,m+1)$ la différence d'énergie entre deux bandes successives d'un même intervalle. La valeur du m^{iene} bit de l'intervalle n est alors définie par :

$$F(n,m) = \begin{cases} 1 & Si \quad \Delta EB(n,m) - \Delta EB(n-1,m) \geq 0 \\ 0 & Si \quad \Delta EB(n,m) - \Delta EB(n-1,m) \leq 0 \end{cases} \tag{3.1}$$

Finalement, la valeur de sous-empreinte de chaque intervalle correspond à la concaténation

des 32 bits $F(n,m), m \in \{1,\ldots,32\}$.

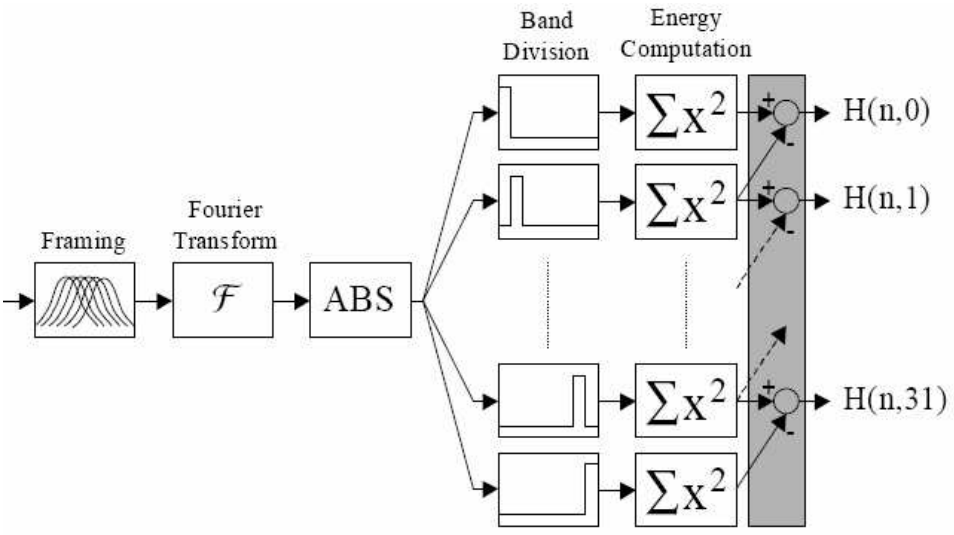


FIG. 3.6 – Empreinte par différences inter-filtres

Dans chacun des cas, l'empreinte finale résulte de la concaténation des valeurs successives de chaque intervalle pour former une séquence de valeurs que l'on va donc comparer avec la base de données pour en rechercher des similitudes.

3.2 Comparaison et Reconnaissance d'identifiants audio

Du choix de la méthode de calcul d'identifiant dépendra le choix de la distance et de l'algorithme d'indexation à mettre en oeuvre pour rechercher rapidement un document audio.

3.2.1 Distances

Comme nous l'avons vu, certaines méthodes caractérisent les documents à l'aide de vecteurs de caractéristiques. On va alors employer une distance qui permet de comparer deux vecteurs entre eux. Pour ce faire, on peut penser à la distance Euclidienne [59, 8] ou dans le cas binaire, à la distance de Hamming [37]. Mihçak et al [45] proposent une mesure d'erreur appelée Exponential Pseudo Norm qui selon eux est plus appropriée car elle accentue l'écart entre les valeurs faibles et élevées de la distance facilitant ainsi distinction entre deux identifiants. Certaines méthodes [1] utilisent également une représentation compacte du modèle

à base de vecteur de quantification et de séquences d'index par Modèles de Markov Cachés (HMM) pour calculer une distance entre un identifiant inconnu et un référent.

Bruck [6] calcule une différence filtre à filtre. Un indice de dissimilarité local est calculé par somme des valeurs absolues des différences. Ils obtiennent ainsi une courbe de dissimilarité entre deux documents (Figure 3.7). L'extraction des minimums de dissimilarité localise alors les positions de correspondances. Ils utilisent finalement un nettoyage itératif afin de faire apparaître les zones de correspondances par des pics.

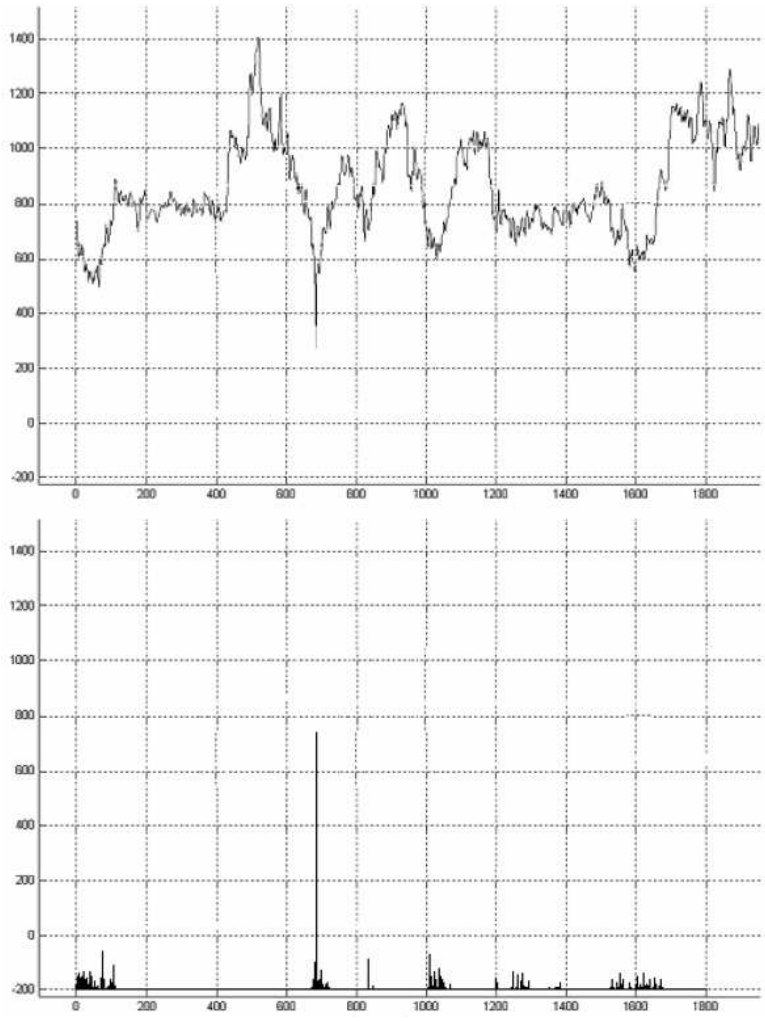


FIG. 3.7 – Figure du haut : courbe de dissimilarité. Figure du bas : localisation des zones de correspondances

Dans le cadre de l'identification de Vidéos, Hoad [32, 31] a proposé des distances entre em-

preintes basées sur la notion de distance d'édition. Une telle distance suppose de considérer chaque empreinte comme une chaîne définie sur un alphabet Σ. Ces chaînes peuvent être modifiées par les opérations suivantes :

1. **substitution** d'un symbole $x \in \Sigma$ par un symbole $y \in \Sigma$, notée $x \to y$
2. **insertion** d'un symbole $x \in \Sigma$, notée $\lambda \to x$
3. **suppression** d'un symbole $y \in \Sigma$, notée $y \to \lambda$

où λ représente le symbole vide.

Une suite de transformations d'une chaîne définit ce que l'on appelle une séquence d'éditions [11] :

Définition 1. Séquence d'édition *Une* ***séquence d'édition*** *S est définie comme une séquence ordonnée d'opérations d'édition élémentaires $s_1, \ldots, s_p$.*

Pour deux chaînes structurellement proches, il existe ainsi une séquence d'édition de faible coût alors que pour deux chaînes dont la structure est très différente une séquence d'édition, dont le coût est important, sera nécessaire.

La **distance d'édition** de deux chaînes est définie comme le chemin d'édition de coût minimum entre celles-ci.

Définition 2. Distance d'édition

Soit c une fonction de coût qui attribue à chaque opération d'édition s une valeur réelle positive $c(s)$. Le coût d'une séquence d'édition est défini comme la somme des coûts de chacune des opérations d'édition élémentaires qui la compose :

$$c(S) = \sum_{i=1}^{p} c(s_i) \tag{3.2}$$

La ***distance d'édition*** *entre deux chaînes X et Y est alors définie comme le coût minimum des séquences d'édition permettant de transformer X en Y :*

$$d(X, Y) = \min\{c(S) : S \textit{ est une séquence d'opérations d'édition qui transforme } X \textit{ en } Y\} \tag{3.3}$$

La distance d'édition permet d'affecter à chaque opération (insertion/suppression/substitution) un coût adapté au contexte applicatif. Dans le cadre de l'identification de vidéos introduit par Hoad, les empreintes sont définies à partir de la détection d'images correspondant à des changements de plans dans les vidéos. Ce problème reste délicat et un algorithme de détection peut facilement détecter des images additionnelles ou ne pas détecter certaines images si l'on modifie légèrement la vidéo. L'utilisation d'opérations d'insertion/suppression dans la distance d'édition permet de prendre naturellement en compte cette faiblesse des algorithmes de calcul d'empreinte utilisé par Hoad.

3.2.2 Techniques d'indexation

Le problème qui se pose maintenant est de comparer efficacement la signature de l'extrait inconnu avec les signatures pré-calculées stockées dans la base de données. Le but est d'organiser les données afin de réduire le nombre de calculs de distance et par la même occasion, le temps de recherche. En effet, un algorithme de recherche comparant chaque combinaison de l'identifiant inconnu avec toute la base pré-calculée prendrait énormément de temps, plusieurs dizaines de minutes voir bien plus suivant la taille de la base de données et la machine employée.

Dans la méthode par ondelettes pyramidales [60], cette méthode génère les coefficients d'ondelettes en approximant le signal à différentes échelles de résolution. Pour la recherche, ils comparent les coefficients d'ondelette de niveau 6 par une distance Euclidienne (Figure 3.8). Ensuite, ils classent par ordre croissant les distances obtenus. Ils conservent alors tous les candidats dont la distance avec le signal inconnu est inférieure à la distance minimale calculée multipliée par un certain facteur. Cela revient à conserver les identifiants dont la distance est proche de la distance minimale trouvée. Puis, ils raffinent la recherche en ajoutant des détails. Ils comparent donc les coefficients au niveau 5 du signal inconnu avec les candidats rescapés de la recherche au niveau 6. Puis ainsi de suite, la méthode devient de plus en plus sélective lorsque l'on passe au niveau inférieur et le nombre de candidats s'amenuise pour n'en garder que quelques uns.

Haitsma et Kalker [36] calculent un identifiant comme étant une suite des valeurs de 32 bits associées à chaque fenêtre du signal. Pour retrouver le bon signal, ils font l'hypothèse

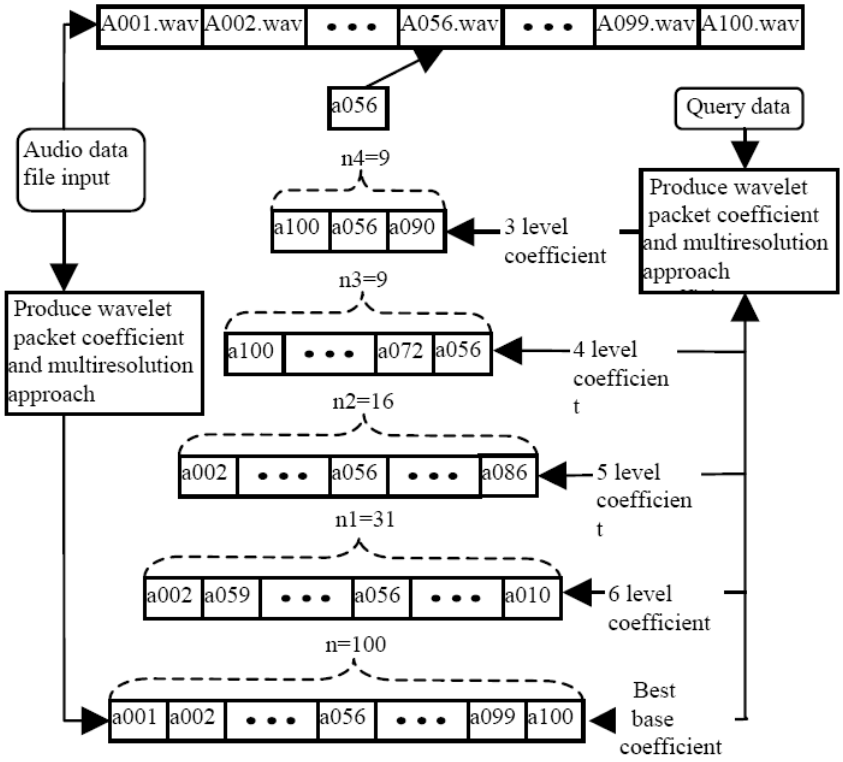


FIG. 3.8 – Reconnaissance par la technique d'ondelettes pyramidales

qu'au moins une fenêtre de l'identifiant pré-calculé correspondant ne contiendra aucun bit erroné. Ils proposent alors d'utiliser un index des valeurs possibles de 32 bits. Chaque entrée dans l'index est associée à un où plusieurs documents ainsi que les positions dans le signal auxquelles apparaissent ces valeurs, comme le montre la Figure 3.9.

Partant d'un extrait inconnu, ces valeurs de sous-empreintes sont calculées et recherchées dans la table d'index afin de trouver une liste de signaux-positions. Ensuite, pour chaque candidat, à la position correspondante, ils calculent la distance de Hamming sur un durée équivalente à 256 valeurs de sous-empreintes. Si une distance de Hamming entre 2 documents est sous un certain seuil déterminé de manière empirique, alors le document associé est considéré comme co-dérivé.

Les heuristiques des principales méthodes exposées dans ce chapitre sont résumées Tab. 3.1.

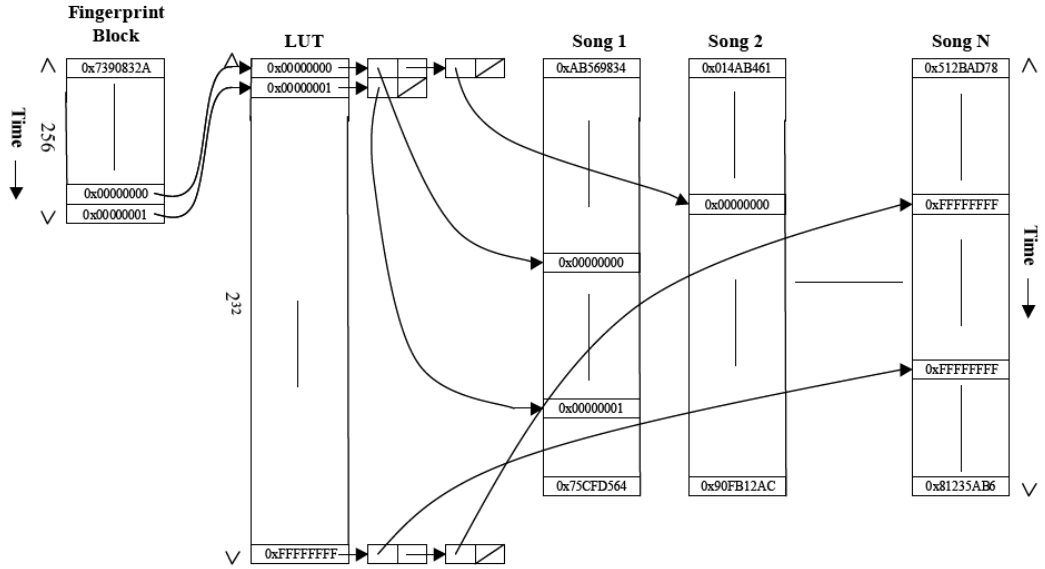


FIG. 3.9 – Reconnaissance par table d'index et distance de Hamming

Méthodes	**Segmentation Temporelle**	**Calcul d'empreinte**	**Indexation**	**Comparaison et reconnaissance**
Li et Hou[41]	Ondelettes pyramidales	Coefficients d'ondelettes à différentes échelles	Indexation par échelle de résolution	Comparaison des coefficients d'ondelettes par distance euclidienne
Kruth[40]	Fenêtres recouvrantes	Bit de différence d'énergie entre l'instant t et t+1	aucune	Distance de Hamming
Burges et al.[8]	Transformée Complexe Modulée	OPCA combinées pour extraire un vecteur de 64 valeurs décrivant 20s de signal	aucune	Reconnaissance par distance euclidienne entre vecteurs de 64 valeurs
Brück et al.[6]	Fenêtres recouvrantes	Energies d'un banc de 8 filtres appliqués à la transformée de Fourier stockées sur 16 bits chacunes, soit 128bits	aucune	Somme des valeurs absolues des différences entre filtres par distance euclidienne et détermination de minima.
Haitsma et Kalker[36]	Fenêtres recouvrantes	Bits de différences entre filtres fréquentiels stockés sur 32bits par intervalle	Indexation par table d'index associés aux documents et positions temporelles auxquelles sont calculées les valeurs	Distance de Hamming calculée à partir d'une valeur de 32 bits identique et sur une longueur de 256 valeurs.

TAB. 3.1 – Tableau Récapitulatif

4

Construction Robuste d'Identifiants Audio

Dans ce chapitre, nous allons détailler les étapes qui nous ont conduit, à l'élaboration d'une nouvelle empreinte de documents audio. Cette empreinte a été développée et a évolué en adéquation avec les contraintes de notre application.

Sommaire

4.1 Introduction

Rappelons brièvement les contraintes qui sont les notres :

1. L'identification doit être indépendante du format du fichier.

   L'utilisation de différents formats (tel que MP3) va essentiellement modifier le taux de compression du document audio. Il faut donc que notre méthode soit robuste aux taux de compression couramment utilisés. Ceci nous a conduit à rejeter un ensemble de méthodes trop peu résistantes vis à vis de ce critère.

2. L'identification doit pouvoir être effectuée à partir d'un échantillon quelconque d'un document audio.

   Cette dernière contrainte est sans doute celle qui a la plus forte incidence dans le choix des outils à utiliser pour l'identification. En effet, beaucoup de documents audio sont assez similaires dans les premières secondes (silence, applaudissements...), par conséquent, l'identification ne doit pas se baser uniquement sur le début d'une oeuvre. De plus, utiliser uniquement le début d'une oeuvre permet de contourner le système de DRM en concaténant quelques secondes d'un morceau connu à un morceau piraté. L'identification d'un morceau doit donc pouvoir être effectuée soit à partir d'une position aléatoire dans le fichier soit périodiquement au cours de l'écoute du document. Cette contrainte induit un décalage de positions entre le document lu et celui présent dans la base des morceaux connus, ce qui impose une synchronisation du calcul d'identifiants. Notre méthode doit donc être robuste à de tels décalages, ce qui nous a conduit

à rejeter l'ensemble (majoritaire) des méthodes présentant une fragilité pour ce type d'altérations.

