Approche hiérarchique pour un alignement musique-sur-partition efficace.

Cyril Joder Slim Essid Gaël Richard

Institut TELECOM – TELECOM ParisTech, CNRS/LTCI

37, rue Dareau, 75014 Paris – FRANCE

{joder, essid, grichard}@telecom-paristech.fr

Résumé

Dans le cadre du problème d'alignement audio-sur-partition, nous utilisons un modèle à états cachés pour modéliser l'évolution du contenu du signal sonore en rapport avec la partition. Nous proposons dans cet article une méthode hiérarchique de réduction de l'espace de recherche pour un tel modèle. Nos expériences menées sur une base de 94 morceaux de musique pop montrent qu'avec cet algorithme, l'utilisation d'un descripteur détectant les attaques de notes permet d'obtenir une précision d'alignement supérieure à celle de l'algorithme de programmation dynamique (DTW), avec une complexité significativement moindre.

Mots clefs

Musique, Alignement, Modèle à états cachés.

1 Introduction

Nous nous intéressons au problème de l'alignement d'une partition musicale polyphonique avec un enregistrement audio de la même pièce. Nous traitons cette tâche par une stratégie "hors ligne", qui permet d'utiliser l'enregistrement dans son ensemble. Nous nous intéressons à un alignement au niveau *symbolique*, dont le résultat sera l'ensemble des positions dans l'enregistrement de chaque note de la partition. L'alignement audio-partition peut être utilisé pour l'indexation d'un morceau de musique par sa partition, l'analyse d'interprétation ou encore comme guide pour une séparation de source informée.

Alors que la plupart des systèmes de suivi de partition en temps-réel utilisent des modèles statistiques qui peuvent être assez élaborés [1, 2, 3, 4], les méthodes "hors-ligne" se contentent souvent de l'algorithme de programmation dynamique DTW (Dynamic Time Warping) ou de variantes [5, 6, 7]. Ces approches sont en général plus simples, et peuvent aussi être appliquées au problème de synchronisation audio-audio.

Cependant, la complexité (en temps et en espace) de la DTW est quadratique en le nombre de trames audio. Ce problème a été étudié dans [8], où une DTW "à court terme" est proposée pour réduire la complexité en espace au prix d'une augmentation de la complexité en temps. Dans [9], Müller *et al* utilisent une DTW "multi-échelle", où certains chemins sont supprimés de façon hiérarchique. Ce procédé diminue la complexité de l'algorithme mais ne garantit plus d'obtenir le chemin d'alignement optimal.

Avec l'algorithme DTW ou ses variantes, l'utilisation de plusieurs descripteurs de nature différente peut être malaisé. Aussi ces systèmes se limitent généralement à l'emploi de vecteurs de chroma. Une exception notable est [7], qui propose une stratégie pour combiner les distances locales issues de descripteurs de chroma ainsi que de détecteurs d'attaque de note. Un modèle statistique à états cachés rend plus naturelle la fusion de ces informations de types différents. Cette structure est souvent utilisée dans des systèmes temps-réel [10, 11] qui modélisent chaque descripteur par un mélange de Gaussiennes.

Le système à états cachés présenté ici exploite un modèle différent pour chaque type de descripteur : un modèle "d'histogramme" (voir 3.1) pour les vecteurs de chroma, et un modèle logistique (voir 3.2) pour la fonction de détection de transitoire. Ce système obtient une très bonne précision d'alignement avec une complexité très inférieure à la DTW.

Nous exploitons en outre une approche hiérarchique de réduction de complexité, qui opère un élagage de l'arbre des états possibles, de manière adaptée aux données. Cette approche, plus souple que celle utilisée dans [9], s'avère bénéfique en terme de complexité globale, sans nuire aux performances d'alignement.

Dans les sections suivantes sont présentés le cadre statistique et les modèles d'observation utilisés. Les trois systèmes d'alignement testés sont comparés dans la section 4. Notre méthode d'élagage hiérarchique pour un décodage approché de ces modèles est proposée en section 5, avant de suggérer quelques conclusions (section 6).

