Fingerprint audio robuste pour la gestion de droits

Jérôme Lebossé
France Télécom R&D,
32 rue des coutures,
14000 Caen, France
jerome.lebosse@orange-ft.com

Luc Brun
GREYC UMR 6072,
ENSICAEN, 6 boulevard du Maréchal Juin,
14050 Caen, France
luc.brun@greyc.ensicaen.fr

Jean Claude Paillès
France Télécom R&D,
32 rue des coutures,
14000 Caen, France
jeanclaude.pailles@orange-ft.com

Résumé

Le fingerprint audio permet d'identifier un document audio éventuellement corrompu, à partir d'un court exemple. Ces méthodes peuvent être utilisées dans le cadre de la gestion des droits numériques (DRM) dans le but d'associer les informations de gestion et de contrôle à chaque document. Dans cet article, nous proposons un nouveau mode de calcul de fingerprint audio qui combine une méthode de segmentation avec un nouveau schéma de construction des codes définissants le fingerprint. La méthode proposée est robuste aux altérations du document audio telles la compression et la suppression de parties ou décalages temporels.

Mots clefs

Audio fingerprint, segmentation, Digital Rights Management, identification.

1 Introduction

Les méthodes de gestion des droits numériques (DRM) empêchent les copies illégales de contenus multimédias et leur distribution par Internet. Cependant, les contenus déjà transmis et copiés avant l'avènement de la DRM sont à jamais perdus pour leurs créateurs. De plus, la conversion numérique-analogique-numérique permet de contourner et de s'affranchir des protections par DRM. Les contenus peuvent alors être transmis sur un réseau non protégé. Des solutions à base de watermarking (ou tatouage) ont alors été proposées [1]. Une marque digitale (watermark) est un message imperceptible ajouté au contenu audio sans altération de sa qualité. Cependant, si l'ajout d'une marque n'altère pas la qualité perceptuelle du document, sa suppression peut généralement s'effectuer également sans altérer la qualité du signal. De plus, à notre connaissance, toutes les méthodes de sécurité basées sur des techniques de watermarking reposent sur la non divulgation de la méthode utilisée pour apposer la marque dans le document. La divulgation ou la découverte de ces méthodes compromet donc la sécurisation des documents basés sur ces techniques de watermarking.

L'identification audio à base de fingerprint représente une approche alternative pour traiter des problèmes de protection de copyrights. Les systèmes de fingerprint multimédia permettent de déterminer la similarité perceptuelle entre deux contenus en utilisant une représentation condensée du signal (le fingerprint). Dans la plupart des applications de fingerprint, un grand nombre de données multimédia sont stockées dans une base de données et associées à leurs métadonnées respectives telles que le nom de l'auteur, le titre, … Le fingerprint peut alors être vu comme un index permettant d'effectuer des requêtes sur le contenu perceptuel des données. Dans le cadre des DRM, les métadonnées associées à un document peuvent inclure des informations sur les opérations autorisées sur celui-ci (e.x. nombre de copies).

L'application de l'audio fingerprint pour la gestion des droits numériques implique certaines exigences. Tout d'abord, le fingerprint se doit d'être le plus invariant possible aux altérations du contenu audio comme la compression ou les décalages temporels. Ensuite, l'algorithme doit respecter des contraintes pour permettre son intégration au sein d'un appareil portable ou d'un ordinateur familial. Plus précisément, la taille de chaque fingerprint doit être la plus concise possible afin de répondre aux exigences de stockage dans la base de données tout en contenant suffisamment d'information discriminante pour caractériser et identifier individuellement chaque document. De plus, le calcul du fingerprint doit pouvoir être réalisé en parallèle à la lecture du document audio. Finalement, l'algorithme de fingerprint doit être capable d'identifier un document à partir d'un échantillon de seulement quelques secondes de signal.

Notons de plus que si l'identification d'un document donne accès à sa lecture, la non reconnaissance d'un document dont le fingerprint est stocké dans la base de données est équivalent à un refus de service. Par conséquent, le taux de faux négatifs du système d'identification se doit d'être très bas.