3. L'identification d'un morceau doit être effectuée en quelque secondes (environ 5s) et doit nécessiter des ressources machine (temps de calcul, espace mémoire, espace disque) compatibles avec les capacités d'un ordinateur familial ou d'un mobile.

   Cette contrainte nous amène à privilégier une identification d'un document audio en parallèle à sa lecture. Nous avons donc rejeté toutes les méthodes utilisant l'ensemble d'un fichier audio pour l'identification. En effet, ces méthodes posent des problèmes en terme de temps de calcul et de ressources (processeur, mémoire, disque) mobilisées. Notons qu'un temps de calcul trop long rendrait inopérante la gestion des droits. En effet, l'identification en 3 minutes d'un fichier audio de 4 minutes remet fortement en cause la pertinence du système DRM qui ne peut alors interdire la lecture qu'à partir des 3/4 de l'oeuvre.

4.2 Analyse des fréquences

Notre première méthode de calcul de sous-empreinte est basée sur le même principe que celle de Haitsma et Kalker [37] exposée en Section 3.1.2 (équation 3.1). Comme ces auteurs, nous avons utilisé la technique des fenêtres recouvrantes afin de séparer le signal en une succession d'intervalles de taille fixe. Puis, nous avons calculé le spectre de chaque intervalle que nous avons ensuite décomposé en une suite de bandes de fréquences avec un espacement logarithmique. Cependant, comme le montrent nos expérimentations (Section 6.2.3), un fort taux de compression peut significativement altérer la robustesse de cet algorithme d'extraction de sous-empreinte. Par exemple, un taux de compression de 128kbps (un des plus répandu dans les réseaux P2P) engendre un taux d'erreur bit à bit de 9% et un taux de valeurs de 32-bits similaires de seulement 24%. Ces tests ont été fait en comparant bit à bit ou valeur entière avec valeur entière l'empreinte d'un signal audio et celle du même contenu préalablement compressé. Ce dernier inconvénient interdit une comparaison directe de deux empreintes de documents audio basée sur le nombre de sous-empreintes communes aux deux signaux. L'empreinte du contenu compressé est en effet trop éloignée de celle de

l'original (24%). L'altération du signal par du bruit, une compression, ou une opération de suppression réduit donc de manière drastique le nombre de valeurs identiques entre l'empreinte d'un document et celle du même document dégradé. Comme nous l'avons vu dans le chapitre précédent, Haitsma et Kalker résolvent ce problème en utilisant la distance de Hamming entre deux séquences de sous-empreintes. Cette stratégie impose toutefois de nombreux calculs de distance de Hamming avant de pouvoir identifier l'extrait inconnu.

Nous avons donc, dans un premier temps envisagé d'améliorer la robustesse de l'algorithme d'extraction de caractéristiques en se basant sur les deux remarques suivantes (Figure 4.1) :

- L'utilisation de deux intervalles successifs afin de calculer la valeur de la sous-empreinte (equation 3.1) implique la corruption de deux sous-empreintes si une erreur se produit dans l'extraction des caractéristiques de l'intervalle qu'ils ont en commun.
- La comparaison des énergies de deux bandes successives d'un spectre est sensible aux erreurs qui peuvent se produire sur une seule bande. On observe alors le même inconvénient qu'au point précédent entre deux valeurs basées sur l'énergie d'une même bande.

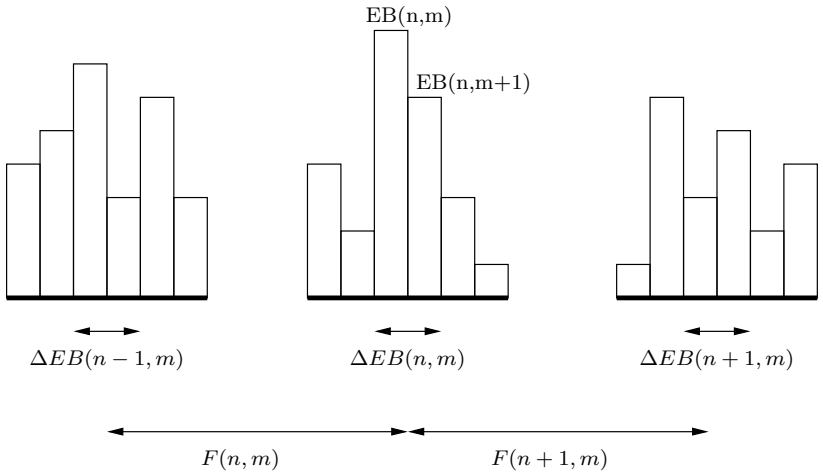


FIG. 4.1 – Relations de dépendances entre les différentes notions utilisées par Haitsma et Kalker.

Nous résolvons la première source d'erreurs en n'utilisant qu'un seul intervalle pour chaque calcul de sous-empreinte. Ceci évite que l'extraction erronée des énergies de bandes

d'une sous empreinte ne perturbe la sous empreinte suivante. La seconde source d'erreur est liée au fait que l'énergie d'une bande du spectre influe sur deux différences d'énergie entre bandes. En effet, en utilisant les mêmes notations que dans dans la section 3.1.2, l'altération de la mesure de l'énergie d'une seule bande ($EB(n,m)$) altère les valeurs de $\Delta EB(n,m-1)$ et $\Delta EB(n,m)$. Cette altération des bandes d'énergie peut être considérée comme la présence d'un bruit aléatoire sur le signal $EB(n,m)_{m\in\{1,..,M\}}$ où M représente l'index de la dernière bande d'énergie.

Si on suppose que le bruit est non corrélé entre les différents échantillons du signal $EB(n,m)_{m\in\{1,..,M\}}$, une méthode classique pour réduire l'influence du bruit consiste à remplacer chaque mesure $EB(n,m)$ par une valeur moyenne de $EB(n,m)$ fonction de m. Nous définissons ainsi l'énergie moyenne $S(n,m)$ d'une bande m, d'un intervalle n, comme la moyenne de toute les énergies des bandes de 0 à m :

$$\forall m \in \{1\ldots,M\}, \quad S(n,m) = \frac{1}{m}\sum_{i=1}^{m} EB(n,i)$$

On remplace alors $EB(n,m)$ par $S(n,m)$ dans le calcul des différences d'énergies entre bandes. Le m^{ieme} bit de la sous-empreinte associée à l'intervalle n ($F(n,m)$) est donc défini par :

$$F(n,m) = \begin{cases} 1 & Si \qquad S(n,m) - S(n,m-1) \geq 0 \\ 0 & Sinon \end{cases}$$

Notons que $F(n,m)$ utilise uniquement les informations de l'intervalle n. Les erreurs ne se propagent donc pas entre deux intervalles. On a de plus la relation suivante :

$$\begin{aligned} S(n,m) - S(n,m-1) &= \tfrac{1}{m}\textstyle\sum_{i=1}^{m} EB(n,i) - \tfrac{1}{m-1}\sum_{i=1}^{m-1} EB(n,i) \\ &= \tfrac{1}{m} EB(n,m) - \tfrac{1}{m(m-1)}\textstyle\sum_{i=1}^{m-1} EB(n,i) \\ &= \tfrac{1}{m}\left(EB(n,m) - S(n,m-1)\right) \end{aligned}$$

Puisque nous utilisons simplement le signe de $S(n,m) - S(n,m-1)$, la formule précédente peut être simplifiée comme suit :

$$F(n,m) = \begin{cases} 1 & Si \qquad EB(n,m) - S(n,m-1) \geq 0 \\ 0 & Sinon \end{cases}$$

La sous-empreinte pour chaque intervalle n est alors définie par la concaténation des M bits $F(n,m)_{m\in\{1,..,M\}}$. Le paramètre M est fixé à 32 afin de faciliter la comparaison de nos empreintes avec celles d'Haitsma et Kalker [37] (chapitre 6). L'empreinte du document audio est enfin définie comme la concaténation de la séquence de sous-empreintes.

La figure suivante (Figure 4.2) montre, à gauche, l'empreinte d'un signal audio dit original. L'axe horizontal correspond au nombre de bits pour chaque empreinte et l'axe verticale représente le temps. De cette manière, nous pouvons visualiser les valeurs de sous empreintes du signal avec $F(n,m) = 0$ représenté de couleur noir tandis que $F(n,m) = 1$ est représenté en blanc. L'empreinte centrale correspond au même contenu ayant subi une compression au taux usuel de 128Kbps. Ainsi, nous observons à droite la distance de Hamming entre ces deux empreintes où les zones blanches correspondent à des bits différents entre les deux empreintes. On observe dans cet exemple une faible variabilité de l'empreinte lorsque le contenu est soumis à une altération de type compression.

4.3 Segmentation temporelle

Nos expérimentations (Chapitre 6, Fig. 6.1) ont montré que la modification précédente de la méthode de Haitsma améliore significativement la robustesse de l'extraction d'empreinte vis à vis de la compression. Cependant, nous nous sommes également aperçu que l'extraction d'empreinte était sensible aux décalages temporels. Ces décalages peuvent être induits par :

1. la suppression d'une partie du morceau de musique ou bien
2. la sélection d'un extrait du morceau pour l'étape de reconnaissance.

Cette sensibilité aux décalages temporels est dûe à l'utilisation de la méthode des fenêtres glissantes appelée fenêtrage ou framing (Section 3.1.1).

En effet, La méthode de fenêtrage assure qu'un nombre fixe et suffisant d'intervalles est sélectionné à partir d'un signal d'entrée. Cependant, la sélection d'une séquence d'intervalles

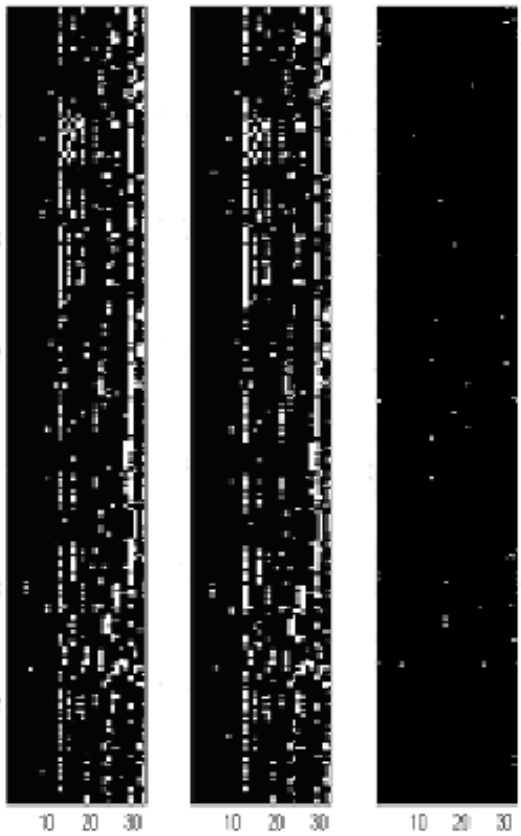


FIG. 4.2 – Distance de Hamming entre l'empreinte d'un contenu et celle de son compressé

contigus est sensible aux décalages temporels qui peuvent être appliqués au document (Section 2). Cet inconvénient est atténué grâce au recouvrement entre fenêtres mais n'est pas complètement résolu. D'un autre côté, les méthodes de segmentation, à base d'onsets par exemple, sont moins sensibles à ces opérations mais ne garantissent pas que suffisamment d'intervalles seront extraits dans un intervalle de temps imparti. En effet, pour détecter un onset, il est nécessaire d'avoir une transition suffisamment nette entre deux parties disctinctes du signal.

La figure 4.3 illustre l'emplacement des changements significatifs du signal audio détectés par des techniques usuelles d'extraction d'onsets ou de tempo. On observe bien qu'il est possible de déterminer des instants particuliers et persistants après dégradation. Toutefois, si on calcule des valeurs caractéristiques du signal uniquement lors de la détection de telles

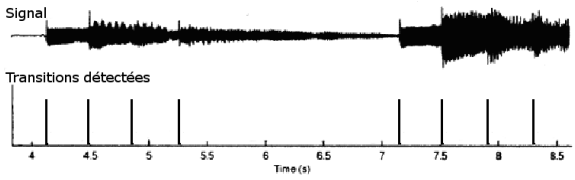


FIG. 4.3 – Détection de transitions le long d'un signal audio.

transitions, nous devons garantir une fréquence minimale de détection pour pouvoir garantir une fréquence minimale de génération de sous empreinte. Ceci est indispensable pour garantir une identification d'un signal audio en un temps imparti. Cependant, le nombre de transitions détectées dans un temps donné dépend de l'allure du signal et est donc imprévisible. On ne peut donc pas garantir a priori qu'un nombre minimal de transitions sera détecté dans un intervalle de temps donné.

L'idée de base de notre méthode est de combiner les avantages respectifs des méthodes de fenêtrage et de détection :

1. Les méthodes de de fenêtrage permettent d'assurer qu'un nombre minimal et suffisant d'intervalles sera extrait dans un temps imparti.
2. Les méthodes de détection permettent de sélectionner les intervalles à des instants particuliers du signal et résistant aux altérations de type compression ou décalages temporels. Ceci induit naturellement une synchronisation rapide de deux empreintes similaires sur ces instants particuliers.

Pour ce faire, la méthode proposée se base sur la définition d'un intervalle de temps de durée déterminée pendant lequel l'enveloppe du signal est analysée afin de rechercher l'instant qui le caractérise au mieux. Cela se traduit par l'algorithme suivant décomposé en 3 étapes (Figure 4.4) :

- Dans la première étape, un intervalle, appelé Intervalle d'Observation (I_o) est sélectionné au début du signal afin d'étudier l'enveloppe du signal sur cet intervalle. La taille de cet intervalle est usuellement égale à quelques centièmes de secondes.
- L'intervalle I_o est ensuite décomposé en une multitude de sous-intervalles de quelques millisecondes appelés Intervalle d'Energie (I_e). L'intervalle I_e va nous servir de fenêtre glissante le long du signal. Chaque intervalle I_e est affecté d'une énergie définie par

l'amplitude moyenne des échantillons sur l'intervalle. L'instant qui caractérise le mieux l'intervalle I_o est défini comme le centre de l'intervalle I_e d'énergie maximum (noté I_{emax})

– Dans la troisième étape, un dernier intervalle, appelé Intervalle de Caractérisation (I_c) est défini au même emplacement (centré) que l'intervalle I_{emax}. Cette intervalle I_c est choisi de longueur égale à quelques dizièmes de secondes afin de calculer la transformée de Fourier associée et d'extraire une valeur de sous-empreinte basée sur la méthode expliquée dans la sous-section précédente.

Finalement, le début de l'intervalle I_o suivant est positionné à la fin de I_{emax}.

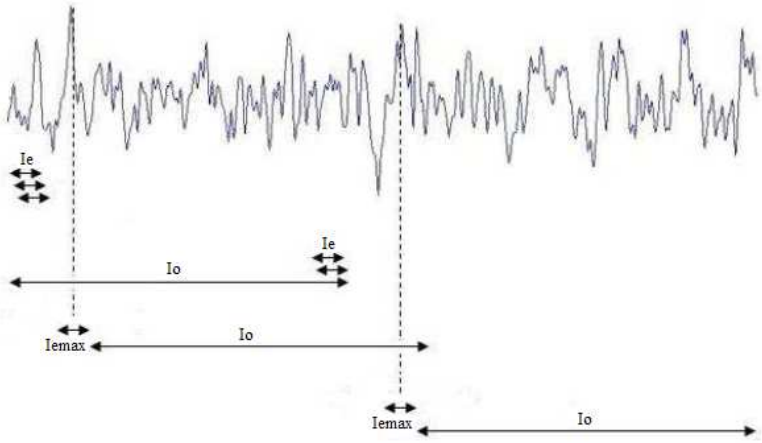


FIG. 4.4 – Méthode de segmentation audio

L'enveloppe du signal est donc analysée par sous-parties. Sur chacune de ces sous-parties est calculée une valeur de sous-empreinte à la position d'énergie maximale. Nous garantissons ainsi qu'une sous empreinte sera calculée sur chaque intervalle I_o et nous synchronisons les calculs de sous empreintes sur des parties remarquables du signal.

Notre heuristique de sélection du prochain intervalle I_o (après I_{emax}) apporte une plus grande robustesse envers les décalages temporels par rapport aux stratégies de base qui consistent à selectionner une séquence consécutive d'intervalles I_o sans tenir compte de la position de l'intervalle I_{emax}. En effet, en utilisant cette dernière stratégie, un I_{emax} situé à la transition entre deux intervalles I_o pourrait ne pas être détecté. De plus, notre stratégie permet de détecter plusieurs intervalles I_e, avec des énergies proches, au sein d'un même

I_o. L'intervalle d'énergie maximum sera détecté mais si d'autres intervalles I_e d'énergie proche lui succèdent, ils correspondront aux maximum sur les intervalles I_o suivants et seront donc également détectés. Cette dernière propriété renforce la robustesse de notre méthode d'extraction d'empreintes. Sans cette propriété, la stratégie de base ne permettrait de sélectionner qu'un seul intervalle I_c pour chaque I_o. Or, une dégradation du signal pourrait échanger la sélection de deux I_e dont les énergies seraient proches et importantes. Notre stratégie renforce donc aussi la robustesse envers d'autres types de dégradations. Elle renforce notamment la robustesse de l'extraction d'empreinte vis à vis de la compression ce qui nous intéresse tout particulièrement.

A noter enfin que la distance entre deux intervalles I_{emax} successifs varie entre I_e et I_o, $1/I_o$ est donc le nombre minimum de valeurs caractéristiques de sous-empreinte calculées en une seconde. Notre méthode garantie donc bien une fréquence minimale $\left(\frac{1}{I_o}\right)$ d'extraction de sous empreinte.

4.4 Conception de l'empreinte

La Transformée de Fourier sur laquelle est basé notre calcul de la valeur de sous-empreinte induit de nombreux calculs et est sensible aux altérations du signal. Ceci induit une variabilité importante de notre empreinte en fonction des dégradations du signal. Or, notre processus de segmentation audio est très fiable (Chapitre 6). Nous avons donc jugé qu'il n'était pas utile de rajouter un second processus (la transformée de Fourrier) source potentielle d'erreurs et avons décidé de baser notre empreinte uniquement sur l'information apportée par le processus de segmentation.

