UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

ALIGNEMENT DU CHANT PAR RAPPORT À UNE RÉFÉRENCE AUDIO EN TEMPS RÉEL

Mémoire de maîtrise
Spécialité : génie électrique

Eric JULIEN

Jury : Roch LEFEBVRE
Éric PLOURDE
Roger GOULET

Sherbrooke (Québec) Canada Janvier 2013

Your file Votre référence
ISBN: 978-0-494-93337-4

Our file Notre référence
ISBN: 978-0-494-93337-4

RÉSUMÉ

Dans l'optique de créer un système de karaoké qui modifie une interprétation chantée à capella en temps réel, il est nécessaire de pouvoir localiser l'interprète par rapport à une référence afin de pouvoir déterminer quelle serait la cible d'un algorithme de modification de la voix. Pour qu'un tel système fonctionne bien, il est nécessaire que l'algorithme d'alignement exploite au maximum les spécificités de la voix, qu'il utilise l'information liée au texte prononcé plutôt qu'aux aspects artistiques du chant, qu'il soit à temps réel et qu'il offre la plus faible latence possible.

Afin d'atteindre ces objectifs, un système d'alignement basé sur le *Dynamic Time Warping* (DTW) a été développé. Une adaptation temps réel simple de l'algorithme ordinaire de la DTW qui permet d'atteindre les objectifs énumérés est proposée et comparée à d'autres approches répertoriées dans la littérature. Cette adaptation a permis d'obtenir de meilleurs résultats que les autres techniques testées.

Une étude comparative de trois types d'analyses spectrales couramment utilisées dans des systèmes de reconnaissance automatique de la voix a été réalisée, dans le cadre spécifique d'un algorithme d'alignement de la voix chantée. Les coefficients évalués sont les *Mel-Frequency Cepstrum Coefficients* (MFCC), les *Warped Discrete Cosine Transform Coefficients* (WDCTC) et les coefficients de l'analyse *Perceptual Linear Prediction* (PLP). Les résultats obtenus indiquent une meilleure performance pour l'analyse PLP.

L'utilisation d'une fonction de transformation linéaire par morceaux, appliquée aux matrices de coûts instantanés obtenues, permet de rendre l'alignement le plus facilement distinguable dans les matrices de coûts cumulés calculées. Les paramètres de la fonction de transformation peuvent être obtenus par l'optimisation en boucle fermée par recherche directe par motif. Une fonction-objectif permettant d'éviter les discontinuités de l'écart quadratique moyen sur l'alignement est développée.

Plusieurs matrices de coûts peuvent être combinées entre elles en effectuant une somme pondérée des matrices de coûts instantanées transformées de chacun des paramètres considérés. La pondération est également obtenue par optimisation. Plusieurs assemblages sont comparés : les meilleurs résultats sont obtenus avec une combinaison de l'analyse PLP et du niveau d'énergie et des dérivées de ceux-ci. L'écart moyen sur l'alignement de référence est de l'ordre de 50 ms, avec un écart-type d'environ 75 ms pour les séquences testées.

Des perspectives permettant d'améliorer la convergence de l'algorithme pour les paires de séquences audio difficiles à aligner, d'obtenir de meilleures matrices de coûts en utilisant d'autres contraintes locales, en considérant l'intégration de nouveaux paramètres tels le *pitch* ou en utilisant une base de données de voix chantée segmentée pour optimiser une mesure de distance sont données.

Mots-clés : alignement, chant, audio, DTW, PLP, MFCC, WDCTC

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

CHAPITRE 1

INTRODUCTION

1.1 Contexte

Le Groupe de Recherche sur la Parole et l'Audio (GRPA) désire développer un système s'apparentant à un dispositif de karaoké intelligent qui modifierait le chant de l'utilisateur afin de le rendre juste. Ce système modifierait notamment la hauteur (*pitch*, en anglais) de la voix de l'usager afin que celle-ci corresponde à la hauteur de l'oeuvre originale. Il pourrait également modifier le timbre et les différentes modulations de la voix (c'est-à-dire le vibrato, la dynamique temporelle de l'amplitude de la voix, etc.) afin que ceux-ci soient plus près de l'oeuvre originale. Il est également désiré que le système n'impose pas le tempo de l'interprétation, permettant des interprétations *a capella*.

De nature ludique, outre un nouveau moyen pour se divertir et oublier les tracas du quotidien, le développement d'un tel système n'a pas vraiment de débouchés directs importants pour la société. Cependant, les nombreux problèmes devant être résolus pour parvenir à un système qui fonctionne bien sont de taille et leurs solutions nécessiteront des contributions originales dans de nombreux domaines, tel le traitement des signaux audio (plus spécifiquement de la modification de la voix chantée), la caractérisation des signaux de voix et/ou de chant et la reconnaissance vocale. Ces contributions seront utiles pour résoudre d'autres problèmes d'intérêts plus directs pour la société.

Parmi les sous-problèmes soulevés par un tel dispositif de karaoké intelligent, il y a le problème de la modification de la voix. En effet, il est nécessaire de développer des algorithmes de modification de la voix chantée qui permettent d'ajuster les caractéristiques qui auront été jugées importantes à une reproduction fidèle d'une oeuvre originale. On peut penser entres autres à la hauteur, au timbre, à la dynamique temporelle de l'amplitude et aux diverses modulations de la voix chantée (par exemple, le vibrato) qui sont toutes des caractéristiques qui, à défaut d'être bien reproduites par l'interprète, nuiront à sa performance. En plus de permettre l'ajustement des caractéristiques choisies, ces algorithmes devront permettre de conserver au maximum le naturel de la voix chantée, c'est-à-dire qu'ils ne devront pas introduire d'artefacts qui risquent de dévoiler la nature synthétique du chant à la sortie du système.

1.2 Problème étudié

L'autre problème important que soulève le système de karaoké intelligent est le problème qui est étudié dans ce document. En supposant que les algorithmes de modification de la voix chantée sont ou seront disponibles (ils sont l'objet d'autres projets au GRPA), il sera nécessaire de leur fournir une cible dynamique qui représentera l'évolution des caractéristiques du chant dans l'oeuvre originale. Puisque le système de karaoké n'imposera pas le tempo de l'interprétation, le système devra fournir ces caractéristiques en fonction de la position calculée de l'interprète dans son interprétation. Pour ce faire, on doit pouvoir savoir à quel indice temporel, par exemple le temps en secondes, correspond la position actuelle de l'interprétation dans l'oeuvre originale.

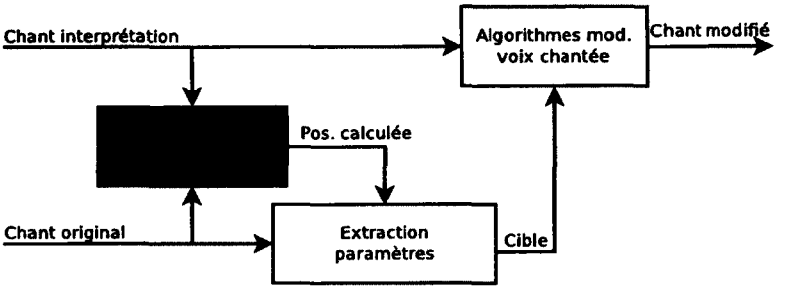


Figure 1.1 Schéma bloc du système de karaoké

L'oeuvre originale peut être représentée de deux façons. La première est d'utiliser une partition musicale et la seconde est d'utiliser un enregistrement audio du chant, isolé, de l'oeuvre originale. Ce qui représente l'original sera désormais désigné sous le nom de référence et le problème est désigné sous le nom d'alignement ; alignement de partitions, dans le cas où la référence est une partition. La figure 1.1 présente le positionnement de l'algorithme d'alignement dans le système global.

Plusieurs travaux connexes, la plupart fonctionnant avec des instruments de musique au lieu de la voix chantée, proposent des solutions au problème de l'alignement de partitions. Cependant, pour la voix, l'utilisation d'une partition laisse à désirer puisque le chant est de nature beaucoup plus continue que la musique produite avec des instruments de musique conventionnels. En effet, pour le *pitch*, par exemple, un instrument a tendance à utiliser un sous-ensemble discret des valeurs de *pitch* possible (hormis les modulations occasionnelles) qui comprends 12 notes par octave, alors que la voix a bien souvent une évolution temporelle de son *pitch* qui est continu [Puckette, 1995] et qui donc est difficile à représenter sur une partition. De plus, la nécessité de transcrire le chant en partition est peu désirable puisque cela doit être effectué de façon manuelle, ce qui peut être long et/ou difficile.

Pour ces raisons, et contrairement à la plupart des travaux connexes répertoriés dans la littérature, le problème abordé par les travaux explicités dans ce mémoire est le problème de l'alignement avec une référence de type audio, c'est-à-dire où la référence est un enregistrement du chant isolé de l'oeuvre originale interprétée. À noter que le problème de l'alignement de partitions pour la voix chantée a fait l'objet d'études au GRPA [Beaudette, 2010].

1.3 Objectifs

L'algorithme d'alignement proposé permet d'atteindre plusieurs objectifs. L'objectif premier et essentiel est évidemment de permettre d'obtenir, à chaque instant de l'interprétation, l'instant correspondant dans la référence, qui représente l'oeuvre originale.

De plus, cet alignement (la série d'instants calculés) doit être très précis. En effet, puisque l'algorithme d'alignement sera en grande partie responsable de commander les algorithmes de modification de la voix (elle partagera cette responsabilité avec certains algorithmes d'extraction de paramètres), toute erreur d'alignement aura une répercussion sur le chant en sortie. Si la correction faite par le système ne correspond pas à ce qui devrait être fait, l'expérience sera désagréable pour l'interprète et/ou son public.

Aussi, l'évolution rapide et continue des caractéristiques de la voix chantée vient aggraver cette nécessité de précision. Par exemple, le vibrato dans la voix d'un chanteur peut atteindre 7 cycles par seconde (hertz) [Puckette, 1995]. Afin de bien suivre la courbe de *pitch* pendant un vibrato, il est donc nécessaire d'avoir une précision excédant de beaucoup 150 millisecondes (environ 1 cycle à 7 hertz).

En plus d'une grande précision, l'algorithme devra pouvoir tourner en temps réel avec une latence assez petite pour être peu ou pas perceptible. Pour un système de karaoké intelligent, il est absolument nécessaire que tous les algorithmes soient des algorithmes temps réel, sinon on ne pourrait pas avoir le chant modifié en sortie du système au fur et à mesure que celui-ci est produit. L'intérêt du système serait donc grandement diminué. De plus, on estime qu'un auditeur peut discerner un délai lorsqu'une de ses actions et sa réponse sonore excède 20 millisecondes [Lago et Kon, 2004]. Or, avant que l'interprète puisse entendre son chant modifié en sortie, le son devra

1. être numérisé et rendu disponible à l'algorithme d'alignement,
2. être aligné à l'aide de l'algorithme qui sera développé,
3. être modifié pour atteindre la cible trouvée à l'aide de l'algorithme d'alignement,

4. être converti en un signal analogique pour être joué sur un haut-parleur,

5. parvenir jusqu'à lui par la voie des airs (environ 3 millisecondes par mètre).

La latence introduite par l'algorithme d'alignement développé s'additionnera donc à la latence totale du système. Par ailleurs, selon [Inoue *et al.*, 1994], 100 millisecondes est trop de retard pour que l'expérience d'un karaoké intelligent soit transparente. Les contraintes de latence sont donc très sévères et seront à considérer pour le développement de l'algorithme d'alignement.

Un autre objectif important au bon fonctionnement du système est la robustesse aux erreurs de l'interprète. En effet, le public visé par un tel système n'est pas seulement les chanteurs professionnels, mais aussi les chanteurs amateurs. La qualité de leurs interprétations d'une pièce peut donc varier grandement. Il est possible que certains faussent complètement, qu'ils soient complètement monotones, qu'ils chantent fort des passages doux et vice-versa, etc. Le système doit être robuste à ces erreurs d'interprétation et se fier davantage à d'autres paramètres, par exemple, le texte, qui risque fort bien d'être bien prononcé par l'interprète.

De plus, certains interprètes risquent de répéter, d'omettre ou de se tromper de mots quelques fois. L'algorithme devra donc pouvoir bien récupérer la position de l'interprète après s'être égaré (soit après des erreurs de l'algorithme ou après des erreurs d'interprétation).

L'algorithme devra également être le plus indépendant possible aux changements de locuteur. Plusieurs systèmes répertoriés dans la littérature sont adaptés spécifiquement pour un seul locuteur. Le système développé doit pouvoir être utilisé avec plusieurs locuteurs différents. Puisque le canal vocal de chaque individu est différent (encore plus entre un homme et une femme) et que chaque individu exerce ses muscles de façon différente pour produire de la parole et du chant, il est difficile d'être insensible au changement de locuteur. Certains systèmes sont entraînés pour chacun des interprètes qui voudront l'utiliser, mais dans ce cas-ci, il est désirable d'éliminer tout entraînement, même automatisé, afin de permettre des changements rapides d'interprète pour qu'il soit possible de l'utiliser dans un bar karaoké ou dans un rassemblement, par exemple, sans que des entraînements nuisent au plaisir que procurera l'utilisation du système.

1.4 Contributions

Même si plusieurs algorithmes permettant de faire de l'alignement (et particulièrement de l'alignement de partitions) sont répertoriés dans la littérature, aucun des algorithmes répertoriés ne fait l'alignement avec une référence audio tout en étant spécialisé pour la voix chantée.

Dans le domaine de la reconnaissance vocale, certains vieux algorithmes le font (la plupart des algorithmes modernes sont basés sur les *Hidden Markov Models(HMMs)*), mais avec la voix parlée et avec des contraintes de délai beaucoup moins grandes.

D'autres algorithmes permettent l'alignement de séquences audio quelconques, par exemple l'algorithme de [Dixon, 2005], mais ces algorithmes ne sont pas aussi performants qu'un algorithme spécialisé pour la voix chantée, pour l'application visée.

La contribution originale principale des travaux détaillés dans cet ouvrage est donc un algorithme temps réel d'alignement audio spécialisé pour la voix chantée, avec une faible latence.

De plus, la démarche suivie dans le développement de cet algorithme a nécessité une comparaison de divers paramètres spectraux caractérisant les phonèmes. Cette comparaison a permis d'identifier quel paramètre est le plus approprié pour identifier un phonème dans le cas de la voix chantée, indépendamment du locuteur. De telles études ont été effectuées pour la voix parlée [Xuan *et al.*, 2002], mais aucune étude répertoriée ne le fait pour la voix chantée.

1.5 Sommaire

Bref, le présent document présentera un algorithme d'alignement audio spécialisé pour la voix chantée qui

1. utilise une référence de type audio,
2. fait la localisation avec une grande précision (excédant 150 millisecondes),
3. fonctionne en temps réel,
4. a une faible latence,
5. est robuste aux erreurs d'interprétation,
6. est robuste aux changements de locuteur

et qui est original par sa spécialisation pour les signaux de voix chantée.

1.6 Plan du document

Ce document est organisé de la façon suivante :

1. Le chapitre 2 décrit l'état de l'art pour le domaine de l'alignement audio. Les différentes techniques pour faire l'alignement de partitions et d'audio et les algorithmes s'y rattachant répertoriés dans la littérature sont exposés et analysés. Les caractéristiques importantes de la voix chantée et leurs représentations sont également présentées.
2. Le chapitre 3 présente en détail l'algorithme global d'alignement utilisé et les ajustements qui y ont été faits.
3. Le chapitre 4 explicite les paramètres utilisés comme entrées au système d'alignement pour représenter la voix chantée. Une étude comparative entre plusieurs paramètres spectraux représentant les phonèmes prononcés y est aussi présentée.
4. Le chapitre 5 porte sur l'étude des fonctions de coût qui permettent de comparer les paramètres choisis entre deux signaux et sur les techniques utilisées pour les optimiser.
5. Le chapitre 6 présente l'assemblage du système ainsi que l'analyse des résultats obtenus et propose des pistes d'amélioration pour les lacunes identifiées dans le système proposé.
6. Le chapitre 7 est un sommaire du document.

CHAPITRE 2

ÉTAT DE L'ART

Dans ce chapitre est présenté un survol de la littérature pertinente dans le cadre du projet de système de karaoké. La littérature étudiée est regroupée selon deux sujets principaux, soit les paramètres permettant de caractériser la voix et les techniques permettant d'aligner des séquences audio.

Un algorithme d'alignement, qu'il utilise une référence sous la forme d'une partition ou d'un signal de référence, n'utilise presque jamais la valeur des échantillons des signaux directement pour faire l'alignement. Pour une référence sous forme d'une partition, il serait très peu pratique de devoir générer un signal complet à partir de la partition pour pouvoir effectuer la comparaison. De plus, très peu d'information pertinente est contenue dans la valeur d'un seul échantillon. Aussi, puisqu'aucun sous-échantillonnage n'est effectué, le nombre de comparaisons impliquées fait en sorte que le temps de calcul est très élevé.

Afin de pouvoir comparer les signaux (ou le signal et la partition), on doit donc convertir l'interprétation et la référence dans un espace de paramètres qui permettra une comparaison avec beaucoup moins de redondance et qui varie le moins possible d'une interprétation à une autre. Pour réduire la quantité de comparaisons nécessaires, on doit travailler avec des trames, de façon à avoir une valeur de paramètre par trame. La figure 2.1 représente un système d'alignement où il y a une telle réduction de l'espace de paramètres.

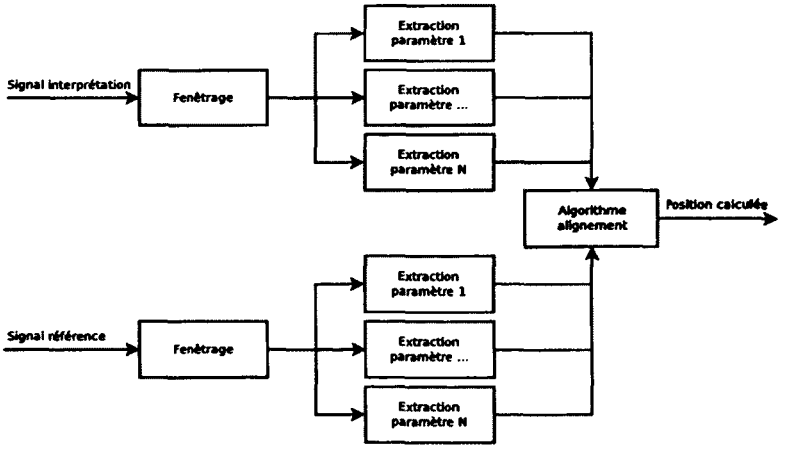


Figure 2.1 Schéma d'un système d'alignement générique

La réduction de l'espace de paramètres est une partie très importante du système d'alignement. Un survol des paramètres caractérisant la voix est présenté dans les prochaines pages.

2.1 Paramètres reliés à la voix

Une grande multitude de paramètres caractérisent les différents aspects de la voix. Certains traduisent bien l'aspect musical de la voix ; le *pitch* et le timbre, par exemple. D'autres concernent plus l'aspect cognitif de la voix ; ils tentent d'exprimer le mieux possible le contenu de la parole. Ils sont les paramètres les plus importants pour le système d'alignement développé puisqu'ils se rapportent au contenu du texte plutôt qu'à la performance vocale, qui risque d'être moins fidèle dans le cas de chanteurs amateurs. Ce sont ces derniers paramètres que la prochaine section étudiera.

2.1.1 Paramètres spectraux - Phonèmes

Les études sur la cognition de la voix, qu'elles aient été fondamentales ou appliquées au domaine de la reconnaissance vocale, ont pu établir plusieurs hypothèses intéressantes sur la production et la cognition de la voix chez l'humain. Il a été établi que la cognition de la parole se faisait à travers des phonèmes, unité de base de la phonétique, qui représentent de courts sons fréquemment produits lors de la locution. Chaque phonème est la prononciation d'une ou plusieurs lettres d'un mot. Plusieurs phonèmes peuvent correspondre à un certain groupe de lettres ; le contexte dicte alors le bon phonème à prononcer. Il existe aussi les ditphongues, qui sont un assemblage de deux ou plusieurs phonèmes avec un glissement continu entre les phonèmes la composant. Un exemple dans la langue anglaise est le mot *I* qui est prononcé en tant que diphtongue avec ses phonèmes constituants [a] et [I].

Chaque phonème est produit en plaçant la langue à un endroit différent et en contrôlant l'ouverture de la bouche. Le *pitch* de la voix n'a pas d'importance au niveau de la cognition des phonèmes. L'effet du déplacement de la langue et de l'ouverture de la bouche modifie la réponse en fréquence du canal vocal. Les multiples résonnances de celui-ci sont déplacées selon le phonème prononcé. Ceci crée ce qu'on appelle des formants (désignés par F1 et F2 à la figure 2.2), c'est-à-dire des lieux de haute énergie observables en analysant la voix dans le domaine fréquentiel. En analysant la position et les caractéristiques (largeur de bande, par exemple) de ceux-ci, on peut arriver à déterminer quels phonèmes sont prononcés.

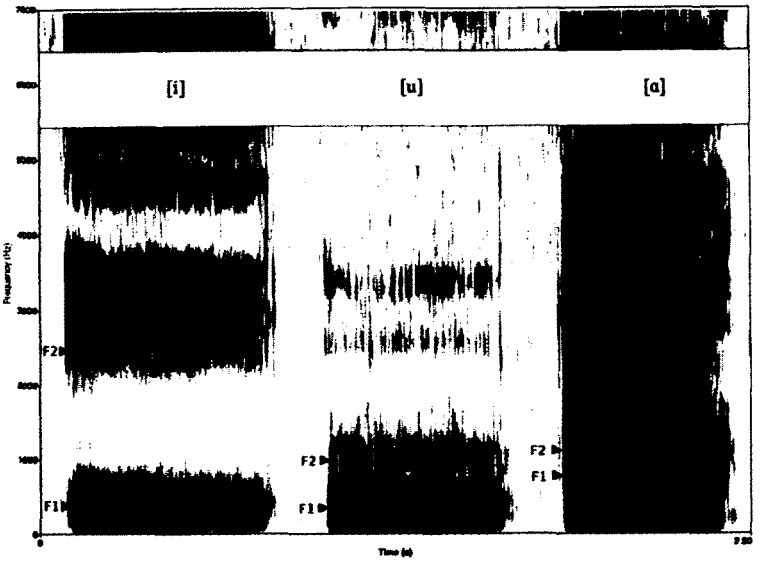


Figure 2.2 Spectrogramme de la séquence de phonèmes [iua] prononcés par un natif de la Louisiane (source : Wikimedia Commons)

Prédiction Linéaire

Il serait possible de trouver la position des formants à partir de la transformée de Fourier discrète (TFD), mais le nombre de pics obtenus pour une taille de transformée procurant une bonne résolution fréquentielle est très élevé. Il est donc difficile de choisir les pics qui sont les formants. Le problème est que la TFD regroupe l'information sur la réponse en fréquence du conduit vocal mais aussi sur la vibration des cordes vocales. On obtient donc une enveloppe spectrale correspondant à la réponse en fréquence du tract vocal surimposée par des séries d'harmoniques.

Les formants étant liés à la réponse en fréquence du tract vocal, il serait désirable d'isoler l'enveloppe spectrale. Il est possible de lisser le spectre de Fourier obtenu, mais il est préférable d'utiliser d'autres méthodes pour extraire seulement l'enveloppe spectrale du signal.

Une de ces méthodes est la prédiction linéaire [Makhoul, 1975]. Un prédicteur linéaire est un filtre à moyenne mobile dont la sortie est une estimation de la valeur du prochain échantillon du signal. Les coefficients du prédicteur linéaire sont calculés de façon à minimiser l'écart quadratique moyen (*Mean Squared Error*, MSE) entre la valeur calculée et la valeur réelle du prochain échantillon. Il fut constaté qu'avec des prédicteurs d'ordres bien choisis, les pôles du filtre de synthèse, qui est autorégressif, se placent de façon à modéliser les résonnances du signal prédit, ce qui revient à modéliser l'enveloppe spectrale du signal. Avec des ordres trop élevés, on risque de voir apparaître des harmoniques du signal dans

la réponse en fréquence du filtre obtenu. La figure 2.3 montre un exemple d'enveloppe spectrale obtenue à l'aide de la technique de la prédiction linéaire.

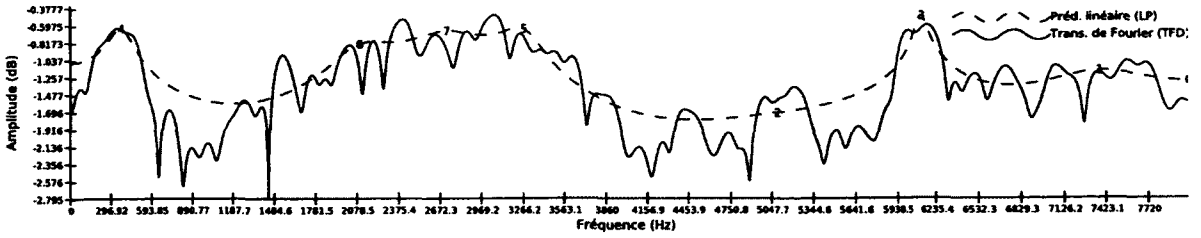


Figure 2.3 Enveloppe spectrale obtenue par prédiction linéaire d'ordre 15 sur un segment de 20 ms du phonème [u]. Les chiffres sur le graphique représentent la position des différents pôles du filtre prédictif.

Plutôt que de choisir les fréquences des formants, il est avantageux d'utiliser directement le spectre, éliminant ainsi les probabilités de mal choisir les pics qui seront choisis comme formants.

Une amélioration de cette technique est détaillée dans [Prahallad *et al.*, 2006]. Celle-ci consiste à mettre en rapport deux spectres d'ordre différents obtenus par prédiction linéaire afin de mieux faire ressortir les résonnances faibles et/ou cachées par l'affaiblissement (*roll-off*) du spectre dans les hautes fréquences.

Mel Frequency Cepstrum Coefficients (MFCC)

L'identification des phonèmes est un problème très important pour le domaine de la reconnaissance de la parole. De nombreuses techniques qui permettent de mieux identifier les phonèmes dans un signal de parole ont été développées. Puisque, sans contredit, le meilleur système de reconnaissance de la parole est l'humain, certaines techniques se sont inspirées de la psychoaccoustique, c'est-à-dire de la façon dont les sons sont perçus par le cerveau humain.

Un des paramètres développés pour caractériser l'aspect phonétique d'un signal est les MFCC [Mermelstein, 1976]. Le MFC (*Mel-Frequency Cepstrum*) utilise l'échelle de Mel, qui est une échelle de fréquences basée sur la perception humaine. Le spectre obtenu, avant d'en extraire le cepstre, est donc plus représentatif de la façon dont le cerveau fonctionne.

L'utilisation des coefficients cepstraux a l'avantage de permettre une comparaison plus rapide entre deux échantillons. En effet, il permet de calculer une distance à partir d'un nombre de coefficients très inférieur à celui d'un spectre fréquentiel conventionnel, l'information sur l'enveloppe spectrale étant compactée dans les premiers coefficients du cepstre. En effet, selon [Juang et Rabiner, 1993], la distance cepstrale tronquée $d_c^2(L)$ approche la

distance spectrale logarithmique quadratique d_2^2 pour des signaux représentables par un modèle autorégressif (ce qui est habituellement vrai pour des enveloppes spectrales) :

$$d_c^2(L) = \sum_{n=1}^{L}(c_n - c'_n)^2 \approx d_2^2 = \int_{-\pi}^{\pi} |log(S(\omega)) - log(S'(\omega))|^2 \frac{d\omega}{2\pi} \tag{2.1}$$

où c_n et c'_n sont les coeffients cepstraux d'un échantillon de chaque signal comparé, L est la longueur utilisée pour tronquer la distance cepstrale et $S(\omega)$ et $S'(\omega)$ sont les spectre de Fourier des échantillons comparés. De plus, en tronquant le cepstre, l'information sur l'enveloppe spectrale, qui est concentrée dans les premiers coefficients, est conservée et l'information spectrale fine est éliminée.

La figure 2.4 présente le schéma bloc haut niveau de l'analyse MFCC.

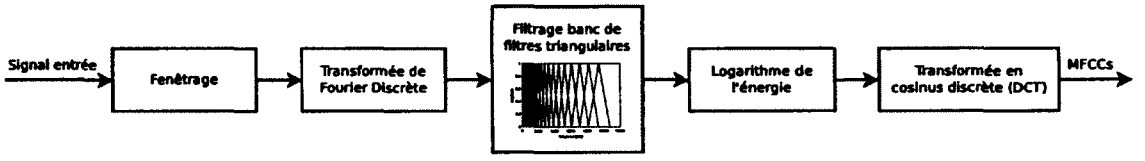


Figure 2.4 Schéma bloc de l'analyse MFCC

Selon [Davis et Mermelstein, 1990], les MFCC donnent de meilleurs résultats pour la reconnaissance de phonèmes que les méthodes traditionnelles basées sur les LFCC (*Linear Frequency Cepstrum Coefficients*), les LPCC(*Linear Prediction Cepstrum Coefficients* ou le LPC (*Linear Predictive Coding* avec distance d'Itakura).

Warped Discrete Cosine Transform Cepstrum (WDCTC)

Plutôt que d'utiliser l'analyse LPC pour extraire l'enveloppe spectrale d'un signal, il est possible d'utiliser des transformées telles que la transformée de Fourier discrète. En extrayant le cepstre du signal à partir de cette transformée, il est aussi possible d'avoir des informations sur l'enveloppe spectrale. En effet, cette information est contenue en grande partie dans les premiers coefficients du cepstre obtenu. Ainsi, en conservant seulement les premiers coefficients, on obtient une version compactée de l'information sur l'enveloppe spectrale (coefficients dénotés LFCC).

La performance de ceux-ci étant inférieure à celle des MFCC pour la reconnaissance de phonème([Davis et Mermelstein, 1990]), on a cherché à l'améliorer. [Muralishankar et Ramakrishnan, 2005] ont adapté la technique à la transformée en cosinus discrète (DCT), obtenant de moins bons résultats que ceux de l'analyse MFCC.

Afin d'améliorer les résultats obtenus, on propose d'utiliser une déformation de l'échelle fréquentielle s'apparentant à l'échelle de Bark [Muralishankar *et al.*, 2005]. L'échelle de Bark est une autre échelle fréquentielle basée sur la perception humaine. La technique est nommée *Warped Discrete Cosine Transform Cepstrum* (WDCTC). La figure 2.5 présente un schéma-bloc haut niveau de cette analyse.

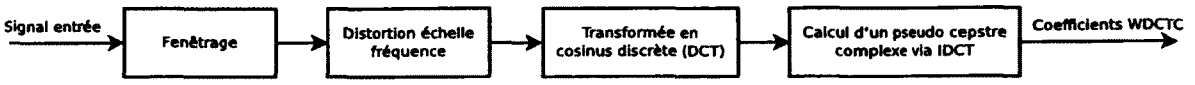


Figure 2.5 Schéma bloc de l'analyse WDCTC

Les résultats obtenus, pour la reconnaissance de phonèmes, sont cette fois-ci légèrement supérieurs aux MFCC, surtout en présence de bruit [Sangwan *et al.*, 2005].

Perceptual Linear Prediction (PLP)

L'analyse *Perceptual Linear Prediction* (PLP) [Hermansky, 1990] est un autre type d'analyse utilisée dans le domaine de la reconnaissance de la parole. Elle combine l'analyse LPC (section 2.1.1), l'utilisation du cepstre et un complexe pré-traitement permettant de rapprocher l'analyse de la perception humaine.