Dans ce papier, nous décrivons une méthode d'extraction de fingerprint robuste qui réponde aux exigences précédemment citées. Après une description des approches alternatives (Section 2), nous décrivons notre méthode de calcul d'identifiant audio dans la Section 3. Sa capacité à satisfaire les contraintes d'une application de gestion des droits numériques est enfin évaluée en Section 4.

2 Etat de l'art

Comme précisé dans la Section 1, une méthode de fingerprint audio doit être capable d'identifier un court échantillon de quelques secondes de signal audio. Un échantillon sur lequel sera appliquée le processus d'identification est appelé *segment*. Typiquement, la taille d'un segment peut varier entre 5 à 10 secondes. De ce fait, un nombre suffisant de caractéristiques discriminantes doit être extrait du segment pendant un intervalle très court. La première phase d'une méthode d'extraction de fingerprint consiste à diviser le signal en intervalles de temps (appelé *frame*) de quelques millisecondes. Une valeur (appelée *sous fingerprint*) est alors associée à chaque frame en codant les caractéristiques acoustiques du signal sur le frame.

La décomposition du signal en frames (appelé *enframing*) doit être robuste aux suppressions de parties du signal et aux décalages temporels qu'elles induisent. Une méthode habituelle pour répondre à cette robustesse consiste à utiliser une fenêtre recouvrante (e.g. [6] utilise des frames de 0,37s avec un taux de recouvrement de 31/32). Cependant, l'utilisation de fenêtres recouvrantes revient seulement à réduire l'influence d'altérations temporelles que peut subir le signal (Section 4). Par exemple, la dégradation d'un signal par un décalage de 12ms d'une suite de frames de 50ms se recouvrant de 50% décalera toutes les fenêtres de 12ms.

Une solution alternative à l'enframing consiste à trouver des positions particulières dans le signal (appelées onsets). Les onsets ([2]) sont définis par un fort gradient calculé sur des caractéristiques perceptuelles du signal traduisant l'apparition d'un changement brusque du signal. Généralement, les techniques de détection d'onsets se basent sur des mesures d'énergies impliquant souvent une pondération fréquentielle. Cette méthode a récemment été améliorée en incorporant une prise en compte de sous bandes fréquentielles ([4]).

Dans [3], les auteurs proposent un schéma de détection d'onsets dans un document musical basé sur les informations apportées par l'énergie fréquentielle du signal combinée à sa phase. En effet, l'utilisation de l'énergie du signal a déjà prouvé son efficacité à détecter d'importants changements du signal, plus particulièrement dans des signaux avec des changements de notes à fortes consonance percussive comme la batterie, puisque l'énergie dénote alors un fort gradient. L'information de phase quant à elle permet de détecter les onsets dans des signaux aux sources mixtes et aux transitions moins franches.

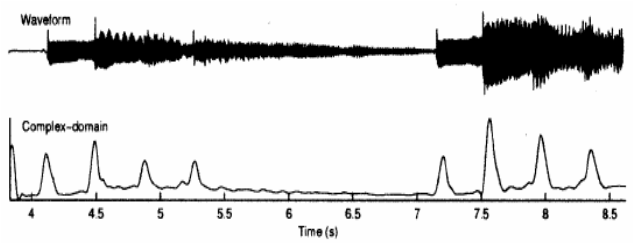

Figure 1 – *Détection d'onsets.*

Comme le montre la figure 1, cette méthode fournit une courbe temporelle décrivant des pics à l'emplacement des onsets et aplanie le reste du temps. L'utilisation d'un filtre médian appliqué à cette courbe permet d'extraire les positions des onsets.

Cependant, l'inconvénient principal de cette approche est dû au fait que le nombre d'onsets détectés dans un laps de temps est imprévisible et est, dans bien des cas, trop faible pour caractériser efficacement un segment. Donc, même si l'approche par onsets permet de synchroniser deux signaux audio basés sur le même contenu, cette approche ne peut pas être utilisée pour le fingerprint audio.