Nous avons donc défini une nouvelle méthode de calcul de la valeur de sous-empreinte basée sur l'écart (en ms) entre les instants particuliers détectés entre deux sous empreintes. Comme nous l'avons vu, cette valeur varie entre I_e et I_o. Une sous empreinte peut donc prendre $I_o - I_e$ valeurs. Étant donné un codage en 44KHz mono, il est nécessaire d'utiliser 14 bits pour coder cette information au lieu des 32 utilisés par la méthode précédente. On a donc une réduction significative de l'espace nécessaire pour stocker une sous empreinte. De plus notre processus de segmentation sélectionne un nombre inférieur d'intervalles que

la méthode des fenêtres recouvrantes. La réduction est donc nettement plus importante que le simple rapport $\frac{14}{22} \approx \frac{2}{3}$.

4.5 Conclusion

Il est possible de jouer avec la valeur de I_o en fonction des contraintes du système. Pour un faible espace de stockage, un I_o élevé réduit le nombre de valeurs contenues dans l'empreinte et donc stockées dans la base de données. A l'inverse, pour de grosses bases de données gérant des milliers de documents, un intervalle I_o plus réduit augmente le nombre de valeurs calculées par seconde et donc la quantité d'information servant à discriminer les signaux.

5

Appariement d'Identifiants Audio

Nous avons présenté notre méthode de calcul d'empreinte dans le chapitre 4. Ces empreintes se présentent comme une suite de nombres entiers, chaque entier codant une distance entre deux instants particuliers du signal. La prochaine étape consiste à comparer efficacement l'empreinte d'un extrait inconnu et une base d'empreintes pré-calculées. Nous allons donc développer ici le cheminement qui nous à conduit à l'élaboration de notre méthode d'identification.

Sommaire

5.1 Introduction

La méthode que nous devons élaborer doit nous permettre de stipuler si une empreinte correspondant au même contenu que le signal original est stockée dans la base ou si aucune empreinte n'est dérivée de l'extrait inconnu. Pour cela, lors de la comparaison, il est nécessaire d'obtenir un score, une distance, ou une mesure de similarité permettant de prendre cette décision. Or, retrouver l'empreinte la plus proche et décider si elle provient d'un même contenu sont deux problématiques différentes. La première appelée indexation consiste à retrouver l'empreinte ayant une plus faible distance ou un meilleur score, suivant la méthode employée. La seconde problématique traite d'identification et de reconnaissance. Il s'agit de fixer un seuil sur le critère de similarité. La position par rapport à ce seuil permettra de savoir si deux empreintes proviennent d'un même contenu original. Cela implique donc une contrainte forte sur le critère et le seuil utilisé afin de s'assurer :

1. que la distance maximale entre deux contenus co-dérivés sera inférieure au seuil et
2. que la distance minimale entre deux contenus différents (non co-dérivés) sera supérieure au seuil.

Ces deux contraintes doivent bien sûr être satisfaites malgré d'éventuelles dégradations (compression, décalages) qui peuvent affecter les signaux. Notons que ces deux contraintes s'adaptent très facilement si on utilise un critère de similarité plutôt qu'une distance.

Comme on l'a vu (Chapitre 3), la méthode utilisée par Haitsma et Kalker pour identifier une empreinte se base sur une comparaison bit à bit par distance de Hamming. Cette distance compare donc membre à membre deux suites de sous empreintes à partir d'une position commune. Nos expérimentations (Section 6.2.2, Fig. 6.1) montrent que deux documents audio co-dérivés ont entre 80 et 90% de valeurs de sous-empreinte en commun en fonction du taux de compression utilisé pour dégrader l'un des deux signaux. Malgré ce nombre important de valeurs communes (largement supérieur a celui obtenu par les méthodes antérieures), ce résultat ne permet pas une comparaison membre à membre de deux empreintes puisqu'une mauvaise détection d'un intervalle I_{emax} revient, dans ce contexte, à décaler une des deux empreintes et fausse complètement le résultat de la comparaison. Toutes les autres méthodes comparant deux empreintes terme à terme (Burges, Bruck) présentent le même inconvénient.

Dans le cadre de la reconnaissance d'empreinte, il existe une méthode alternative qui consiste à considérer l'ensemble de toutes les valeurs de sous-empreintes possibles comme un alphabet. Une empreinte peut alors être interprétée comme une chaîne de caractères. Les méthodes d'appariement de chaînes (string matching) peuvent ainsi être appliquées afin de calculer une distance entre deux chaînes [14, 10]. Le principe de base de ces méthodes est de calculer le degré de similarité entre deux chaînes de caractères à partir du nombre minimum d'opérations (insertion, suppression, inversion) nécessaires pour transformer une chaîne en la seconde. Chaque opération peut être pondérée afin d'établir des coûts adaptés au cadre applicatif de la méthode.

De plus, si seul un court échantillon du signal d'entrée est utilisé pour calculer l'empreinte, la reconnaissance de cette empreinte implique de trouver le plus faible coût de transformation (ou coût d'édition) entre l'empreinte de cet extrait inconnu et une sous-séquence d'une des empreintes stockées dans la base de données. On se retrouve donc confronté à un problème d'*alignement local*.

Considérons une base de données composée de M empreintes de taille n. Si l'empreinte du signal d'entrée est de taille k, une recherche exhaustive dans la base de données de l'alignement de coût minimum serait accomplie en $\mathcal{O}(Mkn)$ [14]. De ce fait, à cause du large nombre M d'empreintes pouvant être stockées dans la base et de la valeur importante de n dans le cadre de notre application, ce type de méthode est inutilisable car trop coûteux

en temps de calculs.

5.2 Scores par quantité d'information

La trop grande complexité de la méthode triviale mentionnée précédemment nécessite la définition d'un premier filtre afin de réduire le nombre de candidats à l'identification. On pourrait comparer ce processus au processus d'embauche. Les recruteurs reçoivent énormément de candidatures spontanées. Après les avoir toutes parcourues assez sommairement, seules certaines d'entre elles sortent du lot grâce à certaines caractéristiques recherchées. Ensuite, les candidats correspondant aux curriculum vitae sélectionnés seront conviés à un entretien afin d'étudier de manière plus approfondie leur profil et évaluer leur candidature. Enfin, le candidat restant ne sera pas celui ayant le meilleur profil, mais celui correspondant parfaitement au profil recherché.

5.2.1 Structuration de la base de données

Cette première phase, dite phase de filtrage doit nous permettre de juger rapidement de la ressemblance entre une sous partie de l'extrait inconnu et n'importe qu'elle sous partie de la base de données. Comme nous l'avons mentionné, il s'agit de définir un indice qui soit simple et rapide à calculer tout en apportant une indication de similarité forte. Dans notre cas, nous avons décidé de nous baser sur une méthode bien connue de la problèmatique de recherche de correspondances textuelles : le filtrage par q-grams [10].

Définition 3. *Soit Σ un alphabet et S_1, S_2 deux chaînes construites sur Σ.*

- *Un q-gram de S_1 est une sous chaîne de S_1 de longueur q,*
- *Un q-gram de S_1 et S_2 est une chaîne de longueur q présente simultanément dans S_1 et S_2.*

Un q-gram commun de S_1 et S_2 est bien évidement un q-gram de S_1 et de S_2.

Dans notre cas, les symboles de l'alphabet correspondent à une valeur de sous empreinte codant l'espace temporel entre 2 instants particuliers du signal audio (sections 4.3 et 4.4).

Un mot de q symboles comprend une succession de q sous-empreintes et caractérise une allure bien précise du signal audio.

La plupart des méthodes de filtration par q-gram sont basées sur le théorème de Jokinen-Ukkonen [34] :

Lemme 1. *Soit $P = P[1..m]$ et $T = T[1..n]$ deux chaînes de longueurs respectives m et n sur un alphabet Σ. Supposons qu'une occurence de P puisse s'apparier avec T avec k erreurs. La sous-chaîne ainsi formée finissant à l'indice j de T, P et $T[j-m+1,..j]$ partagent alors au plus $t = m + 1 - (k + 1)q$ $q-grams$.*

La valeur de t correspond à un seuil qui définit généralement le nombre de q-grams que chacune des sous-chaînes de T et P doivent partager. Toutefois, ce Lemme correspond à une distance d'édition très basique qui n'est pas adaptée à notre application. La notion de q gram restant cependant centrale dans celle-ci, nous avons struturé notre base de données d'empreintes en fonction des q-grams. Dans le cadre d'un stockage des empreintes dans une base de données [27] ceci peut être réalisé en définissant un index supplémentaire de q grams. Cet index permet de retrouver efficacement, pour toute empreinte correspondant à un signal d'entré, les empreintes de la base qui partagent avec celle-ci un nombre minimal de q-grams.

Dans le cadre d'une implémentation par fichiers, un codage sensiblement équivalent peut être réalisé en construisant un tableau de q grams. Ce tableau permet à l'algorithme de recherche de limiter le nombre de séquences à comparer et de considérer seulement les documents contenant ces régions d'intérêt caractérisées par les q-grams.

Le choix de la valeur de q dépend de l'application. Une valeur de q élevée permet de filtrer un nombre maximum d'empreintes tout en augmentant la probabilité de rejeter à tort une empreinte valide. Inversement, une valeur de q faible permet de se prémunir d'une élimination abusive d'un bon candidat mais augmente également le nombre d'empreintes acceptées à tort. Il faut donc définir une valeur de q qui minimise au mieux les taux de «Faux Acceptés» et «Faux Rejetés». Flajolet [22] (dans le cas d'un modèle de Bernoulli uniforme) et Szpankowski [54] (pour le modèle non uniforme) on démontré un résultat intéressant dans ce cadre : pour un texte de taille n généré aléatoirement tout mot de longueur $l < log(n/h)$ apparait presque surement quand n tend vers l'infini. Le symbole h représente ici l'entropie de

Renyi qui peut être mesurée par $log(1/p_{min})$ avec p_{min} la probabilité minimale d'apparition d'une lettre de l'alphabet. Si nous considérons une probabilité uniforme $p = p_{min} = 2^{-14}$ et des chaînes de longeur 5300 (longueur moyenne dans notre base), nous obtenons un l égal à 6.3. La valeur de q devrait donc a priori être supérieure à 7. Nos résultats expérimentaux contredisent partiellement ce résultat puisque une valeur de q proche de 5 semble représenter un bon compromis entre les faux acceptés et les faux rejetés. Ceci est certainement du au fait que nos valeurs de n, bien qu'élevées ne sont pas infinies.

5.2.2 Décision

Les expériences que nous avons menées nous ont montré que le filtrage de notre base d'empreintes par q-gram était extrêmement efficace malgrè la taille (≈ 5300) et le nombre (≈ 400) des empreintes de notre base. Nous pensons que cette efficacité est due essentiellement à la taille de notre alphabet. En effet, si nous supposons que les empreintes sont générées par un processus aléatoire uniforme, la probabilité d'apparition d'un q-gram particulier dans un mot de longueur n sera dans notre cas égal à $(n - q + 1)2^{-14q}$ ce qui est extrêmement petit même pour n grand. A contrario, si on trouve un nombre élevé de q grams communs à deux chaines, on peut en conclure que les deux chaines n'ont pas été générées par des processus indépendants et qu'elles correspondent à des contenus co-dérivés. Ce type de raisonnement s'apparente aux approches a contrario [16].

Supposons que nos empreintes sont générées par un processus aléatoire avec une distribution uniforme pour chaque symbole et considérons la variable aléatoire X_q qui représente le nombre de q grams communs à deux chaines. Étant donné une empreinte correspondant au signal d'entrée et une empreinte de la base, nous mesurons une observation n_q de X_q. Poursuivant l'idée introduite au paragraphe précédent nous définissons un score S proportionnel pour n grand à la quantité d'information induite par l'évènement $X_{min} = n_{min} \& \ldots \& X_{max} = n_{max}$ où $\{min \ldots, max\}$ correspond à une plage de longueurs de $q - grams$. Plus formellement, nous cherchons à définir un score S asymptotiquement équivalent à $Q(n_{min}, \ldots, n_{max}) = P(X_{min} = n_{min} \ldots \& X_{max} = n_{max})$ pour des chaînes de grande taille. La définition d'un tel score suppose au préalable une modélisation de la formation des chaînes et des q-grams afin d'évaluer $P(X_{min} = n_{min} \ldots \& X_{max} = n_{max})$.

5.2.2.1 Modélisation

Commençons par considérer le cas où seulement une longueur de q-gram est prise en compte ($Q(n_q) = P(X_q = n_q)$). Si l'on note par s la taille de l'alphabet, Nicodème [46] a démontré que la loi de probabilité de X_q peut être approximée efficacement par un modèle de balles et d'urnes où chacune des s^q urnes est associée à une valeur possible de q-gram dans s.

En effet, le nombre de q-grams qui se répètent au moins une fois dans une chaîne de longueur n correspond au nombres d'urnes contenant au moins 2 boules (nombre de collisions) après n lancés. Dans le cas de deux chaînes de taille $n + q - 1$ et $m + q - 1$ le nombre de q-gram communs à ces deux chaînes (sans répétition) correspond au nombre d'urnes contenant simultanément des boules noires et blanches (collisions bicolores) après un lancé de m boules noires et n boules blanches.

Dans le même article, Nicodème montre que la distance en variation totale [9] entre la loi de Poisson de X_q et une loi Gaussienne est toujours bornée. De plus, la loi de X_q converge vers une Gaussienne quand les paramètres de la loi de Poisson tendent vers l'infini. On peut donc approximer la loi de X_q par une loi Gaussienne :

$$p(X_q = n_q) = \frac{1}{\sqrt{2\pi}\sigma_{nm}} e^{-\frac{(n_q - \gamma_{nm})^2}{\sigma_{nm}^2}} \tag{5.1}$$

Ou γ_{nm} et σ_{nm} représentent respectivement la moyenne et l'écart type de X_q pour des chaînes à comparer de taille n et m.

Sous cette approximation, estimer la loi de X_q revient à estimer γ_{nm} et σ_{nm}. Nicodème propose d'estimer ces quantités en utilisant une Poissonisation du problème :

Dans le cas d'une seule chaîne, considérons $\phi_n(u)$ la probabilité qu'il existe u urnes sans collisions après n lancés. Si nous considérons que le nombre de lancé n'est plus égal à n mais

[9] la distance en variation totale entre deux variables entières positives aléatoires de fonction de probabilité respectives f_n et g_n est la somme $\sum_n |f_n - g_n|$

suit une loi de Poisson de paramètre z, la transformée de Poisson de $\phi_n(u)$ est définie par :

$$\psi(z,u) = \sum_{n\geq 0} \phi_n(u) \frac{z^n}{n!} e^{-z} = e^{-z} \sum_{n,k} f_{n,k} \frac{u^k z^n}{n!}$$

où $f_{n,k}$ représente la probabilité qu'il existe k urnes sans collision après n lancés.

Soit $F(z,u) = e^z \psi(z,u) = \sum_{n,k} f_{n,k} \frac{u^k z^n}{n!}$. Si l'on connait une forme analytique de $F(z,u)$, on peut obtenir $\phi_n(u)$ par la dépoissonisation :

$$\phi_n(u) = n![z^n]F(z,u)$$

où $[z^n]F(z,u)$ représente le n^{e} coefficient de la décomposition de Taylor de $F(z,u)$.

La moyenne (μ_n) et le moment d'ordre 2 ($m_n^{(2)}$) du nombre d'urnes sans collision peuvent alors être retrouvés par :

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = \sum_n \left(\sum_k k f_{n,k} \right) \frac{z^n}{n!} = \sum_{n\geq 0} \mu_n \frac{z^n}{n!} =_{not} m(z) \Rightarrow \mu_n = n![z^n]m(z)$$

De même :

$$\left.\frac{\partial}{\partial u} u \frac{\partial F(z,u)}{\partial u}\right|_{u=1} = \sum_n \left(\sum_k k^2 f_{n,k} \right) \frac{z^n}{n!} = \sum_{n\geq 0} m_n^{(2)} \frac{z^n}{n!} =_{not} m^{(2)}(z) \Rightarrow m_n^{(2)} = n![z^n]m^{(2)}(z)$$

La moyenne γ_n et la variance σ_n^2 du nombre d'urnes avec collisions s'en déduisent par :

$$\begin{cases} \gamma_n &= m - \mu_n \\ \sigma_n^2 &= m_n^{(2)} - \mu_n^2 \end{cases} \tag{5.2}$$

Ce nombre moyen de collisions correpond aux nombres de q grams qui se répètent dans une chaînes de longueur n.

Dans le cas de la détermination du nombre de q grams communs à deux chaînes, Nicodème utilise une double Poissonisation et montre que :

$$F(z,t,u) = \prod_{0\leq i\leq s^q-1} \left(e^{p_i(z+t)} + (u-1)(e^{p_i z} + e^{p_i t} - 1) \right) = \sum_{k,b,c} f_{kbc} u^k z^b t^c$$

où p_i représente la probabilité d'apparition du i^{e} q-gram, et $f_{k,b,c}$ est la probabilité qu'il existe k urnes sans collision bicolore après un lancé de b boules blanches et c boules noires.

La moyenne et le moment d'ordre 2 du nombre d'urnes sans collision bicolore peuvent être définies comme précédement par :

$$\begin{cases} \mu_{bc} &= [z^b t^c] b!c! \left.\frac{\partial F}{\partial u}\right|_{u=1} \\ m^{(2)}_{bc} &= [z^b t^c] b!c! \left.\frac{\partial}{\partial u} u \frac{\partial F}{\partial u}\right|_{u=1} \end{cases}$$

On en déduit alors la moyenne γ_{bc} et l'écart type σ_{bc} de notre variable X_q en utilisant une méthode similaire à celle de l'équation 5.2.

Nicodème estime les quantités $\left.\frac{\partial F}{\partial u}\right|_{u=1}$ et $\left.\frac{\partial}{\partial u} u \frac{\partial F}{\partial u}\right|_{u=1}$ en supposant que pour tout $i \in \{1, \ldots, s^q\}$, $p_i n$ tend vers une constante θ_i quand n tend vers l'infini. Ceci revient à lier la probabilité (p_i) d'occurence d'un q gram avec la longueur des chaînes entre lesquelles on cherche ces q grams. Ce postulat ne nous a pas semblé cohérent avec nos hypothèses et nous avons préféré supposer une distribution uniforme de q grams. Ceci nous a conduit à formuler la proposition suivante :

Proposition 1. *En utilisant les mêmes notations que Nicodème, si on suppose que les q-grams ont une distribution uniforme $p = \frac{1}{s^q}$, avec s la taille de notre alphabet, la moyenne γ_{nm} et la variance σ^2_{nm} du nombre X_q de q-grams communs entre deux chaînes de longueurs respectives n et m est défini par :*

$$\begin{cases} \gamma_{nm} &= nmp + \mathcal{O}(p^2) \\ \sigma^2_{nm} &= nmp + \mathcal{O}(p^2) \end{cases} \tag{5.3}$$

Démonstration. voir Annexe, section 9.1. □

Comme dans notre cas $s = 2^{14}$, nous obtenons $\gamma_{nm} << 1$ pour des valeurs usuelles de n et m. La moyenne γ_{nm} peut donc être négligée par rapport à l'entier n_q dans l'équation 5.1. La variance étant inversement proportionnelle à s^q, la probabilité $p(X_q = n_q)$ décroit comme on s'y attendait de manière significative quand s augmente et $n_q > 0$.