En effet, l'analyse PLP ne s'arrête pas à l'utilisation d'une échelle de Bark ; elle simule également la perception inégale du volume d'un son de même intensité en fonction de sa fréquence (courbe isotonique) et la relation non linéaire entre l'intensité d'un son et le volume perçu. La figure 2.6 montre le schéma-bloc haut niveau d'une analyse PLP.

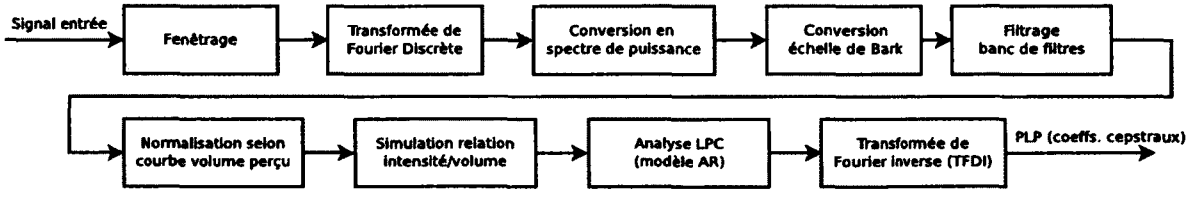


Figure 2.6 Schéma bloc de l'analyse PLP

Les résultats, obtenus, pour la voix, sont nettement supérieurs à ceux de l'analyse LPC traditionnelle [Hermansky, 1990]. Pour le mandarin, ils sont équivalents à ceux des MFCC et même légèrement supérieurs en substituant l'échelle de Bark utilisée dans l'analyse PLP par l'échelle de Mel [Xuan *et al.*, 2002].

2.2 Alignement

2.2.1 Les débuts

L'apparition du domaine du suivi de la partition est due aux travaux de [Dannenberg, 1984] et de [Vercoe, 1984], qui ont été tous deux présentés à l'*International Computer Music Conference* de 1984.

Plus spécifiquement, les travaux de [Vercoe, 1984] portaient sur un système qui permet de générer automatiquement un accompagnement pour un musicien en suivant le tempo de celui-ci. L'instrument étudié était la flûte. Le système développé utilise une partition comme élément de référence et effectue l'alignement en se basant sur le *pitch* seulement. Vercoe utilise des capteurs optiques installés sur les clés de la flûte afin de restreindre le nombre de candidats de *pitch* à 3 et utilise les résultats d'une analyse de *pitch* faite par un processeur audio pour déterminer quel est le «bon» *pitch.*

Aucun détail n'est disponible outre l'extraction du *pitch.* De plus, l'utilisation de capteurs optiques est un grave inconvénient qui rend la méthode impraticable avec la voix chantée.

Les travaux de [Dannenberg, 1984] sont présentés avec beaucoup plus de détails que ceux de Vercoe. Le système développé par celui-ci vise aussi la génération automatique d'accompagnement suivant le tempo d'un musicien. Le *pitch* est aussi le paramètre qui est utilisé pour faire l'alignement ; la référence est une série de *pitch* (aucune information temporelle). L'alignement est fait en ayant recours à la technique de la programmation dynamique qui est une méthode de résolution de problèmes d'optimisation. On fait intervenir une matrice où chaque ligne correspond à un évènement de la partition (une note) et où chaque colonne correspond à une note dans l'interprétation. Chaque élément de la matrice est égal au nombre de notes dont le *pitch* correspond entre la chaîne des n premières notes de la partition et celle des m premières notes de l'interprétation, où n est l'indice représentant la ligne et m l'indice de la colonne considérée. La position dans l'interprétation est donc l'indice de la ligne de la cellule parmis celles de la dernière colonne calculée qui porte la plus grande valeur.

Cette méthode, relativement simple, s'applique uniquement à des signaux monophoniques (qui comportent une seule fréquence fondamentale, par opposition aux accords qui sont polyphoniques) , ce qui ne pose pas de problèmes pour la voix. Cependant, la longueur des notes jouées et des silences ne sont pas pris en compte. Une note jouée qui n'a pas la bonne longueur et qui n'est donc probablement pas celle attendue conduira à une augmentation de la longueur de la meilleure séquence quand même. De plus, la nature continue de la

courbe de *pitch* (représentation *pitch* en fonction du temps) de la voix ne permet pas une telle discrétisation des évènements sonores.

2.2.2 String Matching

Les travaux qui ont suivi ces deux travaux pionniers sont principalement des améliorations de la méthode utilisée par Dannenberg. Plutôt que d'utiliser seulement le *pitch*, on intègre d'autres paramètres, par exemple la durée de la note [Vercoe et Puckette, 1985]. Certains chercheurs tentent aussi de supporter les entrées sonores polyphoniques (composées d'accords)[Bloch et Dannenberg, 1985].

Les travaux de [Vercoe et Puckette, 1985] sont aussi les premiers répertoriés à faire appel à un entraînement. Le calcul de la correspondance n'est plus seulement égal à la longueur de la correspondance entre deux séquence, mais on ajoute des facteurs pénalisants pour les durées non exactes qui sont pondérés par l'écart-type des déviations que les musiciens ayant participé à l'entraînement ont fait par rapport à la durée et au *pitch* des notes.

L'Institut de Recherche et Coordination Acoustique/Musique (IRCAM), un laboratoire de recherche du gouvernement de France, s'est aussi intéressé au problème. Plusieurs suiveurs, c.-a.-d. des algorithmes permettant de déterminer la position courante d'une interprétation par rapport à une référence, ont été implémentés et même utilisés lors de prestations musicales pour déclencher des effets sonores de façon automatique. Un des suiveurs de l'IRCAM est présenté dans [Puckette, 1995]. C'est un suiveur qui opère cette fois-ci sur la voix chantée et qui innove en utilisant deux algorithmes en simultanée, un robuste qui opère avec un certain délai et l'autre qui est rapide, mais moins fiable. L'algorithme rapide est utilisé pour déclencher des effets sonores lorsque la prochaine note est détectée (et qu'elle est associée à un évènement correspondant à un effet). L'algorithme plus lent est celui qui fait le suivi haut niveau, il permet de confirmer que l'avancement qu'a fait l'algorithme rapide est bien correct. De cette manière, on peut obtenir un système qui permet une réaction plus rapide qui est nécessaire pour déclencher des effets.

L'algorithme d'alignement de partition utilisé dans ces travaux, décrits dans [Puckette et Lippe, 1992], est réalisé d'une façon très similaire à celle de [Dannenberg, 1984], avec comme seul paramètre le *pitch*. On conserve toutefois, cette fois-ci, une liste des notes qui ont été sautées. Lorsqu'un évènement sonore (une nouvelle note) se présente dans l'interprétation, on vérifie qu'elle ne figure pas dans la liste des notes sautées. Si c'est le cas, on n'incrémente pas le pointage accordé. Cela permet de rendre le système moins sensible aux inversions de notes.

L'approche est très intéressante puisqu'elle permet d'allouer un délai plus grand tout en maintenant la possibilité de réagir vite ; il est fort probable que la note qui suivra celle qui a été confirmée par l'algorithme lent soit la bonne.

2.2.3 Méthodes stochastiques

Les travaux innovateurs de [Grubb et Dannenberg, 1997] et leur suite [Grubb et Dannenberg, 1998] sont les premiers à avoir considéré les méthodes stochastiques pour faire l'alignement de partition. L'utilisation d'une méthode stochastique implique qu'on tienne compte de l'incertitude inhérente au processus de l'alignement de partition et qu'on calcule la probabilité qu'on se trouve à un endroit donné de la partition plutôt que de simplement estimer qu'on est à un endroit précis. L'algorithme de Grubb, qui opère sur la voix chantée, cherche à calculer la fonction de densité de probabilité de la position actuelle dans la partition. Cette probabilité est calculée à partir de la dernière fonction de densité de probabilité calculée, du tempo (la cadence à laquelle les temps sont joués) estimé, et des observations (mesures des paramètres du signal interprété) qui sont faites sur le signal représentant l'interprétation à l'instant présent. En utilisant l'information sur le tempo, on peut, a priori, sans même observer le signal de l'interprétation, mettre à jour la fonction de densité de probabilité (on peut simplement décaler toute la fonction d'une quantité d'unités qui équivaut au temps écoulé multiplié par le tempo). Les observations sont ensuite utilisées pour corriger la densité de probabilité obtenue. Les régions de fortes probabilités seront bonifiées si les observations correspondent à ce qui est inscrit dans la partition et pénalisées, le cas échéant. Plusieurs observations sont utilisées : le *pitch*, l'enveloppe spectrale (qui contient l'information sur les formants) et la présence ou non d'un début de note.

L'approche utilisée est très intéressante parce qu'elle modélise l'incertitude de l'alignement. De plus, en combinant plusieurs paramètres, Grubb et Dannenberg parviennent à améliorer les résultats qu'ils obtiennent. La technique considère l'influence du temps écoulé sur l'estimation de la position dans la partition, ce qui n'était pas le cas avec les méthodes qui l'ont précédée. Cependant, la méthode requiert de calculer des fonctions de densité de probabilité pour chacun des paramètres à chacune des positions de la partition. Même si plusieurs simplifications peuvent produire des résultats satisfaisants, beaucoup de travail d'alignement manuel est nécessaire pour déterminer ces fonctions de densités de probabilité. De plus, les tests ont été effectués sur des étudiants en chant ayant accompli au moins une année d'études universitaires en chant. Les écarts entre la partition et l'enregistrement risquent d'être beaucoup plus importants avec des chanteurs amateurs.

2.2.4 Modèles de Markov Cachés(HMM)

L'approche stochastique de Grubb et Dannenberg a pavé la voie aux autres méthodes statistiques. Des méthodes basées sur les modèles de Markov cachés, plus communément désignés sous l'acronyme HMM, pour *Hidden Markov Model*, sont apparues avec les travaux de [Raphael, 1999]. Les HMM sont des modèles probabilistes qui permettent de décrire des systèmes qui sont supposés être des processus Markoviens dont l'état interne n'est pas directement mesurable. Un processus Markovien est un processus où la prédiction du futur, étant donné l'état présent, n'est pas dépendante du passé. Évidemment, ceci représente une contrainte gênante ; en utilisant l'information sur le tempo estimé, on pourrait mieux prédire le futur. En dépit de ce désavantage, ceux-ci ont beaucoup été étudiés par les chercheurs oeuvrant dans le domaine du suivi de partition.

Dans un modèle de Markov caché, dont un exemple est représenté à la figure 2.7, à chaque état (les X_i dans la figure 2.7) est associé une probabilité de transition d'état (les a_{ij}) pour chaque état auquel il est possible de passer. À chaque état est également associé une probabilité (les b_{ij}) d'émettre une observation (les y_i). Ces probabilités sont en fait représentées sous forme matricielle, avec une matrice de transition d'états et une matrice d'observations. Avec un tel modèle, on cherche à estimer l'état interne du processus ayant accès seulement aux observations (les y_i). Des algorithmes tels que l'algorithme de Viterbi permettent de déduire l'état le plus probable du système.

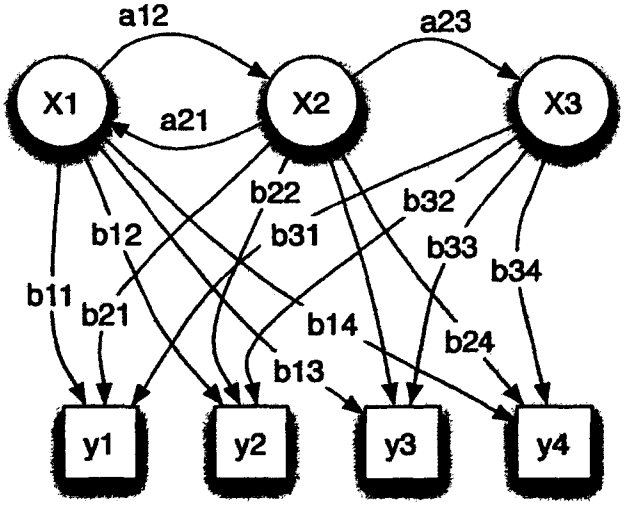


Figure 2.7 Schéma représentant un modèle de Markov caché (Source : Wikipedia)

Dans un contexte de suivi de partition, chaque état représente un évènement (une note, ou une fraction de note). Les probabilités de transitions sont fortement biaisées pour que la transition la plus probable soit celle qui correspond à la prochaine note de la partition mais sans exclure la possibilité de transition vers d'autres états (notes précédentes permises) si une erreur dans l'interprétation survient. Les observations sont les paramètres qui sont extraits de la partition et de l'interprétation (*pitch*, durée de la note, etc.).

Un modèle de Markov caché nécessite de faire une segmentation de la pièce en éléments plus haut niveau (des notes par exemple) et de faire un entraînement avec un algorithme spécialisé permettant de calculer les matrices de transition d'états et d'observations.

L'article de [Raphael, 1999], le premier à utiliser les HMM, propose d'utiliser plus d'un état par note. Ainsi, une note plus longue sera représentée par plus d'un état et cela permettra au système de tenir compte, en quelque sorte, des informations temporelles. Cependant, étant donné que chaque note est représentée par plus d'un état, le modèle devient rapidement gigantesque et il est difficile d'estimer la position en temps réel.

[Cano *et al.*, 1999] utilisent une approche différente. Plutôt que d'utiliser un état par note ou un nombre dépendant du temps de la note, ils utilisent trois états pour chaque note. Les trois états sont l'attaque (le début de la note, où l'énergie croît rapidement), le maintien et la chute (l'énergie décroît, mais la variation est moins brusque que durant l'attaque). Cette séparation est intéressante, puisque les caractéristiques du signal vont changer selon la phase de la note. On va donc pouvoir suivre la partition avec une meilleure précision. Leur système opère également sur la voix chantée en utilisant plusieurs paramètres (observations) :

- le *pitch* et sa dérivée temporelle
- l'énergie et sa dérivée temporelle
- le nombre de passage par zéro par unité de temps

Les auteurs ne se penchent cependant pas sur le problème des erreurs d'interprétations et leur modèle est bien adapté seulement dans le cas où l'interprétation est excellente. Les mêmes auteurs ont développé un autre système [Loscos *et al.*, 1999] pour faire le suivi d'un texte cette fois-ci, au lieu d'une partition. Les états du modèle Markovien sont cette fois-ci chacun des phonèmes (unités de base de la prononciation) du texte qu'on a préalablement analysé. Des états "silence" sont ajoutés entre chacun des phonèmes. Les paramètres utilisés comme observations sont des paramètres qui ont trait à la structure spectrale du signal ; c'est ainsi qu'on peut différencier un phonème d'un autre. Ces paramètres sont les MFCC. En utilisant le texte, l'approche devient plus robuste dans le cas de chanteurs mal entraînés ; il est plus probable que ceux-ci chantent le bon texte qu'il chante chaque note

avec justesse. Les auteurs mentionnent aussi que l'algorithme est à très faible délai qu'ils estiment à 21 millisecondes, pour l'algorithme seul.

L'IRCAM a aussi développé quelques suiveurs basés sur les HMM qui sont brièvement décrits dans [Orio *et al.*, 2003] et [Cont et Schwarz, 2006]. Leurs suiveurs sont adaptés cette fois-ci à des instruments de musique traditionnels monophoniques. Ils innovent en modélisant certains ornements musicaux, tel le trille (alterner rapidement entre deux notes), comme un seul état, plutôt que de le modéliser comme une séquence d'état représentant chacun une note. L'avantage est que lorsqu'un musicien interprète un trille, le nombre d'alternances peut varier. On ne devra donc pas sauter ou répéter certains états dans un modèle qui considère un trille comme un seul état. L'ajout de notes fantômes [Pellegrini et Duée, 2003], qui sont des états facultatifs ajoutés à intervalles réguliers qui sont utilisés lorsque l'interprétation est erronée, permet de mieux récupérer après une erreur de l'interprète.

Au sein de l'IRCAM, [Cont, 2006] a développé un suiveur pour des signaux polyphoniques. La reconnaissance de *pitch* polyphonique est basée sur des techniques de factorisation en matrice non négatives (NNMF) avec des contraintes de parcimonies, ce qui est très innovateur, mais peu pertinent dans le cadre du suivi de la voix. Sans donner beaucoup de détails, par contre, il mentionne qu'il utilise le filtrage particulaire pour estimer la probabilité de se retrouver dans chaque état. C'est une alternative récente et intéressante à l'algorithme de Viterbi qui selon Cont améliore de beaucoup les performances, pour un même temps de calcul.

Bref, les HMM sont des modèles très intéressants pour faire le suivi de partition, tenant compte des incertitudes inhérentes au processus qui sont dues aux déviations dans les différentes interprétations de la même pièce, mais aussi aux incertitudes sur la mesure des paramètres. Cepedandant, un très grand désavantage est qu'il est nécessaire d'entraîner le modèle. Cet entraînement nécessite la segmentation de la pièce en évènements discrets et le calcul des probabilités en créant une base de données à partir de plusieurs interprétations. C'est une procédure qui nécessite beaucoup de travail d'alignement manuel et qui est très lourde. De plus, pour la plupart des techniques mentionnées, l'interprète doit être celui qui effectue l'entraînement, afin que le modèle soit ajusté à sa voix. Finalement, la complexité relativement élevée des algorithmes associés fait en sorte qu'on peut évaluer les probabilités des positions seulement pour une petite fenêtre autour de la position présumée.

2.2.5 Dynamic Time Warping

Les techniques d'alignement de partition basées sur le *Dynamic Time Warping* (DTW) sont très semblables à celles qui étaient inspirées des travaux pionniers de Dannenberg utilisant la programmation dynamique. Dans ces méthodes, on évaluera aussi une matrice pour laquelle chaque cellule contiendra un coût qui représente le coût du meilleur chemin (le chemin détermine l'alignement dans le temps) pour arriver à la cellule (dont la rangée correspond à la position de la note dans la partition et la colonne la position de la note dans l'interprétation). La différence avec l'approche originale de programmation dynamique est qu'on utilise des fonctions de coût qui peuvent être définies de façon à donner les meilleurs résultats plutôt que la longueur de la correspondance entre les deux séquences.

Un système utilisant la méthode est décrit par [Orio et Schwarz, 2001]. Il permet de faire l'alignement même avec des instruments polyphoniques. La fonction de coût dépend de la PSD, pour *Peak Spectral Density*, qui est une mesure de la distance entre le spectre fréquentiel attendu et celui mesuré. Le spectre fréquentiel attendu est calculé en faisant une somme des harmoniques (8 dans ce cas) pour chacun des *pitchs* compris dans l'accord considéré de la partition. La fonction de coût tient également compte de la dérivée de ce paramètre qui permet de bien isoler l'attaque des notes et d'un autre paramètre similaire. La méthode corrige également un des défauts de la DTW, c'est-à-dire un biais pour les chemins diagonaux (ceux-ci indiquent une correspondance 1 pour 1 entre les notes de la partition et celle de la performance, donc un même tempo). Ce biais est dû au fait que le chemin peut aller vers la droite, vers le haut ou en diagonale. En diagonale, on additionne moins de coûts qu'en allant vers la droite ou vers le haut, pour se rendre à la même destination (la distance la plus courte est la ligne droite). Même si ce biais peut sembler favorable (il est plus probable que l'interprétation suive la partition que le contraire), il est trop fort pour que l'algorithme fonctionne bien lors d'écarts entre le tempo de la partition et celle de l'interprétation. On introduit donc des poids pour faire en sorte que les déplacements en diagonale soient plus coûteux.

D'autres travaux [Kaprykowsky et Rodet, 2006] proposent des modifications de l'algorithme de la DTW pour qu'il soit moins exigeant en calcul tout en maintenant les performances à un niveau acceptable. Un temps de calcul plus faible permet de considérer une fenêtre plus grande autour de la position présumée, ce qui rend la méthode plus robuste en cas de sauts rapides.

Un étudiant du Groupe de Recheche sur la Parole et l'Audio (GRPA) de l'Université de Sherbrooke a aussi développé un système d'alignement basé sur la DTW dans le cadre

d'un projet visant à développer un professeur de musique virtuel pour le violon [Gagnon, Brunet et Lefebvre, 2007]. L'algorithme utilise comme paramètres le *pitch*, les instants où sont détectés les débuts de notes et la durée des notes pour faire l'alignement. Plutôt que d'utiliser directement les informations temporelles sur la durée et l'instant de début des notes dans la fonction de coût, il utilise la logique floue pour déterminer si le paramètre est bon, moyen ou mauvais. Cette façon de faire est plus proche de celle qu'un professeur de musique utiliserait, par exemple. Le système utilise une EDTW (*Enhanced Dynamic Time Warping*) qui vise a trouver les meilleurs bouts de chemin plutôt que le meilleur chemin global. De cette façon, l'utilisateur peut répéter une section du morceau plus d'une fois et l'algorithme parviendra à le suivre.

Les techniques utilisant la DTW ont l'avantage de ne pas nécessiter d'entraînement, par rapport aux méthodes basées sur les HMM. De plus, étant donné qu'ils requièrent moins de temps de calcul, on peut considérer une fenêtre plus grande autour de la position présumée, ce qui ajoute à la robustesse des systèmes l'utilisant.

L'algorithme de la DTW est détaillé dans la section 3.1.

2.2.6 Alignement sans partition

Les techniques décrites jusqu'à maintenant partaient toutes d'une référence qui est fournie sous format d'une partition (sauf [Loscos *et al.*, 1999]). Bien qu'on puisse en répertorier beaucoup moins dans la littérature, une autre classe de techniques, partant cette fois-ci d'une référence fournie sous la forme d'un fichier audio, existe.

Toutes les techniques répertoriées utilisent la DTW ou une variante. Plutôt que de segmenter la partition en éléments qui sont des éléments haut niveau comme des notes ou des parties de notes, on procède à un échantillonnage des sources ; on prend un élément à chaque période d'échantillonnage, période qu'on doit déterminer selon la capacité de calcul et la précision requise.

Le système développé par [Dixon, 2005; Dixon et Widmer, 2005] est un système qui permet d'aligner deux séquences audio musicales (peu importe la nature de la musique, on peut même aligner un signal composé de plusieurs instruments polyphoniques en simultané). L'algorithme de la DTW a légèrement été modifé afin de pouvoir être compatible avec une application en ligne, c'est-à-dire qui fonctionne en même temps que l'interprétation est en cours (dans un contexte hors-ligne, on peut faire la localisation en utilisant l'information sur le futur du signal). La complexité de l'algorithme passe de $\mathcal{O}(n^2)$ à $\mathcal{O}(n)$ où n est la longueur des séquences d'éléments considérée. Ceci est possible en mettant à jour la

matrice de coûts seulement aux abords de la cellule qui correspond au meilleur chemin, donc à la position présumée. On perd ainsi une immunité contre les sauts rapides, mais c'est nécessaire étant donné que l'utilisation d'un fichier audio comme référence implique un nombre d'éléments beaucoup plus élevé par rapport à l'utilisation d'une partition où les éléments sont de haut niveau. Les éléments, pour le système de Dixon, sont des vecteurs qui représentent la variation (positive seulement) du spectre fréquentiel du signal. Le précédent paramètre est pris positif seulement, car Dixon estime que le plus important est de faire coïncider les débuts de note, ce qui est plus facilement fait en utilisant une version strictement positive du paramètre.

Une autre approche pour réduire la complexité de l'algorithme est celle de [Muller *et al.*, 2006]. Son approche permet aussi de rendre la complexité de la DTW linéaire par rapport à la longueur des séquences. Cette fois-ci, on fait une itération de l'algorithme à basse résolution (on sous-échantillonne la séquence de paramètres extraits). On utilise le chemin ainsi trouvé comme région où les calculs de coût seront effectués pour l'itération à pleine résolution. On utilise comme paramètre le *chroma* qui est une représentation de l'énergie du signal selon les 12 intervalles musicaux (Do à Si), ce qui permet de bien représenter la musique harmonique (par opposition à mélodique, comme dans le cas de la voix à capella). Cette approche intéressante n'a cependant pas été adaptée pour fonctionner en ligne.

Une autre amélioration de l'algorithme est proposée par [Muller et Appelt, 2008] qui propose de considérer des bouts de chemin optimaux au lieu d'un chemin global optimal comme dans [Gagnon *et al.*, 2007]. Cela permet d'allouer la répétition de certaines sections dans l'interprétation sans que l'algorithme se «perde». L'algorithme n'est cependant pas adapté pour fonctionner en ligne. Dans un cas où la structure est imposée, comme dans un karaoké, cependant, cette amélioration n'est pas très utile.

Bref, les techniques utilisant comme référence un fichier audio utilisent toutes une dérivée de la DTW. Les avantages sont qu'il n'est pas nécessaire de procéder à un entraînement. De plus, on n'est pas forcé de procéder à la transcription de l'oeuvre en partition. Les résultats sont excellents, même si les signaux sont constitués de multiples instruments.

2.2.7 Systèmes de karaoké

Il a été possible de répertorier plusieurs systèmes de karaoké qui visaient aussi à modifier la voix. Les articles de [Inoue *et al.*, 1993, 1994] font état d'un système de karaoké adaptatif (qui modifie la voix). Le système est fait principalement à partir de composants disponibles sur le marché comme un convertisseur audio à MIDI qui s'occupe de faire la segmentation

en notes et un processeur audio de Yamaha qui fait la modification du *pitch*. En plus du *pitch* fourni par le convertisseur audio à MIDI, les auteurs utilisent un détecteur de voyelles pour améliorer leurs résultats. Peu de détails sont donnés sur la façon dont l'alignement est fait, mais on sait que leur système opère avec un délai d'environ 100 millisecondes. Les auteurs estiment que c'est trop pour une telle application. De plus, ils mentionnent que le naturel de la voix n'est pas bien conservé par leur système.

L'autre système répertorié, celui de [Bonada *et al.*, 2000], contourne le problème de l'alignement. Plutôt que de chercher la position de l'interprétation par rapport à la référence, on transforme la voix chantée en unités de transformation. Cette unité représente tous les paramètres nécessaires pour créer une voix synthétique qui se rapproche de celle qui a été analysée (comprends l'information sur le *pitch*, le phonème, la structure spectrale, etc.). En s'aidant d'une analyse phonétique du texte et de la voix de l'utilisateur, on choisit l'unité de transformation la plus ressemblante parmi une base de données créée à partir de l'enregistrement original ou d'un bon imitateur du chanteur original. On utilise ensuite une combinaison des deux unités, celle de l'interprète et celle du chanteur original et on synthétise la voix qui sera plus près de l'originale. On peut donc contrôler à quel point on veut sonner comme l'original. Un inconvénient est qu'on doit entraîner le système avec un imitateur ou en utilisant l'interprétation originale.

2.3 Sommaire

En bref, plusieurs types d'analyses sont possibles pour la reconnaissance de phonèmes, l'élément clé dans la cognition de la parole. Parmi ceux-ci, plusieurs analyses incorporant des notions perceptuelles se distinguent, soit l'analyse PLP, le WDCTC et les MFCC, qui ont des niveaux de performance très similaires pour la voix. Aucune étude ne compare ces paramètres pour la voix chantée.

Aussi, les principaux types d'algorithme utilisés dans la littérature pour le suivi de partitions sont la DTW et les HMM. La DTW est plus simple conceptuellement, requiert moins de temps de calcul (cela permet de considérer une fenêtre de temps plus grande autour de la position présumée) et ne nécessite pas d'entraînement. Les méthodes basées sur les HMM requièrent un entraînement et une segmentation manuelle des pièces. Cet entraînement fait en sorte que l'algorithme est souvent bien adapté seulement pour l'usager l'ayant entraîné. Beaucoup de travail est requis pour effectuer l'entraînement ; on doit effectuer un alignement manuel par paire de signaux utilisés pour l'entraînement.

La DTW a aussi l'avantage d'être applicable à une résolution plus élevée, avec des unités bas-niveau, ce qui permet de l'utiliser avec une séquence audio directement. L'utilisation des HMM avec des états très bas niveau pour une application avec une référence audio n'a pas été étudiée et n'est probablement pas envisageable étant donné la puissance de calcul nécessaire.

CHAPITRE 3

ALGORITHME D'ALIGNEMENT PROPOSÉ

La revue de littérature présentée à la section précédente a permis d'établir les avantages et les inconvénients des deux principales techniques d'alignement répertoriées dans la littérature.

Le problème d'alignement tel que défini dans le présent problème, c'est-à-dire avec une référence audio, présente des particularités qui font en sorte que l'alignement à partir du *Dynamic Time Warping*(DTW) est plus approprié.

D'abord, la nature bas-niveau de la référence, par rapport à une partition, fait en sorte que les vecteurs d'informations qui seront comparés seront échantillonnés à intervalle court. Pour satisfaire les contraintes de faible latence énoncées à la section 1.3, la période d'échantillonnage sera de l'ordre de 20 ms. Un système basé sur les HMMs est beaucoup plus exigeant au niveau du temps de calcul, ce qui fait en sorte que la puissance de calcul nécessaire serait trop élevée. L'entraînement, avec un nombre aussi élevé d'états dans la chaîne de Markov, serait aussi très long.

Pour contrecarrer les problèmes des HMMs avec une référence audio, il serait possible de discrétiser le signal audio de référence en une série d'évènements haut niveau. Par exemple, il serait possible de détecter les débuts et la fin des «phonèmes» chantés et utiliser chaque phonème comme un état de la chaîne de Markov, ce qui permettrait de réduire considérablement le nombre d'états nécessaires.

Cependant, la discrétisation de la référence en une pseudopartition est une opération complexe et sujette à l'erreur. De plus, le problème de la transcription automatique d'un signal audio est un problème auquel s'attardent beaucoup de chercheurs et il n'existe pas de solution vraiment efficace à ce jour. Les erreurs de transcription, dans un système d'alignement, auraient des répercussions sur le résultat du système en entier et il est donc préférable d'éviter cette opération.

Additionnellement, il serait nécessaire de faire une segmentation manuelle de la référence afin d'isoler les évènements haut niveau qui feront partie de la chaîne de Markov. La segmentation manuelle est une opération qui requiert une quantité importante de temps, beaucoup de minutie et une bonne connaissance de la phonétique. Il devient alors très

fastidieux d'ajouter une chanson au répertoire du système de karaoké, ce qui n'est pas désirable.

La DTW, elle, ne requiert aucun entraînement. De plus, la complexité de l'algorithme est compatible avec des vecteurs de comparaison échantillonnés avec une courte période. La faisabilité d'un alignement basé sur la DTW avec des vecteurs bas niveau peu sous-échantillonnés a été démontrée par plusieurs travaux [Dixon, 2005; Dixon et Widmer, 2005; Muller et Appelt, 2008; Muller *et al.*, 2006]. Ce sera donc l'approche utilisée pour réaliser l'alignement.

3.1 *Dynamic Time Warping*

La DTW est une technique qui permet de comparer deux séquences de nature quelconque, que ce soit une séquence de gènes, une séquence de notes d'une partition ou une séquence de valeurs d'amplitude d'un signal, par exemple, tout en permettant que les deux séquences soient déformées l'une par rapport à l'autre.

La technique est une forme de programmation dynamique, c'est-à-dire qu'elle cherche la solution optimale d'un problème en se basant sur des sous-problèmes qui sont eux-mêmes résolus de cette façon. À mesure que les sous-problèmes sont résolus, on s'approche de la solution du problème complet.