L'approche par frames est alors généralement utilisée pour décomposer le signal en courts intervalles. Une fois le signal divisé, des algorithmes d'extraction de caractéristiques sont alors appliqués pour chaque intervalle. La suite de caractéristiques calculées tout au long du signal définie le fingerprint. Certaines approches combinent la pondération des frames par une fenêtre de Hamming avec l'utilisation de caractéristiques extraites à partir du spectre fréquentiel du signal, comme les Mel Frequency Cepstral Coefficients ([5], [6]). Dans [7], l'auteur associe à chaque frame un bit égal à 1 si l'énergie totale d'un frame est supérieure à celle du frame précédent. Le bit est mis à 0 sinon. D'après l'auteur, cette méthode peut servir à accélérer le processus de recherche en éliminant les mauvais candidats. Mais l'utilisation d'un seul bit par frame ne fournit pas assez d'informations discriminantes pour identifier un segment de façon robuste.

Dans [8], une méthode appelée Distorsion Discriminant Analysis est utilisée pour transformer le signal audio en un vecteur de caractéristiques de plus faible dimension. Tout d'abord, une Modulated Complex Lapped Transform (MCLT) est appliquée sur chaque frame. Cette transformée est un cas particulier de la transformée de Fourier discrète puisqu'elle prend des segments du signal, recouvrants à 50% puis calcule les coefficients

d'amplitude de la décomposition spectrale pour un nombre de bandes de fréquences déterminé. Puis, une Analyse en Composantes Principales Orientées (OPCA) est utilisée pour trouver un ensemble de projections du signal qui maximise le ratio Signal sur Bruit. L'auteur combine alors plusieurs couches d'OPCA pour créer un réseau qui extrait alors des caractéristiques robustes au bruit sur un segment. Finalement, pour 20 secondes de signal audio, cette méthode calcule un vecteur de 64 valeurs. Un vecteur de ce type est alors généré toutes les 243,6ms. L'identification est alors effectuée en calculant la distance Euclidienne entre un vecteur et ceux contenus dans une base de données de vecteurs pré-calculés. Cette méthode ne peut toutefois pas être utilisée dans notre application puisqu'elle nécessite un ensemble d'apprentissage pour apprendre les modèles de distorsions. De plus, au moins 20 secondes de signal sont nécessaires pour produire un vecteur caractéristique. Leur algorithme ne peut donc identifier un segment audio de durée plus réduite.

Haitsma et Kalker [9] associent à chaque frame un nombre codé sur 32 bits défini à partir de la décomposition du spectre de chaque frame en bandes de fréquences avec un espacement logarithme. La séquence de bits de chaque frame est définie d'après le signe de la différence d'énergie calculée entre deux bandes consécutives d'un même frame et entre deux frames consécutifs. Plus précisément, définissons EB(n,m) comme étant l'énergie de la $m^{ième}$ bande du $n^{ième}$ frame et $\Delta EB(n,m) = EB(n,m) - EB(n,m+1)$ comme la différence d'énergie de deux bandes successives d'un même frame. La valeur du $m^{ième}$ bit du $n^{ième}$ frame (F(n,m)) est alors définie par :

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

Une table de correspondance est alors crée afin d'associer à chaque sous-fingerprint de la base de donnée la liste des documents audio le contenant et la position du sous-fingerprint dans chaque document. Chaque sous-fingerprint d'un fichier audio inconnu est alors comparé à la table de correspondance pour retrouver la liste de chansons et positions auxquelles elles apparaissent. La distance de Hamming est calculée entre le segment d'entrée et les chansons aux positions sélectionnées. Finalement, un seuil sur la distance de Hamming permet de décider si deux chansons sont dérivées d'un même document. Les expérimentations présentées par les auteurs montrent que leur méthode obtient de bons résultats, même après de forts taux de compression. Cependant, comme nous l'avons déjà mentionné, le framing ne garantie pas des performances robustes en cas de suppression ou de décalage temporel (section 4). Les auteurs ne montrent pas d'expérimentations pour ce type de dégradations.