2 Modèle à états cachés pour l'alignement audio-sur-partition

La partition musicale indique les temps d'attaque et durée de chacune des notes, dans une échelle temporelle – le *tempo*– qui est inconnue et variable. Néanmoins, en négligeant les possibles petites erreurs de synchronisation entre les musiciens, cela nous permet d'effectuer une seg-

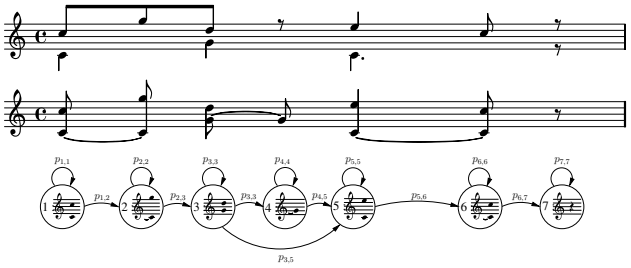

Figure 1 – *Représentations de la partition. Haut : forme graphique originale. Milieu : séquence d'accords. Bas : automate fini correspondant.*

mentation de la partition en *accords*, qui sont des ensembles de notes jouées simultanément (de façon similaire à [2]). À chaque fois qu'une note apparait ou s'éteint, un nouvel accord est créé. La partition est donc vue comme une séquence d'accords, caractérisés par les notes qu'ils contiennent. Cette segmentation en accord est représentée sur la figure 1 (haut et milieu).

Nous faisons ensuite l'hypothèse que les notes présentes à un instant de l'enregistrement dépendent uniquement de l'accord courant. Ainsi, il est possible d'utiliser un modèle à états cachés, dont les états sont les accords précédemment segmentés. Un automate fini est donc construit à partir de la partition, comme illustré sur la figure 1 (bas). La tâche d'alignement revient alors à trouver le chemin optimal (dans un sens explicité par la suite) dans l'automate correspondant à l'enregistrement.

Nous prenons le parti de ne pas utiliser les informations rythmiques de la partition, considérant que nous n'avons aucune connaissance *a priori* sur le tempo. Le critère d'optimalité utilisé est alors le maximum de vraisemblance. Soient $\mathbf{y}=y_1,\ldots,y_N$ la séquence de descripteurs extraits du signal, et S_n la variable aléatoire décrivant l'état courant au temps n. Le chemin optimal $\hat{\mathbf{S}}$, calculé par l'algorithme de Viterbi, est :

$$\hat{\mathbf{S}} = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}}\, P\big(\mathbf{y}|\mathbf{S}\big) = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}} \prod_{n=1}^{N} P(y_n|S_n), \quad (1)$$

où $\mathcal{S}$ est l'ensemble des chemins acceptables. Nous considérons comme acceptables les chemins parcourant tous les accords dans le bon ordre.

3 Modèle d'observation

De façon similaire à [12], deux sortes d'information sont considérées ici : les hauteurs de notes et les informations d'attaque. Ainsi, deux types de descripteurs sont utilisés. Pour modéliser le contenu spectral du signal, nous utilisons des *vecteurs de chroma*, et le *flux spectral* est censé détecter les attaques de notes. Ces deux descripteurs sont extraits à une fréquence de 50 Hz.

3.1 Vecteurs de chroma

Un *vecteur de chroma* est un vecteur à douze dimensions, qui représentent la "puissance" de chaque classe chromatique de la gamme musicale tempérée (de do à si). Les vecteurs de chroma utilisés ici sont calculés par la méthode décrite dans [14]. Bien que ne prenant pas en compte l'information d'octave des note, les vecteurs de chroma fournissent une représentation compacte du contenu "harmonique" du signal, efficace pour l'alignement audio-sur-partition, comme observé dans [13].

Pour chaque accord, une loi de probabilité $\{\tilde{g}(i)\}_{i=1\ldots12}$ sur les douze classes chromatiques est créée, par la superposition de lois élémentaires correspondant aux notes de cet état. Une loi élémentaire est une simple fonction de Kronecker $\{\delta(i,j)\}_{i=1\ldots12}$ où j est la classe chromatique de la note considérée. Une composante constante est ajoutée pour modéliser le bruit, ce qui donne une loi g définie par $g(i) = (1-q)\tilde{g}(i) + \frac{q}{12}$. La valeur $q = 0,7$ a été trouvée satisfaisante. Par exemple, les valeurs de la loi correspondant à l'accord $\{\mathrm{do}_3, \mathrm{mi}_3, \mathrm{sol}_3, \mathrm{do}_4\}$ (représentées en vecteur) seront : $\frac{1-q}{4}(2,0,0,0,1,0,0,1,0,0,0,0) + \frac{q}{12}\mathbb{1}$.