5.2.2.2 Extension à plusieurs tailles de q grams

Considérons maintenant des q-grams de taille i et j avec $i > j$ entre deux empreintes. Chaque i-gram induit donc la présence de $i - j + 1$ j-grams entre ces deux empreintes. Par conséquent, l'existence de n_i i-grams induit l'existence de $n_i(i - j + 1)$ j-grams entre ces mêmes empreintes. A partir du nombre n_j de j-grams, on peut considérer que l'évènement relatif à l'apparition de $n_j - n_i(i - j + 1)$ j-grams restant est indépendant de l'évènement relatif au fait d'avoir n_i i-grams. On obtient alors :

$$\forall i > j \quad p(X_j = n_j | X_i = n_i) = p(X_j = n_j - n_i(i - j + 1)) \tag{5.4}$$

Notons que cette dernière relation est simplement une approximation du cas réel où les différents q-grams ne sont pas indépendants. Cependant, cette approche est en adéquation avec le modèle de balles et d'urnes introduit par Nicodème où les différents q-grams sont considérés comme indépendants.

En utilisant l'équation 5.4, si nous itérons la formule de la probabilité conditionnelle $P(A\&B) = P(A|B)P(B)$, $max - min$ fois sur $Q(n_{min}, \ldots, n_{max})$, on obtient :

$$Q(n_{min}, \ldots, n_{max}) = \prod_{i=min}^{max} p(X_i = u_i)$$

avec $u_{max} = n_{max}, u_{max-1} = n_{max-1} - 2u_{max}$, $u_i = n_i - \sum_{j=i+1}^{max} \alpha_{i,j} u_j$ et $\alpha_{i,j} = i - j + 1$.

L'expression $-\log_2(Q(n_{min}, \ldots, n_{max}))$ peut alors être écrite comme suit :

$$\begin{aligned} -\sum_{i=min}^{max} log_2(p(X_i = u_i)) &= \tfrac{1}{2} log_2(e) \sum_{i=min}^{max} \tfrac{u_i^2}{\sigma_i^2} \\ &+ \tfrac{1}{2} \sum_{i=min}^{max} \log_2(2\pi\sigma_i) \end{aligned}$$

Où σ_i correspond à la déviation standard de X_i. Notons aussi que la moyenne de X_i (équation 5.3) est négligée dans l'équation précédente.

Comme le second terme de l'équation précédente est indépendant de $(n_i)_{i \in \{min, \ldots, max\}}$ et que nous souhaitons uniquement trouver une approximation asymptotique de $-\log_2(Q(n_{min}, \ldots, n_{max}))$, nous définissons notre score $S(n_{min}, \ldots, n_{max})$ entre deux empreintes de la manière suivante :

$$S(n_{min}, \ldots, n_{max}) = \sum_{i=min}^{max} \frac{u_i^2}{\sigma_i^2} = \frac{1}{(mn)^2} \sum_{i=min}^{max} u_i^2 s^{2i} \tag{5.5}$$

Où chaque σ_i a été remplacé par son terme de premier ordre $nm\frac{1}{s^i}$ (equation 5.3)·

Comme $u_i << s$, si nous mettons de côté le facteur $\frac{1}{mn}$, notre fonction score peut être assimilée à la valeur du nombre $(0, \ldots, 0, u_{min}, \ldots, u_{max})$ en base s^2.

L'équation 5.5 fourni alors une mesure de la similarité entre deux empreintes qui permet de pondérer les informations relatives aux nombres de q-grams pour différentes tailles de q. Cependant, en pratique, à cause de la taille élevée de l'alphabet ($s = 2^{14}$), l'équation 5.5 peut renvoyer une valeur dépassant les capacités de stockage des types de variables couramment utilisées en programmation. Par conséquent, l'implémentation pratique de $S(n_{min}, \ldots, n_{max})$ doit être redéfinie de la façon suivante :

$$S_{prat}(n_{min}, \ldots, n_{max}) = \frac{1}{(mn)^2} \sum_{i=min}^{max} u_i^2 b^i \tag{5.6}$$

Où $b < s$ est défini dans la partie expérimentations (section 6.3.2) de manière à être aussi élevé que possible tout en évitant d'introduire des dépassements de mémoire. Notons que si b est suffisamment élevé, l'idée consistant à calculer la valeur de $(0, \ldots, 0, n_{min}, \ldots, n_{max})$ sur une base de taille importante reste valide.

Étant donné une signature I correspondant à un fichier d'entrée et une signature D de la base de données, l'équation précédente peut être réécrite de la façon suivante :

$$S_{prat}(I, D) = \frac{1}{(mn)^2} \sum_{i=min}^{max} u_i^2 b^i \tag{5.7}$$

où les nombres de q-grams $(n_{min}, \ldots, n_{max})$ entre I et D sont déduits de ces deux signatures et n'apparaissent plus en paramètre de notre score.

Comme le montrent les expériences que nous avons mené (Chapitre 6), si I correspond au co-dérivé d'une signature D présente dans la base, $S_{prat}(I, D)$ sera maximum parmi tous les $S_{prat}(I, D')$, D' appartenant à la base. L'équation 5.7 nous permet donc d'identifier un contenu co-dérivé lorsque l'original est présent dans la base de données. Toutefois, il nous faut également être capable de spécifier si une signature d'entrée correspond ou non au

co-dérivé d'une signature de la base. Si l'on dénote notre base de données par $\mathcal{B}$, on peut associer à chaque signature I son plus grand score avec les signatures de la base :

$$Score(I) = \max_{D \in \mathcal{B}} S_{prat}(I, D) \tag{5.8}$$

Il pourrait être tentant de fixer un seuil sur $Score(I)$ au delà duquel on considérerait que I correspond forcément au co-dérivé d'une signature de la base. Malheureusement, comme le montre les expériences menées au chapitre 6, la grande diversité des signatures (et donc des scores) ne nous permet pas de définir un tel seuil. En effet, il est possible de trouver des signatures I et I' telles que I corresponde au co-dérivé d'une signature de la base et I' n'ait pas de correspondant dans la base alors que $Score(I') > Score(I)$.

Toutefois, nos expériences nous ont conduit à conclure que lorsque une signature I correspond au co-dérivé d'un signal D de la base, $S_{prat}(I, D)$ est nettement plus important que n'importe quel score $S_{prat}(I, D')$ avec $D' \in \mathcal{B}$ différent de D. Inversement, si aucune signature $D \in \mathcal{B}$ ne correspond à un co-dérivé de I tous les scores $S_{prat}(I, D)$ seront sensiblement équivalents. Ceci nous a amené a concevoir une règle de décision basée à la fois sur la valeur du score maximal et sur la prédominance de celui-ci vis a vis du score immédiatement inférieur. On considérera donc qu'une signature d'entrée I correspond au co-dérivé d'une signature de la base ssi :

$$Score(I) > S_1 \text{ et } \frac{Score(I)}{S_{prat}(I, D_I^2)} > S_2$$

où S_1 et S_2 sont deux seuils fixés expérimentalement et $S_{prat}(I, D_I^2)$ est le score immédiatement inférieur à $Score(I)$ parmi tous les scores obtenus dans la base de données.

5.3 Score par distance d'édition

Comme nous l'avons vu dans la section précédente, la mesure du nombre de q grams communs entre deux chaînes ne fournit pas une mesure suffisamment discriminante pour identifier un contenu co-dérivé à partir d'un simple seuil. Nous nous sommes donc intéressé à d'autres mesures de distances. Dans le contexte de reconnaissance d'empreintes [32, 31], les

techniques d'appariement de chaînes (section 3.2.1) présentent deux propriétés intéressantes.

- Tout d'abord, contrairement à la distance de Hamming, le poids associé à une substitution de symbole permet de considérer deux valeurs de sous-empreintes comme égales si la différence entre celles-ci est inférieure à un certain seuil. Cette première propriété permet d'introduire une certaine flexibilité dans la comparaison des empreintes en considérant par exemple deux sous empreintes comme similaires si elles sont différentes de quelques dixièmes de milliseconde.
- Secondement, les opérations de suppression et d'insertion constituent un outil approprié pour les problèmes nécessitant de prendre en compte l'addition ou l'oubli de valeurs de sous-empreinte entre deux contenus co-dérivés. Dans notre cadre, cela revient à anticiper le fait que la méthode de segmentation puisse détecter des pics additionnels ou au contraire oublier certain pics entre deux signaux co-dérivés. Ces pics additionnels ou manquant induisent des valeurs d'intervalle I_{emax} erronées entre deux empreintes de contenu co-dérivé.

Des séquences de symboles similaires entre deux chaînes peuvent donc être identifiées en calculant une distance d'édition entre ces deux séquences. A partir de la définition d'un alphabet, la distance d'édition est souvent définie par l'affectation d'un score positif pour une substitution, insertion ou suppression de symbole et un score nul lorsque deux symboles correspondent. Cette notion de coût affecté aux opérations d'édition de chaîne peut être utilisée pour définir une fonction de similarité. Dans ce cas, l'apparition de deux symboles identiques se voit gratifiée d'un coût positif tandis que les opérations de suppression ou insertion entrainent un coût négatif. Une telle fonction de score S peut par exemple être définie par l'équation 5.9.

$$S(i,j) = \begin{cases} S(i-1,j-1) + \alpha & \text{Si } s_i = s_j \\ max \begin{pmatrix} 0, \\ S(i,j-1) - \beta, \\ S(i-1,j) - \beta \end{pmatrix} & \text{sinon} \end{cases} \tag{5.9}$$

Où s_i et s_j correspondent aux deux symboles de rang i et j des deux empreintes comparées et α et β sont deux constantes stritement positives (Section 6.3.3).

La figure 5.1 représente un exemple de matrice de score obtenue avec $\alpha = 5$ et $\beta = 7$. Dans cet exemple, le score de similarité obtenu en localisant le score maximal de la dernière ligne est égal à 23.

	Cible									
Requête	20	3	12	23	15	3	18	21	7	5
3	0	5	0	0	0	5	0	0	0	0
12	0	0	10	3	0	0	0	0	0	0
23	0	0	3	15	8	1	0	0	0	0
15	0	0	0	8	20	13	6	0	0	0
18	0	0	0	1	13	6	18	11	4	0
21	0	0	0	0	6	0	11	23	16	9

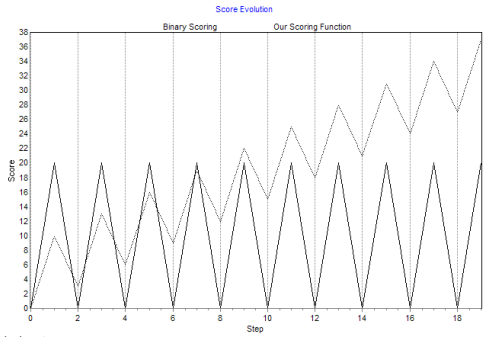
FIG. 5.1 – Matrice de score entre les chaînes 3.12.23.15.18.21 et 20.3.12.23.15.3.18.21.7.5 obtenues à partir de l'équation 5.9 pour $\alpha = 5$ et $\beta = 7$.

A partir d'une telle fonction de score, une longue séquence de correspondances suivie par une séquence de non-correspondances peut fournir le même résultat que deux chaînes alternant symboles identiques et différents. Par exemple, le score entre deux suites `abxy` et `abuv` fournira la même score que celle entre `axby` et `aubv`. Ce comportement est du à l'ajout ou à la soustraction des constantes α et β lors des mises en correspondance ou des suppression de symboles. Ainsi un alignement de deux chaînes nécessitant n mises en correspondances et p opérations de suppressions sera affecté d'un score égal à $n\alpha - p\beta$ et ce indépendamment de l'agencement des opérations de mises en correspondance et des suppression dans l'une ou l'autre chaîne.

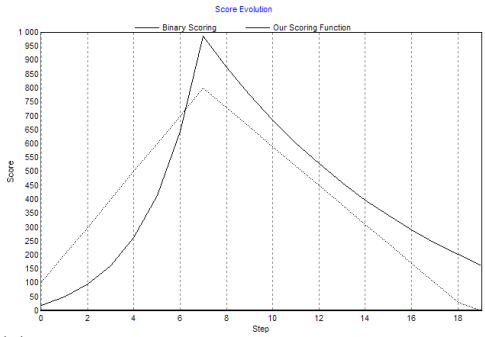
Cependant, dans le contexte de la reconnaissance d'empreintes, deux empreintes co-dérivées partagent de longues séquences de symboles avec peu de symboles erronés. Notre fonction de score doit donc favoriser les longues séquences de symboles identiques entre deux chaînes. Nous avons défini à cette fin une fonction de score pondérée au comportement non-linéaire. La méthode la plus simple pour réaliser une telle fonction consiste à définir $S(i, j)$ comme une fonction affine de $S(i-1, j-1)$ (dans le cas d'une correspondance) ou $S(i-1, j)$ et $S(i, j-1)$ (dans le cas d'une opération de suppression). Chaque case de la matrice de score est donc pondérée par le score de la case précédente afin de permettre une augmentation du score plus franche lors de longue suites de symboles et donc favoriser les empreintes co-dérivées. Cette fonction est définie de la manière suivante :

$$S(i,j) = \begin{cases} \alpha S(i-1,j-1) + \beta & \text{Si } s_i = s_j \\ \frac{1}{\gamma} max \begin{pmatrix} 0, \\ S(i,j-1) - \beta, \\ S(i-1,j) - \beta \end{pmatrix} & \text{sinon} \end{cases} \tag{5.10}$$

Les constantes α, β, γ sont déterminées expérimentalement mais doivent satisfaire la condition $1 < \gamma < \alpha$ afin que le score décroisse plus doucement quand un symbole différent est rencontré qu'il n'augmente pendant une séquence de symboles identiques. Ce comportement est mis en évidence sur la figure 5.2(a). Le score résultant de la mise en correspondance d'une chaîne de longueur n avec elle même est égal à $\beta \sum_{i=0}^{n-1} \alpha^i = \beta \frac{\alpha^n - 1}{\alpha - 1}$ (Figure 5.2(b)). Le score défini par l'équation 5.10 peut être calculé en $\mathcal{O}(nm)$ où n et m représentent les tailles des deux sous-empreintes.



(a) Alternance de mises en correspondances et de suppressions.



(b) Une Suite de correspondances suivi d'une suite de suppressions.

FIG. 5.2 – Comparaison des fonctions de score basées sur les équations 5.9 et 5.10

Dans notre cas, la comparaison d'une empreinte inconnue avec la totalité de la base de données peut être réalisée par des méthodes d'alignement local [31] basées sur notre fonction de score (équation 5.10). Cependant, à partir d'une empreinte inconnue de taille n et N empreintes de taille m stockées dans la base de donnée, une telle recherche exhaustive nécessiterait $\mathcal{O}(Nnm)$ opérations. Nous avons donc repris la notion de filtrage par q-gram déjà utilisée dans la section 5.2.2. Toutefois, le théorème de Jokinen-Ukkonen [34](section 5.2.2) sur lequel est basée cette approche ne s'applique pas à la croissance/décroissance polynomiale de notre fonction de score. Plutôt que de compter simplement le nombre de q-grams communs à deux chaînes, notre algorithme associe à chacun de ces q gram les sous séquences

commençant sur celui-ci dans chacune des deux chaînes (Figure 5.3). Chaque q-gram est alors pondéré par un score défini par l'équation 5.10 et calculé entre deux sous-empreintes contenant le q-gram.

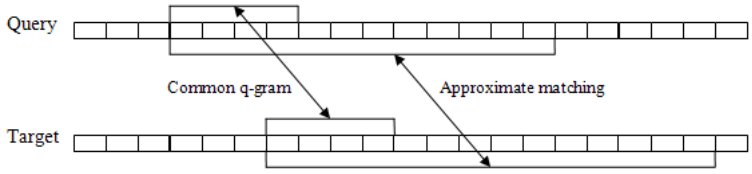


FIG. 5.3 – Filtrage par q-grams et appariement de sous empreintes

Plus formellement, notons $Q_{D,I}$ un q-grams en commun entre l'empreinte inconnue I et une empreinte stockée dans la base de données D. Ce q-gram apparaît aux indices $i_1, \ldots, i_p$ de l'empreinte I et aux indices $j_1, \ldots, j_q$ dans l'empreinte D. Si nous souhaitons comparer les deux empreintes sur une suite de symboles de taille m, le score associé à $Q_{D,I}$ est alors défini par :

$$score(Q_{D,I}) = \sum_{k=1}^{p} \sum_{l=1}^{q} S(I[i_k, i_k + m], D[j_l, j_l + m]) \tag{5.11}$$

Où $S(I[i_k, i_k + m], D[j_l, j_l + m])$ correspond à notre fonction de score calculée entre les deux empreintes sur une longueur m à partir des index i_k et j_l. L'alphabet de l'empreinte étant très large (2^{14}), p et q appartiennent à $\{0, 1\}$ pour la plupart des empreintes de la base de données. Dans les expérimentations, la valeur de m a été fixée à 20 afin de correspondre à une seconde de signal, en fonction de notre taux de calcul de sous-empreinte (Chapitre 6).

Le score codant la similarité entre une empreinte d'entrée I et une empreinte de la base D peut alors être défini comme la somme des scores des q-grams communs à I et D :

$$score(I, D) = \sum_{Q_{D,I} \subset D} score(Q_{D,I}) \tag{5.12}$$

Les empreintes obtenant les scores les plus élevés calculées par notre méthode de filtrage sont alors considérées comme des candidates potentielles aux contenus co-dérivés. Nos expérimentations (Chapitre 6) montrent qu'une empreinte co-dérivée, si elle est contenue dans la base de données, est toujours celle obtenant le meilleur score. Ceci permet de re-

trouver une empreinte co-dérivée quand elle existe. Cependant, une méthode d'identification doit aussi être capable de décider si un extrait inconnu possède ou non un contenu co-dérivé stocké dans la base de données. A partir d'un score attribué à chaque empreinte, il faut alors fixer un seuil tel que :

1. Le score le plus bas obtenu par tout couple d'empreintes co-dérivées doit être au dessus de ce seuil
2. Le meilleur score obtenu par tout couple d'empreinte non co-dérivée doit être inférieur au seuil

Un tel seuil permet a priori de décider si l'extrait inconnu correspond au co-dérivé d'un signal audio stocké dans la base de données. Cependant, comme le montre le chapitre 6, le meilleur score obtenu par deux empreintes non co-dérivées est supérieur au plus mauvais score obtenu entre deux empreintes co-dérivées. Il n'existe donc pas de seuil satisfaisant les deux contraintes précédentes. Toutefois, le score défini par l'équation 5.12 classe toujours en première position l'empreinte co-dérivée quand elle existe. On se retrouve donc avec un problème similaire à celui rencontré dans la section 5.2.2 et on peut comme dans cette dernière section envisager de définir un score basé sur le rapport entre le meilleur et le second score au sens de l'équation 5.12.