3 Extraction de fingerprint robuste

Comme mentionné en Section 2, la conception d'un fingerprint à partir d'un contenu audio nécessite deux étapes. La première consiste à décomposer le signal en séquence d'intervalles. Puis, le calcul d'une valeur de sous-fingerprint intervient pour chaque intervalle. Dans cette section, nous proposons une nouvelle méthode pour chacune des étapes précédentes.

3.1 Segmentation audio

La méthode de framing assure qu'un nombre suffisant de frames est sélectionné à partir d'un signal d'entrée (Section 2). Cependant, la sélection d'une séquence de frames contigus est sensible aux opérations de suppressions de parties du signal et de décalages temporels qui peuvent être appliquées au document (Section 2 et 4). Cet inconvénient est atténué grâce au recouvrement entre frames 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.

L'idée de base de notre méthode est de combiner les avantages respectifs des méthodes de framing et d'onsets en sélectionnant un court intervalle de temps à partir d'un intervalle plus large. L'intervalle plus court permet la détection de caractéristiques particulières du signal alors que l'intervalle plus large assure un taux minimum d'intervalles sélectionnés. Le procédé pourrait être décomposé en trois étapes (Fig. 1):

- Dans la première étape, un intervalle, appelé Intervalle d'Observation (I_o) est sélectionne au début du signal. La taille de cet intervalle est usuellement égale à quelques centièmes de secondes.
- Le signal interne à I_o est analysé pendant la seconde étape. Pendant cette étape, nous parcourons l'intervalle I_o à l'aide d'un sous intervalle de quelques millisecondes appelé Intervalle d'Energie (I_e). L'énergie de chaque intervalle est définie par l'amplitude moyenne des échantillons sur l'intervalle. L'intervalle I_e d'énergie maximale (I_{emax}) sur I_o est alors sélectionné.
- Dans la troisième étape, un dernier intervalle, appelé Intervalle de Caractérisation (I_c) est défini autour de I_{emax}. Finalement, un algorithme d'extraction de caractéristiques est appliqué sur I_c pour calculer une valeur de sous-fingerprint.

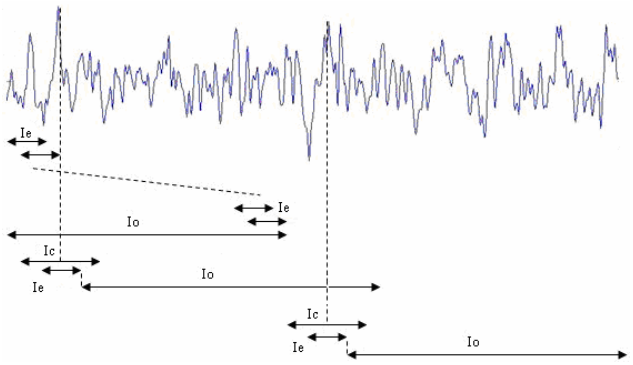

Figure 2 - *Notre méthode de segmentation.*

Partant d'un intervalle I_c sélectionné, le début de l'intervalle I_o suivant est choisi à la fin de I_{emax} (Fig. 2). La distance entre deux intervalles I_c varie alors entre I_e et $I_o - I_e$. Cette méthode apporte une plus grande robustesse envers les décalages temporels par rapport aux statégies de base qui consistent à selectionner une séquence consécutive d'intervalles I_o. 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. Cette dernière propriété permet d'améliorer la robustesse de notre méthode d'extraction de fingerprints. En effet, la statégie de base ne permettrait de sélectionner qu'un seul intervalle I_c. Or, une dégradation de signal pourrait échanger la sélection de deux I_e dont les énergies seraient proches. Notre stratégie renforce donc aussi la robustesse envers d'autres types de dégradations, et plus précisément la compression qui nous intéresse tout particulièrement.