La DTW implique la construction d'une matrice de coûts cumulés. Chaque rangée de la matrice correspond à un indice de la séquence de référence et chaque colonne correspond à un indice de l'autre séquence, qui sera désignée sous le nom de séquence de test. Chaque cellule représente le coût cumulé minimal nécessaire pour partir du début des séquences et se rendre au point indiqué par la position de la cellule. Cette matrice est calculée itérativement (approche par sous-problèmes) à partir du coût instantané correspondant à la cellule calculée et des coûts cumulés des cellules adjacentes (généralement trois cellules). La figure 3.1 illustre ceci.

Le coût de la cellule calculée, $C[i,j]$ est donné par

$$\begin{aligned} C[i,j] &= c[i,j] + C[k',l'] \\ (k',l') &= \underset{(k,l)\in\{(i-1,j),(i-1,j-1),(i,j-1)\}}{\arg\min} C[k,l] \end{aligned} \tag{3.1}$$

où $c[i,j]$ est le coût instantané calculé à la position (i,j), i et j étant respectivement les positions dans la séquence de référence et de test. Le coût instantanné est une mesure qui

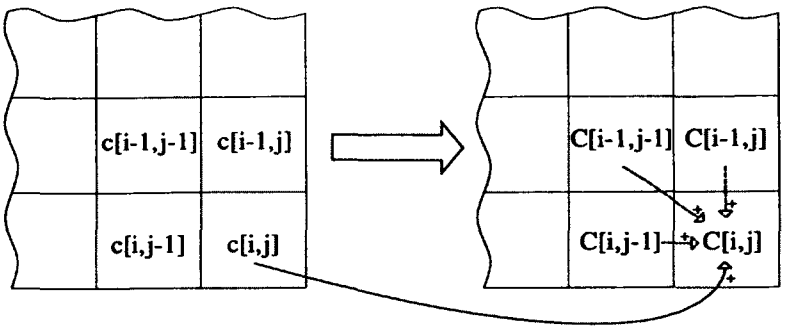


Figure 3.1 Illustration du calcul du coût cumulé à partir des cellules adjacentes et du coût instantané

exprime la distance entre les éléments désignés des deux séquences. Son calcul est propre à chaque problème étudié. À titre d'exemple, on peut utiliser la distance spectrale entre deux trames, dans le cas d'un signal audio ou encore un écart quadratique dans le cas de séquences de nombres, comme à l'équation 3.2.

Une fois la matrice complétée, la cellule finale, d'indice M, N où M est la longueur de la séquence de référence et N est la longueur de la séquence de test, contient une valeur qui est inversement proportionnelle à la «ressemblance» des deux séquences malgré leurs déformations. Avec deux séquences qui diffèrent seulement par le fait que certains éléments sont répétés dans une séquence mais pas dans l'autre, le résultat serait exactement zéro, l'étirement et la compression d'une séquence n'ayant aucun effet sur le résultat de la DTW.

Il est également possible d'effectuer l'alignement de deux séquences grâce à une approche par sous-problèmes. En effet, en partant de la cellule finale, il est possible de reculer d'une cellule à la fois en choisissant la cellule adjacente parmi les trois cellules voisines précédentes qui a le plus faible coût cumulé. Le chemin emprunté permettant d'arriver à la cellule d'origine donne l'alignement entre les deux séquences (la séquence d'indices des cellules du chemin de retour). Un exemple utilisant les séquences de valeurs arbitraires de la figure 3.2 est présenté à la figure 3.3.

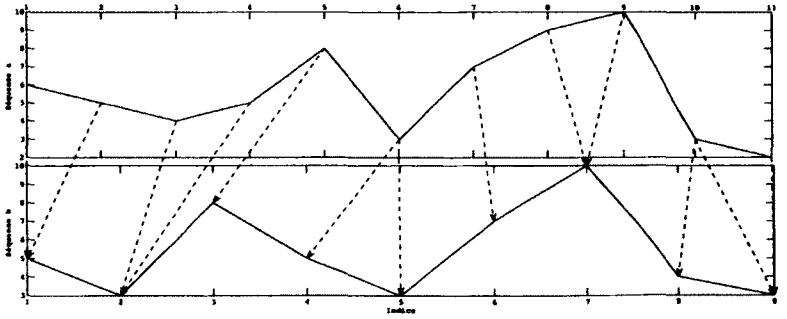


Figure 3.2 Séquences de valeurs numériques arbitraires alignées par DTW

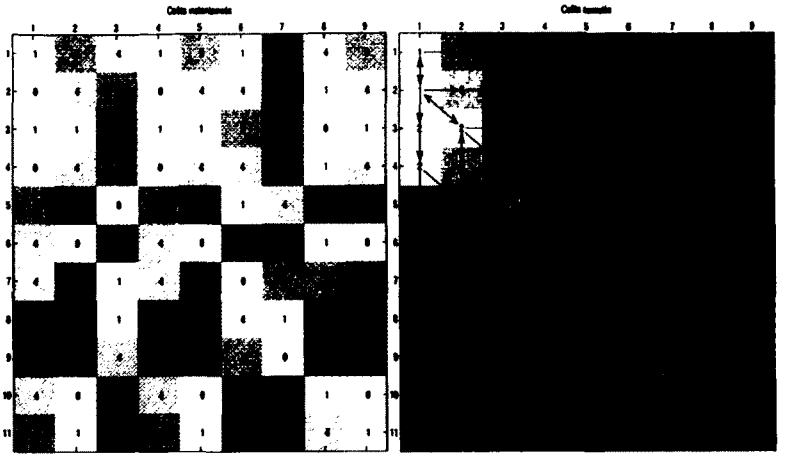


Figure 3.3 Exemple de DTW simple sur une séquence de valeurs numériques

À la figure 3.3, les flèches pointant dans les directions des indices croissants indiquent à partir de quel coût cumulatif la cellule pointée à été calculée. Les autres flèches montrent le chemin de retour reliant la cellule finale et la cellule d'origine. L'alignement des deux séquences est donné par la séquence des indices des cellules du chemin de retour.

Le coût instantanné est calculé en prenant l'écart quadratique entre les valeurs des deux séquences, c'est-à-dire, dans le cas de séquences de nombres,

$$c[i,j] = (a[i] - b[j])^2 \tag{3.2}$$

où a et b sont les séquences à aligner.

À la figure 3.3, on peut remarquer que les directions horizontales et verticales sont priorisées par rapport à la direction diagonale, par exemple, à la cellule d'indice (11,9). Ceci est désirable afin de ne pas pénaliser l'étirement (ou la contraction) d'une séquence par rapport à l'autre. En effet, puisque la diagonale permet de se rapprocher deux fois plus vite de la cellule finale, par rapport aux directions horizontales et verticales, deux fois moins de coûts instantanés seront additionnés à la valeur de la cellule calculée. Ainsi, il y a un biais systématique vers la diagonale. Pour corriger ceci, plusieurs implémentations dont [Dixon, 2005; Muller *et al.*, 2006] réduisent ou éliminent le biais complètement en pondérant le coût instantané selon la direction du déplacement. L'équation 3.1 devient ainsi

$$C[i,j] = \min \begin{cases} w_v c[i,j] + C[i-1,j] \\ w_h c[i,j] + C[i,j-1] \\ w_d c[i,j] + C[i-1,j-1] \end{cases} \tag{3.3}$$

où w_v, w_h et w_d sont respectivement les facteurs de pondération pour les directions verticale, horizontale et diagonale. Pour éliminer le biais complètement, il est nécessaire de fixer $w_d = 2, w_v = 1$ et $w_h = 1$.

Les déplacements possibles sur la matrice de coûts cumulés fixent des contraintes locales sur l'alignement obtenu. Avec les déplacements considérés jusqu'ici, ils imposent que le chemin ou l'alignement soit continu, strictement croissant et borné aux extrémités des séquences. D'autres types de contraintes locales [Juang et Rabiner, 1993], dont certaines sont montrées à la figure 3.4, peuvent être utilisées pour ajouter et/ou changer ces contraintes.

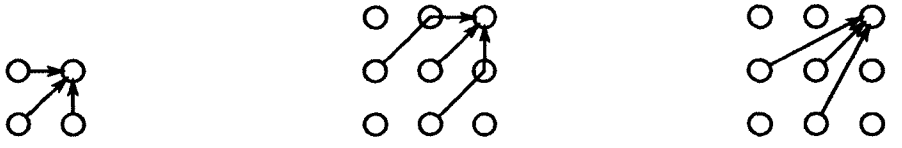


Figure 3.4 Exemple de contraintes locales pour la DTW

La figure 3.4 montre,

1. en 3.4a, la contrainte locale «normale» utilisée jusqu'ici,
2. en 3.4b, une contrainte locale bornant la pente de l'alignement (pente bornée entre 0.5 et 2),
3. en 3.4c, une contrainte locale bornant aussi la pente, mais n'imposant plus la continuité du chemin.

3.2 Adaptation temps réel

Le problème qui est l'objet de ce mémoire nécessite évidemment que l'algorithme fonctionne en temps réel et en ligne. En ligne signifie que l'algorithme s'exécute à chaque nouvelle trame traitée afin de mettre à jour l'alignement au fur et à mesure que l'interprétation progresse. La contrainte de temps réel impose que chaque itération de l'algorithme doive être complètement exécutée dans un intervalle de temps égal à la durée d'une trame du système.

Pour répondre à ces critères, deux options majeures sont à considérer :

1. adapter l'algorithme pour que sa complexité algorithmique devienne constante,
2. limiter la durée des signaux à aligner de façon à ce que la durée maximale de traitement soit suffisamment courte.

L'approche 1 est celle qui a été choisie, n'ayant pas l'inconvénient gênant de limiter la longueur des chansons supportées par le système de karaoké.

La complexité de l'algorithme «standard» de la DTW est calculée et présentée dans le tableau 3.1. Il s'agit de la version hors-ligne de l'algorithme. T_E, T_C, T_A et T sont respectivement le temps de calcul nécessaire pour extraire les paramètres qui composeront les séquences, pour mettre à jour la matrice de coût cumulatif, pour en déduire l'alignement et le temps total nécessaire. T_F est le temps nécessaire pour extraire le vecteur de paramètres qui constituera un élément des séquences. T_D est le temps nécessaire pour calculer un élément de la matrice de coût cumulé. n et m sont les indices courants de la séquence de référence et de test, respectivement, dont N et M sont les longueurs respectives.

Tableau 3.1 Complexité algorithmique de l'algorithme de la DTW standard

A - Extraction paramètres	$T_E(n,m) = (n+m)T_F, T_F \in \mathcal{O}(1)$ $T_E(n,m) \in (n+m)\mathcal{O}(1)$ $T_E(n,m) \in \mathcal{O}(n+m)$
B - Calculs coûts	$T_C(n,m) = nmT_D, T_D \in \mathcal{O}(1)$ $T_C(n,m) \in \mathcal{O}(nm)$
C - Calcul alignement	$T_A(n,m) \in \mathcal{O}(m+n)$ (cas limite)
Total	$T(n,m) \in \mathcal{O}(n+m) + \mathcal{O}(nm) + \mathcal{O}(n+m) \in \mathcal{O}(nm)$

Dans le contexte du problème étudié, il est possible de prétraiter une des deux séquences, celle qui correspond à la référence, en extrayant les paramètres au préalable. On peut aussi utiliser l'algorithme de façon itérative en calculant les coûts à chaque nouvelle trame de la séquence de l'interprétation. Le tableau 3.2 présente la complexité algorithmique alors obtenue.

Tableau 3.2 Complexité algorithmique de la DTW «standard» adaptée pour utilisation en ligne

A - Extraction paramètres	$T_E(m) = NT_F, T_F \in \mathcal{O}(N)$ $T_E(m) \in \mathcal{O}(N)$
B - Calculs coûts	$T_C(m) = NT_D, T_D \in \mathcal{O}(N)$ $T_C(m) \in \mathcal{O}(N)$
C - Calcul alignement	$T_A(m) \in \mathcal{O}(N) + \mathcal{O}(N+m) \in \mathcal{O}(N+m)$ (cas limite)
Total	$T \in \mathcal{O}(N) + \mathcal{O}(N) + \mathcal{O}(N+m) \in \mathcal{O}(N+m)$

Le calcul de l'alignement, dans ce cas, nécessite une étape de plus que dans la version hors-ligne. En effet, l'alignement calculé ne peut plus être borné à l'origine des deux séquences et à la fin des deux séquences puisqu'une des deux séquences n'est pas complète. Pour remédier à ce problème, on peut tout simplement prendre comme borne finale la cellule

de plus faible coût parmi celles qui correspondent à l'indice courant de la séquence de test (l'interprétation, dans le contexte du problème étudié) :

$$n'[m] := \underset{n \in \mathbf{N}, 0<n<N-1}{\arg\min} C[n, m]. \tag{3.4}$$

Cette adaptation, bien qu'ayant une complexité plus intéressante, présente encore une complexité non constante. Elle est cette fois-ci linéaire avec la longueur de la séquence de référence et de test, ce qui implique une limite indésirable dans la longueur des séquences.

3.2.1 Adaptation de Dixon

Une des adaptations temps réel proposées dans la littérature est caractérisée par une complexité algorithmique constante. Il s'agit des travaux de [Dixon, 2005; Dixon et Widmer, 2005] qui utilisent une technique intéressante pour sélectionner les cellules de la matrice de coût cumulé à calculer.

La technique consiste à déterminer, à chaque itération de l'algorithme, si on doit calculer une nouvelle colonne et/ou une nouvelle rangée. Ce choix est effectué en observant où est la cellule de plus faible coût par rapport à la dernière colonne et à la dernière rangée calculée.

Si la cellule se situe dans la dernière rangée, on calcule une nouvelle rangée, si elle est dans la dernière colonne, on calcule une nouvelle colonne. Si la cellule se trouve à la fois dans la dernière rangée et la dernière colonne, on calcule une nouvelle rangée et une nouvelle colonne. On tente ainsi de diriger la zone de recherche vers le meilleur résultat obtenu sur la bordure de la matrice, celui-ci indiquant en quelque sorte le meilleur alignement possible, pour la position courante soit de l'interprétation (colonne) ou de la référence (rangée).

Les rangées et colonnes calculées ont une taille fixe. Le nombre d'éléments calculés pour chaque rangée ou colonne ajoutée est fixé par une constante, choisie de façon à ce que le temps de calcul maximal pour une itération ne dépasse pas la durée d'une trame. Pour un calcul d'une nouvelle colonne, par exemple, on calcule un nombre fixe d'éléments à partir de la rangée d'indice maximal calculée en allant vers le début de la séquence.

Si une nouvelle colonne doit être calculée, l'algorithme attends la prochaine itération (afin d'avoir les données de la prochaine trame). Afin de maintenir la complexité de l'algorithme constante, il est nécessaire de limiter le nombre de rangées calculés consécutivement. Un

seuil est défini et lorsqu'il est excédé, on force le calcul d'une nouvelle colonne (et donc on passe à l'itération suivante).

L'alignement est défini comme la séquence des meilleures cellules sur la bordure de la zone de recherche, à chaque pas de l'algorithme. La figure 3.5 illustre le fonctionnement de l'algorithme. Les chiffres dans les cellules indiquent le coût cumulé et les meilleures cellules, pour tous les pas, y sont représentées en gris.

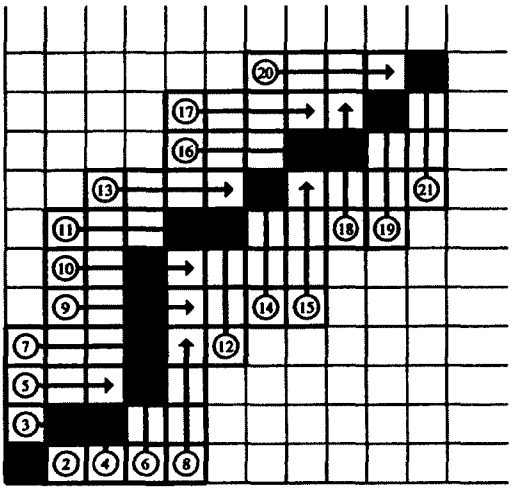
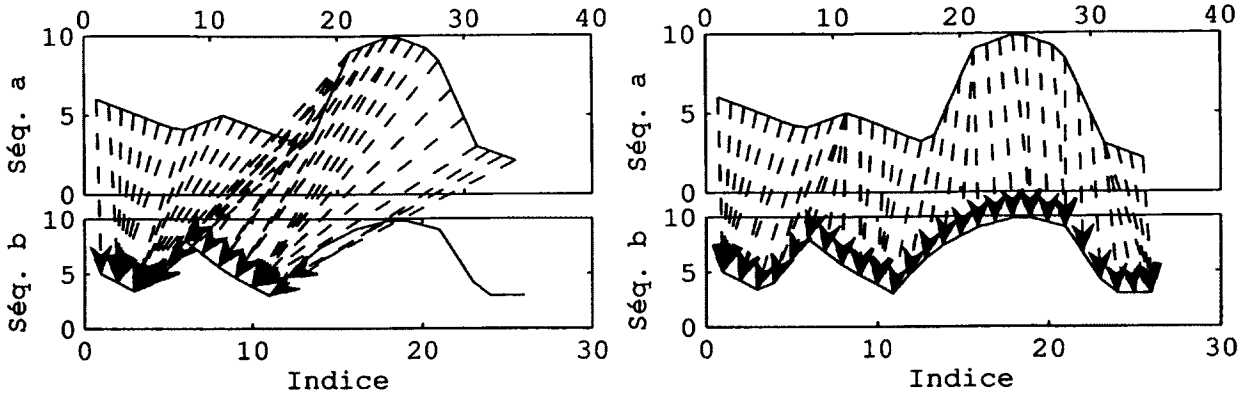


Figure 3.5 Exemple de matrice de coût illustrant l'adaptation de la DTW par Dixon. Source : [Dixon, 2005].

L'intérêt principal de cette adaptation de la DTW est que la fenêtre où la matrice de coût est calculée est ajustée de façon dynamique autour de la position courante estimée, ce qui n'impose aucune contrainte sur la déformation des séquences, contrairement aux techniques qui bornent de façon statique le domaine sur lequel est calculé la matrice de coûts cumulés, décrites dans [Juang et Rabiner, 1993].

L'adaptation de Dixon montre certaines lacunes par rapport à une DTW usuelle. L'estimation en ligne de la position à chaque itération pose problème puisqu'elle peut entraîner l'algorithme dans un chemin peu coûteux à court terme, mais qui devient sous-optimal par la suite. L'exemple présenté aux figures 3.6 et 3.7 illustre ceci. À partir de la position d'indice 25 dans la séquence «a», les coûts cumulés dans la zone considérée par l'algorithme de Dixon sont supérieurs à ceux du chemin trouvé par une DTW sans contraintes.

À la figure 3.7, les cellules marquées d'un cercle indiquent l'alignement obtenu par l'algorithme de Dixon et celles marquées d'un losange indiquent celui qui a été obtenu par une DTW sans modifications. Les cellules grisées sont celles qui ont font partie du domaine de recherche limité de l'algorithme de Dixon. La taille des expansions est de 6 cellules.



(a) DTW avec adaptation de Dixon (b) DTW normale

Figure 3.6 Exemple de résultat d'une DTW illustrant la problématique *cul-de-sac*

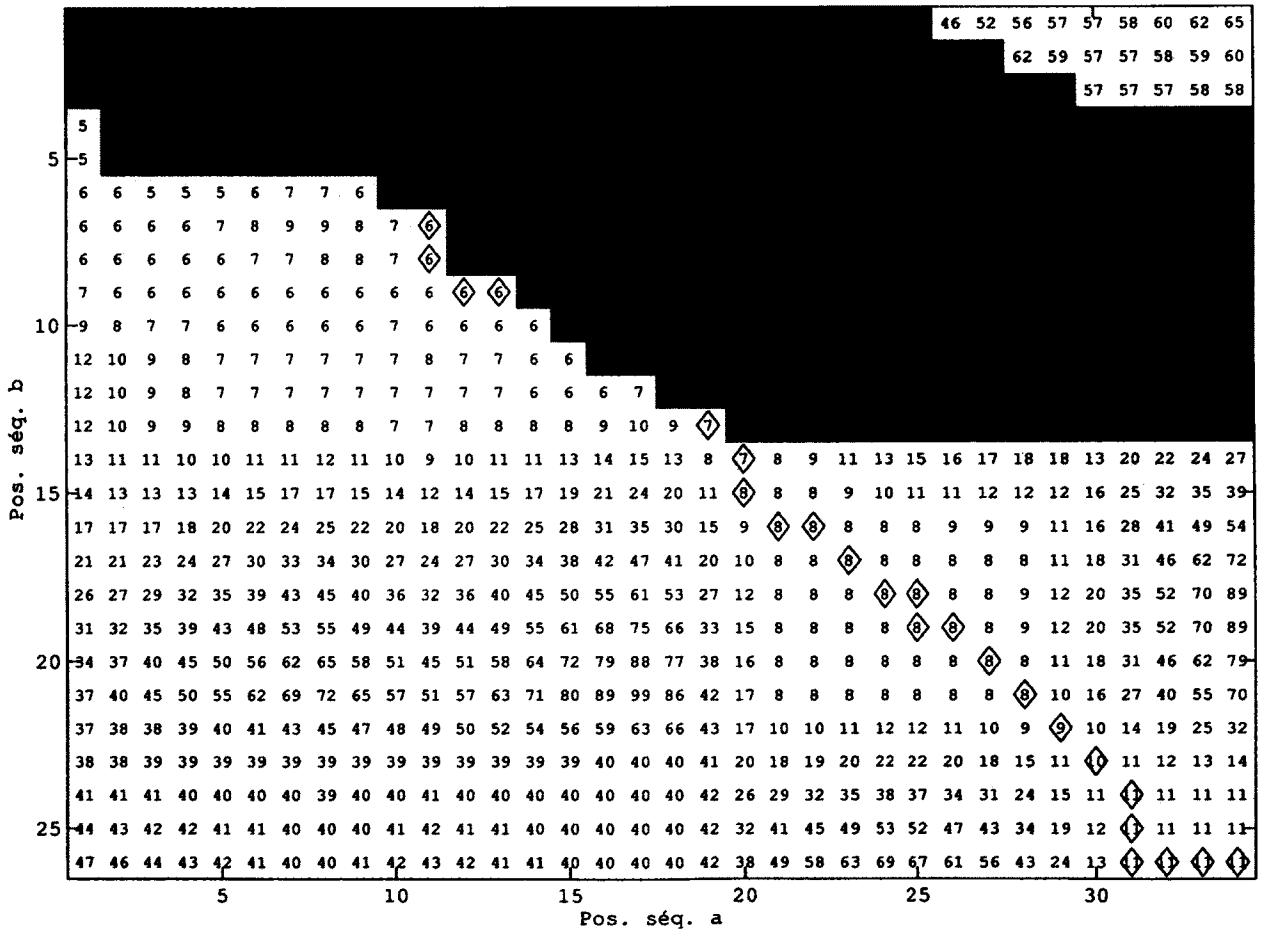


Figure 3.7 Matrice de coûts cumulés pour l'exemple de la figure 3.6

La figure 3.7 montre que l'alignement obtenu par l'algorithme de Dixon ne passe pas par les cellules de plus faible coût puisque l'alignement est défini comme étant la séquence des cellules d'intersection utilisées. Il serait possible de corriger ceci en utilisant la cellule de plus faible coût correspondant à la position courante dans la séquence «a» à chaque fin d'itération. Cela permettrait cependant un alignement discontinu.

Puisque l'algorithme de Dixon considère seulement les cellules qui précèdent la cellule d'intersection ou qui sont dans la colonne ou rangée suivante, l'algorithme ne peut pas faire le «saut» vers le chemin «optimal» dans un court délai. Afin que l'algorithme retrouve le chemin optimal dans le cas présenté aux figures 3.6 et 3.7, il est nécessaire d'allonger les séquences et d'agrandir la taille de la zone de recherche. La figure 3.8 montre le résultat de l'algorithme en utilisant une version allongée de la séquence de la figure 3.6, pour différentes tailles d'expansion. Notez que les séquences ont été allongées avec des valeurs presque identiques d'une séquence à l'autre, mais très différentes du début des séquences, afin de faciliter le retour vers le chemin optimal (coût élevé pour continuer dans le mauvais chemin et coût très faible dans le chemin optimal).

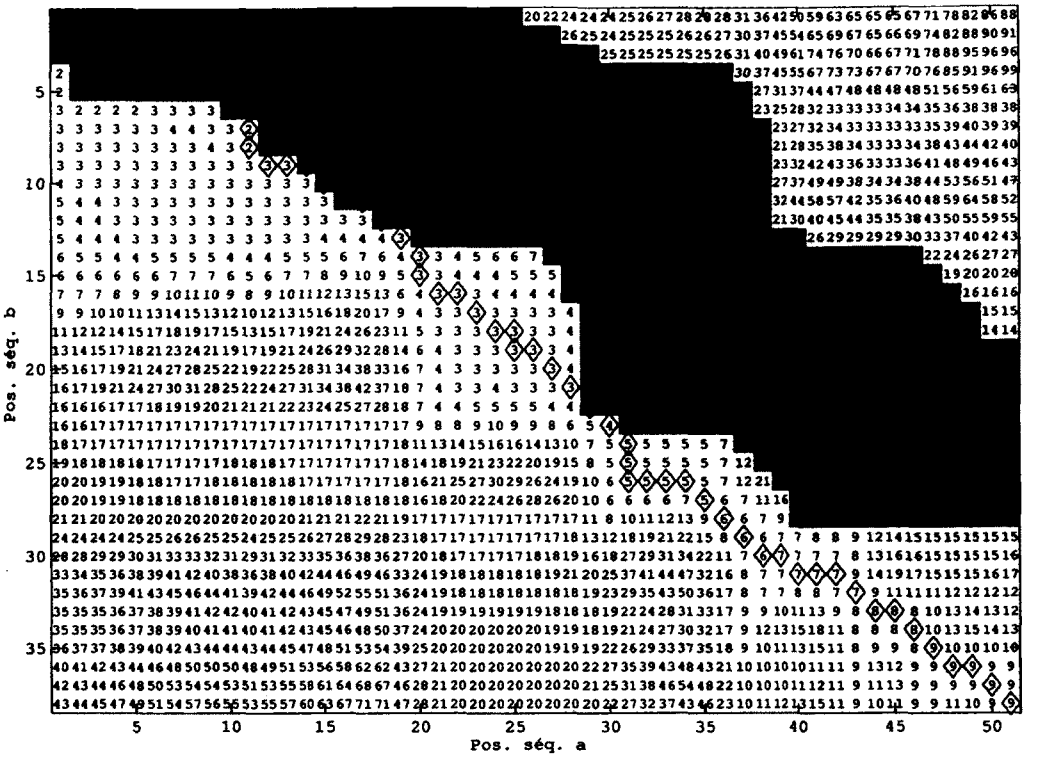
3.2.2 Algorithme proposé

Les lacunes de l'algorithme de Dixon qui ont été identifiées sont gênantes, dans le cadre d'un système de karaoké. La difficulté et la latence que l'algorithme de Dixon lorsqu'il y a des sauts dans l'interprétation ou que l'algorithme entre dans un *cul-de-sac* fait en sorte que l'erreur sur l'alignement peut alors être assez grande. Étant donné le caractère continu de la voix chantée et le fait que ses caractéristiques peuvent changer rapidement, il est nécessaire d'atteindre une latence d'opération faible. Une adaptation temps-réel qui constitue une alternative à l'approche de Dixon est donc proposée dans cette section.

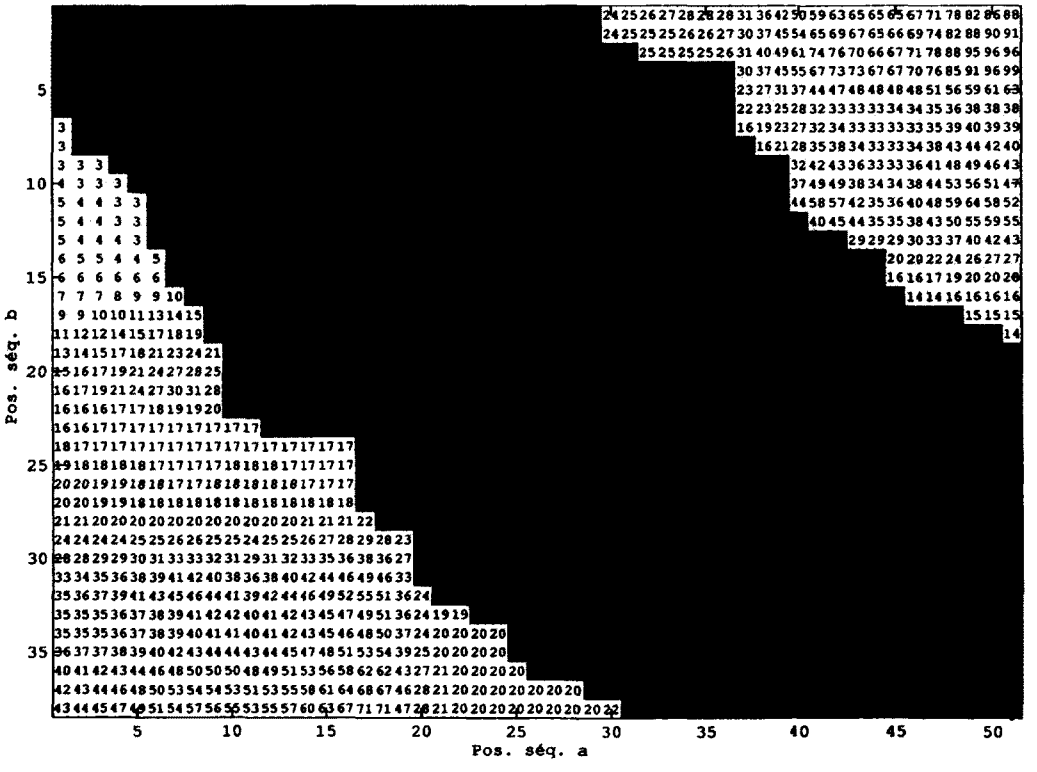
La contrainte de temps réel exigeant une complexité d'ordre constante ($\mathcal{O}(1)$), il est donc requis qu'un nombre fixe de cellules soient calculées pour chaque itération, c'est-à-dire pour chaque échantillon de la séquence d'interprétation.

Proposition A : algorithme *taille fixe*

Plutôt que de calculer des colonnes et/ou des rangées, l'algorithme proposé calcule une nouvelle colonne bornée de la matrice de coût cumulé à chaque itération. Le domaine sur lequel cette colonne est calculée est de taille constante et réparti également autour de la



(a) $L_E = 10$



(b) $L_E = 20$

Figure 3.8 Alignement obtenu pour différentes tailles d'expansion, pour l'exemple de la figure 3.6 allongé

dernière position estimée. La matrice de coût cumulée est donc égale à

$$C[i,j] = \begin{cases} \min \begin{cases} w_v c[i,j] + C[i-1,j] \\ w_h c[i,j] + C[i,j-1] \\ w_d c[i,j] + C[i-1,j-1] \end{cases} & \text{, pour } A[j-1] - \lfloor L_E/2 \rfloor < i < A[j-1] + \lfloor L_E/2 \rfloor \\ \infty & \text{, sinon} \end{cases} \tag{3.5}$$

$$A[j] = \arg\min_x C[x, j-1].$$

où L_E est la taille des expansions de la matrice de coûts (nombre de cellules calculées dans chaque nouvelle colonne). L'expression de la matrice de coût cumulée donnée à l'équation 3.5 est seulement valide pour une valeur de taille d'expansion paire.