Nous avons toutefois privilégié une autre approche. En effet, le fait que notre méthode de filtrage ne fournisse pas une règle de décision valide est principalement dû à la faible longueur m de sous-empreintes considérée pour chaque q-gram (équation 5.11). Puisque l'équation 5.12 nous permet d'isoler l'empreinte co-derivée si elle existe nous pouvons définir notre score en deux temps :

1. Filtrer rapidement la base de données à l'aide de l'équation 5.12 pour extraire l'empreinte potentiellement co-dérivée.
2. Concevoir un critère de décision basé sur un nouveau score éventuellement plus coûteux entre l'empreinte d'entrée et l'empreinte précédemment sélectionnée.

Soit I notre chaîne d'entrée et D la chaîne de la base de données de score maximum au sens de l'équation 5.12. Soit également les deux indices i_{max} et j_{max} dans I et D tels que :

1. $I[i_{max}, i_{max} + q - 1] = D[i_{max}, i_{max} + q - 1]$ et

2. $S(I[i_{max}, i_{max}+m], D[i_{max}, i_{max}+m])$ (équation 5.10) est maximum parmi tous les calculs de score effectués lors de l'évaluation de $Score(I[i_{max}, i_{max}+q-1])$ (équation 5.11).

La chaîne $I[i_{max}, i_{max}+q-1]$ est donc un q gram commun à I et D commençant respectivement en i_{max} et j_{max}. Son score au sens de l'équation 5.10 entre les deux sous-chaînes de I et D de longueur m est maximum. C'est à dire qu'il est le plus élevé parmi tous les calculs de score de longueur m à partir de q grams communs de I et D. Notre score final est basé sur une comparaison sur une durée $M >> m$ des empreintes I et D aux positions i_{max} et j_{max}. Il est donc défini par :

$$score(I) = score(I[i_{max}, i_{max}+M], D[j_{max}, j_{max}+M]) \tag{5.13}$$

où M est une constante supérieure à m. Les expériences menées au chapitre 6 confirment qu'un choix adéquat de M permet d'obtenir un score suffisamment discriminant pour décider si une signature d'entrée possède un contenu co-dérivé dans la base de données.

6

Analyse des résultats

6.1 Introduction

Nous avons présenté dans les chapitres précédents nos méthodes de calcul et de comparaison d'empreintes pour l'identification de fichiers audio. Dans ce chapitre, nous mesurerons les performances de chaque méthode prise individuellement et examinerons également les performances globales de systèmes combinant calcul d'empreinte et méthode d'identification. Les premiers résultats évalueront notre méthode de calcul d'empreinte introduite dans le chapitre 4. Chacune des étapes ayant conduit à l'algorithme final sera évaluée afin de juger les gains induits par nos améliorations successives. Nous comparerons également ces résultats avec des méthodes concurrentes. Dans un second temps, nous évaluerons notre nouvelle méthode de comparaison d'empreinte sur une base de données d'empreintes précalculées. La capacité d'identifier un fichier audio induite par nos méthodes de comparaisons d'empreinte sera également évaluée.

Sommaire

6.2 Évaluation de la robustesse de l'empreinte

Pour faire ces expérimentations, nous avons utilisé un ensemble de documents de genres très divers qui représentent 24h de signal sonore. Certains morceaux de musiques sont représentés par une version studio ainsi qu'une version live. Ils sont donc considérés comme des documents différents. Pour évaluer la robustesse des empreintes, nous avons soumis tous ces documents musicaux à des compressions/décompressions de taux variables afin d'étudier la variabilité d'une empreinte en fonction de ce paramètre. Ces documents compressés ont aussi subi des décalages temporels par ajout de segments de silence ou d'autres signaux audio de durée variables au début du signal à reconnaître.

6.2.1 Taille de l'empreinte

Les tailles des intervalles I_e et I_o ont été respectivement fixées à 1 et 100 millisecondes. Ces valeurs offrent un bon compromis entre la taille et la robustesse des empreintes. En effet, I_o égal à $100ms$ permet d'avoir suffisamment d'échantillons pour trouver un intervalle I_e significatif sur cette période tout en assurant un nombre de valeurs calculées par secondes suffisant pour, par la suite, reconnaître tout morceau à partir d'un extrait de quelques

secondes. Quand à I_e, sa taille doit être suffisamment petite pour correspondre à la détection d'un point particulier du signal tout en restant suffisamment importante pour caractériser celui-ci de façon robuste.

En utilisant ces valeurs pour I_0 et I_e et en considérant un taux d'échantillonnage du signal audio T_e égal à 44100kbps, le taux moyen d'intervalles I_{emax} détectés sur l'ensemble de la base de données est égal à $21,9$ intervalles par seconde. L'écart type de cette mesure est égal à $3,5$. Les valeurs minimales et maximales de détection d'intervalles I_{emax} trouvées sur notre base sont respectivement égales à 18 et 34 valeurs de sous-empreinte par seconde.

De plus, à partir de ces valeur d'intervalles prédéfinies, la valeur maximale V_{max} d'une sous-empreinte est définie par $V_{max} = (I_o - I_e) * T_e = 4365.9ms$. Cela permet de déduire qu'une valeur de sous empreinte peut être codée et stockée sur 13 bits. Si l'on reprend notre taux moyen de sous-empreintes détectées et calculées par seconde soit 21,9, on en déduit que la taille nécessaire pour stocker une empreinte correspondant à une minute de signal est égale à $21.9 * 13 * 60 = 17082$ bits par minutes soit 2,13Ko/min

D'un autre côté, la méthode de Kalker et Haitsma, la principale référence dans ce domaine, utilise des intervalles de 370 ms avec un taux de recouvrement de 31/32, ce qui correspond à une valeur de sous-empreinte calculée toutes les $11,56$ ms. Comme nous l'avons vu dans la section 3.1.2, cette méthode calcule un ensemble de différences inter filtres stockées dans une empreinte de 32 bits. L'espace de stockage nécessité par cette méthode pour stocker l'empreinte d'une minute de signal est donc égal à $(60/0.01156) * 32 = 20,7$ Ko/min.

Par conséquent, notre méthode introduit un gain d'espace de stockage d'environ 90% par rapport à la méthode de Kalker et Haitsma. Ces résultats ont été confirmés lors de nos expérimentations en mesurant le nombre total de sous empreintes calculées multiplié par le nombre de bits requis par le stockage d'une sous empreinte (soit 14) divisé par le nombre de minutes de documents sonores utilisées pour calculer ces empreintes (le résultat obtenu est enfin divisé par 8 pour avoir une valeur en octets).

6.2.2 Mesures de performances

Définissons T_i comme étant l'ensemble des intervalles du signal s_i servant à calculer son empreinte. Comme on l'a vu précédemment, avec la méthode de Kalker et Haitsma [37] , ce

nombre est égal au nombre de fenêtres glissantes utilisées. Avec notre méthode, ce nombre est égal au nombre d'intervalles I_{emax} détectés par notre méthode de segmentation.

A partir d'un signal audio s_i, définissons aussi $SP_i \subset T_i$ l'ensemble d'intervalles détectés à la même position dans le signal s_i et une version dégradée de ce signal. En nous basant sur le comportement de notre algorithme, nous considérons deux instants particuliers comme étant détectés à la même position si la distance qui sépare ces instants est inférieure à $0,25ms$. Considérons de plus, l'ensemble $SV_i \subset SP_i$ des instants particuliers ayant la même position et la même valeur de sous-empreinte entre s_i et sa version dégradée.

Si l'on applique une dégradation à un contenu musical, plusieurs mesures peuvent nous fournir un premier indice de performances de nos algorithmes :

Taux de segmentation : Cette mesure représente la valeur moyenne d'intervalles I_{emax} détectés localisés à la même position entre un contenu original et sa version dégradée. Cette quantité, qui revient à évaluer les performances de notre algorithme de segmentation, est définie par :

$$SR = \frac{1}{N} \sum_{i=1}^{N} \frac{|SP_i|}{|T_i|} \tag{6.1}$$

où $|.|$ dénote le cardinal de l'ensemble et N le nombre de fichiers audio contenus dans la base de données

Taux de reconnaissance : La robustesse de la méthode utilisée pour calculer les valeurs de sous-empreinte peut être mesurée, pour chaque signal s_i, par le ratio entre la valeur de SV_i et SP_i. Nous mesurons donc parmi les instants correctement détectés le taux de ceux dont la valeur de sous-empreinte reste inchangée malgré la dégradation appliquée au contenu. La valeur moyenne de ce ratio sur toute la base de donnée est définie par :

$$RR = \frac{1}{N} \sum_{i=1}^{N} \frac{|SV_i|}{|SP_i|} \tag{6.2}$$

On considérera également le **taux de variations** défini par $1 - RR$ et qui représente le taux d'intervalles différents entre les deux signaux.

Taux de reconnaissance total : Le taux de reconnaissance défini ci-dessus mesure la robustesse de notre méthode de calcul d'empreinte indépendemment de l'étape de seg-

mentation. Une mesure combinant ces deux étapes peut alors être défini en calculant, pour chaque signal s_i, le ratio entre SV_i et T_i. Cette valeur correspond donc au taux d'invariance global de notre méthode de calcul d'empreinte. Elle est définie par :

$$TRR = \frac{1}{N} \sum_{i=1}^{N} \frac{|SV_i|}{|T_i|} \tag{6.3}$$

La méthode de Kalker et Haitsma [37] utilise des fenêtres glissantes et ne comporte pas d'étape de segmentation. La seule mesure de performance applicable dans ce cas est donc le taux de reconnaissance total.

6.2.3 Résistance à la compression

Les fichier musicaux de notre base de données ont été encodés à 705 $Kbps$. Les taux de compression utilisés peuvent alors être définis par le nombre de bits par secondes servant à encoder les fichier ou le ratio par rapport à l'encodage du document original. Les encodages utilisés pour nos expérimentations sont 48, 64, 96, 128, 192 et 256 $Kbps$ ce qui correspond à une compression de 14.7, 11.02, 7.35, 5.5, 3.67 et 2.75% du fichier original.

Nous avons tout d'abord comparé la méthode d'Haitsma avec notre première méthode inspirée de celle-ci ainsi qu'avec notre proposition finale. La figure suivante montre le taux de reconnaissance total de ces 3 méthodes :

La courbe du bas de la figure 6.1 (—) représente le taux de reconnaissance total (équation 6.3) obtenu par la méthode d'Haitsma, calculée en fonction de différents taux de compression. Le taux de reconnaissance obtenu par cette méthode oscille alors entre 5 et 30% de sous-empreintes communes entre deux contenus co-dérivés, un original et un compressé. Précisons encore que pour cette méthode, aucune méthode de segmentation n'est appliquée. Par conséquent, cette courbe montre un taux de reconnissance total inférieur à 30% pour un taux de compression usuel voir plutôt faible. Cet taux décroit significativement lorsque le taux de compression augmente.

La seconde courbe (- -) représente le taux de reconnaissance total obtenu par notre première idée de calcul d'empreinte basée sur une amélioration de la comparaison de filtre fréquentiels (Section 4.2). Cette courbe monte à 62% pour un encodage à 256$Kbps$ et décroît

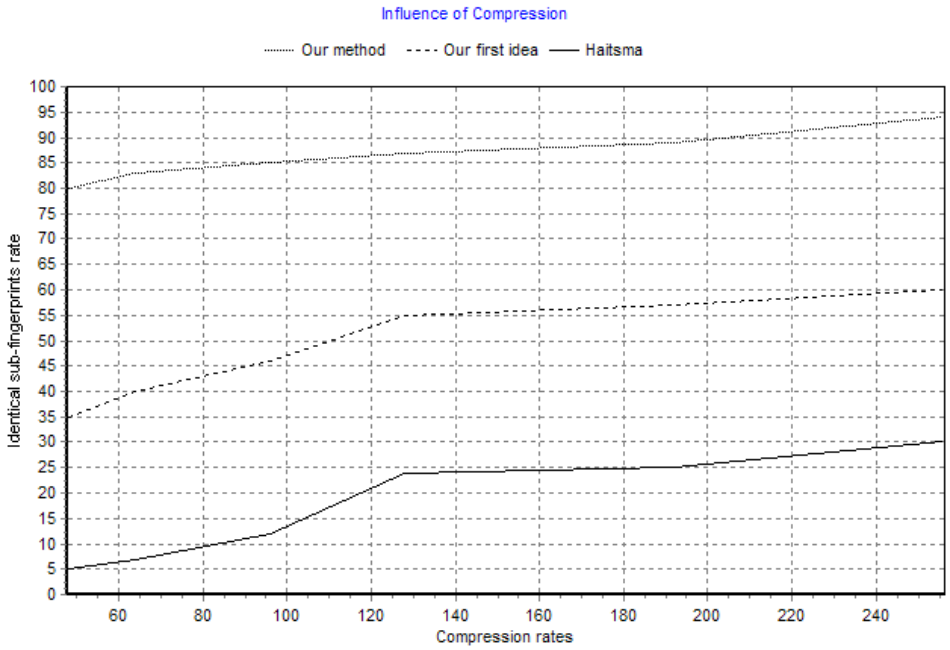


FIG. 6.1 – Taux de valeurs de sous-empreinte communes entre un original et sa version compressée (à 48, 64, 96, 128, 192 et 256 Kbps)

jusqu'à 34% pour $48Kbps$. Cette courbe ne prend pas en compte l'algorithme de segmentation et compare les empreintes calculées entre deux contenus co-dérivés lors d'instants détectés au même emplacement dans le signal.

Cette courbe permet donc de comparer efficacement la méthode d'Haitsma avec notre première proposition en faisant abstraction de la partie segmentation. Nous pouvons donc observer que cette première proposition obtenait déjà un taux de reconnaissance plus important (35% minimum) que celle d'Haitsma (30% maximum).

La courbe du haut (·····) correspond au taux de segmentation obtenu par notre méthode (Section 4.3). Comme nous l'avons expliqué, nous considérons ici le taux d'intervalles de haute énergie correctement détectés, c'est à dire à la même position dans le signal original et dans le compressé. Notre méthode de segmentation obtient un taux de reconnaissance total variant de 93% pour une compression à $256Kbps$ à seulement 80% pour une compression $48Kbps$ en passant à 87% pour un taux de compression de $128Kbps$. Nous avons aussi observé que lorsqu'un instant est mal détecté, il entraîne une légère succession d'instants (environ 3 à 4) faussement détectés le temps pour l'algorithme de se recaler sur des instants

significatifs aussi compris dans l'original.

De plus, cette courbe montre l'efficacité de notre méthode de segmentation, qui permet d'obtenir un très faible taux d'instants mal détectés (pas plus de 20%). En effet, cette méthode se base sur des propriétés moins sensibles à la compression que d'autres parties du signal moins pertinentes. C'est ce résultat qui nous a mis sur la voix de notre proposition finale. En effet, pourquoi ajouter à cela une seconde étape rajoutant elle aussi une perte d'information quand la première étape fournit une information caractéristique, discriminante et robuste ?

Nous avons donc défini nos valeurs de sous empreinte comme étant l'écart entre deux instants détectés. Cela revient à considérer cette courbe de taux de segmentation comme étant sensiblement égale au taux de reconnaissance total obtenu par notre proposition finale de calcul de sous-empreinte. Par conséquent, notre proposition finale obtient un taux de valeur de sous-empreinte erronées de seulement 20% dans le pire des cas considéré.

6.2.4 Invariance aux décalages temporels

Les fichiers utilisés précédemment ont aussi été soumis à un autre type de dégradation. Cette dégradation est appelée « décalage temporel ». Ce type de dégradation peut intervenir de différentes manière :

coupure de parties : en effet, un morceau de musique peut être reformaté et ainsi subir l'ablation de parties inutiles. Tel est le cas d'applaudissements interminables en début de titre live ou encore, lors de passages à la radio, de suppression de solo guitare trop longs voir ennuyeux afin de réduire la durée d'un morceau de musique.

insertion : il n'est pas rare non plus qu'un morceau de musique comporte des séquences musicales qui ne lui appartiennent pas, c'est le cas de morceaux techno par exemple où une chanson peut être interrompue pour laisser place à un extrait ou un son particulier, mais encore des chansons passées à la radio interrompues en pleine écoute par un jingle.

prise intantannée : c'est un cas particulier du premier cas. Ici, le fait de calculer l'empreinte en cours de lecture est simulée par le fait de couper une partie du début, de rajouter un blanc, ou de considérer un extrait pris aléatoirement dans un morceau.

Par conséquent, le calcul d'empreinte doit être le plus invariant possible aux décalages temporels afin que l'empreinte calculée permette de reconnaître le contenu co-dérivé malgré cette altération.

Dans un premier temps, nous avons testé le degré d'invariance vis à vis des décalages temporels en ajoutant, en début de morceaux, un blanc de taille variable. La durée de ces blancs était de 1, 2, 3, 5 et 6.25ms. La figure 6.2 représente le taux de reconnaissance total des différents algorithmes c'est à dire le pourcentage de valeurs de sous-empreintes identiques et positionnées au même moment dans le signal.