3.2 Calcul de fingerprint

Notre méthode pour calculer un sous-fingerprint pour chaque intervalle I_c est basée sur le même principe que celle de Haitsma et Kalker [9] (Section 2). Comme ces auteurs, nous utilisons donc une décomposition du spectre de I_c en une suite de bandes de fréquences avec un espacement logarithme. Cependant, comme le montrent nos expérimentations (Section 4), un fort taux de compression peut significativement altérer la robustesse de cet algorithme d'extraction de sous-fingerprint. Ce dernier inconvénient interdit une comparaison directe de deux documents audio qui soit simplement basée sur le nombre de sous-fingerprint communs aux deux signaux. En effet, l'altération du signal par du bruit, une compression, ou une opération de suppression réduit drastiquement le nombre de valeurs identiques entre un document et le même document dégradé. Haitsma et Kalker résolvent ce problème en utilisant la distance de Hamming entre deux séquences de sous-fingerprint[9]. Cette stratégie impose toutefois de nombreux calculs de distance de Hamming.

Nous nous proposons d'améliorer la robustesse de l'algorithme d'extraction de caractéristiques en se basant sur les deux remarques suivantes:

- L'utilisation de deux intervalles successifs afin de calculer la valeur du sous-fingerprint implique la corruption de deux sous-fingerprint si une erreur se produit dans l'extraction des caractéristiques de l'intervalle Ic 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.

Nous résolvons la première source d'erreurs en n'utilisant qu'un seul intervalle pour chaque calcul de sous-fingerprint. La seconde source d'erreur est liée au fait que l'énergie d'une bande du spectre de I_c ayant subit trop de variation implique une valeur de sous-fingerprint erronée. En effet, en utilisant la même notation que dans la section 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 basique pour réduire l'influence du bruit consiste à remplacer chaque mesure EB(n,m) par le calcul d'une valeur moyenne de EB(n,m) fonction de m. Nous définissons alors 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:

$$S(n,m) = \frac{1}{m}\sum_{j=1}^{m} EB(n,i)$$

On remplace alors EB(n,m) par S(n,m) dans le calcul des différences des énergies des bandes. Le $m^{ième}$ bit du sous-fingerprint associé à l'intervalle n (F(n,m)) est donc défini par:

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

Notons que F(n,m) utilise uniquement les informations de l'intervalle n, Les erreurs ne se propagent donc pas. On peut alors facilement montrer que S(n,m)-S(n,m-1)=(EB(n,m)-S(n,m-1))/m. La formule précédente peut alors être simplifiée comme suit:

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

Le sous-fingerprint pour chaque frame n est alors défini par la concaténation des M bits $F(n,m)_{m \in \{1,..,M\}}$. Le paramètre M est fixé à 32 dans nos expérimentations (Section 4). Le fingerprint du document audio est défini comme la concaténation de la séquence de sous-fingerprint.

4 Expérimentations

Notre base de données contient 357 chansons de tous genres d'approximativement 4 minutes chacune (environ 5300 valeurs par chanson). Toutes ces chansons ont été soumises à une compression/décompression MP3 à 128kps. Les versions compressées ont ensuite été décalées temporellement en ajoutant un silence d'environ 6ms au début de chaque chanson. Les intervalles I_o, I_e et I_c ont été définis respectivement à 100ms, 1ms, 80ms pour ces expérimentations.
Les taux minimum et maximum d'extraction d'intervalles I_c pour une seconde sont alors respectivement égaux à 10 et 1000 intervalles par seconde. Le taux de détection moyen d'intervalles I_c sur l'ensemble de la base de données est égal à 21,9 intervalles par seconde. L'écart type associé à cette moyenne est égal à 3,5. Les valeurs minimales et maximales calculées sur notre base de chansons sont respectivement égales à 18 et 34.

	Moyenne	**Ecart-type**	**Min**	**Max**
Nb I_c par seconde	21,9	3,5	18	34

Table 1 – *moyenne, écart-type, valeurs min et max du nombre d'intervalles I_c détectés par secondes sur notre base de données*