La vraisemblance de chaque état est ensuite calculée grâce au modèle décrit dans [15]. La valeur d'un vecteur de chroma v extrait de l'audio est considérée comme l'histogramme d'un tirage aléatoire d'après la loi g, correspondant à un accord c. La probabilité d'obtenir cet histogramme est alors

$$p\big(v|c\big) = Z \prod_{i=1}^{12} g(i)^{\alpha v(i)}, \quad (2)$$

où Z dépend uniquement de l'observation et α est un paramètre d'échelle. La valeur de ce paramètre n'influant pas sur l'optimum cherché, sa valeur est fixée à 1.

3.2 Flux spectral

Pour prendre en compte les transitoires présents aux attaques de notes, nous utilisons le descripteur de *flux spectral*, dont l'efficacité pour la tâche de détection de tempo est illustrée dans [16]. Nous employons ce descripteur pour un détecteur de transitoires "probabilisé".

Tout d'abord, les valeurs du flux spectral sont normalisées afin que la valeur maximale soit 1. Un seuil local est alors calculé par un filtre d'ordre aux 67$^{\text{ème}}$ centile, sur une longueur de 200 ms. Notre fonction de détection d'attaque est alors obtenue en retranchant ce seuil au flux spectral. Enfin, la vraisemblance d'une attaque est calculée grâce à un simple modèle logistique. Soir A la variable aléatoire de Bernoulli représentant l'évènement "attaque". Pour une valeur f de notre fonction de détection, on a

$$p\big(A=1|f\big) = \frac{e^{bf}}{1+e^{bf}} \quad (3)$$

où b est un paramètre positif à déterminer, qui contrôle la "confiance" accordée au détecteur de transitoire : plus il est grand, plus la décision sera proche d'un détecteur déterministe (valeurs 0 ou 1).

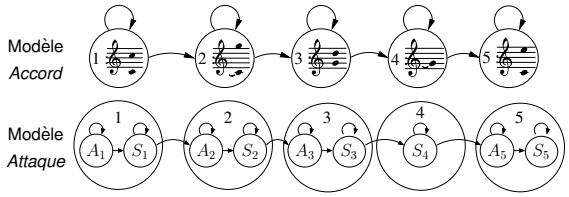

Figure 2 – *Structure des systèmes* Accord *et* Attaque *pour la même partition (*A *et* S *représentent respectivement* attaque *et* soutien*).*

4 Performance des systèmes d'alignement

4.1 Système *Accord* et système *Attaque*

Dans le cadre statistique présenté précédemment, deux structures de modèles sont utilisées. Dans la plus simple, qui constitue le système *Accord*, un accord est représenté par un unique état, quels que soient son contenu ou sa durée théorique. Seule les observations de chroma sont alors considérées. Le flux spectral n'est donc pas utilisé, et les vraisemblances des accords sont calculées d'après (2).

Le système *Attaque* est une modification du système précédent qui prend en compte les transitoires. Dans ce modèle, un "niveau de hiérarchie" inférieur est ajouté afin de pouvoir modéliser deux phases différentes à l'intérieur d'un accord : la phase d'*attaque* et la phase de *soutien*. Un accord comprenant au moins une attaque de note est donc représenté par deux états successifs, qui partagent le même modèle de chroma. Les descripteurs d'attaque et de chroma sont supposés indépendants. Un état S est alors un couple accord/phase (C, A), et sa vraisemblance est exprimée par

$$p\big(v, f|C, A\big) \propto p\big(v|C\big)p\big(A|f\big)$$

d'après les équations (2) et (3). Six valeurs différentes sont testées pour le paramètre b de l'équation (3) : 0 ; 0,1 ; 1 ; 10 ; 50 et 100.

Dans les deux cas, seules deux transitions sont autorisées à partir d'un état : vers lui-même ou vers l'état suivant. La figure 2 illustre les différences de structure entre les deux systèmes. Dans le système *Attaque*, le "bouclage" sur l'état d'attaque est utile pour modéliser une phase d'attaque (valeur élevée du flux spectral) durant plusieurs trames.