$A[j]$ représente l'alignement, c'est-à-dire la position estimée de la séquence de référence (séquence «b» dans l'exemple de la figure 3.6) à la position j de la séquence de test (séquence «a»). Puisqu'un nombre fixe de cellules de la matrice de coût cumulé est calculé pour chaque position de la séquence de test, l'alignement est calculé en recherchant la valeur minimale parmi un nombre fixe de coûts cumulés ce qui fait en sorte que la complexité globale de l'algorithme est constante ($\mathcal{O}(1)$).

La figure 3.9 montre le résultat obtenu en utilisant l'algorithme *taille fixe*, pour les séquences de l'exemple *cul-de-sac* (figure 3.6). L'alignement obtenu avec l'algorithme usuel de la DTW est cette fois-ci représenté par les cellules marquées d'un losange. L'alignement obtenu avec l'algorithme *taille fixe* est beaucoup plus près de l'alignement idéal qu'avec l'algorithme de Dixon et ce, avec une faible taille d'expansion L_E.

L'algorithme a cependant une lacune majeure. En utilisant deux nouvelles séquences pour former un nouveau problème qui sera désigné *saut*, cette lacune apparaît plus évidente. La figure 3.10 montre les séquences en question ainsi que l'alignement obtenu avec l'algorithme *taille fixe*. Il est important de noter que les séquences sont très différentes au début, ce qui explique la performance médiocre obtenue, le but étant d'illustrer une autre lacune.

On remarque, à la figure 3.11 que certaines cellules qui auraient dû être calculées n'ont pas été calculées, ou plus précisément ont généré un coût cumulé infini. Ces cellules sont marquées d'un «X» à la figure 3.11b. Elles ont généré des valeurs infinies parce que les trois cellules voisines considérées à l'équation 3.1 (illustrées à la figure 3.4) ont également toutes une valeur infinie. La zone effective sur laquelle la recherche du meilleur alignement est faite s'en trouve réduite, ce qui a un impact sur la performance globale de l'algorithme.

De plus, à la figure 3.11b, l'écart entre la matrice de coûts cumulés obtenue et celle qui est obtenue si on applique la DTW usuelle est calculé pour chacune des cellules de la

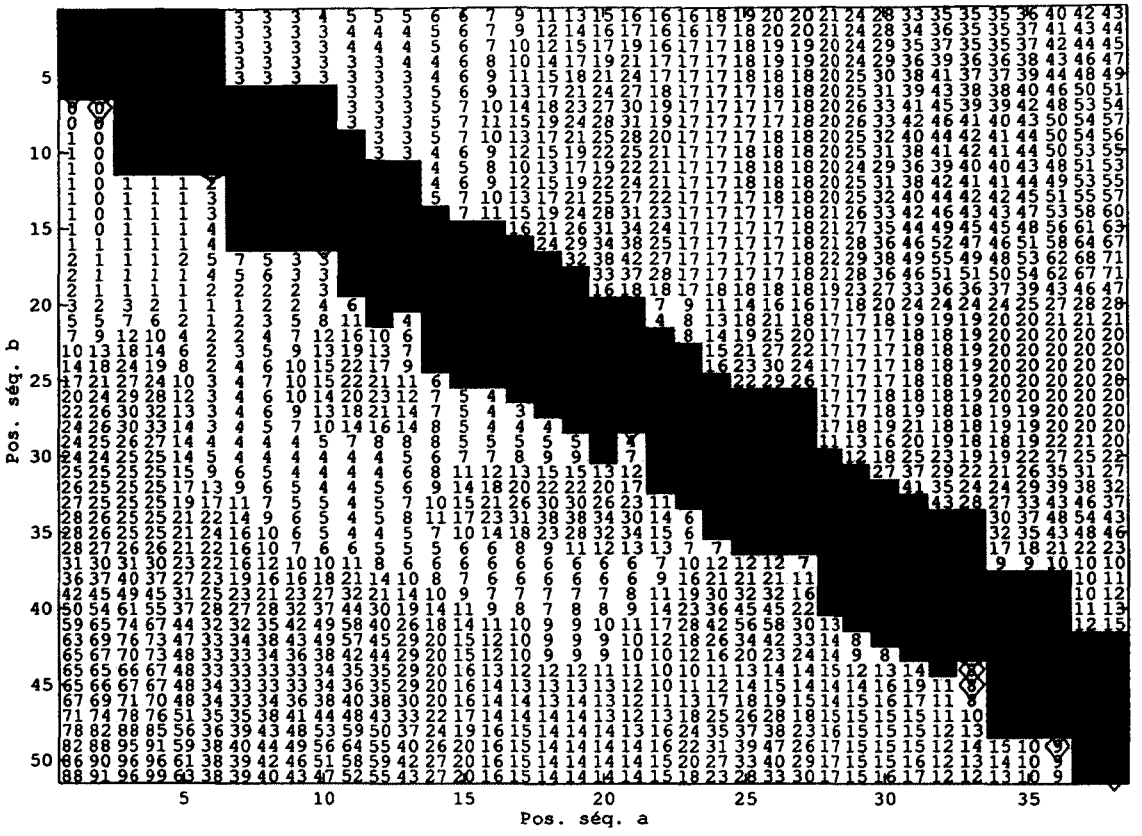


Figure 3.9 Matrice de coûts cumulés obtenue pour la problématique *cul-de-sac* : algorithme *taille fixe*

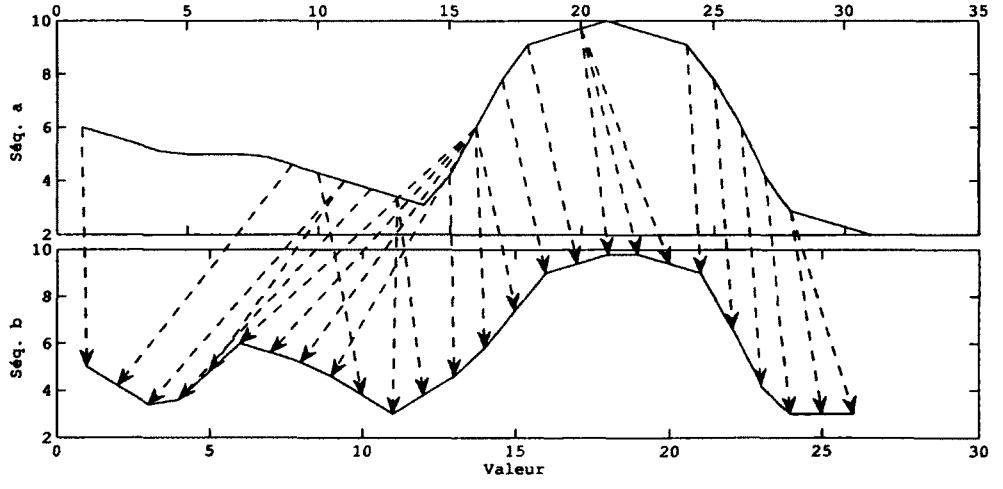
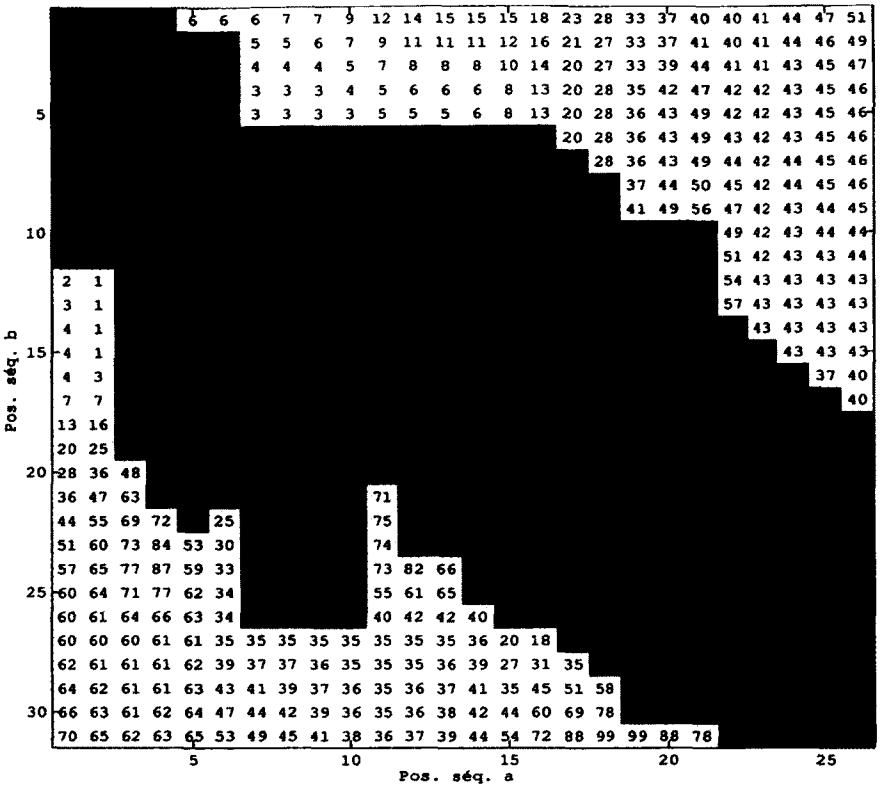
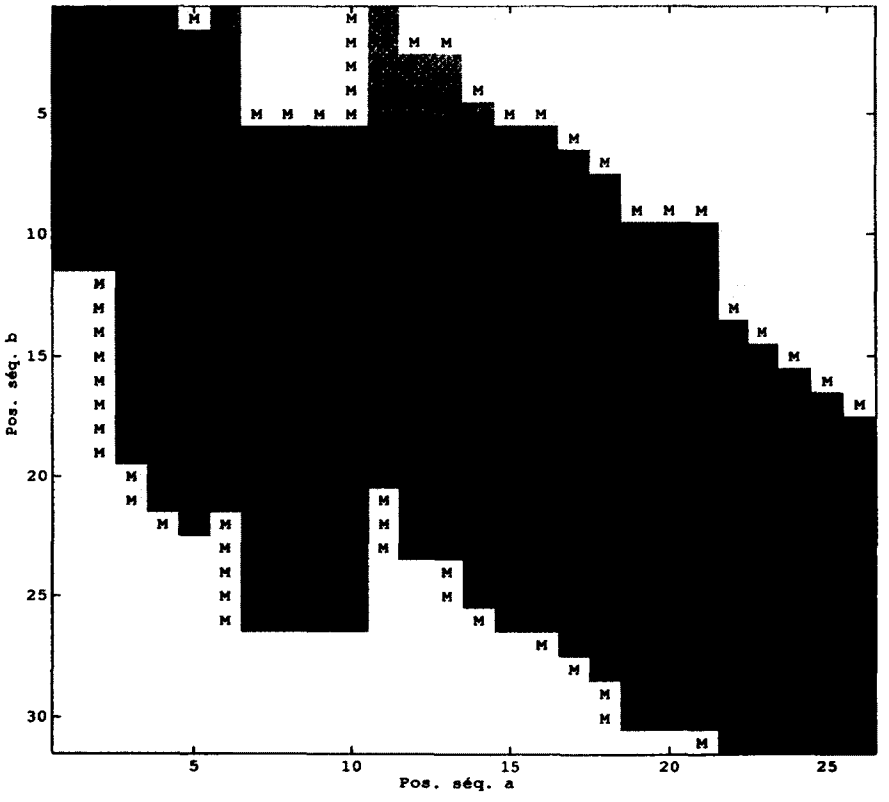


Figure 3.10 Séquences de la problématique *saut* et alignement obtenu pour l'algorithme «B»



(a) Matrice de coûts cumulés et alignement obtenus



(b) Différence entre la matrice obtenue et la matrice qui aurait été obtenue sans contraintes

Figure 3.11 Résultats obtenus pour la problématique *saut* : algorithme *taille fixe*

matrice. Cet écart est causé par les cellules de l'équation 3.1 dont les valeurs ont été considérées comme infinies puisqu'elles n'avaient pas été calculées. Ces cellules dont le coût est manquant sont indiquées par un «M» si elles ne font pas partie de la région de recherche considérée par l'algorithme et d'un «X» si elles en font partie.

On peut observer que les écarts et cellules manquantes dans la zone de recherche (marquées d'un «X») apparaissent lorsque la cellule centrale autour de laquelle la zone de recherche est définie fait un saut suivi d'un retour en arrière. En effet, un saut, comme à la colonne 7 de la figure 3.11 fait en sorte que les cellules des rangées d'indice faible (2 à 5, dans ce cas) ne sont plus calculées. Avec le retour en arrière à la colonne 11, il serait nécessaire d'avoir les valeurs des cellules de la colonne 10 pour les rangées de 1 à 20, afin de pouvoir calculer toutes les cellules de la zone de recherche. Étant donné que celles-ci n'ont pas été calculées pour les rangées 1 à 5, il est impossible de calculer plusieurs cellules de la zone de recherche. Tout retour en arrière provoquera un tel manque, mais le précédent saut exacerbe le problème.

Une modification de l'algorithme permettant de mitiger le problème des cellules de valeur infinie et d'écart de la matrice de coûts cumulés est proposée dans la prochaine sous-section.

Proposition B : algorithme *taille fixe à déplacement proportionnel*

L'algorithme *taille fixe à déplacement proportionnel* est une simple modification à l'algorithme *taille fixe*. Afin de limiter la fréquence des sauts instantanés dans la position des colonnes de la matrice à calculer, dans les deux directions, la nouvelle position de la cellule centrale est déterminée par l'écart entre la position centrale et la position estimée dans les colonnes précédentes. Plus l'écart est élevé dans les colonnes précédentes, plus la zone de recherche se déplace vers les rangées d'indices supérieurs. Aussi, puisque le retour en arrière risque d'entraîner une réduction effective de la zone de recherche, un écart négatif n'entraîne pas un retour en arrière ; la nouvelle cellule centrale est prise dans la même rangée que la précédente.

L'équation 3.5 devient

$$
\begin{aligned}
C[i,j] &= \begin{cases} \min \begin{cases} w_v c[i,j] + C[i-1,j] \\ w_h c[i,j] + C[i,j-1] \\ w_d c[i,j] + C[i-1,j-1] \end{cases} & \text{, pour } P[j-1] - \lfloor L_E/2 \rfloor < i < P[j-1] + \lfloor L_E/2 \rfloor \\ \infty & \text{, sinon} \end{cases} \\
A[j] &= \arg\min_x C[x, j-1] \\
d[j] &= \max\left(d[j-1] + \frac{A[j] - P[j-1]}{D}, 0\right) \\
P[j] &= \lfloor P[j-1] + d[j] \rfloor
\end{aligned}
\tag{3.6}
$$

où $P[j]$ est la position de la cellule centrale pour la position j de la séquence de test, $d[j]$ est l'incrément de cette position pour l'itération j et D est une constante d'amortissement limitant le déplacement instantané de la cellule centrale ($P[j]$).

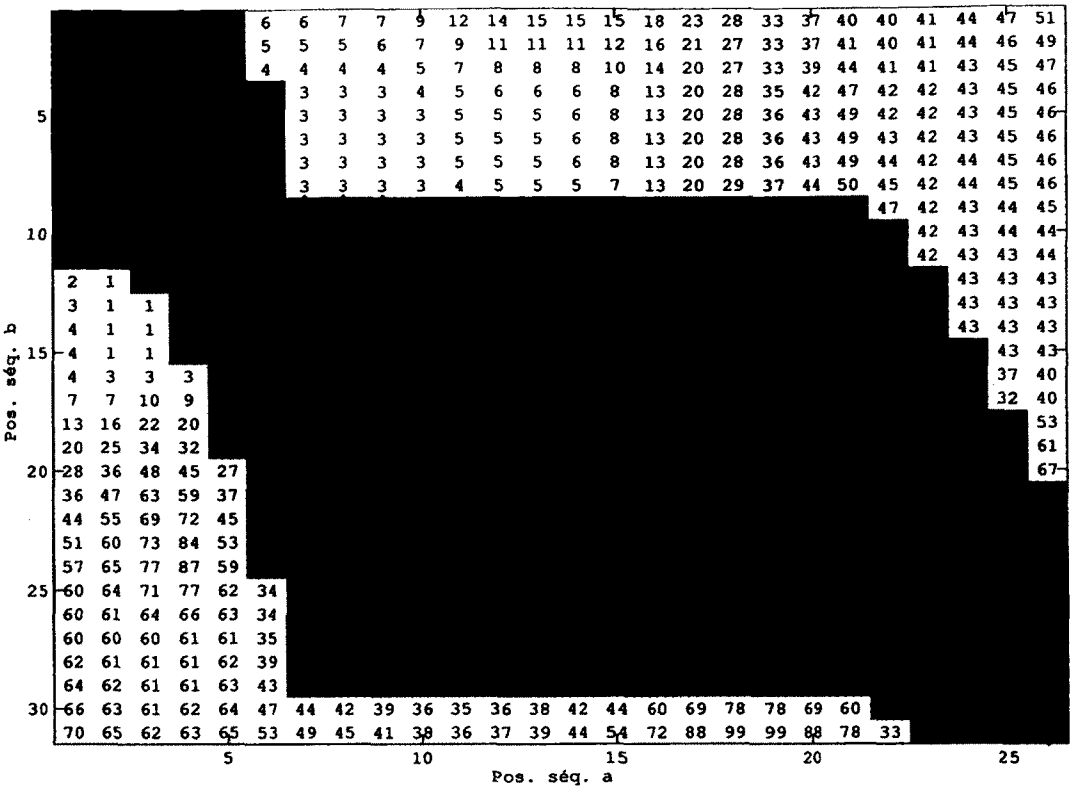
La figure 3.12 montre les résultats obtenus par l'algorithme *taille fixe à déplacement proportionnel.* La valeur de la constante d'amortissement D a été fixée à 5 pour cet exemple. On peut remarquer une amélioration au niveau de la zone de recherche et des écarts entre la matrice de coûts cumulés obtenue par rapport à celle qui est obtenue avec une DTW usuelle (à la figure 3.12b) d'abord puisque qu'un coût a pu être calculé pour toutes les cellules de la zone de recherche (pas de diminution de la zone de recherche effective) et que l'écart maximal parmis les cellules calculées est de 28 comparé à l'écart maximal de 39 observé à la figure 3.11.

3.3 Sommaire

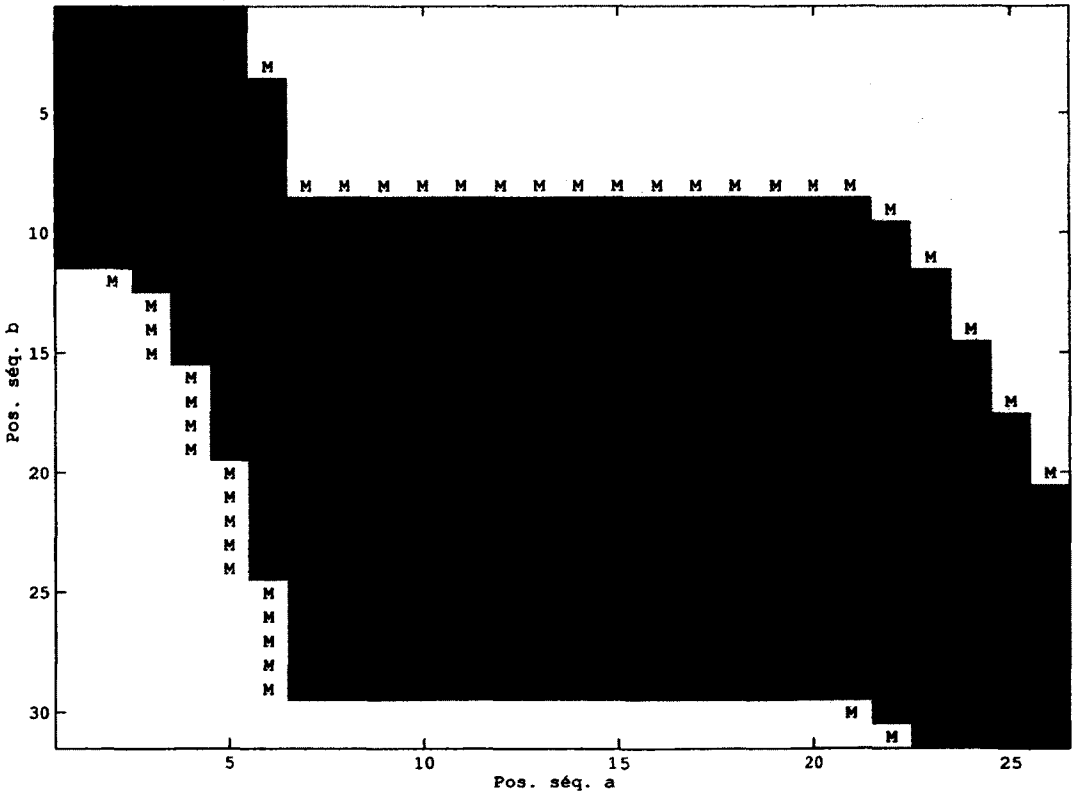
Une technique d'alignement de séquences de nature quelconque pouvant être appliquée aux séquences audio a été présentée. Dans sa forme ordinaire, celle-ci requiert deux séquences qui sont bornées et présentent une complexité d'ordre $\mathcal{O}(nm)$ où n et m sont les longueurs des deux séquences en question.

L'algorithme de Dixon propose une alternative à l'algorithme original qui permet l'utilisation en ligne et en temps réel, proposant une complexité constante, à chaque itération.

Cet algorithme présente cependant certaines lacunes importantes qui font en sorte que l'alignement obtenu peut être retardé significativement par rapport à l'alignement idéal. Deux algorithmes simples à complexité constante, applicables pour des problèmes en ligne et en temps réel, ont été proposés comme alternative à l'algorithme de Dixon. Ceux-ci ont fourni de meilleurs résultats sur des problèmes synthétiques.



(a) Matrice de coûts cumulés et alignement obtenus



(b) Différence entre la matrice obtenue et la matrice qui aurait été obtenue sans contraintes

Figure 3.12 Résultats obtenus pour la problématique *saut* : algorithme *taille fixe à déplacement proportionnel*

La performance de ces algorithmes pour des problèmes concrets d'alignement audio sera étudiée au chapitre 6.

CHAPITRE 4
ESPACE DE PARAMÈTRES

Dans ce chapitre, on s'attarde au choix de l'espace de paramètres pour le système d'alignement des signaux de voix chantée réalisé. Plusieurs types de paramètres seront considérés, dont entre autres les paramètres spectraux ayant un lien avec la cognition de la voix, mais aussi d'autres paramètres tels que le niveau d'énergie des signaux. Le détail de l'implémentation de l'extraction de ces différents paramètres sera explicité.

4.1 Paramètres spectraux

La première catégorie de paramètres considérés est les paramètres spectraux. Tel que mentionné à la section 2.1, on estime que les paramètres les plus importants pour l'algorithme d'alignement sont les paramètres spectraux. Ceux-ci sont liés à la cognition de la voix et donc plus au texte qu'aux autres aspects plus artistiques de la performance tels que le *pitch* ainsi que les modulations qui y sont associées tel le trémolo. On peut assumer beaucoup plus facilement qu'un chanteur amateur prononcera le bon texte qu'il chantera «juste».

Plusieurs jeux de paramètres spectraux ont été considérés. Ces jeux de paramètres ont été tirés des différents travaux ayant trait à la reconnaissance automatique de la parole. Trois types d'analyse ont été mis à l'épreuve dans le cadre d'un système d'alignement de la voix chantée.

Dérivées temporelles

En ajoutant l'information sur l'évolution des paramètres au jeu de paramètres utilisé par un algorithme d'alignement, on parvient améliorer les résultats. Pour cette raison, tous les paramètres spectraux retenus seront évalués en examinant également leurs dérivées temporelles. Puisque les analyses spectrales utilisées produiront un vecteur par trame, on obtiendra une séquence de données discrètes. Comme la dérivée n'est pas définie pour une fonction discrète, une approximation sera utilisée. Étant donné que les vecteurs obtenus sont de nature discontinue dans le temps, une approximation de la dérivée par différence finie ne donne pas de bons résultats. L'approximation qui a été choisie est la pente du modèle linéaire minimisant l'écart quadratique moyen considérant les K dernières valeurs

d'un coefficient donné dans les vecteurs de coefficients obtenus. Le modèle linéaire est donné par

$$\hat{y}_i[n] = a_i[n] \cdot x_i[n] + b_i[n] \tag{4.1}$$

où, en utilisant l'estimateur des moindres carrés, on a

$$a_i[n] = \frac{\sum_{k=0}^{K-1} x_i[n-k] \sum_{k=0}^{K-1} y_i[n-k] - K \sum_{k=0}^{K-1} x_i[n-k] \cdot y_i[n-k]}{(\sum_{k=0}^{K-1} x_i[n-k])^2 - K \sum_{k=0}^{K-1} x_i^2[n-k]} \tag{4.2}$$

et donc

$$\Delta x_i[n] = a_i[n] \tag{4.3}$$

où $x_i[n]$ et $\Delta x_i[n]$ sont respectivement le $i^{\text{ème}}$ coefficient du paramètre considéré et l'estimateur de sa dérivée, à la trame d'indice n.

Étant donné que l'approximation utilise les K dernières valeurs et que l'approximation sera plus juste au centre, l'approximation de la dérivée sera retardée de K/2 trames.

Exemples

Pour chacun des paramètres évalués, un exemple d'extraction du paramètre et les matrices de coûts qui y sont associées seront montrés. L'exemple qui sera utilisé dans ce chapitre utilise les segments audio représentés aux figures 4.1 et 4.2. Les deux signaux sont des extraits d'interprétations de la chanson *You Will You Won't* du groupe *The Zutons*. Ces extraits sont tirés de la fin du premier couplet et correspondent aux paroles : «But all the time you're thinking, well you're tricking your mind. You will you won't.». Le premier, utilisé en tant que référence, est interprété par un homme de 28 ans et le second, utilisé comme séquence de test, est interprété par femme de 29 ans. Les interprètes ont chanté intégralement le texte, sans fausser de façon intentionnelle et sans omettre ni répéter de mots.

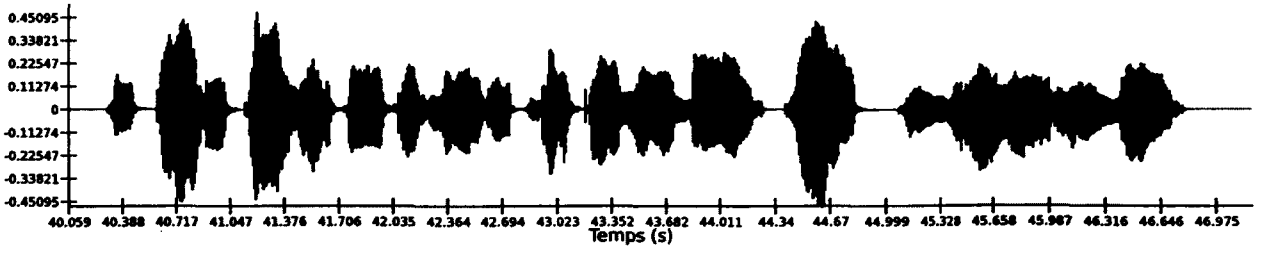


Figure 4.1 Segment audio de référence pour l'exemple

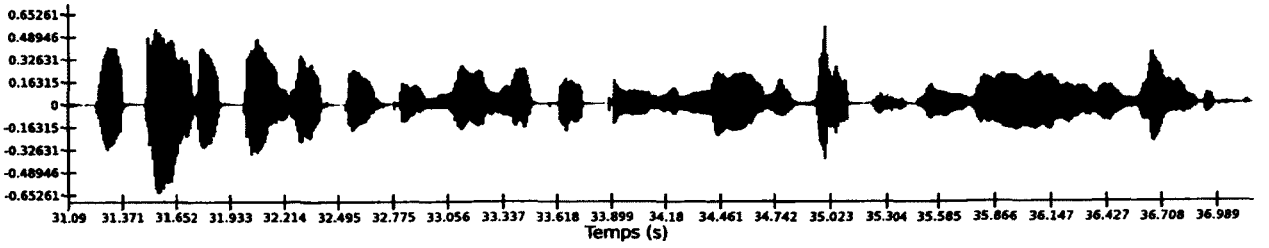


Figure 4.2 Segment audio de test pour l'exemple

Résultats et choix des paramètres d'analyse

Afin de comparer les paramètres spectraux étudiés et de déterminer quelles valeurs de leurs différents paramètres permettent d'obtenir les meilleurs résultats, plusieurs tests ont été effectués. Les tests ont été effectués en intégrant les paramètres évalués dans l'algorithme d'alignement complet et en mesurant une métrique de performance dont la valeur diminue lorsque le coût cumulé des cellules faisant partie de l'alignement idéal est plus faible que les cellules voisines. Cette métrique sera décrite plus loin à la section 5.2.

Les séquences de test sont deux fichiers audio constitués d'un assemblage de six paires de séquences audio chantées par différents interprètes, parfois avec différents débits, pour un total d'environ 30 secondes par fichier. Une des interprétations comporte un mot répété et une autre est une interprétation monotone qui ne respecte pas du tout le débit original de la chanson.

Les coûts obtenus en comparant les vecteurs de coefficients résultants sont modifiés en utilisant une non-linéarité. De plus, les coûts obtenus pour la version directe et la version dérivée des paramètres spectraux seront pondérés afin de refléter une importance relative différente pour chacun des coûts obtenus. Tel que décrit plus loin, au chapitre 5, un algorithme d'optimisation sera utilisé pour déterminer les paramètres optimaux des non-linéarités et les facteurs de pondérations à utiliser. Chaque résultat présenté a fait appel à ce processus d'optimisation.

4.1.1 *Mel-Frequency Cepstrum Coefficients* (MFCC)

Le premier paramètre considéré est les MFCC. L'analyse MFCC est une analyse communément utilisée dans le domaine de la reconnaissance automatique de la parole. On attribue la technique aux travaux de Paul Mermelstein [Mermelstein, 1976].

La technique est une amélioration de l'analyse *Linear Frequency Cepstrum Coefficients* (LPCC) qui est décrite dans [Juang et Rabiner, 1993]. La figure 4.3 montre le schéma bloc des deux types d'analyse.

La cognition de la voix est associée à la forme générale du spectre d'amplitude du signal de parole. Les deux méthodes emploient une TFD afin de passer au domaine fréquentiel. Ensuite, un banc de filtre regroupe l'énergie des bandes de fréquences, l'intérêt étant d'isoler la forme générale du spectre d'amplitude plutôt que sa structure fine. On passe ensuite au domaine cepstral en prenant le logarithme de la sortie des filtres et en effectuant une transformée en cosinus discrète (DCT) du résultat.

On utilise une DCT-II pour obtenir le cepstre plutôt qu'une TFD puisque le signal de sortie est réel et que le spectre correspondant présente une symétrie paire. Dans un tel cas, la DCT-II produit le même résultat, mais a l'avantage de nécessiter moins de temps de calcul.

En tronquant le résultat de la DCT, on élimine des composants hautes fréquences de l'enveloppe obtenue, ce qui a pour effet de la lisser. De plus, en réduisant le nombre de coefficients obtenus, les calculs de coûts effectués dans le cadre de l'algorithme d'alignement seront plus rapides.