Les deux premières colonnes de la Figure 2 montrent la séquence de sous-fingerprint calculées par notre méthode sur une version originale puis altérée par compression d'un même contenu audio. Pour chaque valeur de F(n,m), F(n,m)=1 est représenté par un point blanc sur la ligne n de la colonne m. La troisième colonne de cette figure représente la différence (ou exclusif) entre les deux premières colonnes. Les lignes blanches signifient qu'un intervalle I_c détecté dans un signal ne l'était pas dans l'autre. On considère alors que le sous-fingerprint est erroné. On remarque alors que très peu d'erreurs apparaissent entre les deux séquences de fingerprint. Les principales différences entre les colonnes 1 et 2 sont induites par des non correspondances des intervalles. La quatrième colonne représente le fingerprint obtenu à partir d'une version compressée puis décalée du même contenu original. La dernière colonne est une comparaison entre les colonnes 1 et 4. On peut noter visuellement que l'ajout d'un décalage temporel n'augmente pas le nombre d'intervalles détectés erronés (représentés par des lignes blanches). Sur cet exemple, le taux de bit erronés est égal à 0,22%.

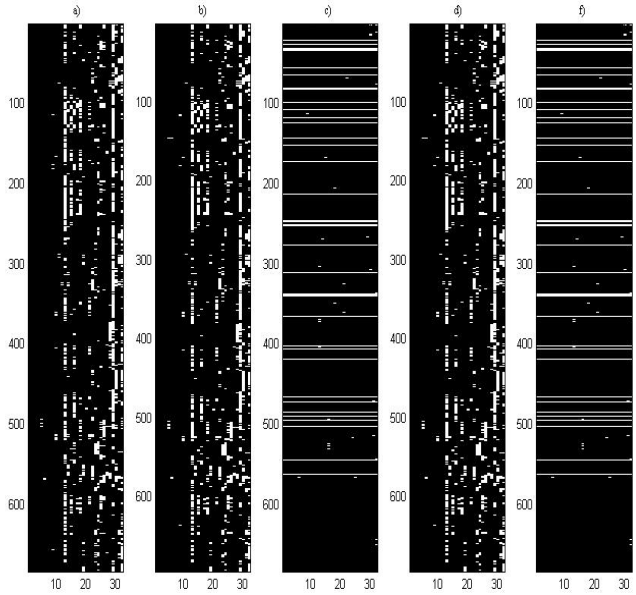

Figure 3 - *(a) fingerprint du document audio original. (b) fingerprint de la version compressée, puis décalée (d). Erreurs entre (a) et (b) puis entre (a) et (d) respectivement représentées par (c) et (e).*

Nous comparons dans la Table 1 les documents audio originaux contenus dans notre base de données avec leurs versions compressées (COM) et compressées/décalées (C&D). Les trois premières colonnes de cette table (SC, SFC et BER) sont divisées en deux sous-colonnes, chaque sous colonne représentant les performances d'un algorithme vis-à-vis des fichiers audio compressés (colonne COM) ou compressé et décalés (colonne C&D). Les colonnes SC (Segmentation Correcte) et SFC (Sous-Fingerprint Correct) représentent respectivement le pourcentage d'intervalles I_c en commun et d'intervalles communs avec des valeurs de sous-fingerprint identiques, c'est-à-dire sans un seul bit erroné, entre le fichier audio original et ses versions dégradées. La colonne BER (Bit Error Rate) représente le pourcentage de bit erronés entre les fingerprints des signaux comparés. La dernière colonne (Ko/min) correspond au nombre moyen de kilo octets nécessaires pour chaque méthode pour coder le fingerprint d'une minute de signal.

	SC		SFC		BER		Ko/ min
	COM	C&D	COM	C&D	COM	C&D	
Kalker [6]			29.9	16.9	5.8	7.3	20
Méthode Hybride	90.7	88.3	16.6	16.3	6.7	7.1	5.3
Notre Méthode	90.7	88.3	66.8	66.4	1.1	1.1	5.3

Table 2 – *Résultats d'expérimentations*

La première ligne de ce tableau illustre les performances de la méthode de Haitsma [9]. Cette méthode utilise des frames de 370ms avec un taux de recouvrement de 31/32. Les différences entre les sous-colonnes COM et C&D à l'intérieur des colonnes SFC et BER montrent la dégradation des performances de cette méthode induites par le décalage temporel. La colonne SC est laissée vide car elle n'a aucune signification pour la méthode de framing.