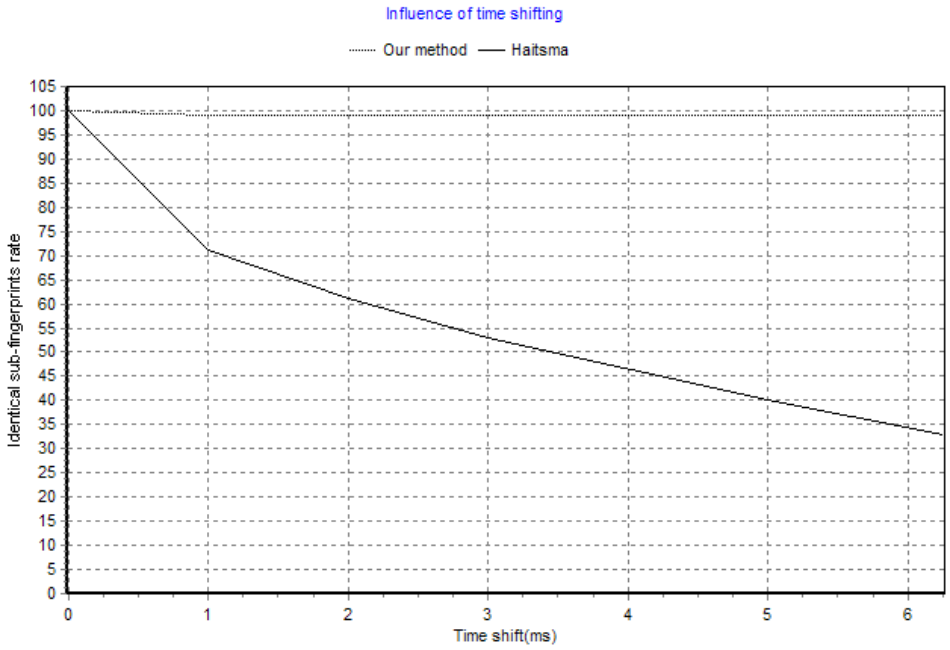


FIG. 6.2 – Taux de valeurs identiques entre un original et sa version décalée (de 1, 2, 3, 5 et 6.25 ms)

La courbe du haut (····) de la Figure 6.2 représente le taux de segmentation obtenu par notre méthode vis à vis des décalages temporels. On rappelle en effet que cette méthode a été développée pour synchroniser le processus de calcul de sous-empreinte afin de le rendre robuste à ce genre de dégradations. Calculer le taux de segmentation, dans notre méthode finale, revient alors à calculer le comportement global de notre méthode puisque notre calcul de sous-empreinte est basé sur l'efficacité de cette segmentation. Nous pouvons alors observer un taux de segmentation légèrement inférieur à 100%, mais supérieur à 99%, lorsqu'un

décalage temporel est inséré au début d'un extrait musical. Ce léger écart est simplement la conséquence du temps requis par notre méthode pour se resynchroniser sur un instant significatif du signal, c'est à dire quelques valeurs de sous-empreinte. Une fois l'empreinte d'entrée resynchronisée par rapport à l'original, peu importe la durée du décalage introduit, le taux d'instants détectés reste constant. Cette constatation est resté vraie lorsque nous avons introduit un blanc de durée plus importante (10, 25, et 50ms). Nous pouvons en conclure que la taille du décalage n'influe pas sur la resynchronisation ni sur la segmentation d'après synchronisation. Nous pouvons aussi en déduire que la resynchronisation, une fois le blanc terminé, s'effectue très rapidement. En effet, en moyenne, l'équivalent de 2 à 3 sous-valeur erronées sont nécessaires pour ensuite se resynchroniser). L'autre courbe de cette Figure 6.2 (—) représente le taux de reconnaissance de la méthode de Kalker et Haitsma servant de référence. Nous pouvons observer que cette technique souffre d'une baisse significative de son taux de sous-valeurs identiques dès lors qu'un décalage est introduit (de 100 à 70% pour un décalage d'1ms). Cette courbe reflète aussi une décroissance importante lorsque la taille du décalage augmente pour enfin atteindre un taux de sous-valeurs identiques de 33% lorsque le blanc inséré au début est de $6,25ms$. Ces performances sont dues à leur méthode à base de fenêtre recouvrante, méthode figée qui atténue les conséquences d'un décalage temporel sans les supprimer.

Dans un second temps, nous avons soumis notre méthode de segmentation aux autres types de décalages temporels afin de confirmer les premiers résultats. Nous avons utilisés des extraits musicaux piochés aléatoirement au cours de la lecture du morceau de musique. Nous avons aussi inséré des extraits d'autres morceaux de musique ou encore supprimé des parties au milieu d'un même signal sonore. Peu importe le type de dégradation temporelle introduite, notre méthode de segmentation montre le même comportement que précédemment. En effet, lorsqu'il s'agit de cours échantillons provenant d'autres documents musicaux par exemple, cela n'influe que sur la durée de ce signal étranger, lui même obtenant des valeurs de sous-empreintes identiques à son original si celui-ci est contenu dans la base de données. Lorsqu'il s'agit d'extraits piochés aléatoirement, seulement quelques valeurs de sous-empreinte au début du processus de calcul d'empreinte sont faussement extraites avant que l'algorithme ne se resynchronise. De la même manière, couper un échantillon au milieu de l'extraction

d'empreinte revient à fausser quelques valeurs extraites puis se resynchroniser sur un instant significatif et ainsi reprendre le calcul de valeurs de sous-empreinte identiques.

Nous pouvons donc conclure que notre méthode de segmentation offre une réponse efficace au problème de décalage temporel avec un taux de sous-valeurs identiques extrêmement élevé voir quasi-invariant.

6.3 Identification d'empreinte

L'efficacité de notre méthode de reconnaissance a été testée en extrayant aléatoirement de cours extraits de 5s dans les documents audio de la base de données. Ces extraits ont alors été compressés à 128kbps, cette dégradation étant la plus commune appliquée aux documents audio numériques transitant sur l'Internet. Nous avons enfin calculé l'empreinte de chacun de ces extraits compressés.

6.3.1 Pertinence des q-grams

Avant d'exposer les performances de nos méthodes pour un taux d'identification quelconque, il convient d'apporter la preuve de la pertinence des q-grams. Nous utilisons pour cela les extraits compressés à 128 kbps des morceaux présents dans la base. Soit $\mathcal{C}$ cet ensemble d'extraits. Pour chaque empreinte compressée $e \in \mathcal{C}$ nous avons sélectionné les trois empreintes de la base de score le plus élevé (au sens de l'équation 5.8). Ces empreintes sont respectivement notées $s_1(e), s_2(e)$ et $s_3(e)$. Pour tout rang $i \in \{1, 2, 3\}$ on définit également les deux empreintes :

$$\begin{cases} min_i &= argmin_{e \in \mathcal{C}} S_{prat}(e, s_i(e)) \\ max_i &= argmax_{e \in \mathcal{C}} S_{prat}(e, s_i(e)) \end{cases}$$

Notons que pour chaque version compressée le fichier original est toujours présent dans la base. Nous comparons donc une empreinte de signal non compressé avec sa version compressée. Dans cette expérience l'empreinte de la version originale à toujours été celle obtenant le meilleur score au sens de l'équation 5.8. L'empreinte $s_1(e)$ correspond donc dans tous les cas à l'empreinte originale de e. Les empreintes min_1 (resp. max_1) correspondent donc aux

empreintes compressées les plus éloignées (resp. proche) de leur original (toujours au sens de l'équation 5.8).

L'empreinte de la base obtenant le second score représente la meilleure « mauvaise candidate ». On peut donc interpréter son score comme le score qui serait obtenu si on avait enlevé le fichier original de notre base de données. Dans ce dernier cas, la version compressée de notre empreinte ne devrait pas être reconnue. L'empreinte max_2 représente dans ce cadre l'empreinte la plus proche de son meilleur « mauvais candidat ».

L'écart entre la deuxième empreinte et la troisième comparé à celui entre la première et la deuxième permet d'évaluer de combien une empreinte se détache du reste de la base (en terme de score ou de q-grams) lorsque on la compare à un co-dérivé.

La table 6.1 présente différentes mesures sur les q grams obtenues en comparant chaque empreinte compressée avec l'ensemble de la base. Les groupements de lignes $1^{\text{er}}, 2^{\text{e}}, 3^{\text{e}}$ représentent les mesures calculées entre les empreintes $e \in \mathcal{C}$ et respectivement $s_1(e), s_2(e)$ et $s_3(e)$. Pour chaque groupement i :

- la ligne (i,moy), représente le nombre moyen de q-grams entre les empreintes compressées e et $s_i(e)$ pour $e \in \mathcal{C}$,
- La ligne (i,min) représente le nombre de q grams en communs entre min_i et $s_i(min_i)$.
- La ligne (i,max) représente le nombre de q grams en communs entre max_i et $s_i(max_i)$.

		3	4	5	6	7	8	9	10	11	12	13	14
1^{er}	min	91	6	0	2	0	1	0	0	1	0	0	0
	moy	120	13	3.65	1.54	1.15	0.81	0.55	0.46	0.33	0.34	0.24	0.21
	max	124	4	3	0	0	0	1	0	0	0	1	1
2^{e}	min	0	0	0	0	0	0	0	0	0	0	0	0
	moy	74	9.2	2.19	0.17	0.03	0	0	0	0	0	0	0
	max	180	40	7	4	1	2	0	0	0	0	0	0
3^{e}	min	0	0	0	0	0	0	0	0	0	0	0	0
	moy	48	5.5	0.96	0.09	0.01	0	0	0	0	0	0	0
	max	233	25	5	1	0	2	0	0	0	0	0	0

TAB. 6.1 – Longueur Q et nombre N de q-grams partagés entre l'extrait inconnu et ceux de la base de données. Le premier correspond à l'original, le second et troisième correspondent aux meilleurs faux positif.

Les lignes $(1^{\text{re}}, min)$ et $(2^{\text{e}}, max)$ de la table 6.1 montrent que pour chaque valeur de q on peut toujours trouver deux empreintes compressées s et s' tel que le nombre de q gram entre s et son co-dérivé classé premier est plus petit que le nombre de $q - grams$ entre s'

et l'empreinte classée seconde (qui ne correspond donc pas à un co-dérivé). Ce type résultat interdit de distinguer les contenu co-dérivé uniquement sur le nombre de q grams communs. Notons toutefois que le nombre moyen de q-grams partagés par deux co-dérivés est nettement supérieur à celui obtenu avec les candidats classés second et troisième. Le nombre moyen de q-grams de taille q est ainsi représenté sur la figure 6.3 pour une meilleure visibilité de l'écart entre courbes :

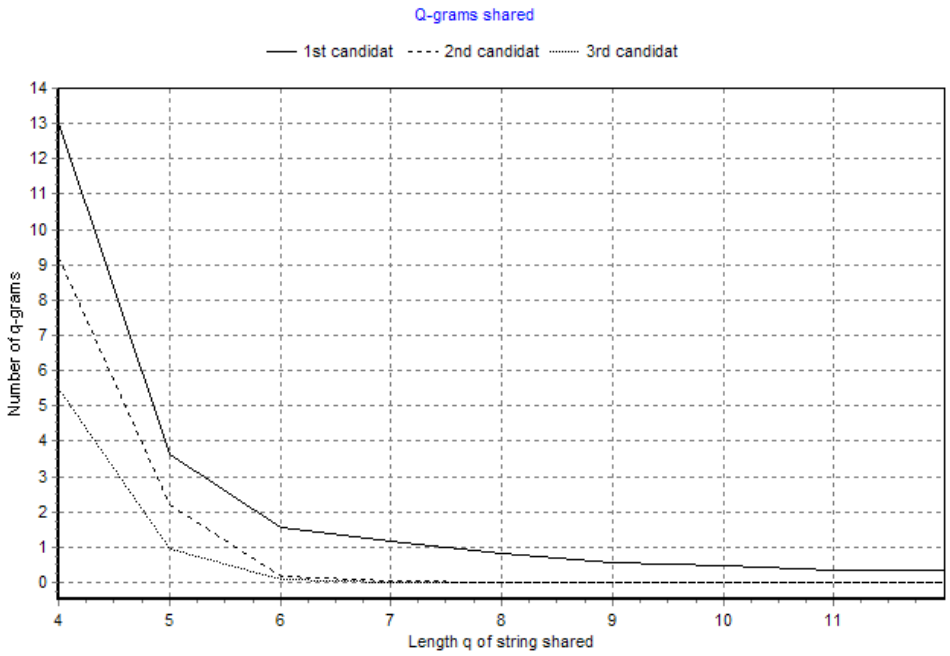


FIG. 6.3 – Nombre et taille de q-grams en commun entre un extrait compressé et les empreintes de la base ayant obtenus les meilleurs scores, le co-dérivé arrivant toujours premier candidat

La courbe du haut (—) de la Figure 6.3 correspond à la ligne ($1^{re}, moy$) de la table 6.1. Le nombre de q-grams d'un co-dérivé décroît légèrement lorsque la taille de q augmente tout en restant non nul. Les seconde et troisième courbes (- - et ····) correspondent respectivement aux nombre de q gram moyen des secondes et troisièmes empreintes. On remarque que le nombre de q-grams est significativement inférieur a celui obtenu par les contenus co-dérivés originels. L'information portée par les q grams est donc pertinente pour notre problématique d'identification. On peut également noter que la longueur de q-grams partagés entre deux empreintes non co dérivées n'excède jamais un certaine taille ($q_{max} = 8$) alors que lorsque

l'on compare l'empreinte d'un extrait avec l'empreinte de son co-dérivé, cette taille est largement dépassée.

6.3.2 Scores par quantité d'information

Nous avons montré dans la section précédente l'intérêt des q-grams et leur potentielle capacité à discriminer une empreinte co-dérivée. Fort de ces résultats, nous avons appliqué notre première méthode d'identification basée sur la quantité d'information apportée par les q-grams (Section 5.2). Pour cela, nous avons utilisé nos mêmes extraits de 5 secondes compressés que nous avons comparé avec chaque empreinte de la base de données. Pour chacune de ces empreintes, nous avons calculé son score suivant l'équation 5.6. Comme nous l'avons déjà expliqué, la taille de notre alphabet est de $s = 2^{14}$. Cependant, pour éviter des problèmes de dépassement de la capacité de représentation des types usuels, la valeur de b dans l'équation 5.6 a été fixée à 5. Cette valeur expérimentale correspond à la valeur la plus élevée qui nous permette d' éviter le problème de dépassement précédemment cité dans tous les cas.

Nous avons ensuite calculé les scores à partir de l'équation 5.6 pour des tailles de q-grams allant de $q_{min} = 4$ à $q_{max} = 20$. La table 6.2 représente les scores calculés entre l'empreinte compressée et les quatre empreintes de la base ayant obtenues les meilleurs scores. Comme dans le cas précédent (Section 6.3.1), l'empreinte co-dérivée à toujours été celle obtenant le meilleur score. Pour chaque candidat, nous représentons sur la première partie de la table 6.2 les scores minimum, moyens et maximums. La seconde partie représente les ratios des scores des quatre premiers candidats.

	Candidats				Ratios		
Scores	1^{er}	2^{e}	3^{e}	4^{e}	$1^{er}/2^{e}$	$2^{e}/3^{e}$	$3^{e}/4^{e}$
Min	**28 750**	3 125	625	625	**199.8**	1	1
Moy	$1\ 832.10^{6}$	3 475	1 877	883	$1\ 113.10^{6}$	1.48	1.13
Max	$4\ 294.10^{6}$	**188 750**	162 500	91 250	$4\ 294.10^{6}$	**23**	6

TAB. 6.2 – Scores des 4 meilleures empreintes et ratio inter-candidats

Comme le montre ce tableau, le score moyen obtenu par un contenu co-dérivé est largement supérieur aux scores obtenus par les autres contenus. Cependant, le score minimum obtenu par le contenu co-dérivé dans un cas peut être inférieur à un score d'un contenu différent obtenu dans un autre cas. Par conséquent, une règle de décision uniquement basée

sur l'utilisation d'un seuil pour départager les candidats est impossible puisqu'on ne peut scinder l'espace des scores.

Toutefois, comme le montre la deuxième partie de la table un seuil de décision peut facilement être positionné en considérant non plus les scores mais les ratios. entre les scores. Ce phénomène est du au fait que lorsque une empreinte possède un contenu co dérivé dans la base, le score entre celle-ci et son co dérivé va dominer nettement les autres scores. Inversement lorsque aucun co dérivé n'est présent dans la base, tous les scores sont sensiblement équivalents.

6.3.3 Scores par distance d'édition

Nous rappelons que pour réaliser ces expérimentations, 5 sec de signal audio ont été extraites aléatoirement de notre base de données, compressées à 128Kbps et leurs empreintes calculées. Notre méthode de reconnaissance (Section 5.3) a été mise en oeuvre pour identifier chaque empreinte. Notons que pour ces tests, chaque empreinte a un contenu co-dérivé dans la base de données correspondant à son original non compressé.

La première étape de notre algorithme consiste à rechercher les positions des q-grams communs aux deux empreintes et à calculer pour chaque q gram un score local de potentielle correspondance (équation 5.11). La somme de ces scores est ensuite calculée afin de réaliser notre étape de filtrage (équation 5.12). L'empreinte avec le score le plus élevé est alors sélectionnée et toutes les autres sont filtrées. La position dans les deux chaînes ayant donné lieu à un score d'édition maximal sur des sous chaînes de longueur m sera utilisée comme point de synchronisation entre les deux chaînes pour calculer une distance sur des sous chaînes de taille plus importante. Les expérimentations ont montré qu'une taille minimale de $q = 5$ symboles identiques successifs offre le meilleur compromis entre discrimination et souplesse. Les valeurs de α,γ et β ont été respectivement positionnées à : $\alpha = 1.5$; $\gamma = 1.1$; $\beta = 20$.

	Classements de filtrage			Ratios	
Scores	1^{ier}	2^{nd}	3^{ieme}	$1^{ier}/2^{nd}$	$2^{nd}/3^{ieme}$
Min	**14878**	0	0	**34.74**	0
Mean	11.10^5	240.72	143.74	8.10^5	4.19
Max	4.10^6	$\mathbf{2.10^4}$	1.10^4	4.10^6	**206.2**

FIG. 6.4 – Score de filtrage par q-grams

Nous avons relevé dans la Table 6.4 les scores et ratios obtenus par les 3 meilleures empreintes de la base de données en fonction de l'extrait inconnu identifié. Les scores minimums, moyens et maximums y sont indiqués. L'empreinte ayant obtenu le meilleur score est, dans chaque cas, celle correspondant au contenu co-dérivé. Par conséquent, le second score obtenu revient à simuler le fait de ne pas avoir de contenu co-dérivé dans la base de données. Il s'agit dont du meilleur faux positif. Comme le montre cette table, le score du co-dérivé est toujours plus élevé que ceux obtenus par les autres empreintes. Cependant, les cases (Min, 1^{er}) et (Max, 2^{e}) mettent en évidence le fait que le meilleur score obtenu par un faux positif peut être dans un cas, encore supérieur à celui obtenu par un co-dérivé dans un autre cas. Ce dernier point ne permet donc pas de vérifier la présence d'un co-dérivé dans la base de données. De plus, les rations $(Min, 1^{\text{er}}/2^{\text{e}})$ et $(Max, 2^{\text{e}}/3^{\text{e}})$ montrent que contrairement aux scores par quantité d'information (Section 6.3.2, Tab 6.2) ces ratios ne permettent pas de caractériser un contenu co-dérivé. Nous nous trouvons face à 2 cas :

– l'empreinte ayant obtenu le meilleur score correspond au contenu co-dérivé.

– l'empreinte ayant obtenu le meilleur score correspond au meilleur faux positif.