La seule différence entre les MFCC et les LFCC est l'utilisation d'un banc de filtre dont les filtres ont été spécifiés dans l'échelle de Mel plutôt que dans une échelle linéaire. Plus précisément, les fréquences centrales et de «bordure» sont réparties de façon linéaire sur une échelle de Mel. On obtient alors des filtres dont la bande passante n'est pas uniforme ; elle augmente avec la fréquence, ce qui correspond à la façon dont les sons sont perçus par l'humain.

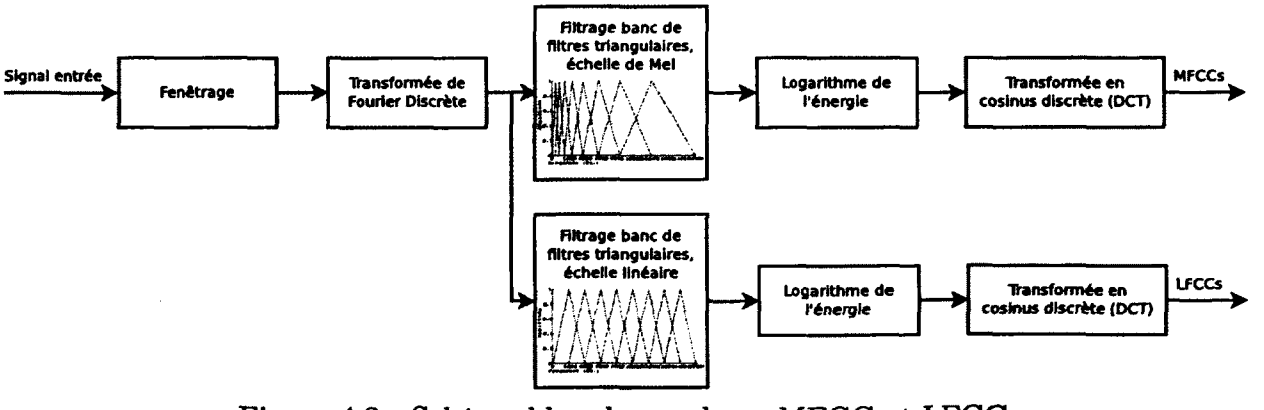


Figure 4.3 Schéma bloc des analyses MFCC et LFCC

Implémentation de l'analyse MFCC

L'algorithme utilisé dans le cadre des présents travaux pour le calcul des MFCC est une implémentation en langage C faisant partie de la librairie *libxtract* [Bullock, 2007]. Cette librairie contient une multitude de fonctions permettant l'extraction de divers paramètres des signaux audio, dont entre autres les MFCC.

Cette librairie utilise également une autre librairie, *FFTW* (pour *Fastest Fourier Transform on the West*) [Frigo et Johnson, 1998] afin de calculer les TFDs et DCTs requises.

Les spécificités de l'algorithme de la MFCC ne sont pas clairement explicitées dans l' article de Mermelstein [Mermelstein, 1976], ce qui fait en sorte qu'on peut répertorier des implémentations avec des différences majeures dans la littérature. [Zheng *et al.*, 2001] présentent une comparaison des différentes implémentations possibles de l'analyse MFCC.

L'implémentation de l'analyse utilisée dans ces travaux emploie des trames de 20 millisecondes ou 960 échantillons à 48 kHz. Avant l'opération de fenêtrage, les hautes fréquences sont accentuées à l'aide d'un filtre décrit par :

$$A(z) = 1 - 0.99z^{-1}; \tag{4.4}$$

La fenêtre utilisée est une fenêtre de Hamming. On effectue ensuite une TFD sur la trame obtenue afin d'obtenir une représentation spectrale à court terme du signal. Le spectre d'amplitude est ensuite isolé et filtré avec le banc de filtres triangulaires défini par

$$H_k(f) = \begin{cases} 0 & \text{, pour } f < l(k) \\ \frac{f-l(k)}{l(k+1)-l(k)} & \text{, pour } l(k) \leq f < l(k+1) \\ \frac{f-l(k+1)}{l(k+2)-l(k+1)} & \text{, pour } l(k+1) \leq f \leq l(k+2) \\ 0 & \text{, pour } f > l(k+2) \end{cases} \quad ,k = 1, ..., N_f - 1 \tag{4.5}$$

où $l(k) = F(M(f_{min}) + k \cdot \Delta m)$ est la fréquence à laquelle la bande passante du filtre d'indice k débute, $\Delta m = \frac{M(f_{max}) - M(f_{min})}{N_f + 1}$ est l'écart, en mels, entre deux filtres consécutifs. Les fonctions $F(m)$ et $M(f)$ sont respectivement les fonctions de conversion de mel à Hz et de Hz à mel :

$$\begin{aligned} M(f) &= 2595 \cdot \log_{10}(1 + \tfrac{f}{700}) \\ F(m) &= 700 \cdot (10^{\frac{m}{2595}} - 1) \end{aligned} \tag{4.6}$$

Puisque les différents filtres du banc de filtre décrit en (4.5) n'ont pas la même bande passante, l'énergie à la sortie de chaque filtre ne sera pas la même pour un signal dont le spectre d'amplitude à une valeur constante. Afin d'éliminer ce biais, on multiplie chaque

filtre par un gain qui ramène l'aire de chaque filtre à 1 :

$$G_k(f) = \frac{H_k(f)}{l(k+2) - l(k)}. \tag{4.7}$$

La figure 4.4 montre la réponse en fréquence d'un exemple de banc de filtre décrit par l'équation 4.5, pour un nombre de filtres N_f de 8, répartis entre $f_{min} = 80$Hz et $f_{max} = 18000$Hz.

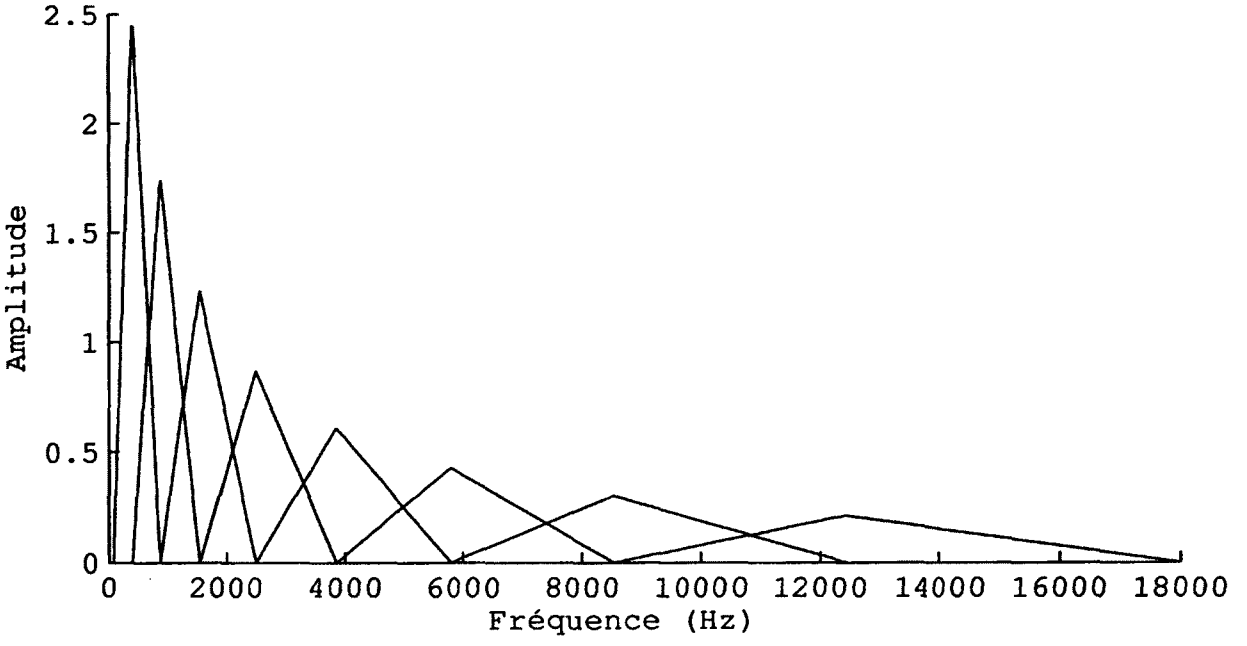


Figure 4.4 Réponse en fréquence du banc de filtre triangulaire utilisé par l'analyse MFCC

À la sortie du banc de filtres triangulaires, en appliquant l'opérateur logarithme, on obtient :

$$Y_k = \log_{10}(\sum_{n=0}^{N-1} \widehat{G_k}[n] \cdot |X_k[n]|), k = 1, ..., N_f \tag{4.8}$$

où N est la longueur des trames utilisées, soit 960 échantillons, et $\widehat{G_k}[n]$ sont les versions discrétisées des filtres triangulaires $G_k(f)$ obtenus précédemment.

On obtient ensuite une représentation cepstrale en appliquant une DCT-II, pour finalement obtenir les coefficients MFCC

$$C_i = \sum_{k=1}^{N_f} Y_k \cdot cos(i \cdot (k - 1/2) \cdot \pi / N_f), i = 1, ..., N_c \tag{4.9}$$

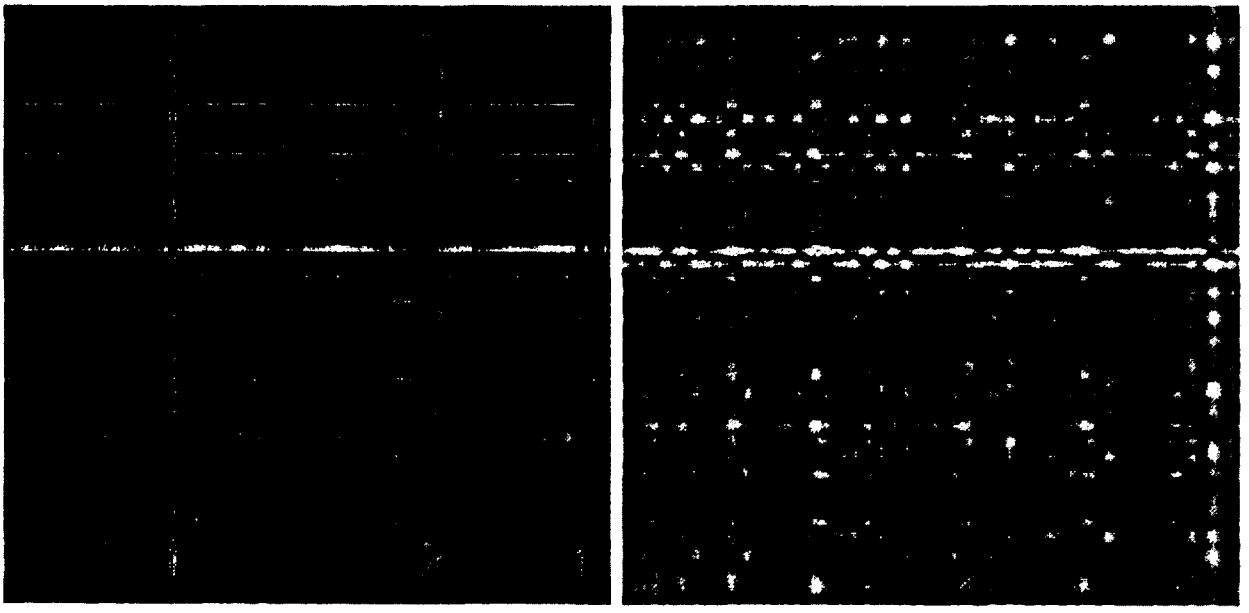
où N_c est le nombre de coefficients MFCC retenus.

La librairie *libxtract* utilise l'implémentation de la DCT-II de la librairie *FFTW* pour passer au domaine cepstral, après l'opération logarithme.

Les figures 4.5 et 4.6 montrent les résultats de l'extraction des coefficients MFCC et les matrices de coût associées à l'exemple (figures 4.1 et 4.2).

(a) Séquence de référence (b) Séquence de test

Figure 4.5 Comparaison des spectrogrammes MFCC obtenus pour l'exemple. Chaque rangée montre la valeur d'un coefficient différent de l'analyse, en partant du haut de l'image, pour chaque trame (colonne) analysée.



(a) Énergie (b) Dérivée discrète de l'énergie

Figure 4.6 Matrices de coûts obtenues pour le paramètre spectral MFCC pour l'exemple

Choix des paramètres d'analyse

Plusieurs variables de l'analyse MFCC sont configurables. Parmi ceux-ci, on compte le nombre de coefficients et le nombre de filtres utilisés. Aussi, dans le contexte de l'algorithme d'alignement, on s'intéresse aussi à l'ordre de l'estimateur de la dérivée (décrit

au paragraphe «Dérivées Temporelles», en 4.1) et au type de pondération des coefficients utilisé pour le calcul de coût, car leurs effets ne sont pas indépendants du type d'analyse effectuée. Afin de déterminer les valeurs de ces paramètres qui permettent d'obtenir les meilleurs résultats, plusieurs essais ont été faits.

Le premier paramètre dont l'impact a été évalué est la pondération des coefficients dans l'évaluation du coût. Le coût utilisé est l'écart quadratique tel que donné par

$$c[i,j] = \sum_{n=0}^{N_C-1} p_n \cdot (T_n[i] - R_n[j])^2 \tag{4.10}$$

où p_n sont les coefficients de pondération évalués, **T[i]** et **R[j]** sont respectivement le vecteur de paramètres extrait de la séquence de test et de référence pour les trame i et j et $N{+}C$ est la longueur des ces vecteurs, égale au nombre de coefficients à la sortie de l'analyse.

Plusieurs types de pondérations appliqués à l'analyse MFCC sont comparées dans [Zheng *et al.*, 1997]. Dans cette étude, la pondération par le logarithme de l'indice des coefficients donne les meilleurs résultats. L'effet d'une telle pondération est d'augmenter l'importance des coefficients d'indice plus élevés mais d'une façon plus modérée que la pondération par l'indice. Les coefficients de pondération pour la pondération par le logarithme de l'indice sont donnés par

$$p_n = ln(n+2). \tag{4.11}$$

Les essais effectués dans le cadre des présents travaux corroborent ceux de [Zheng *et al.*, 1997]. La figure 4.7 montre plusieurs résultats obtenus en faisant varier différents paramètres (nombre de filtres, de coefficients et ordre de l'estimateur de la dérivée), sans pondération (tous les poids à 1) et avec une pondération par le logarithme de l'indice. La moyenne des résultats de l'écart quadratique moyen est de 0.2424 ms sans pondération et de 0.2370 ms avec une pondération par le logarithme de l'indice.

La plupart des points sont sous la droite, indiquant un écart quadratique moyen inférieur pour la pondération par le logarithme de l'indice. Les prochains essais, comparant les autres paramètres de l'analyse MFCC, utilisent donc tous la pondération par le logarithme de l'indice.

D'autres essais ont été réalisés en faisant varier le nombre de coefficients, le rapport du nombre de filtres sur le nombre de coefficients retenus et l'ordre de l'estimateur de la dérivée utilisé. Les résultats présentés sont des valeurs d'une métrique de performance

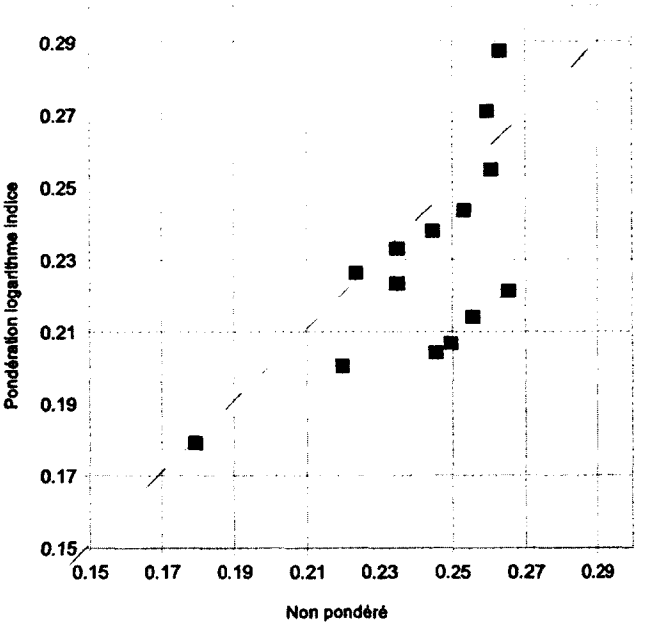


Figure 4.7 Comparaison de l'erreur quadratique moyenne avec pondération par le logarithme de l'indice et sans pondération

calculée à partir de la matrice de coûts cumulés, donnée à l'équation 5.5, au chapitre suivant. Plus la valeur est faible, plus la valeur du coût cumulé aux positions alignées est faible par rapport aux positions non alignées, ce qui permet de plus facilement distinguer l'alignement optimal. Le tableau 4.1 présente les résultats obtenus, pour fins de référence. Les résultats seront analysés selon différents groupement dans les prochains paragraphes.

Tableau 4.1 MFCC : Métrique de performance obtenue pour différentes valeurs des paramètres de l'analyse

N. coeffs.	10			12			14			16			18		
N. filtres	10	15	20	12	18	24	14	21	28	16	24	32	18	27	36
Δ ordre 3	-.926	-.921	-.905	-.936	-.909	-.910	-.907	-.916	-.914	-.896	-.916	-.921	-.928	-.926	-.894
Δ ordre 5	-.929	-.928	-.900	-.933	-.894	-.940	-.916	-.931	-.914	-.916	-.931	-.904	-.915	-.921	-.900
Δ ordre 7	-.934	-.945	-.929	-.939	-.928	-.932	-.959	-.933	-.941	-.930	-.942	-.958	-.929	-.929	-.920

La figure 4.8 et le tableau 4.2 présentent les moyennes des résultats obtenus groupés selon le nombre de coefficients et le ratio du nombre de filtres sur le nombre de coefficients. L'efficacité du paramètre diminue avec l'augmentation du nombre de coefficients jusqu'à concurrence de 14 coefficients et augmente légèrement par la suite.

La figure 4.9 et le tableau 4.3 montrent l'effet de l'ordre de l'estimateur de la dérivée. Les meilleurs résultats ont été obtenus avec un estimateur d'ordre 7.

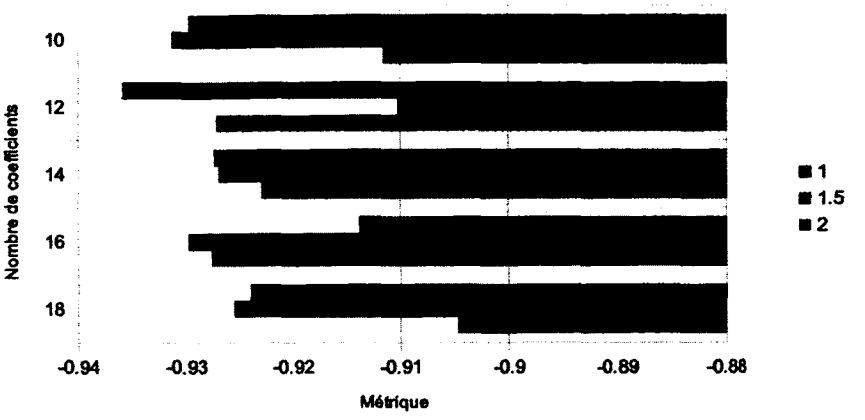


Figure 4.8 MFCC : Moyenne de la métrique de performance en fonction du nombre de coefficients et du ratio du nombre de filtres sur le nombre de filtres

Tableau 4.2 MFCC : Moyenne de la métrique de performance obtenue pour différentes valeurs du nombre de coefficients

N. coeffs	10	12	14	16	18
Moy. métrique	-0.9242	-0.9244	-0.9258	-0.9237	-0.9180

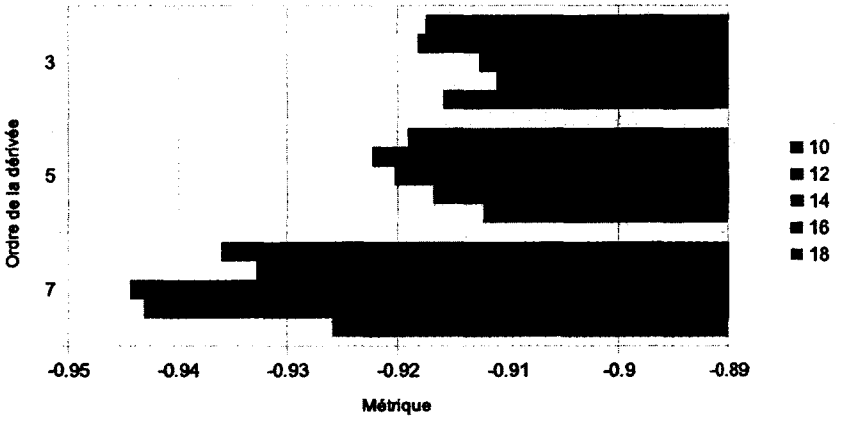


Figure 4.9 MFCC : Moyenne de la métrique de performance en fonction de l'ordre de l'estimateur de la dérivée et du nombre de coefficients

Tableau 4.3 MFCC : Moyenne de la métrique de performance obtenue pour différentes valeurs d'ordre de l'estimateur de la dérivée

Ordre Δ	3	5	7
Moy. métrique	-0.9150	-0.9181	-0.9365

Le tableau 4.4 présente la moyenne des résultats obtenus pour différentes valeurs du ratio du nombre de filtres sur le nombre de coefficients. Les meilleurs résultats sont obtenus avec un ratio égal à 1.

Tableau 4.4 MFCC : Moyenne de la métrique de performance obtenue avec différentes valeurs du ratio du nombre de filtres sur le nombre de coefficients

Ratio N_f/N_c	1	1.5	2
Moy. métrique	-0.9261	-0.9247	-0.9188

Le meilleur résultat obtenu dans les essais réalisés et répertoriés au tableau 4.1 utilise 14 coefficients, 14 filtres (ratio N_f/N_c de 1) et un estimateur de la dérivée d'ordre 7, ce qui correspond aux paramètres identifiés précédemment.

4.1.2 *Warped Discrete Cosine Transform Cepstrum* (WDCTC)

L'analyse WDCTC, pour *Warped Discrete Cosine Transform Cepstrum* [Muralishankar *et al.*, 2005], est une modification du *Discrete Cosine Transform Cepstrum* (DCTC) décrit par [Muralishankar et Ramakrishnan, 2005]. Cette dernière technique est inspirée de [Hassanein et Rudko, 1984], où les auteurs proposent d'utiliser la DCT plutôt que la TFD pour calculer le cepstre complexe d'un signal.

Le cepstre d'un signal est utile pour passer d'un domaine où plusieurs signaux sont obtenus par convolutions à un domaine où ces signaux s'additionnent. Il devient alors possible, dépendamment des signaux en cause, de plus facilement isoler la contribution de chacun des signaux. Si le cepstre complexe est utilisé, il est possible de faire la transformation inverse et d'obtenir chacun des signaux isolés dans le domaine temporel. L'opération s'appelle filtrage homomorphe.

Le cepstre complexe se défini par

$$\begin{aligned} \hat{x}[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{X}(e^{j\omega})e^{j\omega n}d\omega = \mathcal{F}^{-1}(\hat{X}) \\ \hat{X}(e^{j\omega}) &= ln(X(e^{j\omega})) \\ X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \mathcal{F}(x) \end{aligned} \tag{4.12}$$

où $\hat{x}[n]$ est le cepstre complexe du signal $x[n]$ et $\mathcal{F}$ et $\mathcal{F}^{-1}$ représentent respectivement une transformée de Fourier à temps discret (TFTD) et son inverse, la transformée de Fourier à temps discret inverse (TFDI).

Le cepstre réel, non réversible, mais aussi utile pour l'analyse de signaux, utilise l'amplitude du spectre fréquentiel seulement :

$$\begin{aligned} \hat{x}_r[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{X}_r(e^{j\omega})e^{j\omega n}d\omega = \mathcal{F}^{-1}(\hat{X}_r) \\ \hat{X}_r(e^{j\omega}) &= ln\left|X(e^{j\omega})\right| \\ X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \mathcal{F}(x). \end{aligned} \tag{4.13}$$

Dans le cas particulier où on a un signal réel à phase minimale (dont tous les pôles et zéros sont à l'intérieur du cercle unitaire du plan z), le cepstre complexe correspond au cepstre réel à un facteur près [Oppenheim et Schafer, 1968].

L'extraction du cepstre complexe est une opération coûteuse puisque pour effectuer le logarithme complexe du spectre de fréquence obtenu, il est nécessaire de dérouler la phase de celui-ci. En faisant l'hypothèse que le signal analysé est un signal réel pair, la TFD est remplaçable par une DCT-II.

La transformée de Fourier discrète d'un signal est donnée par

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{k}{N}n}. \tag{4.14}$$

En considérant un signal $x[n] = \{x[0], x[1], x[2], ..., x[N-1]\}$ et une séquence $y_t[m_t]$ construite à partir de $x[n]$ et définie par

$$\begin{aligned} y_t[m_t] &= y[m_t - 1/2] \\ y[m] &= \begin{cases} x[-(m-1)] & \text{, pour } m = -N, -N-1, ..., -1 \\ x[m] & \text{, pour } m = 0, 1, ..., N-1 \end{cases} \end{aligned} \tag{4.15}$$

La transformée de Fourier discrète de ce signal est donnée par

$$
\begin{aligned}
Y_t[k] &= \sum_{m_t=-N+\frac{1}{2}}^{N-\frac{1}{2}} y\left[m_t-\frac{1}{2}\right] e^{\frac{-j2\pi m_t k}{2N}} \\
Y_t[k] &= \sum_{m_t=-N+\frac{1}{2}}^{N-\frac{1}{2}} y\left[m_t-\frac{1}{2}\right] cos\frac{\pi m_t k}{N} - j\sum_{m_t=-N+\frac{1}{2}}^{N-\frac{1}{2}} y\left[m_t-\frac{1}{2}\right] sin\frac{\pi m_t k}{N} \\
Y_t[k] &= \sum_{i=0}^{N-1}\left(y[-i-1]cos\left(-\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right)+y[i]cos\left(\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right)\right) \\
&\quad -j\sum_{i=0}^{N-1}\left(y[-i-1]sin\left(-\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right)+y[i]sin\left(\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right)\right) \\
Y_t[k] &= \sum_{i=0}^{N-1}(y[i]+y[-i-1])cos\left(\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right) - j\sum_{i=0}^{N-1}(y[i]-y[-i-1])sin\left(\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right) \\
Y_t[k] &= 2\sum_{i=0}^{N-1} y[i]cos\left(\pi\left(i+\frac{1}{2}\right)\frac{k}{N}\right) = 2\sum_{n=0}^{N-1} x[n]cos\left(\pi\left(n+\frac{1}{2}\right)\frac{k}{N}\right).
\end{aligned}
\tag{4.16}
$$

La transformée est donc égale à un facteur près à la DCT-II du même signal

$$
C[k] = \sum_{n=0}^{N-1} x[n]cos\left[\pi\left(n+\frac{1}{2}\right)\frac{k}{N}\right] = \frac{Y_t[k]}{2}. \tag{4.17}
$$

En plus d'être plus rapide à calculer que la TFD, l'opération du déroulement de la phase est facilitée puisque la phase non déroulée ne peut prendre que deux valeurs : $-\pi$ et 0, selon le signe de la valeur du spectre réel obtenu. Une technique pour ce faire est décrite dans [Hassanein et Rudko, 1984].

L'analyse DCTC de [Muralishankar et Ramakrishnan, 2005] n'effectue cependant pas le déroulement de la phase, elle utilise seulement la valeur principale d'arctangeante de chaque échantillon du spectre ; on obtient donc un spectre «pseudocomplexe». Sous forme polaire, on a

$$
Y_t[k] = |Y_t[k]|\, e^{\xi(k)}. \tag{4.18}
$$

Puisque $Y_t[k]$ est réel, le terme $e^{\xi(k)}$ ne peut prendre que deux valeurs :

$$
e^{j\xi(k)} = \begin{cases} 1 & \text{, pour } Y_t[k] \geq 0 \\ -1 & \text{, pour } Y_t[k] < 0 \end{cases}. \tag{4.19}
$$

On a donc

$$\xi(k) = \frac{\pi(sgn(Y_t[k]) - 1)}{2}$$
$$\hat{C}[k] = ln(C[K]) = ln\,|Y_t[k]| + j\xi(k). \tag{4.20}$$

On obtient ensuite le cepstre «pseudocomplexe» en prenant la partie réelle de la transformée inverse de la DCT-II, soit la DCT-III, souvent simplement nommée transformée en cosinus discrète inverse (IDCT).

Les résultats obtenus par la DCTC sont supérieurs aux LPCC, mais inférieurs aux MFCC [Muralishankar et Ramakrishnan, 2005]. En remplaçant la DCT par la WDCT [Cho et Mitra, 2000], qui utilise une échelle fréquentielle non linéaire ajustée pour approcher l'échelle de Bark, l'analyse WDCTC obtient de meilleurs résultats que les MFCC lorsqu'intégré à un système de reconnaissance de la parole [Muralishankar *et al.*, 2005].

La WDCT utilise une cascade de filtres passe-bande du premier ordre désignés sous le nom de réseau de Laguerre. Ces filtres sont utilisés pour réaliser la déformation de l'échelle fréquentielle, en remplaçant les éléments z^{-1} par des filtres décrits par

$$A(z) = \frac{-\beta + z^{-1}}{1 - \beta z^{-1}}. \tag{4.21}$$

La DCT peut être implémentée à l'aide d'un banc de filtres discrets décrits par

$$F_k(z) = \sum_{n=0}^{N-1} cos\left(\frac{\pi(2n+1)k}{2N}\right) z^n. \tag{4.22}$$

Les sorties de ces filtres sont décimées par un facteur N puisque les coefficients de la DCT sont obtenus à chaque début de trame (à intervalle de N échantillons).

En appliquant la transformation, on obtient la WDCT :

$$W_k(z) = F_k(A(z)^{-1}) = \sum_{n=0}^{N-1} cos\left(\frac{\pi(2n+1)k}{2N}\right) A(z)^{-n}. \tag{4.23}$$

Le schéma-bloc de l'analyse résumant WDCTC est présenté à la figure 4.10.

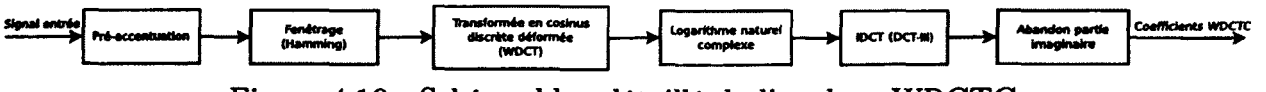


Figure 4.10 Schéma-bloc détaillé de l'analyse WDCTC

Implémentation de l'analyse WDCTC

Avant de calculer la WDCT, le signal d'entrée est préaccentué à l'aide d'un filtre passe-haut décrit par $H(z) = 1 - 0.98z^{-1}$. Chaque trame (960 échantillons à 48kHz pour une période de 20 ms) est ensuite fenêtrée à l'aide d'une fenêtre de Hamming.