4.2 Système de référence : *Multi-scale Dynamic Time Warping*

Les modèles précédents sont comparés à un système utilisant l'algorithme multi-scale Dynamic Time Warping (MsDTW) [9, 17], qui ne correspond pas au cadre statistique présenté. L'algorithme MsDTW cherche l'alignement de coût minimal entre deux séquences, d'abord à un niveau grossier (avec une faible résolution temporelle des descripteurs), puis à un niveau plus fin. À chaque niveau de précision, l'alignement est calculé en explorant uniquement un voisinage de δ trames autour du chemin d'alignement du niveau supérieur.

Cet algorithme est utilisé pour synchroniser la séquence de vecteurs de chroma extraite de l'audio avec une séquence construite à partir de la partition (l'information d'attaque n'est pas prise en compte). Pour cela, on effectue une "pseudo-synthèse" de la partition, en associant à chaque accord un vecteur de chroma type. Les valeurs de vecteur-type sont les valeurs de la loi de probabilité vue en 3.1. Cette pseudo-synthèse est ensuite dilatée pour que sa durée soit égale à celle de l'audio.

Trois niveaux de précision sont utilisés pour l'algorithme MsDTW : le plus fin utilise les vecteurs de chroma initiaux, avec une résolution temporelle de 50 Hz. Les niveaux supérieurs utilisent ces descripteurs moyennés sur respectivement 10 trames (200 ms) et 50 trames (1 s), avec des résolutions respectives de 10 Hz et 2 Hz. La mesure de distance locale utilisée est la distance cosinus. La largeur du voisinage considéré pour la réduction hiérarchique de l'espace de recherche est fixée à $\delta = 1000$, dans le but de supprimer le moins de chemins possible, tout en conservant une complexité raisonnable.

4.3 Évaluation

Nous évaluons les performances des systèmes d'alignement grâce à une base de données de 94 chansons de 2 à 6 minutes, tirées de la base RWC-pop [18]. Ces morceaux sont polyphoniques, multi-instrumentaux et la plupart contient des percussions. Afin de réduire la taille des données, ces morceaux (initialement échantillonés à 44,1 kHz) sont sous-échantillonés à 16 kHz. La vérité-terrain est fournie par des fichiers MIDI synchronisés avec l'audio. Les partitions utilisées sont ces mêmes fichiers MIDI, dans lesquels plusieurs changements de tempo arbitraires ont été ajoutés.

Les scores sont mesurés par le taux de reconnaissance, défini comme la proportion des attaques de notes détectées correctement, dans un intervalle de 300 ms autour de l'instant d'attaque réel. Ce seuil de 300 ms est choisi égal à celui de l'évaluation MIREX'06 [19].

Les scores ainsi que la complexité moyenne sont compilés dans le tableau 1. La complexité est mesurée par le nombre de cellules (couples trame audio - état ou trame audio - trame de pseudo-synthèse) évaluées ramené au nombre de cellules nécessaires à l'algorithme DTW (le carré du nombre de trames audio).

Tout d'abord, on peut noter que la MsDTW obtient de meilleurs résultats que le système *Accord*. La raison en est que la MsDTW modélise implicitement les durées des notes dans la phase de "pseudo-synthèse", alors que le modèle statistique n'en tient pas compte. Cela augmente la précision, mais aussi la complexité (68,4% contre 16,2%).

Cependant, on voit que l'utilisation du flux spectral permet au système *Attaque* de surpasser la MsDTW, en conservant une complexité significativement moindre. En effet, un taux de reconnaissance de 87,2% est obtenu avec la valeur $b = 50$, contre 78,8% pour la MsDTW, alors que la complexité reste à 26,3%.