Quoi qu'il en soit, la conclusion est qu'à partir de cette étape, seule l'empreinte ayant obtenu le meilleur score nous intéresse. Reste à décider s'il s'agit ou non d'un contenu co-dérivé. Pour cela, nous utilisons notre q-gram repère, c'est à dire la sous chaîne de longueur q comune aux deux chaînes I et D à partir des positions i et j telles que $S(I[i, i+m], D[j, j+m])$ est maximum parmi tous les scores calculés pour évaluer les équations 5.11 et 5.12. Nous allons calculer à partir de ces positions une distance d'édition sur un durée plus importante de 5 secondes (équation 5.13). Afin de démontrer l'efficacité de cette mesure, nous donnons dans le tableau 6.5, les scores des trois meilleures empreintes au sens du score. Encore une fois le contenu co-dérivé est classé premier tandis que les deux autres empreintes correspondent aux meilleurs faux positifs.

	Candidats			Ratios	
Scores	1^{er}	2^{e}	3^{e}	$1^{\text{er}}/2^{\text{e}}$	$2^{\text{e}}/3^{\text{e}}$
Min	**100000**	0	0	1980	0
Mean	3.10^{15}	16	3	8.10^{14}	4
Max	5.10^{17}	**2800**	590	5.10^{16}	300

FIG. 6.5 – Score final obtenu à partir de 5 secondes d'extrait comparé avec la base de données

L'objectif de cette technique est ainsi de creuser l'écart des scores entre co-dérivé et faux

positifs afin de pouvoir définir un simple seuil qui soit discriminant. Cet objectif est atteint comme le montre les scores des cases (Min, 1^{ier}) et (Max, 2^{nd}). Le score obtenu par un co-dérivé est en effet désormais toujours très largement supérieur au meilleur faux positif. Un seuil défini entre le score minimal d'un co-dérivé et le score maximal du meilleur faux positif permet donc d'établir une règle de décision simple et efficace afin de décider de la présence d'un document audio co-dérivé dans la base de données. Par exemple, un seuil de 50.000 suffit pour identifier à 100% un extrait inconnu si un contenu co-dérivé est présent dans la base de données ou bien, le cas échéant, confirmer que cet extrait est bien inconnu du système.

7

Un scénario pour la gestion des droits

7.1 Introduction

L'objectif de ce chapitre est de définir un ensemble de moyens de contrôle régissant l'usage de contenus sur un terminal dit « conforme ». Ces règles comprennent l'acceptation ou le refus de jouer un contenu, le stockage et la copie d'un contenu, ainsi que son échange vers un autre terminal. Ces contenus peuvent alors provenir de différentes sources identifiées qui sont : Internet, un réseau local, un média (CD, DVD, cartes, ...). Ces moyens de contrôle contiennent des techniques de DRM mais peuvent ne pas se limiter à ces techniques.

Le but est d'offrir un système où les techniques actuelles de DRM collaboreraient avec de nouvelles techniques de contrôle basées sur la reconnaissance du contenu. Ces techniques se reposeraient sur la détermination d'un identifiant unique pour chaque document multimédia qui soit insensible aux dégradations dues au format de compression ou à la suppression de parties du contenu. Cette méthode ainsi que son comportement face aux tests réalisés ont été décrits dans les chapitres précédents.

Cette technique d'identification reconnaît un document original à partir d'un court extrait d'un document compressé et permet de contrôler l'utilisation de ce document même si

celui-ci est paru antérieurement à la mise en place de techniques de DRM ou si sa protection par DRM aurait été inefficace ou contournée. Le vocable ADRM a ainsi été choisi pour représenter les deux aspects du contrôle : Analogiques basés sur l'identification du contenu, et techniques de DRM.

Sommaire

7.2 Cadre de confiance

L'ADRM opère dans le contexte défini par le TCG, le Trusted Computing Group, l'informatique de confiance. Ce groupe a été lancé en 2003 à l'initiative conjointe de plusieurs grands acteurs du secteur informatique, et non phonographique, dont IBM, Hewlett-Packard et Intel afin de proposer une méthode de certification universelle pour la sécurisation matérielle et logicielle à travers plusieurs types de plate-forme (terminaux, PC familiaux, téléphones mobiles, assistants personnels...). Concrètement, leurs spécifications définissent la manière d'exploiter un composant de surveillance qui, lors de la phase de démarrage de la machine, se charge de vérifier la conformité des éléments matériels et logiciels qu'elle utilise au regard d'une liste d'outils préalablement définie. Les applications de cette technologie sont nombreuses. Elle peut permettre de contrôler l'utilisation qui est faite d'un logiciel en autorisant l'exécution de ce dernier aux seuls détenteurs de licences. Mais également de vérifier la conformité d'un logiciel afin d'empêcher l'installation de logiciels piratés, de cracks, ou de logiciels en contradiction avec les impératifs de droits d'auteur en traquant les éventuels codes malicieux susceptibles de s'immiscer sur les disques durs.

Cela permet de considérer un cadre matériel et logiciel au sein duquel ne pourraient être utilisés que des logiciels conformes ayant satisfait les conditions de leur homologation et donc de leur installation, comme par exemple l'acceptation d'une brique logicielle pour la vérification des droits d'auteurs avant et/ou pendant la lecture d'un contenu (ADRM). Ce cadre interdit donc tout logiciel pirate, de lecture ou de copie de contenu, qui ne répondrait

pas aux conditions de mise en oeuvre de notre plug-in de gestion de l'utilisation de documents multimédia dans le respect des droits d'auteur.

7.3 Reconnaissance Audio

Comme il l'a été précisé, notre proposition de gestion de documents audio s'appuie sur notre méthode de reconnaissance basée sur le contenu lui-même, le signal audio. Le principe est celui d'une liste blanche de documents audio dont l'utilisateur aurait acquis les droits, cette liste étant stockée sur la machine de l'utilisateur et exportable sur chacun de ses appareils. Si on lit un document audio compressé type MP3 téléchargé, on va vérifier si ce document est dérivé de l'un de ceux figurant dans la liste et dont on a acquis les droits, si non, on arrête la lecture. C'est le fingerprinting. Cette technique est, comme son nom l'indique, comparable au principe de contrôle d'accès d'un bâtiment qui se baserait sur l'empreinte digitale des individus ayant l'autorisation d'y accéder. Il s'agit donc ici bien d'identification et non d'authentification.

Comme nous l'avons vu, cette technique étant la pierre angulaire du système, elle implique certaines exigences. Tout d'abord, le calcul de l'empreinte se doit d'être le plus invariant possible aux altérations du signal telle la compression afin de pouvoir être reconnu efficacement. L'algorithme doit respecter d'autres contraintes pour permettre son intégration sur un ordinateur familial voir un téléphone mobile. La vitesse d'exécution des étapes de calcul de l'empreinte ou de reconnaissance doit être la plus rapide possible pour pouvoir être réalisées en parallèle à la lecture. La taille de l'empreinte doit être la plus réduite possible pour son stockage sur un ordinateur ou encore un mobile. Enfin, l'identification d'un morceau de musique par rapport à son empreinte doit intervenir à partir d'un court échantillon du signal (environ 5sec) pris à n'importe quel moment de la lecture du signal. Enfin, et surtout, si l'identification d'un document donne accès à sa lecture, la non-reconnaissance d'un document dont l'empreinte est dans la base de données est équivalent à un refus de service. Par conséquent, le taux de mauvais refusés se doit d'être extrêmement bas. Or nous avons vu que notre technique, tant au niveau de l'algorithme de calcul de l'empreinte que de la méthode utilisée pour identifier les extraits, satisfait l'ensemble de ces pré-requis et

apporte une réelle amélioration de l'existant sur ces points. En effet, nous sommes maintenant capables de dire si les quelques secondes de musique que notre lecteur multimédia est en train de jouer correspondent à un des documents présents dans notre liste blanche de fichiers acquis légalement ou bien s'il s'agit d'un document inconnu pour la machine et, donc, probablement illégal.

7.4 Description

La figure 7.1 ci-dessous détaille ces modes de contrôle en fonction des différents types de contenus reçus par un terminal conforme protégé par ADRM. Les tests que le terminal doit faire avant d'accepter ce contenu sont de deux types, d'un côté les test DRM classiques et de l'autre les tests prenant en compte la technique d'identifiant de contenu. On appelle CD signé un CD normal ayant un fichier de signature (donc qui reste lisible sur tout équipement de lecture de CD classique). Cette signature est calculée par une autorité sur idf, et hash du contenu originel.

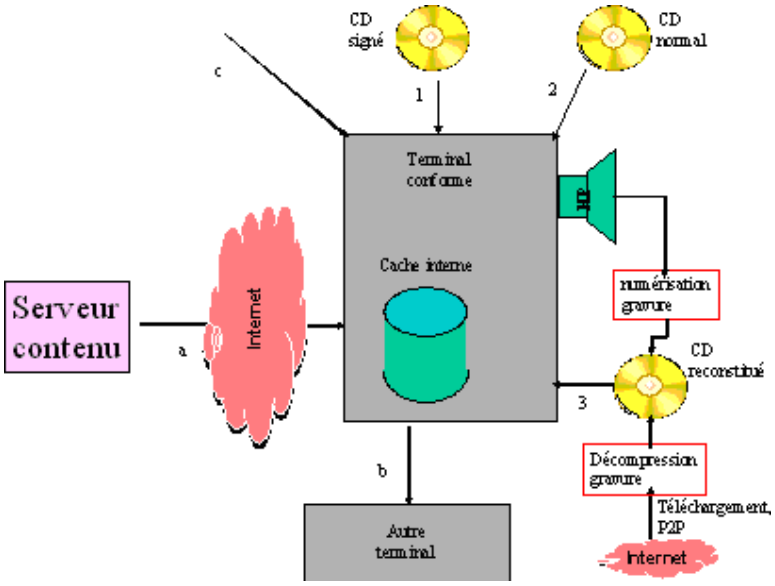


FIG. 7.1 – Contrôles de l'utilisation

Dans le cas a), le terminal fait l'acquisition d'un contenu acheté sur un serveur de distribution de musique en ligne par le biais d'internet. Le contenu acheté et dont on a acquis les

droits est protégé par des techniques de type DRM afin de contrôler les échanges et copies de ce contenu. Lors de cette acquisition, la signature du document est aussi fournie afin que l'identifiant calculé sur le contenu du document puisse être ajouté au cache interne du terminal.

Le cas b) présente l'échange à partir d'un appareil conforme vers un autre terminal (par internet, réseau local, flash, ...). Cet échange est régis par des technique de DRM classiques concernant les fonctions de super distribution

Le cas c) montre l'import d'un contenu vers le terminal. Ce type d'import peut provenir de différentes sources et doit être géré de manière adéquate suivant la source.

Le contenu est protégé par une DRM classique (pouvant donc provenir d'un autre terminal du réseau local), on revient alors au cas a) Le contenu n'est pas protégé, on va chercher à identifier la source et suivant le cas, se référer aux cas 1, 2 ou 3.

Cas 1 : il existe sur le média un fichier signature. Il s'agit de ce que l'on appelle un CD signé. La signature contient les empreintes des fichiers audio contenu sur le CD. On accepte alors la lecture et si c'est la première lecture de ce média, les identifiants de chaque document contenu sur le média seront lus et enregistrés dans a liste blanche.

Cas 2 : Le cd ne contient pas de fichier signature mais est reconnu comme étant un CD original produit par une maison de disque homologuée et acheté légalement sur le marché grâce à la lecture de son International Standard Recording Code. Ce CD étant reconnu comme original, son contenu peut être lu et s'il s'agit de sa première lecture, les identifiants de forme de chaque document seront calculés et ajoutés dans la liste blanche.

Cas 3 : L'ISRC n'est pas présent, il s'agit donc d'un CD gravé à partir de MP3. Lors de la lecture d'un document contenu sur ce CD, son identifiant sera alors calculé et comparé avec ceux contenus dans la liste blanche du terminal. Si l'identifiant est reconnu comme correspondant à un de ceux de la liste blanche (signifiant qu'on a les droits dessus), la lecture continu. Si on ne reconnaît pas l'identifiant parmi ceux du cache, c'est qu'aucun original de ce document n'a été lu par le terminal, donc aucune preuve de possession de l'original n'a été apportée et, par conséquent, la lecture s'arrête.

La lecture d'une copie bit à bit d'un original non régis par des DRM sera alors vérifiée par le contrôle de la Burst Cutting Area qui est une section proche du centre du CD ou

DVD où des informations ne peuvent être écrites que par un laser de haute puissance. Par conséquent, un graveur classique ne pourra écrire dans cette zone du CD ce qui permettra de savoir qu'il s'agit d'une copie bit à bit.

7.5 Prototypage

Afin de montrer la validité de ces concepts nous avons construit un démonstrateur. Celui-ci à été réalisé en collaboration avec Yves Feuillet de France Telecom qui a conçu le squelette du player audio.Pour ce démonstrateur, on a voulu un design de player qui soit proche d'un player classique (Figure 7.2).

FIG. 7.2 – Prototype de lecteur audio développé

Pour simuler le scénario ADRM, on recherche dans l'arborescence un document audio numérique. Si on coche la case « contenu licencié », on simule le fait que le contenu choisi est un contenu légalement acquis. Lorsqu'on lance la lecture de ce document, en quelques secondes à peine, la signature est calculée et sauvegardée. Si on lit ensuite une version compressée du même contenu, la lecture se déroulera sans perturbation pour l'utilisateur. Par contre, si on tente de lire un autre contenu que celui ou ceux dont la signature a été sauvegardée, la lecture sera stoppée.

Finalement, ce démonstrateur simule l'utilisation de notre technique d'identification dans le cadre de la gestion des droits numériques avec pour résultat la gestion et le contrôle de l'utilisation des documents audio.

8

Conclusion et perspectives

Sommaire

8.1 Contributions

Le but de cette thèse était de définir une méthode d'identification de contenus audio qui soit à la fois légère (en termes de temps de calcul et de capacité disque) et qui présente un faible taux de faux négatifs. Nous avons établi un état de l'art des diverses méthodes de calcul et de reconnaissance d'empreintes. Cet état de l'art nous a conduit à proposer plusieurs méthodes de calcul d'empreintes et deux méthodes d'identification de celles-ci.

8.1.1 Calcul d'empreinte audio

La première contribution de cette thèse concerne la définition des empreintes. Celles-ci sont basées sur l'extraction régulières de caractéristiques du signal audio. La concaténation de ces mesures définit une empreinte qui caractérise le signal. Nous avons étudié plusieurs pistes et plus particulièrement une étude spatio-fréquentielle utilisant des mesures déduites de la transformée de Fourrier sur de courtes fenêtre prises le long du signal (Section 4.2). Cependant, nous avons observé qu'il était plus rapide d'utiliser uniquement une description temporelle et que cette seule propriété apportait finalement de meilleurs résultats qu'une utilisation combinée de propriétés fréquentielles et temporelles. Nous avons donc défini une méthode de détection d'intervalles de haute énergie dans le signal et construit notre empreinte comme la concaténation des écarts entre ces pics de haute énergie (Section 4.3). Cette segmentation contrainte du signal induit des empreintes présentant une très forte

résistance à la compression et aux décalages temporels. De plus la taille des empreintes produites est inférieure d'un facteur au moins 10 aux empreintes basées sur des fenêtres glissantes. Ce type d'empreinte n'est pas robuste aux altérations qui accélèrent ou ralentissent un signal. Toutefois, très peu d'internautes téléchargeant des fichiers illégaux appliquent de telles dégradations qui modifient de façon importante le signal et donc la qualité d'écoute du document audio téléchargé.

8.1.2 Identification audio

Le seconde étape consiste à comparer une empreinte avec une base de données. Nous avons proposé deux méthodes. La première (Section 5.2) est basée sur un score qui approxime la quantité d'informations induite par la présence de q-grams de longueurs différentes entre deux empreintes. Cette méthode est très efficace pour indexer des documents puisque dans toutes nos expériences le score entre un document original et un document dégradé est toujours supérieur au score établi entre ce même document original et un document différent (non co-dérivé). Cette propriété garantit que le document original, s'il est présent dans la base sera celui obtenant le meilleur score. Toutefois, ce score n'est pas suffisant pour identifier un document puisque le score entre deux documents co-dérivés peut être inférieur au score entre deux documents qui ne le sont pas. On ne peut donc pas fixer un seuil sur le score pour décider si le document de la base classé premier correspond à un contenu co-dérivé. On peut toutefois identifier tout de même des document en considérant non pas les scores mais le rapport des scores entre le premier et le second document de la base classés par ordre croissant.

Cette première expérience nous a conduit à conclure qu'un simple comptage des q-grams bien qu' extrêmement utile, n'était pas suffisant pour identifier des contenu co-dérivés. Nous avons donc élaboré une nouvelle méthode (Section 5.3) qui combine l'indexation par q-grams à une nouvelle distance d'édition entre chaînes. Cette distance permet de favoriser de longues séquences de symboles communs caractéristiques de contenus co-dérivés. Les expériences que nous avons menées nous ont montré que cette méthode classe toujours premier un document co-dérivé et que l'écart entre deux contenu co-dérivés et deux contenus qui ne le sont pas est suffisamment important pour que l'on puisse facilement positionner un seuil permettant

d'identifier la présence d'un contenu co-dérivé dans la base.

8.2 Perspectives

Ce travail nous permet d'envisager diverses possibilités d'évolutions. Ces évolutions concernent d'une part les améliorations que l'on peut apporter à notre méthode d'identification et d'autre part les nouvelles applications industrielles qui pourraient être imaginées en l'état ou en complémentarité d'autres techniques.

8.2.1 Améliorations

La principale amélioration qui pourrait être apportée à ce travail de recherche serait une technique permettant d'accélérer le processus de recherche dans la base de données. Ce travail avait été initié lors d'un projet d'école d'ingénieur mais n'a pas assez abouti. Il s'agirait de combiner notre technique de filtrage par q-grams avec une indexation de la base de données en fonction de ces mêmes q-grams. Pour une taille de q-gram fixée (disons 5), une table des valeurs possible de ces q-grams serait construite afin d'accéder plus rapidement aux couples document-position recherchés. De nouvelles fonctions de scores pourraient aussi être étudiées afin, d'améliorer encore les résultats de l'identification par appariement de chaînes. En fonction des applications envisagées, on pourrait également renforcer la robustesse de notre méthode de segmentation vis a vis de dégradations spécifiques. Enfin, les résultats obtenus étant réellement très encourageants une utilisation sur une base de données contenant des millions de documents ainsi qu'une mise en oeuvre au sein d'un dispositif portable nous permettrait de valider nos contributions sur une plus grande échelle.