La WDCT fait intervenir des filtres à réponse impulsionnelle infinie. Afin de pouvoir calculer les coefficients de la WDCT à l'aide d'un produit matriciel, il est désirable d'utiliser une approximation des filtres permettant d'obtenir les coefficients de la WDCT (eq. 4.23) à l'aide d'un filtre à réponse impulsionnelle finie de longueur N correspondant à la taille d'une trame. En utilisant le théorème de Plancherel, on a

$$\begin{aligned} \sum_{n=-\infty}^{\infty} \overline{x[n]} f[n] &= \sum_{k=-\infty}^{\infty} \overline{X[k]} F[k] \\ \sum_{n=0}^{N-1} \overline{x[n]} f[n] &\approx \sum_{k=0}^{N-1} \overline{X[k]} F[k] \end{aligned} \tag{4.24}$$

où $\overline{\square}$ représente le complexe conjugué.

Avec $x[n] = \delta[n-i]$, on obtient

$$\begin{aligned} \sum_{n=0}^{N-1} \delta[n-i] f[n] &\approx \sum_{k=0}^{N-1} \overline{\left(\sum_{n=0}^{N-1} \delta[n-i] e^{\frac{-j2\pi kn}{N}} \right)} F[k] \\ f[i] &\approx \sum_{k=0}^{N-1} F[k] e^{\frac{j2\pi ki}{N}} \\ \mathbf{f} &\approx \mathcal{F}^{-1}(\mathbf{F}). \end{aligned} \tag{4.25}$$

Les coefficients de l'approximation de la WDCT peuvent donc être calculés à partir de l'équation 4.25 :

$$\widehat{\mathbf{w_k}} = \mathcal{F}^{-1}(\mathbf{W_k}) \tag{4.26}$$

où

$$\mathbf{W_k} = \left[W_k(e^0) \;\; W_k\left(e^{\frac{2\pi}{N}}\right) \;\; W_k\left(e^{\frac{4\pi}{N}}\right) \;\; \cdots \;\; W_k\left(e^{\frac{2(N-1)\pi}{N}}\right) \right]. \tag{4.27}$$

Le calcul de la WDCT est effectué sous forme d'une multiplication matricielle entre un vecteur dont les éléments sont les échantillons d'une trame et la matrice de transforma-

tion $\widehat{\mathbf{W}}$:

$$\begin{aligned}\mathcal{X} = \mathbf{x}\widehat{\mathbf{W}} &= \mathbf{x}\begin{bmatrix}\mathbf{W_0}\\ \mathbf{W_1}\\ \vdots\\ \mathbf{W_{N-1}}\end{bmatrix}\\ &= \begin{bmatrix}x_0 & x_1 & \dots & x_{N-1}\end{bmatrix}\begin{bmatrix} W_0(e^0) & W_0\left(e^{\frac{2\pi}{N}}\right) & \cdots & W_0\left(e^{\frac{2(N-1)\pi}{N}}\right)\\ W_1(e^0) & W_1\left(e^{\frac{2\pi}{N}}\right) & \cdots & W_1\left(e^{\frac{2(N-1)\pi}{N}}\right)\\ \vdots & \vdots & \ddots & \vdots\\ W_{N-1}(e^0) & W_{N-1}\left(e^{\frac{2\pi}{N}}\right) & \cdots & W_{N-1}\left(e^{\frac{2(N-1)\pi}{N}}\right)\end{bmatrix}\end{aligned} \tag{4.28}$$

La valeur du facteur β à l'équation 4.21, qui permet d'approcher l'échelle de Bark est donnée par [Muralishankar *et al.*, 2005] :

$$\beta = 1.0211\ \left(\frac{2}{\pi}arctan(0.076 f_s)\right)^{\frac{1}{2}} - 0.19877 = 0.82224. \tag{4.29}$$

Le spectre obtenu est ensuite converti sous sa forme polaire avant d'effectuer le logarithme complexe, tel que décrit par l'équation 4.20, en substituant le spectre Y pour $\mathcal{X}$.

La transformée par cosinus discrète inverse (IDCT, DCT-III) est ensuite appliquée au spectre obtenu afin d'obtenir les coefficients cepstraux. Le premier coefficient est abandonné et le vecteur constitué des N_c coefficients subséquents est conservé.

Choix des paramètres d'analyse

Le tableau 4.5 présente les métriques de performance obtenues pour différentes valeurs du nombre de coefficients retenus, d'ordre de l'estimateur de la dérivée et de type de pondération. La figure 4.11 présente les résultats obtenus sans pondération ; ceux-ci sont largement supérieurs, avec une moyenne de -0.9261 contre -0.7468 pour une pondération par le logarithme de l'indice.

Les meilleurs résultats sont obtenus avec 15 coefficients et un estimateur de dérivée d'ordre 5.

Tableau 4.5 WDCTC : Métriques de performances obtenues avec différentes valeurs des paramètres de l'analyse

	Nbre coefficients	15	20	25
Pond. unif.	Δ ordre 3	-0.932814	-0.919292	-0.91576
	Δ ordre 5	-0.943679	-0.935877	-0.926415
	Δ ordre 7	-0.937617	-0.923807	-0.926056
Pond. log ind.	Δ ordre 3	-0.747043	-0.72532	-0.739172
	Δ ordre 5	-0.742433	-0.739249	-0.731361
	Δ ordre 7	-0.757489	-0.774105	-0.746807

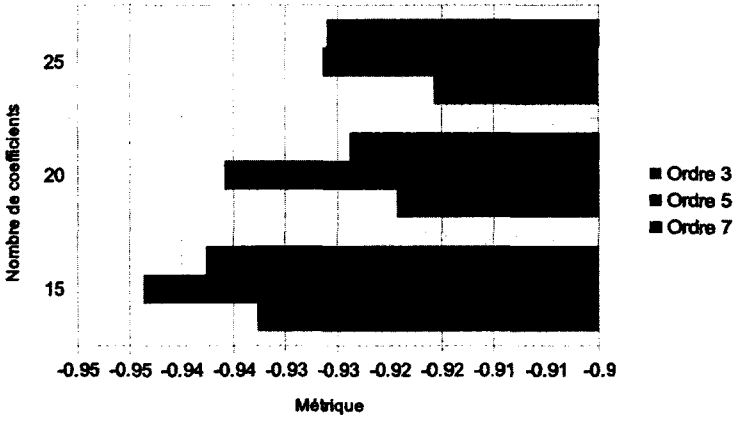


Figure 4.11 WDCTC : Moyenne de la métrique de fonction pour différentes valeurs du nombre de coefficients et de l'ordre de l'estimateur de la dérivée

4.1.3 *Perceptual Linear Prediction* (PLP)

L'analyse PLP [Hermansky, 1990] est une amélioration de l'analyse LPCC (*Linear Prediction Cepstral Coefficients*). Cette dernière utilise l'analyse LP afin de caractériser un modèle autorégressif du signal. En choisissant judicieusement l'ordre de l'analyse LP utilisée, une version aplanie de la réponse en fréquence est obtenue, permettant de distinguer les formants. Plutôt que d'utiliser les coefficients LP obtenus directement, on utilise les coefficients cepstraux correspondants puisque ceux-ci sont plus compacts en plus de permettre une séparation source-filtre lorsque les premiers coefficients sont conservés. De plus, l'évaluation de la distance entre deux signaux est simplifiée puisqu'on peut utiliser un simple écart quadratique moyen dans le cas des coefficients cepstraux.

L'analyse PLP, illustrée à la figure 4.12, vient ajouter des éléments de psychoaccoustique à l'analyse LPCC, dont un des problèmes est qu'elle modélise le spectre du signal analysé avec la même précision sur toute la plage d'analyse, ce qui ne correspond pas à l'audition humaine, pour laquelle l'importance des différentes bandes de fréquence varie. La technique effectue donc une série de manipulations au spectre de fréquence du signal afin de tenir compte de certains phénomènes psychoaccoustiques.

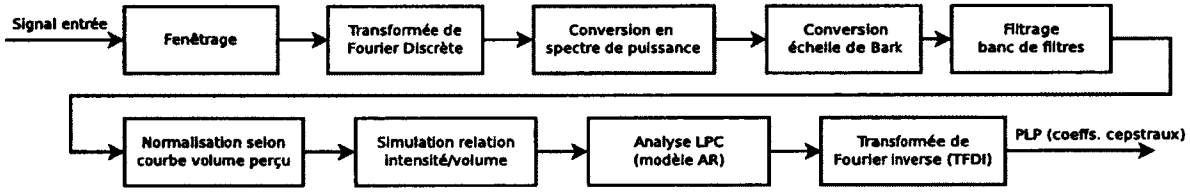


Figure 4.12 Schéma bloc de l'analyse PLP

Chaque trame est d'abord fenêtrée à l'aide d'une fenêtre de Hamming. C'est une étape nécessaire, étant donné la présomption de périodicité faite par la TFD qui suit. La densité spectrale de puissance est ensuite extraite à partir du spectre fréquentiel obtenu :

$$P(\omega) = Re\,[X(\omega)]^2 + Im\,[X(\omega)]^2\,.$$

Les opérations subséquentes visent à intégrer certaines notions de psychoaccoustique.

D'abord, la densité spectrale de puissance obtenue est déformée afin que son échelle fréquentielle corresponde à l'échelle de Bark. La conversion Hertz-Bark utilisée est celle de [Schroeder, 1977] et s'exprime par

$$\Omega(f) = 6 \cdot ln\left(\frac{f}{600} + \sqrt{\left(\frac{f}{600}\right)^2 + 1}\right). \tag{4.30}$$

Le spectre obtenu est ensuite convolué avec la courbe de masquage des bandes critiques décrite par

$$\Psi(\Omega) = \begin{cases} 0 & , \text{pour } \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)} & , \text{pour } -1.3 \leq \Omega \leq -0.5 \\ 1 & , \text{pour } -0.5 < \Omega < 0.5 \\ 10^{-1.0(\Omega-0.5)} & , \text{pour } 0.5 \leq \Omega \leq 2.5 \\ 0 & , \text{pour } \Omega > 2.5 \end{cases} . \tag{4.31}$$

La convolution du spectre et de la courbe de masquage permet d'obtenir la densité spectrale de puissance des bandes critiques :

$$\theta(\Omega) = \sum_{\Omega_i=-1.3}^{2.5} P(\Omega_i - \Omega)\Psi(\Omega_i). \tag{4.32}$$

Cette opération de masquage réduit significativement la résolution spectrale de $\theta(\Omega)$. Un sous-échantillonnage d'importance (typiquement un échantillon par Bark) est donc ensuite réalisé.

Ensuite, le spectre échantillonné $\Theta(\Omega)$ est préaccentué à l'aide d'une courbe isotonique, qui représente le niveau de puissance sonore nécessaire étant perçu de même intensité en fonction de la fréquence.

$$\Xi(\Omega) = \Theta(\Omega)E(\Omega) \tag{4.33}$$

La courbe utilisée est décrite par

$$E(\omega) = \frac{(\omega^2 + 56.8 \cdot 10^6)\omega^4}{(\omega^2 + 6.3 \cdot 10^6)^2 \cdot (\omega^2 + 0.38 \cdot 10^9)}. \tag{4.34}$$

Une compression de l'amplitude du spectre est ensuite effectuée afin de simuler la relation non linéaire entre l'intensité d'un son et le volume perçu. La compression utilisée est la racine cubique :

$$\Phi(\Omega) = \sqrt[3]{\Xi(\Omega)}. \tag{4.35}$$

Finalement, la densité spectrale de puissance modifiée du signal est modélisée par un modèle LP autorégressif. Les coefficients LPC sont calculés à partir de l'autocorrélation $R[n]$ du signal $\Phi(\Omega_k)$ pour

$$\Omega_k = \Omega_{MIN} + k \cdot \frac{\Omega_{MAX} - \Omega_{MIN}}{N_B - 1} \tag{4.36}$$

où N_B est le nombre de bandes considérées (nombre d'échantillons pris dans $\theta(\Omega)$) et Ω_{MIN} et Ω_{MAX} sont respectivement la limite inférieure et supérieure de la plage d'analyse, en Barks. En résolvant les équations de Yule-Walker, les coefficients LPC sont obtenus. Le cepstre correspondant à cette représentation tout pôle du signal est ensuite calculé. Le cepstre obtenu est tronqué à N_C coefficients, ce qui permet d'arrondir davantage le spectre correspondant en plus de réduire la quantité de stockage nécessaire.

Implémentation de l'analyse PLP

L'analyse PLP implémentée utilise des trames de 960 échantillons, à une fréquence d'échantillonnage de 48kHz.

Il est possible d'obtenir $\Xi[l]$, la version discrète de $\Xi(\Omega)$ par multiplication matricielle avec la matrice $\mathbf{\Pi}$ à partir de la densité spectrale de puissance $P[k]$.

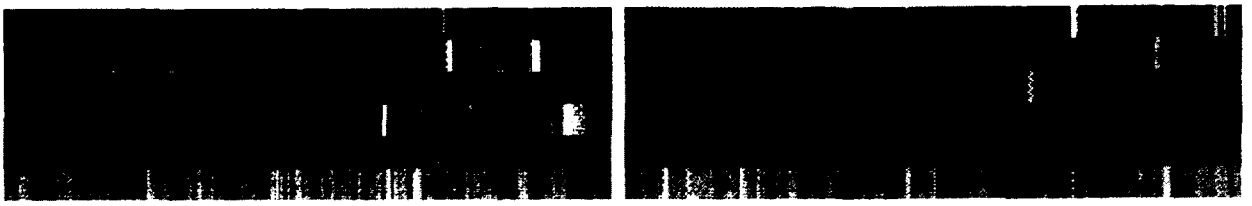
$$\mathbf{\Xi} = \mathbf{P\Pi} \tag{4.37}$$

$\mathbf{\Xi}$ et $\mathbf{P}$ sont les vecteurs contenant les échantillons de $\Xi[l]$ et $P[k]$, respectivement. $\mathbf{\Pi}$ est la matrice de transformation linéaire combinant le filtrage, le sous-échantillonnage et la pré-accentuation par une courbe isotonique. $\mathbf{\Pi}$ est de taille $N \times N_B$ où N est la longueur de la trame, soit 960 échantillons.

Après la compression par la racine cubique, les coefficients d'autocorrélation sont obtenus en prenant la TFDI de $\Phi[l]$. La TFDI est effectuée élément par élément plutôt qu'à l'aide de la FFT, étant donné qu'un nombre limité de coefficients d'autocorrélation est nécessaire. Les coefficients LPC, pour un prédicteur d'ordre P, sont ensuite calculés à l'aide des équations de Yule-Walker, résolues récursivement à l'aide de la récursion de Levinson-Durbin. Les coefficients cepstraux s'obtiennent à partir des coefficients LPC par la récursion suivante

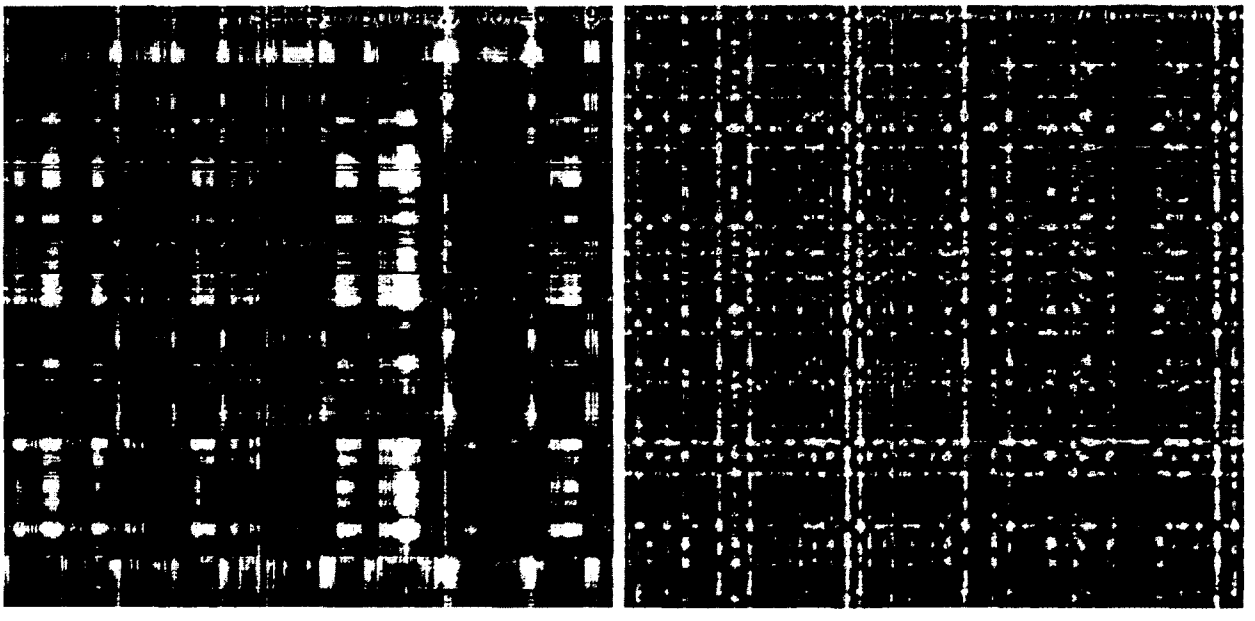
$$\begin{aligned} c[m] &= -a[m] + \tfrac{1}{m}\sum_{k=1}^{m-1}(-(m-k)\cdot a[k]c[m-k]) && \text{, pour } 1 \leq m \leq P \\ c[m] &= \tfrac{1}{m}\sum_{k=1}^{P}(-(m-k)\cdot a[k]c[m-k]) && \text{, pour } P < m \leq N_C \end{aligned} \tag{4.38}$$

Les figures 4.13 et 4.14 montrent les résultats de l'extraction des coefficients PLP et les matrices de coûts obtenus pour l'exemple (figures 4.1 et 4.2).



(a) Séquence de référence (b) Séquence de test

Figure 4.13 Comparaison des spectrogrammes PLP obtenus pour l'exemple. Chaque rangée montre la valeur d'un coefficient différent de l'analyse, en partant du haut de l'image, pour chaque trame (colonne) analysée.



(a) Énergie (b) Dérivée discrète de l'énergie.

Figure 4.14 Matrices de coûts obtenues pour l'analyse PLP pour l'exemple

Choix des paramètres d'analyse

Plusieurs paramètres de l'analyse PLP sont configurables. La présente section s'intéresse à l'influence de ceux-ci sur les performances de l'algorithme. Comme pour l'analyse MFCC et WDCTC, les résultats présentés dans cette section utilisent la métrique de performance donnée par l'équation 5.5.

La figure 4.15 présente plusieurs résultats obtenus avec différentes valeurs du nombre de bandes, de l'ordre du prédicteur linéaire, du nombre de coefficients et de l'ordre de l'estimateur de la dérivée. La position de chaque point montre le résultat obtenu à partir du vecteur résultat de l'analyse PLP sans pondération par rapport au résultat obtenu avec une pondération selon le logarithme de l'indice. La pondération par le logarithme de l'indice obtient de meilleurs résultats pour toutes les combinaisons de paramètres testées, avec une moyenne de -0.9669 contre -0.9503, sans pondération.

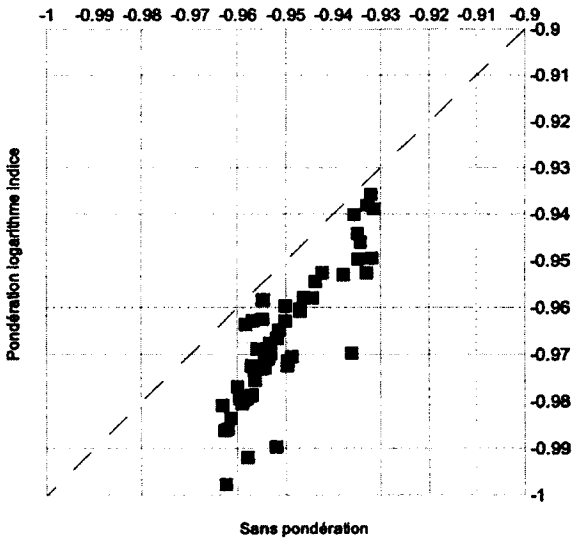


Figure 4.15 PLP : Comparaison des métriques de performance obtenues avec différents types de pondération

Tous les résultats subséquents utiliseront la pondération par le logarithme de l'indice. Les résultats obtenus pour les différentes combinaisons de valeurs de paramètres considérées sont présentés dans le tableau 4.6. Ces résultats seront analysés par groupements logiques dans les paragraphes suivants.

La figure 4.16 et les tableaux 4.7 et 4.8 montrent la moyenne des résultats obtenus pour différentes valeurs du nombre de bandes spectrales et de l'ordre du modèle LP autorégressif utilisé. Les meilleurs résultats sont obtenus en moyenne avec 16 bandes et un prédicteur

Tableau 4.6 PLP : Métrique de performance obtenue pour différentes valeurs de paramètres de l'analyse

	N. coeffs.	16			18			20		
Ord. Δ	Ord. pred. / N. coeffs.	5	10	15	5	10	15	5	10	15
5	5	-.9797	-.9709	-.9730	-.9794	-.9666	-.9724	-.9755	-.9714	-.9705
	6	-.9769	-.9544	-.9526	-.9804	-.9530	-.9497	-.9788	-.9526	-.9495
	7	-.9794	-.9977	-.9919	-.9806	-.9862	-.9836	-.9786	-.9809	-.9859
7	5	.9688	-.9597	-.9629	-.9694	-.9608	-.9580	-.9676	-.9603	-.9578
	6	.9724	-.9401	-.9381	-.9710	-.9442	-.9389	-.9699	-.9461	-.9358
	7	.9726	-.9896	-.9648	-.9728	-.9628	-.9636	-.9697	-.9583	-.9626

linéaire d'ordre 5 mais on remarque que le meilleur résultat parmi les essais effectués est obtenu avec un prédicteur d'ordre 10. Pour cette raison, ces deux jeux de paramètre seront testés dans le cadre du système complet, au prochain chapitre.

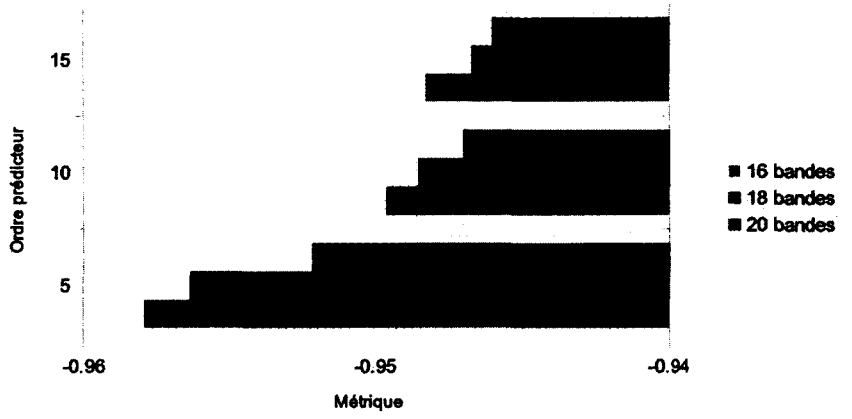


Figure 4.16 PLP : Moyennes des métriques de performance obtenues pour différentes valeurs de l'ordre du prédicteur et du nombre de bandes

Tableau 4.7 PLP : Moyennes de la métrique de performance obtenues pour différentes valeurs du nombre de bandes

N. bandes	16	18	20
Moy. métrique	-0.9692	-0.9663	-0.9650

La figure 4.17 et les tableaux 4.9 et 4.10 montrent cette fois-ci la moyenne des résultats obtenus en faisant varier le nombre de coefficients et l'ordre de l'estimateur de la dérivée. Les meilleurs résultats ont été obtenus avec 7 coefficients et un estimateur de la dérivée d'ordre 5.

Tableau 4.8 PLP : Moyennes de la métrique de performance obtenues pour différentes valeurs de l'ordre du prédicteur linéaire

Ord. pred.	5	10	15
Moy. métrique	-0.9746	-0.9642	-0.9618

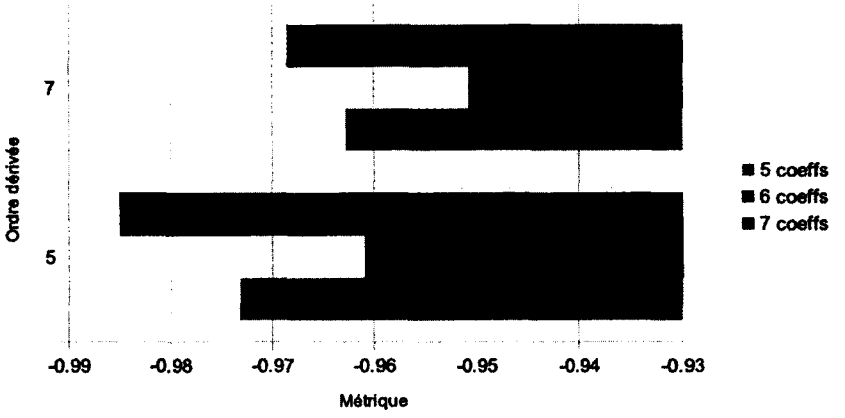


Figure 4.17 PLP : Moyennes des métriques de performance obtenues pour différentes valeurs de l'ordre de l'estimateur de la dérivée et du nombre de coefficients.

Tableau 4.9 PLP : Moyenne de la métrique de performance en fonction du nombre de coefficients

N. coeffs.	5	6	7
Moy. métrique	-0.9680	-0.9558	-0.9768

Tableau 4.10 PLP : Moyenne de la métrique de performance pour différentes valeurs de l'ordre de l'estimateur de la dérivée

N. coeffs.	5	7
Moy. métrique	-0.9730	-0.9607

Selon les moyennes obtenues, les meilleurs résultats seraient obtenus en effectuant l'analyse PLP avec 16 bandes, 5 coefficients, un modèle LP d'ordre 5 et un estimateur de la dérivée d'ordre 5. Cependant, le meilleur résultat obtenu l'a été avec 16 bandes, 7 coefficients, un modèle LP d'ordre 10 et un estimateur de la dérivée d'ordre 5. Pour cette raison, les deux combinaisons seront testées dans le système complet, au chapitre 6.

4.2 Autres paramètres

4.2.1 Niveau d'énergie

En plus des paramètres spectraux, il est utile de considérer l'énergie instantanée d'un signal. L'énergie du signal donne l'enveloppe temporelle des signaux à aligner. Bien que l'enveloppe temporelle ne permette pas d'effectuer un alignement très précis, l'énergie ne changeant pas nécessairement lors d'une transition de phonème vers un autre, elle permet de bien mesurer la structure grossière du signal. Les pauses, l'augmentation de l'énergie au début d'un mot, bref la dynamique vocale qui est particulièment présente dans le chant, seront corrélées à l'énergie du signal. De plus, certaines transitions entre différents phonèmes, particulièrement les transitions d'un phonème voisé à non-voisé et vice-versa seront marqués par une variation significative de l'énergie.

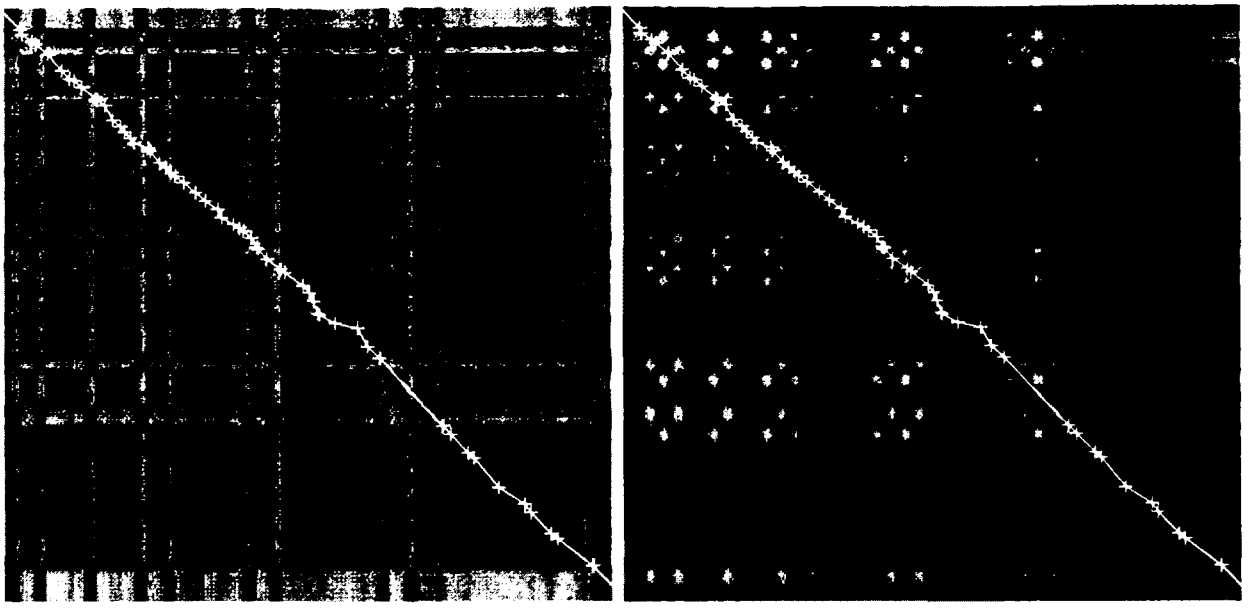
L'énergie moyenne d'une trame est calculée comme suit :

$$E = 10 \cdot \log_{10} \left(\frac{\sum_{n=0}^{N-1} (x[n] - \overline{x})^2}{N} \right) \tag{4.39}$$

où $\overline{x}$ est la valeur moyenne des échantillons $x[n]$ de la trame courante et N est la longueur d'une trame.

Les figures 4.18 et 4.19 montrent les résultats de l'extraction de l'énergie moyenne et les matrices de coûts obtenus pour l'exemple (figures 4.1 et 4.2).

Figure 4.18 Comparaison des énergies obtenues pour l'exemple



(a) Énergie (b) Dérivée discrète de l'énergie.

Figure 4.19 Matrices de coûts obtenues pour le paramètre *énergie* pour l'exemple

Le tableau 4.11 présente les résultats obtenus pour différentes valeurs de l'ordre de l'estimateur de la dérivée. La même métrique de performance que pour les paramètres spectraux est utilisée (décrite à la section 5.2).

Tableau 4.11 Énergie : Moyenne de la métrique de performance obtenue pour différentes valeurs de l'ordre de l'estimateur de la dérivée

Ord. dérivée	3	5	7
Moy. métrique	-0.8480	-0.7849	-0.8459

4.3 Sommaire

Dans ce chapitre, trois types d'analyses spectrales ont été détaillés. Pour chacune de celles-ci, au moins une configuration a été retenue pour le choix de l'algorithme final. Le tableau 4.12 présente ces candidats.

Une analyse reliée au niveau d'énergie des signaux a aussi été présentée et permet d'obtenir, à elle seule, une valeur de la métrique de performance de -0.8480.

Tableau 4.12 Résultats obtenus avec les meilleures configurations de chacune des analyses spectrales

Analyse	Param. analyse	Ord. derivée	Métrique
MFCC	$N_C = 14$, $N_F = 14$, pond. log. indice	7	−0.953
WDCTC	$N_C = 15$, sans pond.	5	−0.944
PLP	$N_B = 16$, $N_C = 5$, $N_{LPC} = 5$, pond. log. indice	5	−0.980
PLP	$N_B = 16$, $N_C = 7$, $N_{LPC} = 10$, pond. log. indice	5	−0.990

CHAPITRE 5

FONCTIONS DE COÛT

Idéalement, afin d'obtenir les meilleurs résultats possible, les matrices de coûts obtenues à l'aide des algorithmes d'extraction de paramètres présentés au chapitre précédent (chapitre 4) ne compteraient que des cellules dont le coût est nul pour tous les points alignés et infinis pour le reste. De cette façon, même un algorithme d'alignement peu performant parviendrait à obtenir un résultat parfait. Dans ce chapitre sera détaillé l'ajout d'une transformation non linéaire des coûts obtenus afin de se rapprocher le plus possible de la matrice de coût idéale. De plus, une métrique de performance permettant d'optimiser les paramètres de cette transformation ainsi que la technique d'optimisation utilisée seront présentés.