Système	Score	Complexité
MsDTW	**78,8**%	68,4%
Accord	64,5%	16,2%
Attaque ($b = 0$)	69,7%	26,3%
Attaque ($b = 0, 1$)	70,5%	
Attaque ($b = 1$)	73,1%	
Attaque ($b = 10$)	82,9%	
Attaque ($b = 50$)	**87,2**%	
Attaque ($b = 100$)	84,7%	

Tableau 1 – *Taux de reconnaissance et complexité moyenne (proportion de la complexité DTW) en fonction du système d'alignement.*

L'augmentation de la complexité peut être bénéfique à la précision de l'alignement. En effet, même avec la valeur $b = 0$ (le flux spectral n'est pas pris en compte), le taux de reconnaissance passe de 64,5% (système *Accord*) à 69,7% (système *Attaque*). Cela tient au fait que la plupart des accords sont alors représentés par deux états. Ainsi, la durée minimale passée dans chaque accord est de deux trames au lieu d'une. Cela évite au système de passer très rapidement d'un état à un autre état éloigné, et conduit à un chemin d'alignement plus régulier.

5 Approche hiérarchique pour un décodage approché

5.1 Principe

Afin d'accélérer encore la phase de décodage du modèle statistique, nous présentons ici une approche hiérarchique de réduction de complexité inspirée de l'algorithme MsDTW. Comme dans cette méthode, l'idée est d'effectuer un alignement d'abord à un niveau grossier, puis d'affiner ce résultat en considérant uniquement le voisinage du chemin d'alignement obtenu.

Pour ces alignement grossiers, nous tirons parti de structures musicales plus longues que les accords, à savoir les *temps* (ou *beats*, que nous emploierons pour éviter des confusions) et les *mesures*. Pour chacun de ces niveaux, on peut construire un automate, dont les états correspondent respectivement aux beats et aux mesures de la partition. Ces automates forment des modèles à états cachés dont le décodage fournira un alignement. Puisque les unités temporelles considérées sont plus grandes, on utilise des descripteurs calculés sur des fenêtres plus longues et ayant une résolution temporelle plus faible. La figure 3 illustre la construction des automates et le calcul des observations aux trois niveaux de hiérarchie utilisés.

L'algorithme se déroule comme suit : on calcule le chemin optimal $\hat{\mathbf{S}} = \hat{S}_1, \ldots, \hat{S}_N$ dans l'automate de plus haut niveau. Une passe "retour" est ajoutée à l'algorithme de Viterbi, pour calculer

$$\tilde{P}(n, s) = \max_{\mathbf{S} \in \mathcal{S}, \hat{S}_n = s} \{P(\mathbf{y}|\mathbf{S})\}$$

pour tout état s et tout instant n. $\mathcal{S}$ est l'ensemble des chemins acceptables et $\mathbf{y}$ est la séquence d'observations. Cette valeur est la vraisemblance du meilleur chemin passant par l'état s à l'instant n.

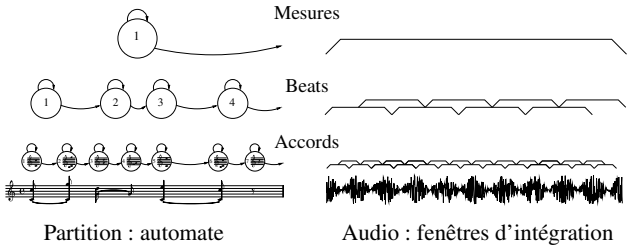

Figure 3 – *Automates finis (modélisant la partition) et fenêtres d'intégration (sur lesquelles sont calculées les observations) aux trois niveaux de hiérarchie considérés.*

La structure de l'automate est gauche-droite, on peut donc définir un ordre total sur ses états : $s \leq s'$ ssi il existe un chemin de s vers s'. On calcule alors, pour chaque instant n, les "états admissibles les plus lointains" S_n^- et S_n^+, définis par :

$$S_n^- = \min \{s \,/\, \tilde{P}(n, s) \geq \frac{P(\mathbf{y}|\hat{\mathbf{S}})}{\eta}\}$$

$$S_n^+ = \max \{s \,/\, \tilde{P}(n, s) \geq \frac{P(\mathbf{y}|\hat{\mathbf{S}})}{\eta}\}$$

où η est un paramètre contrôlant la vraisemblance minimale considérée au niveau plus bas. On définit alors les *rayons de tolérance* δ_- et δ_+ comme le maximum (pour n dans $\{1, \ldots, N\}$) du nombre d'états séparant respectivement S_n^- de $\hat{S}_n$ et $\hat{S}_n$ de S_n^+.