La seconde ligne présente les performances d'une méthode hybride combinant notre méthode de segmentation avec les calculs de sous-fingerprint proposé par Haitsma. Cette ligne montre que notre méthode de segmentation obtient en moyenne un taux d'extraction d'intervalles communs d'environ 90% (colonne SC). De plus, même si le taux de bits erronés est assez bas (colonne BER), le nombre de valeurs de sous-fingerprints sans erreurs est aussi très bas (16,6%, colonne SFC). Les erreurs sont alors parsemées sur la plupart des sous-fingerprints. On peut, de plus, noter que les performances de cet algorithme chutent légèrement après décalage temporel (colonne C&D à l'intérieur des colonnes SFC et BER). Ce comportement est principalement dû à la méthode de segmentation et au calcul de sous-fingerprint qui nécessite deux intervalles consécutifs. Notons de plus, que l'utilisation de la méthode de segmentation divise aussi par un facteur de 4 le nombre de sous fingerprints calculés (colonne Ko/min).

La dernière ligne montre les performances de notre méthode (Section 3). Le BER est beaucoup plus bas qu'en utilisant l'algorithme de calcul de sous-fingerprint de Haitsma, et plus de la moitié des sous-fingerprint (66,8%) sont extraits sans erreur. Ces performances peuvent donc permettre une comparaison de fichier audio en comptant simplement le nombre de valeurs de sous-fingerprints identiques en commun.

5 Conclusion

Nous avons présenté un nouvel identifiant audio basé sur un algorithme de segmentation du signal audio et un nouveau calcul des identifiants définissant les sous fingerprint L'algorithme de segmentation détermine des positions caractéristiques à l'intérieur du signal tout en maintenant une fréquence d'extraction de telles positions relativement constante. La fréquence d'extraction est également suffisamment élevée pour permettre une identification efficace du signal.
Notre méthode de calcul d'identifiants est basée sur l'énergie des bandes de fréquence calculées sur un court intervalle autour des positions sélectionnées. La suite des énergies de chacune des bandes est considérée comme un signal dont déduit des caractéristiques robustes au bruit susceptible de l'affecter.
La méthode de calcul d'identifiants proposée renforce la robustesse de notre méthode globale vis-à-vis de la compression du contenu. Notre méthode de sélection des positions ajoute à cela une robustesse vis-à-vis des altérations temporelles. Enfin, comparé à une simple sélection de frames consécutifs, notre méthode réduit la taille de la base de données de fingerprints.
Dans nos prochains travaux, nous envisageons de proposer une méthode d'indexation et de recherche appropriée qui, combinée à notre méthode de calcul d'identifiant, permettra une identification rapide de documents audio avec un très faible taux de faux négatif.

Références

[1] Secure Digital Music Initiative (SDMI), http://www.sdmi.org, 2001.

[2] S.Hainsworth and M.Macleod. Onset Detection in Musical Audio Signal. Dans *Proceeding of the Internationl Computer Music Conference*, 2003.

[3] F.Gouyon, A.Klapuri, S.Simon, M.Alonso, G.Tzanetakis, C.Uhle and P.Cano. An Experimental Comparison of Audio Tempo Induction Algorithms. IEEE Transactions on Audio, Speech and Language Processing. 2006.

[4] J.P.Bello, C.Duxbury, M.Davies, M.Sandler. On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain. IEEE Signal Processing Letters, vol. 11, NO.6, Juin 2004.

[5] B.Logan. Mel Frequency Cepstral Coefficients for Music Modelling. Dans *Proceeding of the International Symposium on Music Information Retriaval (ISMIR)*, 2001.

[6] S.E.Johnson and P.C.Woodland. A method for direct audio search with application to indexing and retrieval. IEEE International Conference on Acoustics, Speech and Signal. 2000.

[7] F.Kurth. A ranking technique for fast audio identification. Dans *The International Workshop on Multimedia Signal Processing*, 2002.

[8] C.Burges, J.Platt and S.Jana. Distorsion Discriminant Analysis for Audio Fingerprinting. IEEE Transactions on Speech and Audio Processing. 2002.

[9] J.Haitsma and T.Kalker. A highly robust audio fingerprinting system. *ISMIR*, pages 144-148, 2002