5.1 Transformation appliquée au coût

La figure 5.1 montre la densité de probabilité des coûts idéalisée, à la différence près que les coûts hors alignement sont fixés à 1 plutôt qu'à l'infini. L'intérêt de fixer le coût hors alignement à 1 est qu'il sera plus facile de combiner différents paramètres en utilisant des poids qui exprimeront l'importance relative de chaque paramètre. Si la valeur hors alignement n'était pas normalisée, la valeur des poids ne serait pas nécessairement indicatrice de l'importance relative du paramètre.

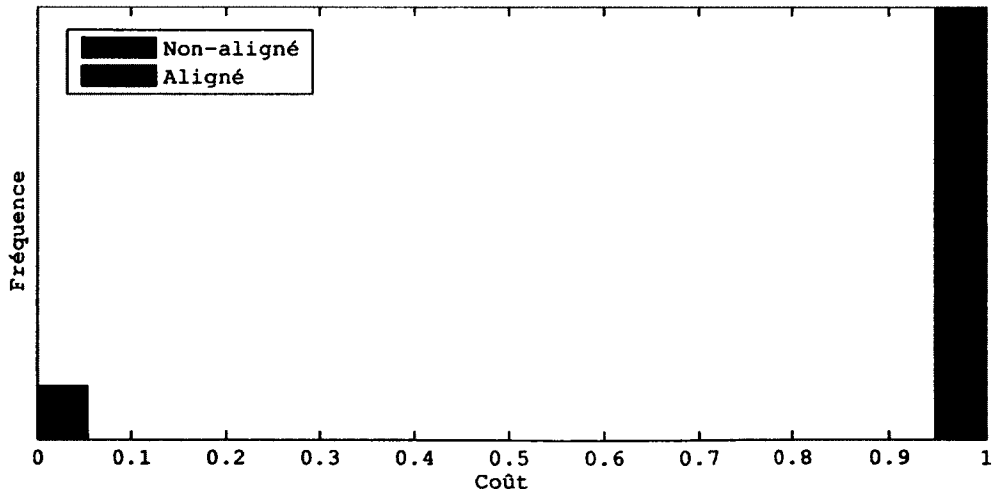


Figure 5.1 Densité de probabilité de coût idéale pour l'alignement

En utilisant des poids pour combiner plusieurs matrices de coût, on aura

$$c_t[i,j] = \sum_{p=1}^{N_p} w_p \cdot \hat{c_p}[i,j] \tag{5.1}$$

où N_p est le nombre de paramètres considérés et $\hat{c_p}$ et w_p sont respectivement le coût transformé et le poids correspondant au paramètre d'indice p.

La fréquence des coûts nuls, à la figure 5.1, est de beaucoup inférieure à la fréquence des coûts maximaux, étant donné qu'il y a beaucoup plus de cellules qui ne font pas partie de l'alignement des deux signaux que de cellules qui en font partie.

Les figures 5.3a et 5.3c montrent un exemple de matrice de coût matrice de coût obtenue pour une analyse MFCC ainsi que la densité de probabilité et les probabilités cumulées associées. Afin d'approcher la densité de probabilité idéale de la figure 5.1, une transformation non linéaire simple sera utilisée. La transformation doit nécessairement être non linéaire, sans quoi la forme de la densité de probabilité des coûts obtenue ne pourrait changer.

La figure 5.2 présente un graphique du coût de sortie de la transformation en fonction du coût d'entrée, pour la transformation non linéaire utilisée. La sortie de la transformation utilisée est donnée par

$$\hat{c} = \begin{cases} 0 & \text{, pour } c \leq a' \\ \frac{c-a'}{b'} & \text{, pour } a' < c \leq a' + b' \\ 1 & \text{, pour } c > a' + b' \end{cases} . \tag{5.2}$$

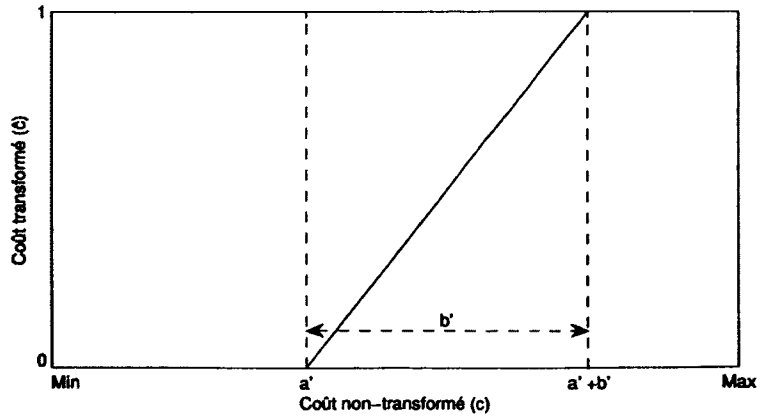


Figure 5.2 Transformation non linéaire appliquée aux coûts

Les valeurs des paramètres a' et b' seront obtenus à l'aide d'une optimisation en boucle fermée, c'est-à-dire en effectuant la recherche de la combinaison qui fera en sorte que l'alignement à la sortie du système soit optimal. Le faible nombre de paramètres nécessaire pour caractériser la transformation est un avantage puisqu'il limite le nombre de degrés de liberté du système à optimiser, ce qui accélérera grandement le processus. Une fonction non linéaire plus complexe aurait pu également être choisie, une expression linéaire par morceaux avec plus de segments ou une expression polynomiale, par exemple.

En réalité, l'optimisation est effectuée à partir d'une expression alternative de la transformation qui fait intervenir des coûts d'entrée normalisés entre 0 et 1 décrite par

$$\hat{c} = \begin{cases} 0 & \text{, pour } c_n \leq a \\ \frac{c_n - a}{b} & \text{, pour } a < c_n \leq a + b \\ 1 & \text{, pour } c_n > a + b \end{cases} . \tag{5.3}$$

où

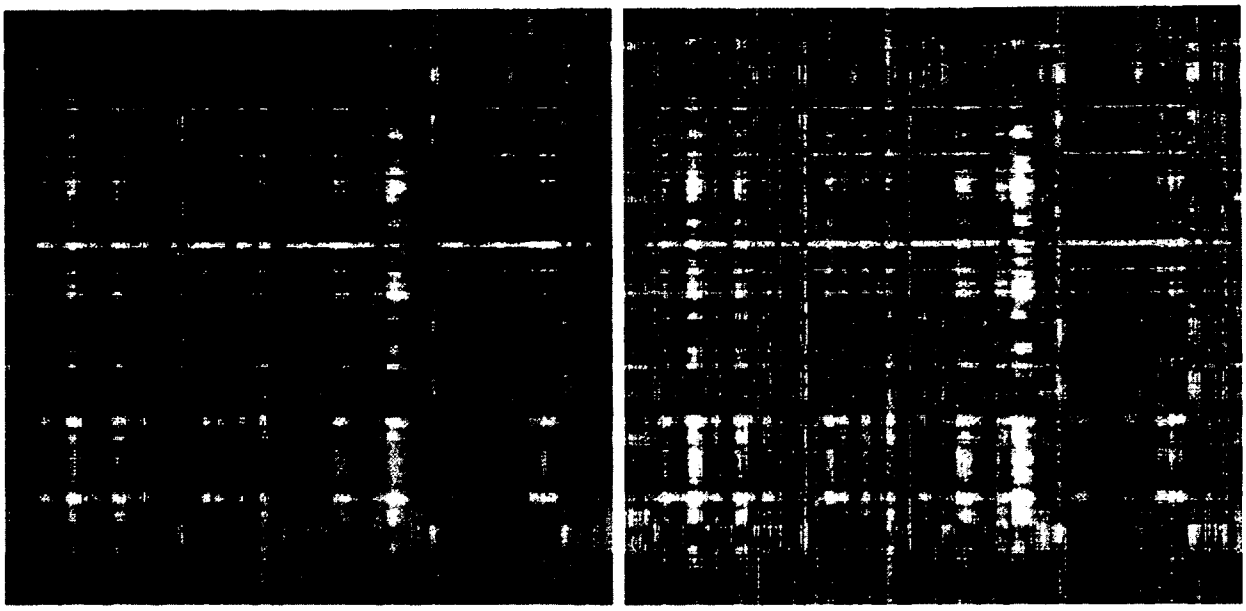
$$c_n = \frac{c - min(\mathbf{c})}{max(\mathbf{c}) - min(\mathbf{c})}. \tag{5.4}$$

Les coûts minimum et maximum utilisés pour la normalisation du coût d'entrée sont ceux de la matrice de coût obtenue avant transformation.

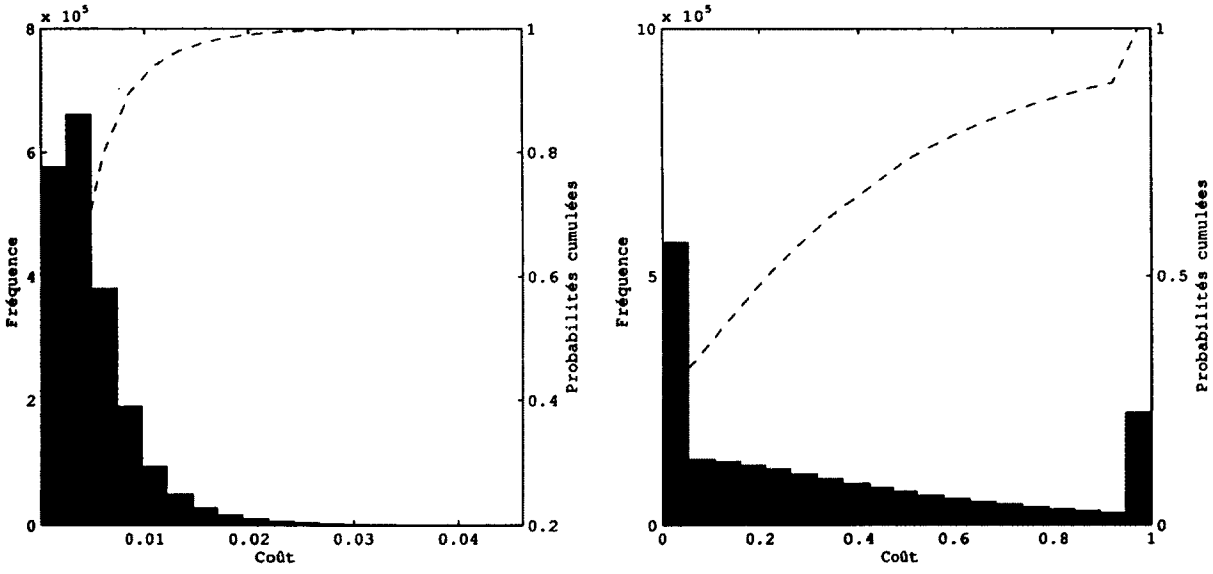
Les figures 5.3b et 5.3d montrent le résultat obtenu après transformation en utilisant des valeurs choisies à la main qui permettent d'obtenir de bons résultats d'alignement. Même si ces valeurs ne sont pas optimales, la figure 5.3d montre une densité de probabilité qui est beaucoup plus près de la densité idéale (figure 5.1) qu'avant la transformation (figure 5.3c). De plus, le chemin aligné (approximativement la diagonale, dans ce cas précis) est plus clair dans la figure 5.3b.

5.2 Effet des paramètres et métrique de performance

L'efficacité de la fonction de transformation sera mesurée en boucle fermée, c'est-à-dire en intégrant la fonction de transformation au système d'alignement complet. Afin de s'assurer d'avoir des résultats représentatifs de la plus grande variété de signaux possibles, les deux séquences décrites à la section 4.1, c'est-à-dire deux assemblages de plusieurs paires de séquences avec différents interprètes et différentes qualités d'interprétations, seront utilisées. Dans cette section, l'effet des paramètres de la fonction de transformation sur les résultats obtenus sera analysé et une métrique de performance qui permettra une calibration automatique des paramètres à l'aide de techniques d'optimisation sera développée.



(a) Matrice de coût avant transformation (b) Matrice de coût après transformation



(c) Densité de probabilité avant transformation (d) Densité de probabilité après transformation

Figure 5.3 Exemple d'application de la transformation non linéaire sur la matrice de coût et sur la densité de probabilité du coût

La figure 5.4 montre l'erreur quadratique moyenne sur l'alignement obtenue en fonction des paramètres de la fonction de transformation utilisée. Cette figure a été produite en échantillonnant la réponse du système sur 70 valeurs du paramètre a distribuées uniformément entre 0 et 0.3 et 40 valeurs du paramètre b distribuées uniformément entre 0 et 1. Les résultats augmentent de façon similaire pour les valeurs de a entre 0.3 et 1 et ne sont donc pas présentés afin de mieux pouvoir voir les résultats dans la zone d'intérêt.

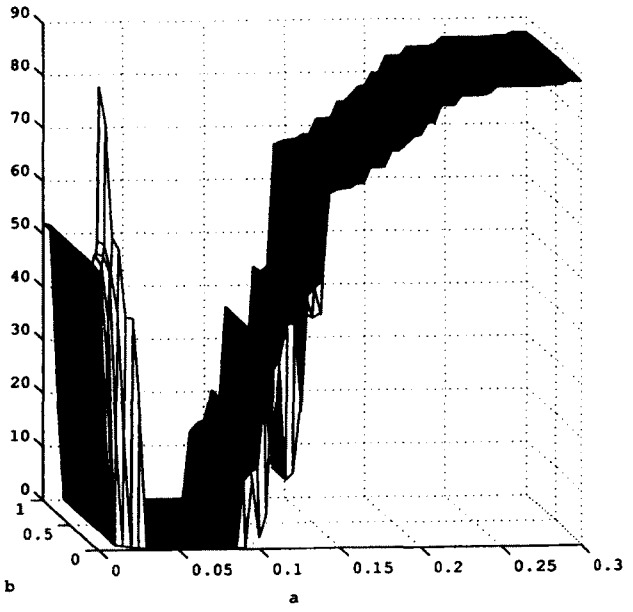


Figure 5.4 Erreur moyenne quadratique sur l'alignement en fonction des paramètres de la fonction de transformation utilisée

La figure 5.4 montre que le résultat obtenu sans transformation, qui est équivalent à utiliser les paramètres $a = 0$ et $b = 1$, produit des résultats médiocres. On peut également remarquer que l'influence du paramètre b est limitée, le résultat obtenu variant très peu pour une valeur de a donnée. La figure montre aussi que le système atteint un plancher à une valeur s'approchant de 0 et que ce plancher est atteint sur une grande partie du domaine observé.

Le grand plancher observé à la figure 5.4 pose un problème pour l'optimisation du système. En effet, même si l'alignement résultant est excellent sur ce domaine, il est très difficile de savoir quelle est réellement la meilleure combinaison de a et b, étant donné la nature discontinue du résultat de l'alignement. La nature de l'algorithme de la DTW fait en sorte qu'une faible variation d'un seul coût peut faire basculer l'alignement dans un chemin totalement différent. Il est donc plus prudent de mesurer les performances de l'algorithme différemment.

La mesure de performance qui a été utilisée compare le coût cumulé obtenu aux points d'alignement connus avec les coûts cumulés des cellules environnantes. Plus la matrice de coût obtenue permet de bien distinguer le chemin optimal, plus les coûts cumulés des points d'alignement devraient être faibles en comparaison avec les cellules voisines. Les valeurs de coûts cumulés sont centrées et réduites afin de mieux exprimer la performance des cellules alignées par rapport aux cellules environnantes. En ne faisant pas cette dernière opération, une cellule de faible coût cumulé pourrait indiquer une bonne performance même si ses voisines ont un coût plus faible. On cherche à obtenir la plus faible valeur possible. La mesure de performance utilisée est calculée selon

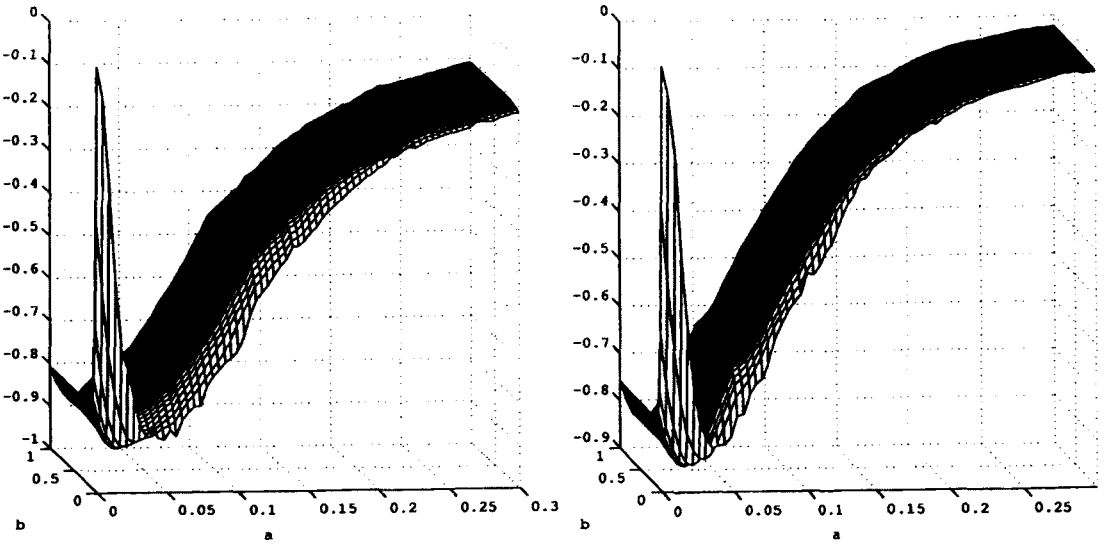
$$V = \sum_{i}^{N} \frac{C[\widehat{m_i}, n_i] - \overline{\mathbf{C_i}}}{Var(\mathbf{C_i})} \qquad (5.5)$$
$$\mathbf{C_i} = \left[\; C[\widehat{m_i} - W, n_i] \quad \ldots \quad C[\widehat{m_i}, n_i] \quad \ldots \quad C[\widehat{m_i} + W, n_i] \; \right]$$

où W est la largeur de la zone autour de la cellule alignée considérée, $\widehat{m_i}$ est la position de la séquence de référence considérée comme alignée au point d'alignement i et n_i est la position de la séquence de test au point d'alignement.

L'alignement qui est utilisé dans le calcul est un alignement qui a été réalisé de façon manuelle. Dans le cas de la paire de séquences assemblées, qui est utilisée pour réaliser les optimisations qui suivent, l'alignement manuel est constitué de 172 points généralement placés au début et à la fin de chaque mot et aux transitions entre les différents phonèmes prononcés.

Les figures 5.5 montrent la performance obtenue avec la métrique de l'équation 5.5 pour différentes valeurs des paramètres a et b pour deux paires de séquences différentes. En 5.5a, les performances obtenues pour la paire de séquences assemblées sont montrées. La surface semble cette fois-ci continue et ne présente pas le plancher de la figure 5.4, ce qui permettra à l'algorithme d'optimisation de fixer les paramètres de façon plus optimale.

La valeur du métrique de performance a aussi été évaluée sur le même domaine pour d'autres paires de séquence, afin de vérifier que les paramètres optimaux obtenus se généraliseront bien pour d'autres paires de séquences. La figure 5.5b montre le résultat obtenu pour la paire de séquences de l'exemple 1, qui est décrite à la section 4.1. Les résultats obtenus sont très semblables et semblent donc indiquer que les paramètres de la fonction de transformation obtenus pour la paire de séquences assemblées produiront aussi de bons résultats avec les autres paires de séquences.



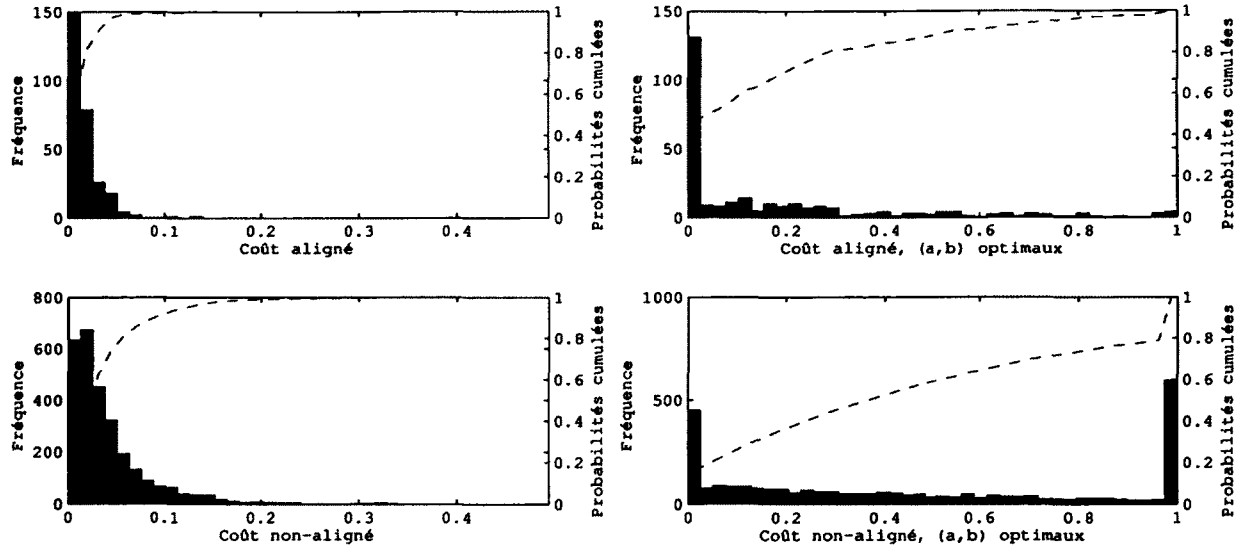
(a) Assemblage (b) Exemple 1

Figure 5.5 Moyenne du coût cumulé centré réduit aux points d'alignement, en fonction des paramètres de la fonction de transformation utilisée

L'effet de la transformation sur la densité de probabilité des coûts est illustré à la figure 5.6. La figure 5.6a, qui présente la même forme que le cas $(a, b) = (0, 1)$, sauf pour l'échelle en x, permet d'expliquer les résultats médiocres obtenus sans transformation. En effet,la densité de probabilité des coûts non alignés montre que ceux-ci ne sont pas suffisamment élevés, ce qui fait en sorte de ne pas pénaliser suffisamment les chemins passant par des cellules non alignées, même lorsque les vecteurs de paramètres correspondants sont très distants.

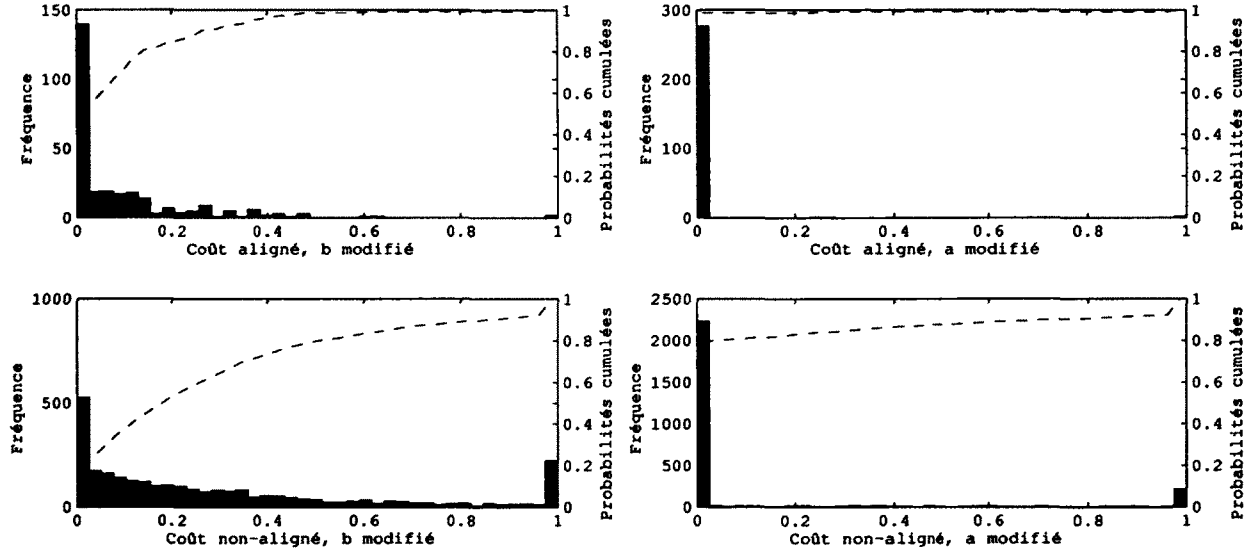
La figure 5.6c, où la valeur de b a été bonifiée de 0.1 par rapport à la valeur optimale trouvée, montre un résultat un peu supérieur, semblable à celui qui est obtenu pour la fonction de transformation optimale (5.6b), mais la pente plus faible de son segment central dûe à la plus grande valeur de b fait en sorte qu'un nombre moins élevé de cellules non alignées atteignent le coût plafond.

La figure 5.6d, où la valeur de a a été bonifié de 0.1 par rapport à la valeur optimale trouvée, montre un résultat beaucoup moins bon qui peut s'expliquer par le fait que presque toutes les cellules non alignées ont un coût nul.



(a) Sans transformation

(b) Optimale ($a = 0.0174$, $b = 0.1026$)



(c) b modifié ($a = 0.0174$, $b = 0.2026$)

(d) a modifié ($a = 0.1174$, $b = 0.1026$)

Figure 5.6 Densités de probabilité du coût pour différentes valeurs des paramètres de la fonction de transformation

5.3 Optimisation des paramètres de la transformation

Afin de déterminer quels sont les paramètres a et b de la fonction de transformation qui permettent d'obtenir la meilleure valeur de la métrique de performance de l'équation 5.5, une technique d'optimisation directe est utilisée.

Étant donné que l'expression des dérivées partielles de la métrique de performance n'est pas connue, voire impossible à obtenir, les méthodes d'optimisation classiques tels la descente du gradient, la méthode du gradient conjugué ou la méthode de Newton-Raphson ne sont pas applicables directement. Dans le cas de l'optimisation basée sur les simulations, même s'il était possible d'estimer les dérivées avec des différences finies, les erreurs sur son estimation et la nature bruitée de la fonction à optimiser font en sorte que les techniques d'optimisation classiques ne sont pas adaptées et ne produiront pas de bons résultats [Kolda *et al.*, 2003].

Différentes techniques qui n'utilisent pas les dérivées sont disponibles. Parmi celles-ci, on retrouve entre autres le recuit simulé, les algorithmes génétiques et la recherche directe par motif (*direct search* ou *pattern search*, en anglais). L'algorithme qui a été utilisé est la recherche directe par motif. La principale raison qui justifie ce choix est que le nombre moyen d'évaluations de la fonction à optimiser s'est révélé être inférieur aux deux autres techniques, pour le problème étudié. L'évaluation d'une valeur de la métrique de performance nécessite de démarrer l'algorithme complet puisqu'il est implémenté en tant que programme indépendant. L'alignement de deux séquences et l'enregistrement des matrices de coûts cumulés prennent de deux à trois secondes, pour des séquences d'environ trente secondes, d'où l'importance de minimiser le nombre d'évaluations de la fonction objectif, surtout lorsqu'on désire l'évaluer pour plusieurs valeurs de paramètres différents, comme au chapitre 4. De plus, la recherche directe par motif se prête bien à la parallélisation, ce qui permet d'évaluer la fonction sur plusieurs points à la fois, tirant avantage des architectures de processeur à coeurs multiples. La figure 5.7 montre un exemple de la vitesse de convergence des différents algorithmes d'optimisation considérés.

La technique de la recherche directe par motif est apparue avec les travaux de [Hooke et Jeeves, 1961]. Une généralisation de ces travaux a conduit à la recherche généralisée par motif (traduit de l'anglais *Generalized Pattern Search (GPS)*). La méthode consiste à définir un treillis autour d'un point central, appelé centre de sonde, défini par

$$M_k = \{\mathbf{x_k} + \Delta_k \mathbf{d_i} : i \in \mathbb{N} \mid 1 \leq i \leq N_D\} \tag{5.6}$$

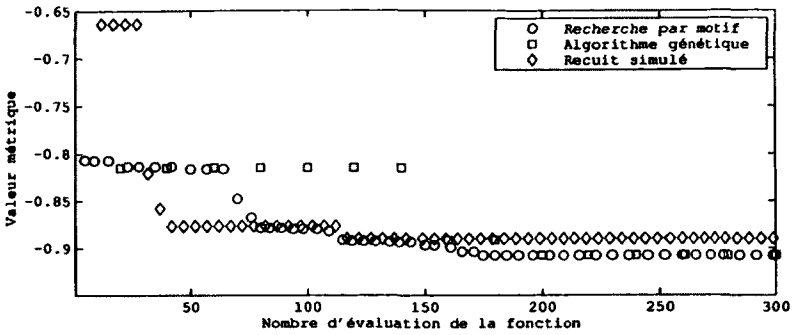


Figure 5.7 Comparaison de la vitesse de convergence des différents algorithmes d'optimisation considérés

où $\mathbf{x_k}$ et Δ_k sont respectivement le centre de sonde et la taille du pas à l'itération k et $\mathbf{d_i}$ est le $i^{\text{ème}}$ vecteur de l'ensemble générateur $\mathbf{D}$. L'ensemble générateur $\mathbf{D}$, de taille N_D, doit former une base positive de $\mathbb{R}^N$, N étant la dimension de l'espace sur lequel l'optimisation est faite, soit le nombre de paramètres à optimiser.

Une base positive est formée par un ensemble de vecteurs positivement indépendants qui engendrent un espace donné. Un ensemble de vecteurs est positivement indépendant s'il n'existe pas de combinaison positive de vecteurs de cet ensemble qui puisse générer un vecteur de cet l'ensemble, en excluant la combinaison triviale (le vecteur lui-même). On définit une combinaison positive d'un ensemble D comme

$$\mathbf{c} = \sum_{i=1}^{N_D} a_i \mathbf{d_i} \mid a_i \geq 0. \tag{5.7}$$

Une base positive doit contenir au moins N_D+1 et au plus $2N_D$ éléments. La nécessité que la base soit positive s'explique par le fait que la taille du pas Δ_k est strictement positive et qu'il serait donc impossible de parcourir tout l'espace avec une base non positive.

L'algorithme GPS débute avec le choix d'un premier centre de sonde. Dans une première étape facultative appelée recherche, on peut tenter de choisir une meilleure valeur du centre de sonde à l'aide d'un autre algorithme. La recherche peut même être globale et faire intervenir un algorithme complexe tel un algorithme génétique. Si un meilleur point est trouvé, il sera utilisé comme centre de sonde pour l'étape de la sonde locale.

Ensuite, on évalue la fonction f à optimiser sur le treillis M_k défini autour du centre de sonde $\mathbf{x_k}$. Si on trouve un point sondé tel que $f(\mathbf{m_{ki}}) < f(\mathbf{x_k})$), celui-ci sera utilisé comme prochain centre de sonde. Il serait possible d'arrêter aussitôt qu'un point sondé satisfait

le critère, mais pour le problème étudié, étant donné qu'il est avantageux de paralléliser la sonde, le meilleur point sondé sera choisi.