Ces rayons de tolérance définissent un ensemble d'états possibles autour du chemin optimal, qui est utilisé pour réduire l'espace de recherche au niveau inférieur. Un alignement au niveau inférieur est alors calculé, en explorant uniquement le domaine défini précédemment. La figure 4 illustre ce processus de réduction de l'espace de recherche.

La séquence de descripteurs audio correspondant aux niveaux supérieurs est constituée de versions intégrées (moyennées) des vecteurs de chroma initiaux, avec une résolution temporelle réduite. Le flux spectral n'est pas pris en compte à ces niveaux. Les durées d'intégration sont choisies en fonction des tempos les plus rapides acceptables. Pour le niveau beat, cette durée est de 200 ms, correspondant à un tempo très rapide de 300 bpm. Pour le niveau mesure, la fenêtre d'intégration est de 1 s, soit une mesure à 4 temps à un tempo de 240 bpm. Une recouvrement de moitié est utilisé, ce qui donne des résolutions temporelles respectives de 10 Hz et 2 Hz. Le même modèle d'observation vu en 2 est utilisé. Pour un état (beat ou mesure) la loi g est la superposition des lois associées aux accords que contient cet état, pondérées par leurs durées.

La grande différence entre cette approche et la méthode à l'origine de la MsDTW est que les rayons de tolérance δ ne sont pas des paramètres, mais sont calculés à partir des données de façon adaptative, contrôlé par le paramètre η. Il

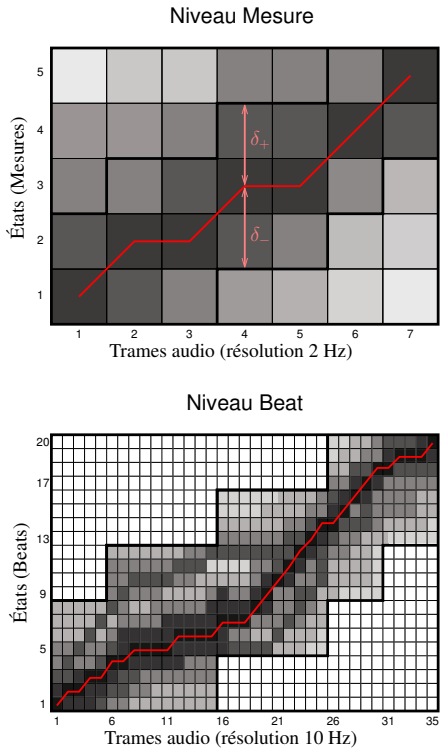

Figure 4 – *Principe de la méthode d'élagage hiérarchique. Le niveau de gris d'une cellule temps-état correspond à la vraisemblance maximale des chemins passant par cette cellule. Au niveau* beats*, seul le domaine grisé est exploré.*

est souvent plus avantageux de régler la tolérance en terme de score (le paramètre η) plutôt qu'en terme de déviation du chemin d'alignement (paramètres δ). En effet, il est possible qu'un mauvais chemin obtienne un score légèrement supérieur à celui du "vrai" chemin d'alignement à un niveau grossier, par exemple s'il suit une différente répétition d'une phrase musicale. Si ce chemin est trop éloigné du "vrai" alignement, ce dernier pourra être supprimé si on considère un rayon de tolérance fixe. En revanche, on peut supposer que le "vrai" chemin a un score élevé, et qu'il n'est pas supprimé avec notre méthode.

5.2 Expériences

Nous testons cette approche hiérarchique sur la base de données déjà présentée. Le système de bas-niveau utilisé est le système *Attaque* avec $b = 50$. Plusieurs valeurs du paramètre η sont testées et les principaux résultats sont présentés dans le tableau 2. La complexité est présentée en fraction du nombre de cellules explorées en rapport au nombre de cellules de l'algorithme DTW. La complexité au niveau mesure est égale à 0,16% pour la MsDTW, et à 0,04% pour tous les autres systèmes. Le temps d'exécution total et le nombre d'"erreurs d'élagage" sont présentés. Une erreur d'élagage se produit si une partie du "vrai" chemin d'alignement se trouve supprimée dans le processus de réduction de l'espace de recherche. Notre implémentation de l'algorithme est en MATLAB, et a été exécuté sur un