8.2.2 Utilisations

De multiples utilisations peuvent être envisagées. La surveillance de réseaux et l'archivage ont déjà été cités en introduction de ce mémoire. Cependant, d'autres applications pourraient voir le jour. En effet, une vidéo contenant une bande son, notre technique pourrait être utilisée pour identifier une vidéo à partir de sa bande sons. Elle pourrait aussi collaborer avec une technique d'extraction d'empreintes vidéo basée image afin de définir une empreinte

image-audio qui décrive parfaitement un document vidéo. Nous pourrions également étudier les applications de nos empreintes à la classification de sons (hors de la problématique de l'identification). Une application de ces techniques pourrait par exemple consister à insérer une base de données associant sons-objets dans un téléphone portable afin d'apporter une aide aux personnes sourdes ou malentendantes. Ainsi, les sons de la vie quotidienne, appareils ménagers ou bruits extérieurs, pourraient faire vibrer le téléphone portable et ainsi la personne pourrait lire à l'écran que tel objet a émis tel bruit. Les applications possibles d'une telle adaptation sont tellement variées qu'il est difficile de les énumérer. Cependant, si les empreintes et les fonctions de scores que nous avons proposées sont suffisament flexibles pour être adaptées au cadre de la classification , cette thèse pourait servir de socle commun à de nombreuses applications.

8.3 Publications

Communication internationale avec actes et comité de lecture :

Jérôme Lebossé, Luc Brun and Jean Claude Pailles, *"A Robust Audio Fingerprint Extraction Algorithm"*, Proceedings of SPPRA'2006, Pages 185-192, Editors Robert Sablatnig and O. Scherze, Acta Press, Innsbruck (Austria), Février, 2006.

Jérôme Lebossé, Luc Brun and Jean Claude Pailles, *"A Robust Audio Fingerprint's Based Identification Method"*, Proceedings of IbPRIA'2007, Pages 185-192, Editors Joan Marti, Jose Miguel Benedi, Ana Maria Mendonca and Joan Serrat, LNCS, Volume I, number 4477, Girona (Spain), Juin, 2007.

Jérôme Lebossé et Luc Brun, *"Audio Fingerprint Identification by Approximate String Matching"*, Proceedings of ISMIR 2007, Vienna (Austria), Septembre, 2007.

Communication nationale avec actes et comité de lecture :

Jérôme Lebossé, Luc Brun et Jean Claude Pailles, *"Fingerprint audio robuste pour la gestion de droits"*, Actes de CORESA 2006, Caen, Novembre, 2006.

Jérôme Lebossé, Luc Brun et Jean Claude Pailles, *"Identification de signaux audio par appariement de chaînes"*, Proceedings of GRETSI 2007, Troyes, Septembre, 2007.

Brevet :

J.Lebossé et J-C. Pailles, *"Détermination d'identification de signal"*, France Télécom R&D Caen, Février, 2006.

Présentations :

Présentations internes à France Télécom concernant la valorisation du projet de laboratoire commun LATEMS en exposant l'avancement des travaux de recherches et résultats obtenus.

Présentation à la journée thématique du GdR ISIS sur la protection des données multimédia. *"Contrôle de l'utilisation de documents audio dans le respect des droits d'auteurs grâce à l'identification de contenus par empreinte"*, Novembre, 2007.

Bibliographie

[1] E. Allamanche, J. Herre, O. Helmuth, B. Fröba, T. Kasten, and M. Cremer. Content-based identification of audio material using mpeg-7 low level description. *Proc. Of the Int. Symp. Of Music Information Retrieval*, 2002.

[2] M. Alonso, B. David, and G. Richard. Tempo and beat estimation of musical signals, 2004.

[3] M. Alonso, G. Richard, and B. David. Extracting note onset from musical recordings, 2005.

[4] J. Bello, C. Duxbury, M. Davies, and M. Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Procesing letters*, 11(6) :553–556, June 2004.

[5] W. Birmingham, R. Dannenberg, G. Wakefield, M. Bartsch, D. Bykowski, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart : Music retrieval via aural queries, 2001.

[6] J. Bruck, S. Bres, and D. Pellerin. Construction d'une signature audio pour l'indexation de documents audio visuels. In *Actes de Coresa 2004*, 2004.

[7] C. Burges, D. Plastina, J. Platt, E. Renshaw, and H. Malvar. Using audio fingerprinting for duplicate detection and thumbnail generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005.

[8] C. Burges, J. Platt, and S. Jana. Distorsion discriminant analysis for audio fingerprn-ting. *IEEE Transactions on Speech and Audio Processing*, 11(3) :165–174, 2003.

[9] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied. Robust sound modeling for song detection in broadcast audio, 2002.

[10] W. Chang and T. Marr. Approximate string matching and local similarity. In M. Crochemore and D. Gusfield, editors, *Combinatorial Pattern Matching, Proceedings of the 5th Annual Symposium*, volume 807 of *Lecture Notes in Computer Science*, pages 259–273. Springer Verlag, 1994.

[11] C. Charras and T. Lecroq. *Handbook of Exact string matching algorithms*. King's College London Publications, 2004.

[12] S. Consortium. Secure digital music initiative, 2001. http ://www.sdmi.org.

[13] M. Covell and S. Baluja. Known audio detection using waveprint : spectrogram fingerprint by wavelet hashing. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP-2007)*, 2007.

[14] M. Crochemore and W. Rytter. *Text algorithms*. Oxford University Press, 1995.

[15] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *Signal Processing, IEEE Transactions on*, 53(8) :2961–2974, Aug. 2005.

[16] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1) :7–23, 2000.

[17] S. Dixon. Automatic extraction of tempo and beat from expressive performances, 2001.

[18] S. Dixon. Classification of dance music by periodicity pattern, 2003.

[19] P. Doets, M. Gisbert, and R. Lagendijk. On the comparison of audio fingerprints for extracting quality parameters of compressed audio. In *In Proceedings of SPIE*, volume 6072, February 2006.

[20] P. Doets and R. Lagendijk. Theoretical modeling of a robust audio fingerprinting system. In *IEEE Benelux Signal Processing Symposium*, 2004.

[21] D. Duxbury, M. Sandler, and M. Davies. A hybrid approach to musical note detection. In *Proc. of Digital Audio Effects Workshop (DAFx)*, pages 33–38, Hamburg, Germany, 2002.

[22] P. Flajolet. Random tree models in the analysis of algorithms. In *Performance*, pages 171–187, 1987.

[23] B. Gajic and K. Paliwal. Robust feature extraction using subband spectral centroid-histograms, 2001.

[24] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *J. of New Music Research, vol30(2)*, pages 159–171, 2001.

[25] F. Gouyon, A. Klapuri, S. Dixon, G. Tzanetakis, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Speech and Audio Processing*, 14(5), 2006.

[26] F. Gouyon, F. Pachet, and O. Delerue. the use of zerocrossing rate for an application of classification of percussive sounds, 2000.

[27] L. Gravano, H. J. P.G. Ipeirotis, and D. Srivastava. Approximate string joins in a database almost for free. In M. K. P. Inc., editor, *In Proceedings of the International Conference on Very Large Databases*, pages 491–500, 2001.

[28] S. Hainsworth and M. Macleod. Onset detection in musical audio signals. In *Proceedings of the International Computer Music Conference*, Singapore, September 2003.

[29] P. Herrera, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds, 2003.

[30] P. Herrera, X. Serra, and G. Peeters. Audio descriptors and descriptor schemes in the context of mpeg-7, 1999.

[31] T. Hoad. *Video Representations for Effective Rettrieval From Large Collections*. PhD thesis, RMIT University, Melbourne, Australia, 2004.

[32] T. Hoad and J. Zobel. Video similarity detection for digital rights management. In *ACSC*, pages 237–245, 2003.

[33] O. Izmirli. Using a spectral flatness based feature for audio segmentation and retrieval, 2000.

[34] P. Jokinen and E. Ukkonen. Two algorithms for approximate string matching in static texts. In *Lecture Notes in Computer Science*, pages 240–248, 1991.

[35] B. Juang. Speech, acoustics and audio processing for multimedia, 1997.

[36] T. Kalker and J. Haitsma. A highly robust audio fingerprinting system. In *Proceedings of ISMIR 2002*, pages 144–148, 2002.

[37] T. Kalker and J. Haitsma. A highly robust audio fingerprinting system. In *Proceedings of ISMIR'2002*, pages 144–148, 2002.

[38] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge.

[39] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals, 2005.

[40] F. Kurth. A ranking technique for fast audio identification. In *Proceedings of the International Workshop on Multimedia Signal Processing*, 2002.

[41] Y. Li and Y. Hou. Search audio data with the wavelet pyramidal algorithm. In *Inf. Proc. Letters, vol 91(1)*, pages 49–55, 2004.

[42] D. Liu, L. Lu, and H. J. Zhang. Automatic mood detection from acoustic music data, 1998.

[43] B. Logan. Mel frequency cepstral coefficients for music modelling. In *Proc. of the Int. Symposium on Music Information Retriaval*, 2001.

[44] M. F. Mckinney. Features for audio and music classification. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 151–158, 2003.

[45] M. Mihçak and R. Venkatesan. A perceptual audio hashing algorithm : a tool for robust audio identification and information hiding. In *4th Workshop on Information Hiding*, 2001.

[46] P. Nicodeme. Q-grams analysis and urn models. In *Proceedings of Discrete Random Walks DRW*, pages 243–258, 2003.

[47] F. Pachet. Kknowledge management and musical metadata, 2005.

[48] F. Pachet. Knowledge management and musical metadata. In *Encyclopedia of Knowledge Management*, 2005.

[49] G. Peeters. Toward automatic music audio summary generation from signal analysis, 2002.

[50] RIAA. Request for audio fingerprint technologies, 2001.

[51] E. Scheirer. Tempo and beat analysis of acoustic musical signals, 1998.

[52] C. E. Shannon. A mathematical theroy of communication, 1948.

[53] S. R. Subramanya, R. Simha, B. Narahari, and A. Youssef. Transform-based indexing of audio data for multimedia databases. In *International Conference on Multimedia Computing System*, pages 3–6, 1997.

[54] W. Szpankowski. Asymptotic properties of data compression and suffix trees. *IEEE-TIT : IEEE Transactions on Information Theory*, 39, 1993.

[55] G. Tzanetakis, G. Essl, and P. Cook. Audio analysis using the discrete wavelet transform. In *"In Proceedings of of WSES International Conference, Acoustics and Music : Theory and Applications (AMTA)"*, 2001.

[56] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *"Proc. International Symposium for Audio Information Retrieval (ISMIR)"*, 2002.

[57] C. Uhle and J. Herre. Estimation of tempo, micro time and time signature from percussive music, 2003.

[58] A. Wang. An industrial-strength audio search algorithm. *Proc. of the International Symposium on Music Information Retrieval*, 2003.

[59] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE MultiMedia*, 3 :27–36, 1996.

[60] L. Ying. Search audio data with wavelet packet best base and pyramidal algorithm. *Congress on Image and Signal Processing*, 2008.

[61] A. Zils and F. Pachet. Extracting automatically the perceived intensity of music titles, 2003.

9

Annexe

9.1 Calcul de la moyenne γ_{bc} et de la variance σ^2_{bc}

Lemme 2. *Le coefficient en* $[z^r t^s]$ *de la série entière* e^{az+bt} *est égal à :*

$$\frac{a^r b^s}{r!s!}$$

Démonstration :

$$\begin{aligned} e^{az+bt} &= \sum_{n=0}^{+\infty} \frac{1}{n!}(az+bt)^n \\ &= \sum_{n=0}^{+\infty} \frac{1}{n!} \sum_{k=0}^{n} C_n^k (az)^k (bt)^{n-k} \end{aligned}$$

Les relations $k = r$ et$n - r = s$ imposent $n = r + s$. On obient alors :

$$\begin{aligned} [z^r t^s] e^{az+bt} &= \frac{1}{(r+s)!} C_{r+s}^r a^r b^s \\ &= \frac{a^r b^s}{r!s!} \end{aligned}$$

□

L'article nous indique que $F(z,t,u)$ est donné par :

$$F(z,t,u) = \Pi_{i=0}^{m-1} \left(e^{p_i(z+t)} + (u-1)(e^{p_i z} + e^{p_i t} - 1) \right)$$

Si nous supposons tous les q grams équiprobable on obtient :

$$p_i = p = \frac{1}{m}, \quad \forall i \in \{0 \ldots, m-1\}$$

et $F(z,t,u)$ s'écrit :

$$F(z,t,u) = \left(e^{p(z+t)} + (u-1)(e^{pz} + e^{pt} - 1)\right)^m$$

On a donc :

$$\frac{\partial F}{\partial u} = m\left(e^{p(z+t)} + (u-1)(e^{pz} + e^{pt} - 1)\right)^{m-1}(e^{pz} + e^{pt} - 1)$$

d'où :

$$\begin{aligned} \frac{\partial F}{\partial u}|_{u=1} &= me^{p(m-1)(z+t)}(e^{pz} + e^{pt} - 1) \\ &= me^{pt(m-1)+z} + me^{pz(m-1)+t} - me^{p(m-1)(z+t)} \end{aligned}$$

où l'on a utilisé la relation $pm = 1$.

En utilisant le fait que F est issue d'une double Poissonisation, nous obtenons la définition suivante de la moyenne μ_{bc} :

$$\mu_{bc} = [z^b t^c] b! c! \frac{\partial F}{\partial u}|_{u=1}$$

En utilisant le lemme précédent, on obtient donc :

$$\begin{aligned} \mu_{bc} &= m\left(p^c(m-1)^c + p^b(m-1)^b - (p(m-1))^{b+c}\right) \\ &= m\left((1-p)^b + (1-p)^c - (1-p)^{b+c}\right) \end{aligned}$$

On obtient donc :

$$\begin{aligned} \gamma_{bc} &= m - \mu_{bc} \\ &= m\left(1 - (1-p)^b - (1-p)^c + (1-p)^{b+c}\right) \\ &= \tfrac{1}{p}\left(1 - (1-p)^b - (1-p)^c + (1-p)^{b+c}\right) \end{aligned}$$

Un développement limité de γ_{bc} à l'ordre 2 nous donne :

$$\gamma_{bc} \approx bcp - \frac{1}{2}bc(c+b-2)p^2 + \mathcal{O}(p^3)$$

Calculons à présent σ_{bc}. On a :

$$\sigma^2_{bc} = m^{(2)}_{bc} - \mu^2_{bc} \text{ avec } m^{(2)}_{bc} = b!c![z^b t^c]m^{(2)}(z,t)$$

où :

$$m^{(2)}(z,t) = \frac{\partial}{\partial u} u \frac{\partial F(z,t,u)}{\partial u}|_{u=1} = \frac{\partial F(z,t,u)}{\partial u}|_{u=1} + \frac{\partial^2 F(z,t,u)}{\partial u^2}|_{u=1}$$

La dérivé seconde de F est donnée par :

$$\frac{\partial^2 F(z,t,u)}{\partial u^2} = m(m-1)\left(e^{pt}+e^{pz}-1\right)^2\left[e^{p(z+t)}+(u-1)(e^{pt}+e^{pz}-1)\right]^{m-2}$$

Donc :

$$\begin{aligned}
\frac{1}{m(m-1)}\frac{\partial^2 F(z,t,u)}{\partial u^2}|_{u=1} &= \left(e^{pt}+e^{pz}-1\right)^2 e^{p(m-2)(z+t)} \\
&= \left(1+(e^{pt}+e^{pz})^2-2(e^{pt}+e^{pz})\right)e^{p(m-2)(z+t)} \\
&= (1+e^{2pz}+e^{2pt}+2e^{p(z+t)}-2e^{pt}-2e^{pz})e^{p(m-2)(z+t)} \\
&= e^{p(m-2)(z+t)}+e^{p(m-2)z+t}+e^{p(m-2)t+z}+2e^{p(m-1)(z+t)} \\
&\quad -2e^{p(m-2)z+p(m-1)t}-2e^{p(m-2)t+p(m-1)z}
\end{aligned}$$

Donc la série $m^{(2)}(z,t)$ est donnée par la fonction :

$$\begin{aligned}
\frac{\partial}{\partial u}u\frac{\partial F(z,t,u)}{\partial u}|_{u=1} &= me^{pt(m-1)+z}+me^{pz(m-1)+t}-me^{p(m-1)(z+t)} \\
&\quad +m(m-1)e^{p(m-2)(z+t)}+m(m-1)e^{p(m-2)z+t} \\
&\quad +m(m-1)e^{p(m-2)t+z}+2m(m-1)e^{p(m-1)(z+t)} \\
&\quad -2m(m-1)e^{p(m-2)z+p(m-1)t}-2m(m-1)e^{p(m-2)t+p(m-1)z}
\end{aligned}$$

En appliquant le Lemme 1 on obtient :

$$\begin{aligned} m_{bc}^{(2)} &= mp^c(m-1)^c + mp^b(m-1)^b - mp^{b+c}(m-1)^{b+c} + m(m-1)p^{b+c}(m-2)^{b+c} \\ &\quad +m(m-1)p^b(m-2)^b + m(m-1)p^c(m-2)^c + 2m(m-1)p^{b+c}(m-1)^{b+c} \\ &\quad -2m(m-1)p^{b+c}(m-1)^b(m-2)^c - 2m(m-1)p^{b+c}(m-1)^c(m-2)^b \end{aligned}$$

$$\begin{aligned} pm_{bc}^{(2)} &= (1-p)^c + (1-p)^b - (1-p)^{b+c} + (m-1)(1-2p)^{b+c} \\ &\quad +(m-1)(1-2p)^b + (m-1)(1-2p)^c + 2(m-1)(1-p)^{b+c} \\ &\quad -2(m-1)(1-p)^b(1-2p)^c - 2(m-1)(1-p)^c(1-2p)^b \end{aligned}$$

$$\begin{aligned} \tfrac{p^2}{1-p} m_{bc}^{(2)} &= p(1-p)^{c-1} + p(1-p)^{b-1} - p(1-p)^{b+c-1} + (1-2p)^{b+c} \\ &\quad +(1-2p)^b + (1-2p)^c + 2(1-p)^{b+c} \\ &\quad -2(1-p)^b(1-2p)^c - 2(1-p)^c(1-2p)^b \end{aligned}$$

De plus :

$$\mu_{bc}^2 = \frac{1}{p^2}\left((1-p)^c + (1-p)^b - (1-p)^{b+c}\right)^2$$

On a donc :

$$\begin{aligned} p^2\sigma_{bc}^2 &= p^2 m_{bc}^{(2)} - p^2\mu_{bc}^2 \\ &= (1-p)\left(\begin{array}{l} p(1-p)^{c-1} + p(1-p)^{b-1} - p(1-p)^{b+c-1} + (1-2p)^{b+c} + (1-2p)^b + (1-2p)^c \\ +2(1-p)^{b+c} - 2(1-p)^b(1-2p)^c - 2(1-p)^c(1-2p)^b \end{array}\right) \\ &\quad - \left((1-p)^c + (1-p)^b - (1-p)^{b+c}\right)^2 \end{aligned}$$

Un développement limité à l'ordre 4 de l'expression de droite nous permet de conclure que :

$$\sigma_{bc}^2 \approx bcp - \frac{1}{2}(3cb^2 + (3c^2 - 4c)b)p^2 + \mathcal{O}(p^3)$$

Index