Système	Complexité		Temps	Erreurs
	Beats	Accords	d'exécution	(nb)
MsDTW δ=150	2,24%	14,02%	1180 s	0
Attaque b=50	–	26,26%	482 s	0
$\delta = 60$	0,81%	7,93%	362 s	0
$\eta = 1000$	0,42%	4,53%	300 s	0
$\eta = 200$	0,35%	4,07%	276 s	0
$\eta = 100$	0,33%	3,82%	265 s	0
$\eta = 50$	0,30%	3,59%	256 s	0
$\eta = 20$	0,26%	3,22%	240 s	0
$\eta = 10$	0,23%	2,97%	229 s	2
$\eta = 5$	0,19%	2,59%	215 s	2

Tableau 2 – *Performance de notre implémentation du système d'alignement, avec différents paramètres pour l'algorithme hiérarchique de réduction de complexité. Les erreurs comptent le nombre de morceaux pour lesquels le "vrai" chemin est supprimé au cours de l'élagage.*

Intel Core2, 2,66 GHz avec 3,6 Go de RAM, sous linux.
Trois systèmes de références sont considérés. Le premier utilise l'algorithme MsDTW avec le paramètre $\delta = 150$, qui est la valeur minimale n'entrainant aucune erreur d'élagage sur notre base de données. Le second système utilise le modèle *Attaque* sans élagage. Le dernier effectue un élagage hiérarchique, mais avec des rayons de tolérance constant $\delta_- = \delta_+ = 60$. Cette valeur est la plus basse pour laquelle aucune erreur n'est comptée.
En terme de précision d'alignement, tous les systèmes qui ne font pas d'erreurs d'élagage obtiennent le même score que le système de référence (87.16%). Ainsi, cette méthode approchée de décodage n'affecte pas les performances d'alignement.
Les résultats montrent l'avantage de cette approche, car la complexité et le temps d'exécution de tous les systèmes l'utilisant sont inférieurs à celle de la méthode de référence (sans élagage). Comme prévu, la complexité diminue avec la valeur de η. Aucune erreur d'élagage n'est à déplorer jusqu'à la valeur $\eta = 20$, dont le temps d'exécution correspondant est la moitié de celui du système de référence (240 s contre 484 s).
Le gain de cette méthode par rapport à l'utilisation d'un rayon de tolérance δ fixe est visible. En effet, le système utilisant un rayon fixe optimal $\delta = 60$ s'exécute en 362 s et présente une complexité en espace supérieure à notre stratégie d'élagage "adaptative".
Sur la figure 5 sont représentés les complexités en espace de trois systèmes d'alignement : sans élagage, avec un rayon fixe $\delta = 60$ et avec $\eta = 20$. Les deux stratégies d'élagage entrainent une réduction significative de la complexité sur tous les morceaux. De plus, il est visible que la taille de l'espace de recherche obtenu avec notre stratégie peut fortement varier suivant les morceaux, alors qu'elle est à peu près constante avec un rayon δ fixe (les variations sont dues aux différents nombres de notes par beat). Ces variations ne sont pas corrélées avec le nombre d'états initial du modèle, indiquant que notre approche adapte le processus d'élagage aux données. Ainsi, alors que dans cer-

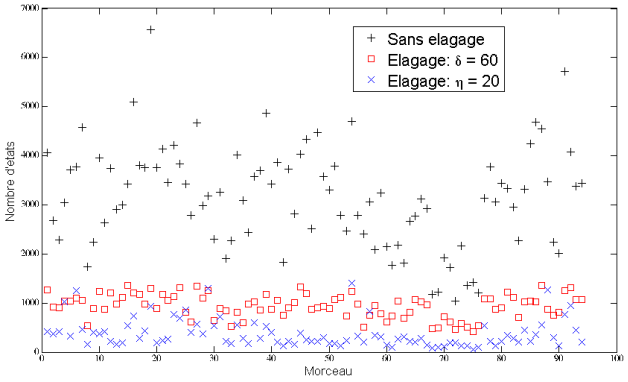

Figure 5 – *Nombre d'états explorés par trame, au cours l'alignement au plus bas niveau en fonction du morceau traité.*

tains cas la complexité de notre méthode est supérieure à celle d'un rayon fixe, elle est significativement plus faible pour la majorité des morceaux.

6 Conclusion

Dans cet article, nous montrons qu'une approche hiérarchique pour le décodage approché d'un modèle à états cachés peut fournir un alignement d'une très bonne précision pour une faible complexité. Nos expériences indiquent que les taux de reconnaissance sont supérieurs à ceux obtenus avec un système par DTW, quand une description des attaques de notes est utilisée en plus des vecteurs de chroma, et cela en conservant une complexité inférieure pour la phase de décodage.

La méthode hiérarchique d'élagage de l'arbre de recherche réduit encore la complexité, sans affecter la précision du système. L'avantage de notre stratégie par rapport à celle utilisée dans [9, 17] est que les rayons de tolérance peuvent s'adapter aux données, ce qui conduit à une meilleure efficacité globale.

La suite de ces travaux sera consacrée à l'utilisation de modèles plus élaborés pour le niveau le plus bas, dont l'utilisation est rendue abordable grâce à la réduction de complexité engendrée par l'élagage. Nous tenterons aussi de réduire encore le nombre d'états du modèle statistique en tirant parti des répétitions dans la structure musicale.

Références

[1] Lorin Grubb et Richard Dannenberg. A stochastic method of tracking a vocal performer. Dans *Proc. of ICMC*, 1997.

[2] Christopher Raphael. A probabilistic expert system for automatic musical accompaniment. *Journal of Computational and Graphical Statistics*, 10, 2001.

[3] Diemo Schwarz, Nicola Orio, et Norbert Schnell. Robust polyphonic midi score following with hidden markov models. Dans *Proc. of ICMC*, 2004.

[4] Arshia Cont. A coupled duration-focused architecture for realtime music to score alignment. *IEEE Trans. on Pattern Analysis and Machine Intelligence on*, 32(6) :974–987, June 2010.

[5] Ning Hu, Roger B. Dannenberg, et George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. Dans *Proc. of IEEE WASPAA*, 2003.

[6] Christian Fremerey, Michael Clausen, Sebastian Ewert, et Meinard Müller. Sheet music-to-audio identification. Dans *Proc. of ISMIR*, 2009.

[7] Sebastian Ewert, Meinard Müller, et Peter Grosche. High resolution audio synchronization using chroma onset features. Dans *Proc. of IEEE ICASSP*, 2009.

[8] Hagen Kaprykowsky et Xavier Rodet. Globally optimal short-time dynamic time warping : Application to score to audio alignment. Dans *Proc. of IEEE ICASSP*, 2006.

[9] Meinard Müller, Henning Mates, et Frank Kurth. An efficient multiscale approach to audio synchronization. Dans *Proc. of ISMIR*, 2006.

[10] Pedro Cano, Alex Loscos, et Bonada. Score-performance matching using hmms. Dans *Proc. of ICMC*, 1999.

[11] Arshia Cont, Diemo Schwarz, et Norbert Schnell. Training ircam's score follower. Dans *Proc. of IEEE ICASSP*, 2005.

[12] Nicola Orio et Diemo Schwarz. Alignment of monophonic and polyphonic music to a score. Dans *Proc. of ICMC*, 2001.

[13] Cyril Joder, Slim Essid, et Gaël Richard. A comparative study of tonal acoustic features for a symbolic level music-to-score alignment. Dans *Proc. of IEEE ICASSP*, 2010.

[14] Yongwei Zhu et M.S. Kankanhalli. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Trans. on Multimedia*, 8(3) :575–584, June 2006.

[15] Christopher Raphael. Aligning music audio with symbolic scores using a hybrid graphical model. *Machine Learning Journal*, 2006.

[16] Miguel Alonso, Gaël Richard, et Bertrand David. Extracting note onsets from musical recordings. Dans *Proc. of ICME*, 2005.

[17] Stan Salvador et Philip Chan. Fastdtw : Toward accurate dynamic time warping in linear time and space. Dans *KDD Workshop on Mining Temporal and Sequential Data*, pages 70–80, 2004.

[18] M. Goto. Rwc music database : Popular, classical, and jazz music databases, 2002.

[19] Music information retrieval evaluation exchange 2006, score following task : `http://www.music-ir.org/mirex/2006/index.php/Score_Following_Proposal`.