À la fin de chaque itération, la taille du pas de l'itération suivante Δ_{k+1} est calculée selon

$$\begin{aligned} \Delta_{k+1} &= \alpha\Delta_k \quad \text{, si } f(\mathbf{m_{ki}}) < f(\mathbf{x_k})) \\ \Delta_{k+1} &= \beta\Delta_k \quad \text{, sinon} \end{aligned} \tag{5.8}$$

où α et β sont respectivement les facteurs d'expansion et de contraction du treillis. α est un nombre réel supérieur à 1 tandis que β est un réel compris entre 0 et 1. La figure 5.8 montre un exemple d'évolution du treillis de sonde pour l'algorithme GPS avec une base positive canonique de taille $N_D = 2N$ et des facteurs d'expansion et de contraction du treillis $\alpha = \frac{3}{2}$ et $\beta = \frac{2}{3}$. À l'étape initiale, «A», un des points de sonde obtient une valeur inférieure, soit 2, à celle du centre de sonde, pour lequel la fonction a la valeur 3. Le trellis est donc recentré sur ce point de sonde et subit une expansion de facteur $\alpha = \frac{3}{2}$, tel que montré à l'étape «B». Cette fois-ci, puisqu'aucun point de sonde n'a une valeur inférieure à celle du centre de sonde, le trellis subit une contraction de facteur $\beta = \frac{2}{3}$ (étape «C»). À l'étape suivante, une autre translation et expansion du trellis aurait eu lieu, un point de sonde ayant révélé une valeur inférieure au centre de sonde.

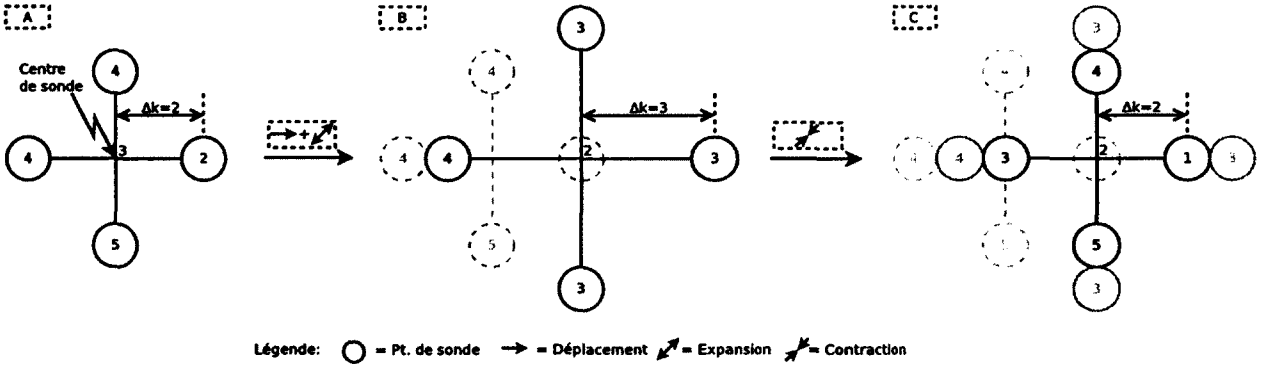


Figure 5.8 Exemple d'évolution du treillis dans l'algorithme de recherche généralisée par motif

Plusieurs critères d'arrêts sont possibles, mais généralement la taille du pas est utilisée. Si celle-ci devient inférieure à un seul prédéterminé, l'algorithme arrête.

Afin de faire l'optimisation de la fonction de transformation du coût, une alternative de l'algorithme GPS appelée *Generating Set Search (GSS)* [Kolda *et al.*, 2003] est utilisée. Une différence avec l'algorithme GPS est que la sonde est jugée réussie lorsqu'elle permet une décroissance suffisante (par rapport à une décroissance simple, pour GPS) de la valeur de la fonction, pour un point sondé.

Aussi, l'ensemble générateur $\mathbf{D}$, qui devient $\mathbf{D_k}$ peut changer à chaque itération de l'algorithme et est modifié dans le cas où on impose des frontières ou contraintes linéaires aux valeurs des paramètres. Dans ces cas, l'ensemble générateur est adapté de façon à mieux approcher la contrainte sans sortir du domaine d'optimisation, ce qui serait un problème de l'algorithme GPS.

[Kolda *et al.*, 2003] présente certaines conditions à respecter afin de s'assurer de la convergence de l'algorithme vers un point stationnaire (dont le gradient est nul) du problème. L'analyse formelle de ces conditions n'a pas été faite pour le problème étudié, mais en pratique, à grande échelle, la fonction de transformation apparaît assez lisse comme en témoigne la figure 5.5 et dans tous les cas, l'algorithme d'optimisation a pu converger vers un résultat satisfaisant.

L'implémentation de l'algorithme GSS qui a été utilisée est celle du logiciel Matlab. Afin de limiter la taille du domaine de recherche et de s'assurer que le coût instantané de certaines cellules des matrices de coûts sature, la contrainte linéaire $a + b \leq 1$ a été utilisée et la valeur des paramètres a et b a été restreinte à l'intervalle $[0, 1]$. La base choisie est une base positive à $N_D = 2N$, soit 4 vecteurs. L'étape optionnelle de recherche n'a pas été utilisée. Les facteurs α et βont été fixés respectivement à 2 et à $0,5$.

5.4 Sommaire

Dans ce chapitre, une transformation appliquée aux coûts instantanés permettant de rendre plus robuste l'alignement a été proposée. Celle-ci est une simple fonction linéaire par morceaux paramétrable via deux paramètres. Les coûts obtenus à la sortie de ces fonctions de transformations possèdent une densité de probabilité plus près de la densité idéale. Les coûts obtenus sont aussi contraints à la plage $[0, 1]$, ce qui permet de combiner plus facilement des matrices, en utilisant des poids qui représentent l'importance relative de chacune des matrices constituantes.

Afin d'obtenir les valeurs optimales des paramètres des fonctions de transformation, il a été nécessaire de développer une métrique qui présente une variation beaucoup plus continue que l'erreur sur l'alignement. Cette métrique est caractérisée par une bonne corrélation avec l'écart quadratique moyen sur l'alignement.

Une méthode d'optimisation des paramètres des fonctions de transformation a été élaborée, se basant sur l'algorithme de recherche directe par motif, une méthode d'optimisation globale.

CHAPITRE 6

ASSEMBLAGE ET RÉSULTATS

Dans ce chapitre, l'assemblage du système complet et les procédures d'optimisation requises seront détaillés. Plusieurs assemblages différents seront testés afin d'arriver au meilleur système d'alignement possible. Les résultats pour chacun des algorithmes mis à l'épreuve seront présentés et des pistes d'amélioration seront présentées.

6.1 Assemblage du système

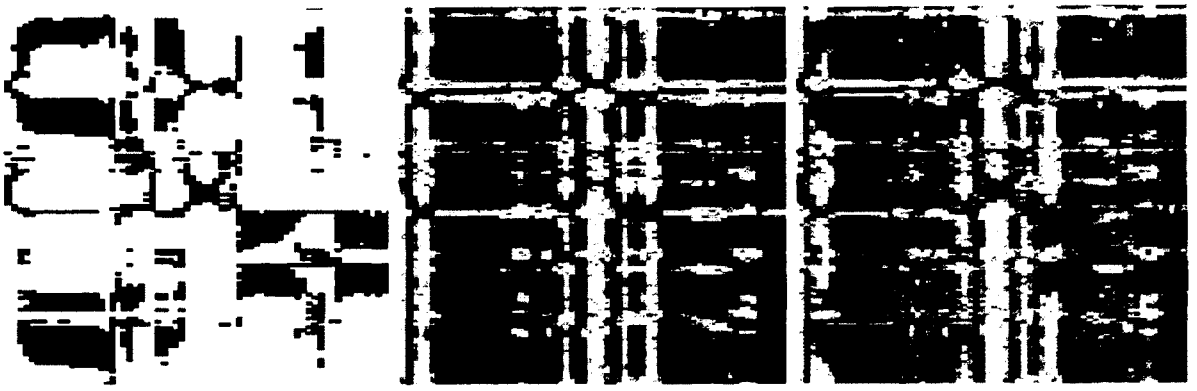
Les deux chapitres précédents ont traité des techniques d'analyse permettant d'extraire l'information pouvant être utilisée dans un contexte d'alignement de la voix chantée et de techniques qui permettent d'améliorer les matrices de coût associées à ces informations. Plusieurs matrices de coûts pourront être combinées afin d'obtenir un meilleur résultat global.

À la section 5.1, la fonction de transformation du coût instantané a été présentée. Celle-ci a la caractéristique de fournir une valeur de sortie qui est bornée entre 0 et 1. Cela fait en sorte qu'il est plus facile de combiner plusieurs matrices de coût, en permettant de les additionner avec une pondération qui reflète l'importance relative de chacune des contributions à la matrice de coût instantanée globale (équation 5.1).

La figure 6.1 montre un exemple de combinaison d'une matrice de coût direct et d'une matrice de coût de la dérivée, pour l'analyse PLP.

La combinaison de plusieurs matrices permet d'obtenir une valeur de la métrique de performance de l'équation 5.5 plus faible, donc meilleure, qu'avec les matrices de coût prises individuellement. La valeur des poids pour chaque matrice de coût, qui expriment l'importance relative du paramètre du signal correspondant, est obtenue par un processus très similaire à celui utilisé pour l'obtention des paramètres des fonctions de transformations présenté à la section 5.2.

En effet, la méthode qui a été utilisée est encore une fois une optimisation à l'aide de la technique de recherche par motif. Les paramètres optimaux pour chaque fonction de transformation ayant été obtenus préalablement pour chacune des matrices de coût utilisées, seuls les poids relatifs seraient à déterminer. Cependant, en optimisant tous les



(a) Coût direct (b) Coût de la dérivée (c) Coût combiné

Figure 6.1 Exemple de combinaison de la matrice de coût direct et de la dérivée, pour l'analyse PLP

paramètres des fonctions de transformation et les poids relatifs de façon simultanée, il est possible d'obtenir de meilleurs résultats. Cela s'explique par le fait que le domaine sur lequel une solution est recherchée est beaucoup plus grand. À titre d'exemple, pour une analyse PLP avec $N_B = 16$, $N_{LP} = 5$, $N_C = 5$, un estimateur de dérivée d'ordre 5 et une pondération par le logarithme de l'indice, en optimisant les poids et chacune des fonctions de transformation de façon indépendante, un résultat de -0.9939 a été obtenu, pour une combinaison de la matrice directe et de la matrice «dérivée». En optimisant tous les paramètres en même temps, la métrique de performance est passée à -0.9955. La figure 6.2 présente une comparaison des résultats obtenus pour les deux méthodes d'optimisation, pour quelques-uns des systèmes évalués.

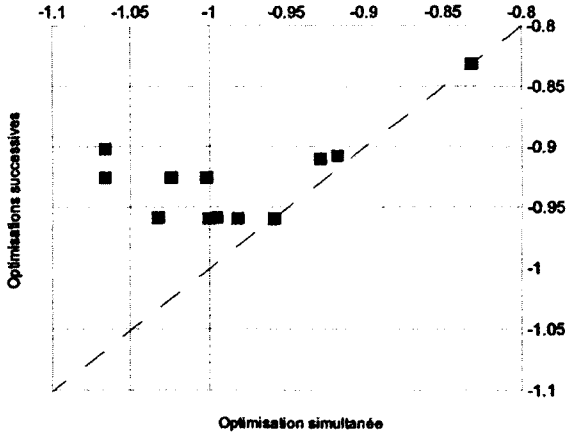


Figure 6.2 Métrique de performance avec une optimisation simultanée par rapport à plusieurs optimisations successives

Pour chacune des matrices de coûts, le poids associé est contraint entre 0 et 3. L'optimisation utilisée est également l'algorithme GSS, tel qu'implémenté par Matlab, avec une base positive à $N_D = 2N$ vecteurs, avec $N = 3N_M$ (2 paramètres pour la fonction de transformation et un poids relatif pour chaque matrice de coût). Les facteurs α et βont été fixés respectivement à 2 et à 0.5. L'optimisation a aussi été effectuée en utilisant la même paire de séquences assemblées, afin de fournir une bonne diversité de signaux et ainsi d'éviter une surspécialisation des paramètres optimisés.

Il aurait été plus intéressant de contraindre les poids des différentes matrices entre 0 et 1, de telle sorte que la somme de ces poids soit égale à 1. De cette façon, la plage de valeurs possible dans la matrice de coûts combinée serait de 0 à 1. En pratique, cependant, en imposant la contrainte linéaire $\sum_i w_i = 1$ où w_i sont les poids de chacune des matrices de coûts considérées, la performance observée diminue, pour un même nombre d'itérations de l'algorithme GSS. La déformation du treillis requise pour se conformer à cette contrainte explique cette diminution de performance. En pratique, cependant, il est possible de tout simplement utiliser diviser les poids obtenus par la somme des poids pour obtenir le même résultat.

6.2 Résultats obtenus

Au chapitre 4, plusieurs paramètres spectraux ont été identifiés et comparés. Les meilleurs résultats obtenus, pour chacune des analyses étudiées, seront repris dans le cadre du système d'alignement complet. Le tableau 6.1 présente les différents paramètres spectraux qui sont considérés comme des candidats pour l'algorithme d'alignement final. À ce tableau s'ajoute l'analyse du niveau d'énergie, présentée en 4.2.1, qui sera désignée par «ME» et qui utilise un estimateur de la dérivée d'ordre 3.

Tableau 6.1 Analyses spectrales candidates pour l'algorithme final

Désignation	Analyse	Paramètres utilisés	Ord. derivée
MFCC1	MFCC	$N_C = 14$, $N_F = 14$, pond. log. indice	7
WDCTC1	WDCTC	$N_C = 15$, sans pond.	5
PLP1	PLP	$N_B = 16$, $N_C = 5$, $N_{LP} = 5$, pond. log. indice	5
PLP2	PLP	$N_B = 16$, $N_C = 7$, $N_{LP} = 10$, pond. log. indice	5

Les séquences audio qui ont été utilisées pour évaluer la performance des systèmes évalués sont décrites dans le tableau 6.2. Ces séquences sont utilisées pour former les paires de séquences jumelées du tableau 6.3. Pour chacune de ces paires, un alignement manuel

comptant approximativement un point pour chaque phonème a été réalisé. Ces alignements manuels permettent d'évaluer les performances du système d'alignement assemblé.

Puisque le système sera évalué avec des paires d'interprétations qui ont été enregistrées et alignées dans le cadre de ce projet, il n'est pas possible de comparer les résultats obtenus avec les autres algorithmes répertoriés dans la littérature. Au moment d'écrire ces lignes, aucun cadre d'évaluation commun aux problèmes d'alignement de la voix chantée n'était disponible.

Tableau 6.2 Liste des interprétations utilisées pour l'évaluation et l'analyse du système d'alignement

Dés.	Chanson	Interprète	Description
YWD1	You Will You Won't(The Zutons)	David B.	Interprété le plus fidèlement possible à l'originale. Aucune erreur dans le texte, tous les mots sont bien intelligibles. 1m07s
YWD2	You Will You Won't(The Zutons)	David B.	Interprété assez fidèlement. 2 erreurs détectables. Un mot mal prononcé et un mot dont le début est erroné. 1m05s.
YWE1	You Will You Won't(The Zutons)	Eugénie P.	Interprété assez fidèlement, mais avec une déviation importante du rythme original sur une courte période. Un mot répété et un mot erroné. Quelques débuts de fou rire. 52s.
YWR1	You Will You Won't(The Zutons)	Eric J.	Interprété intentionnellement à un tempo beaucoup plus rapide que l'originale. Sur un extrait d'environ une seconde, les mots sont mal prononcés et difficiles à comprendre. 31s.
MFD1	More Than A Feeling(Boston)	David B.	Quelques déviations importantes du tempo par rapport à l'originale. Quelques moments où le *pitch* est intentionnellement constant (monotone). Mots parfaitement intelligibles. 24s.
MFR1	More Than A Feeling(Boston)	Eric J.	Interprétation assez fidèle sauf pour quelques fausses notes lorsque l'originale est très aigüe. Les mots sont tous bien intelligibles. 21s.
CR1	Creep(Radiohead)	Eric J.	Interprétation assez fidèle. Quelques fausses notes, mais le texte est bien intelligible. 40s.
CF1	Creep(Radiohead)	François B.	Interprétation assez fidèle. Tous les mots sont bien prononcés. 35s.

Tableau 6.3 Paires de séquences utilisées pour évaluer le système d'alignement

Dés.	Séquences référence	Durée	Séquences test	Durée
YW1	YWD1	1m07s	YWR1	31s
YW2	YWD1	1m07s	YWE1	52s
YW3	YWD1	1m07s	YWD2	1m05s
ASM2	YWD1, YWD1, YWD1, MFR1	37s	YWR1, YWE1, YWD2, MFD1	30s

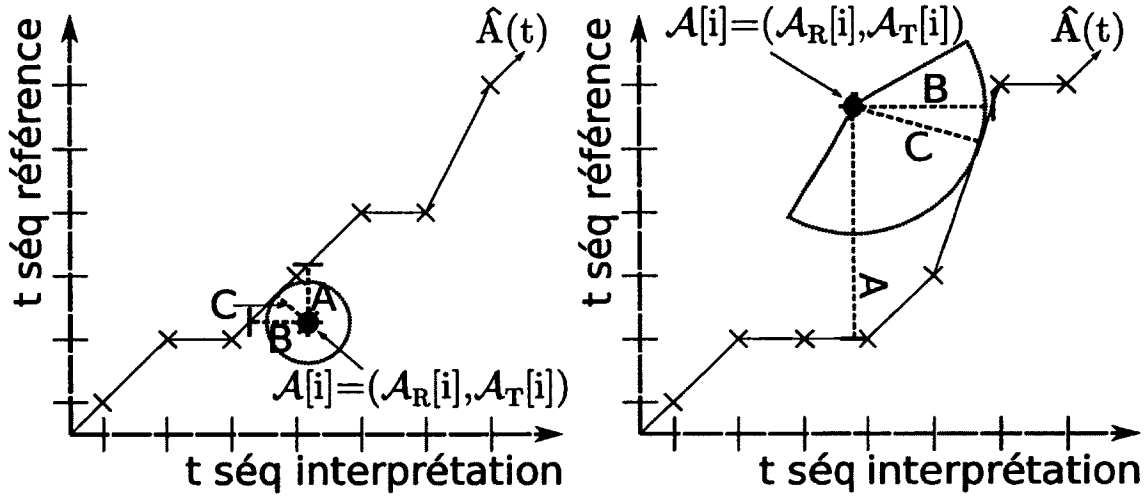
Afin d'évaluer la performance des systèmes d'alignement proposés, l'écart quadratique moyen entre l'alignement obtenu et l'alignement manuel est utilisé :

$$MSE = \frac{\sum_{i=1}^{N_A} e^2[i]}{N_A}$$
$$e[i] = \min(\mathcal{A}_R[i] - \hat{A}(\mathcal{A}_T[i]), d_x(\hat{A}, \mathcal{A}[i])) \tag{6.1}$$

où $\mathcal{A}[i] = (\mathcal{A}_R[i], \mathcal{A}_T[i])$ est le $i^{\text{ème}}$ des N_A points d'alignement manuels, donnés en secondes, $\hat{A}$ est une version interpolée de l'alignement A obtenu et d_x est la distance minimale en x entre la fonction $\hat{A}(t)$ et le point $\mathcal{A}[i]$ (écart sur la position de la séquence de test, pour une position dans la séquence de référence donnée). La distance d_x est calculée selon

$$d_x(\mathcal{A}[i], \hat{A}) = \min(\mathbf{t} - \mathcal{A}_T[i]), \mathbf{t} = \left\{ t | \hat{A}(t) = \mathcal{A}_R[i] \right\}. \tag{6.2}$$

La figure 6.3 illustre la mesure de distance utilisée pour deux exemples de points d'alignement. Les mesures A et B correspondent respectivement aux termes $\mathcal{A}_R[i] - \hat{A}(\mathcal{A}_T[i])$ et $d_x(\hat{A}, \mathcal{A}[i])$ de l'équation 6.1. La plus faible de ces deux mesure donne l'erreur sur l'alignement pour le point considéré.



(a) Exemple 1 (b) Exemple 2

Figure 6.3 Illustration du calcul de l'erreur d'alignement pour un point

Il aurait sans doute été préférable d'utiliser la distance absolue (non restreinte à l'axe x ou y), désignée par C à la figure 6.3, pour calculer l'écart quadratique, mais pour des raisons de complexité, cette option a été écartée.

L'écart quadratique moyen est surtout utilisé pour comparer les différents algorithmes d'alignement puisque d'un algorithme à l'autre, la matrice de coûts cumulés risque de très peu changer. La valeur de la métrique de performance de l'équation 5.5 sera privilégiée pour le choix du jeu de paramètres utilisés, puisque sa valeur dépend beaucoup moins de la performance de l'algorithme d'alignement que l'écart quadratique moyen ; la métrique de l'équation 5.5 n'utilise pas l'alignement obtenu.

Le tableau 6.4 présente les résultats obtenus pour différents systèmes, en utilisant l'algorithme proposé «B», présenté à la section 3.2.2. Pour chaque candidat du tableau 6.1, quatre systèmes ont été testés. D'abord, le paramètre spectral direct a été testé, puis on a ajouté la composante dérivée estimée, désignée par l'ajout du symbole Δ (voir section 4.1). Ensuite, la matrice de coût directe relative à l'énergie des signaux a été ajoutée et enfin sa dérivée a été intégrée. Notez qu'une taille d'expansion T_E (eq. 3.6) de 400 cellules a été utilisée.

L'analyse PLP obtient les meilleurs résultats, suivie de la WDCTC et finalement des MFCC. Les deux combinaisons testées pour l'analyse PLP obtiennent des résultats semblables, mais l'analyse PLP2 obtient de meilleurs résultats pour les combinaisons moins complexes. Pour cette raison, la configuration de paramètres retenue sera celle du système 8, soit le jeu de paramètres PLP2+ΔPLP2+ME+ΔME.

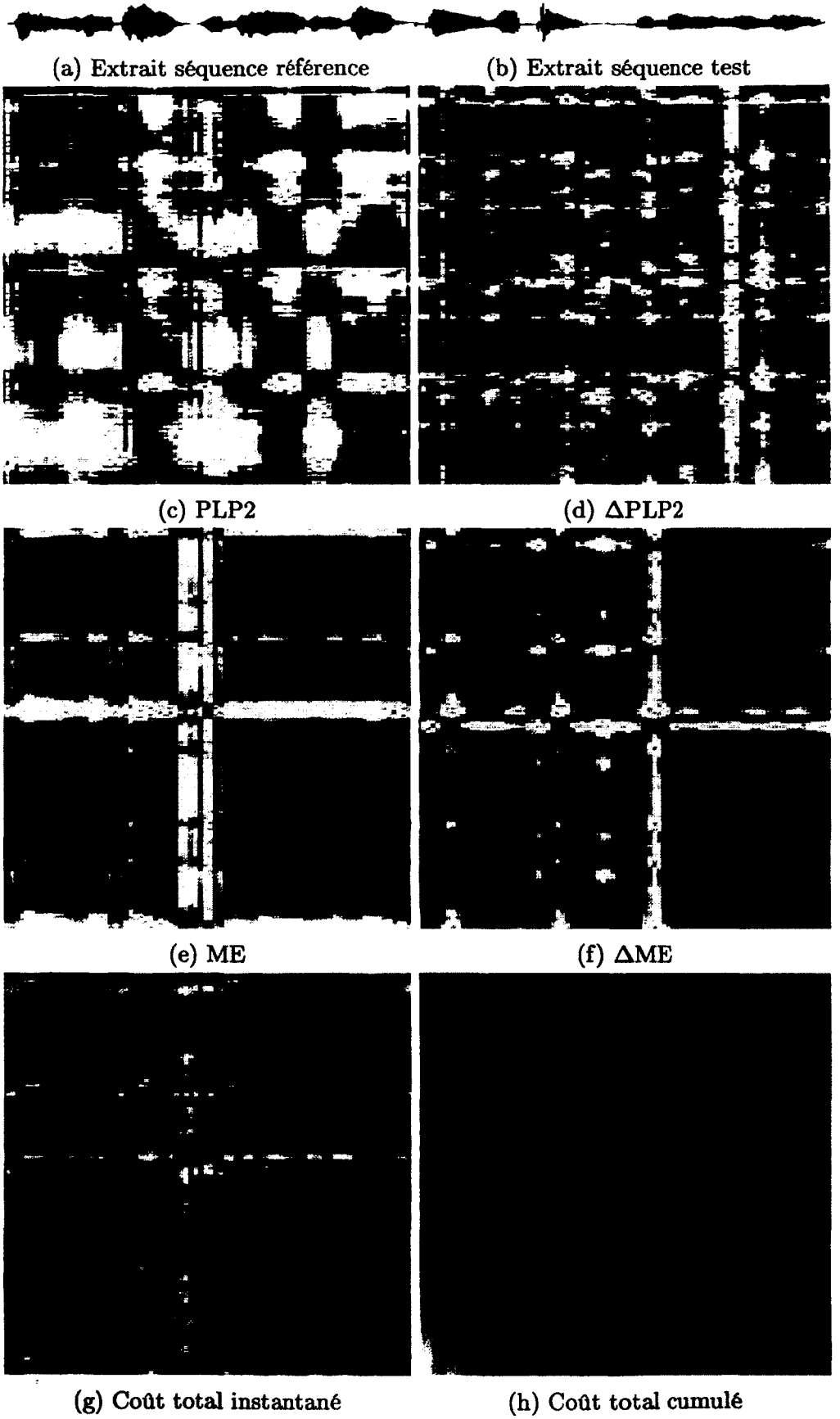
La figure 6.4 montre des agrandissement des matrices de coûts pour chaque paramètre de la configuration retenue ainsi que la matrice de coûts totale et la matrice de coûts cumulés obtenue (en utilisant l'algorithme «B»). Les poids utilisés pour chacune des matrices, obtenus par optimisation, sont décrits par $0.2587 \cdot \mathrm{PLP2} + 0.2500 \cdot \Delta\mathrm{PLP2} + 0.2266 \cdot \mathrm{ME} + 0.2500 \cdot \Delta\mathrm{ME}$.

Plusieurs variantes de l'algorithme de la DTW ont été détaillées au chapitre 3. Le tableau 6.5 montre les résultats obtenus avec ces différents algorithmes, pour la configuration de paramètres retenue. Pour l'algorithme de Dixon, décrit à la section 3.2.1, le paramètre déterminant la taille de la zone de recherche, T_E, a été fixée à 400 cellules, ce qui est aussi le cas pour les algorithmes «A» et «B». De plus, le nombre d'expansions consécutives dans la même direction, pour l'algorithme de Dixon, a été limité à 3. Le facteur d'amortissement D utilisé par l'algorithme «B» a été fixé à 5.

Les statistiques pertinentes sur l'erreur d'alignement, pour les paires YW2, YW3 et ASM2 et le système 8B, sont présentées au diagramme à moustache à la figure 6.5 où la médiane et les quartiles 1 et 3 sont représentés. Les points représentés par des «+» sont les erreurs

Tableau 6.4 Résultats obtenus pour différents systèmes complets

			Moyenne écart quadratique				Métrique de performance			
Sys.	Analyses	Algo.	YW2	YW3	ASM2	Moy.	YW2	YW3	ASM2	Moy.
1	PLP 1	B	11.018	0.007	21.122	10.716	-0.854	-0.980	-0.918	-0.917
2	$(1+\Delta)$PLP1	B	0.041	0.007	0.017	0.022	-0.985	-1.018	-0.995	-0.999
3	$(1+\Delta)$PLP1+ME	B	0.010	0.006	0.017	0.011	-1.047	-1.027	-1.033	-1.035
4	$(1+\Delta)$PLP1+$(1+\Delta)$ME	B	0.006	0.006	0.023	0.011	-1.056	-1.042	-1.067	-1.055
5	PLP2	B	9.231	0.006	0.058	3.098	-0.882	-0.971	-0.929	-0.927
6	$(1+\Delta)$PLP2	B	0.018	0.006	0.006	0.010	-0.984	-1.027	-1.002	-1.004
7	$(1+\Delta)$PLP2+ME	B	0.009	0.006	0.005	0.007	-1.051	-1.053	-1.024	-1.043
8	$(1+\Delta)$PLP2+$(1+\Delta)$ME	B	0.005	0.007	0.023	0.012	-1.054	-1.045	-1.066	-1.055
9	MFCC1	B	0.083	0.045	0.312	0.147	-0.805	-0.804	-0.832	-0.813
10	$(1+\Delta)$MFCC1	B	0.050	0.019	0.104	0.058	-0.929	-0.957	-0.958	-0.948
11	$(1+\Delta)$MFCC1+ME	B	0.009	0.011	0.102	0.041	-1.007	-0.978	-0.982	-0.989
12	$(1+\Delta)$MFCC1+$(1+\Delta)$ME	B	0.006	0.010	0.101	0.039	-1.000	-0.996	-1.001	-0.999
13	WDCTC1	B	0.039	0.019	0.061	0.039	-0.902	-0.911	-0.899	-0.904
14	$(1+\Delta)$WDCTC1	B	0.009	0.009	0.025	0.014	-1.040	-1.070	-0.938	-1.016
15	$(1+\Delta)$WDCTC1+ME	B	0.020	0.011	0.006	0.012	-1.046	-1.042	-0.986	-1.025
16	$(1+\Delta)$WDCTC1+$(1+\Delta)$ME	B	0.016	0.009	0.103	0.043	-1.036	-1.036	-0.973	-1.015



(a) Extrait séquence référence (b) Extrait séquence test

(c) PLP2 (d) ΔPLP2

(e) ME (f) ΔME

(g) Coût total instantané (h) Coût total cumulé

Figure 6.4 Matrices de coûts obtenues pour la configuration 8, pour la paire de séquences ASM2 (agrandissements)

Tableau 6.5 Moyenne de l'écart quadratique pour différents types d'algorithme d'alignement

			Moyenne écart quadratique			
Dés.	Analyses	Algo.	YW2	YW3	ASM2	Moy.
8Z	$(1+\Delta)$PLP2+$(1+\Delta)$ME	Dixon	0.9783	0.0290	0.0067	0.3437
8A	$(1+\Delta)$PLP2+$(1+\Delta)$ME	A	0.0054	0.0154	0.0227	0.0124
8B	$(1+\Delta)$PLP2+$(1+\Delta)$ME	B	0.0053	0.0067	0.0237	0.0119

qui sont une distance de l'intervalle interquartille ($[Q1, Q3]$) de plus d'une fois et demie l'écart interquartille ($Q3 - Q1$).

L'erreur moyenne, son écart-type ainsi que les valeurs maximales observées sont donnés au tableau 6.6. L'erreur d'alignement moyenne est de 56.7 ms avec un écart-type moyen de 86.7 ms. L'erreur maximale positive est supérieure à l'erreur maximale négative, ce qui peut s'expliquer par le fait que les plus grands écarts sont généralement observés au début des mots, après des périodes de silence. La position dans la séquence de référence est alors généralement sous-estimée, ce qui produit une erreur positive.

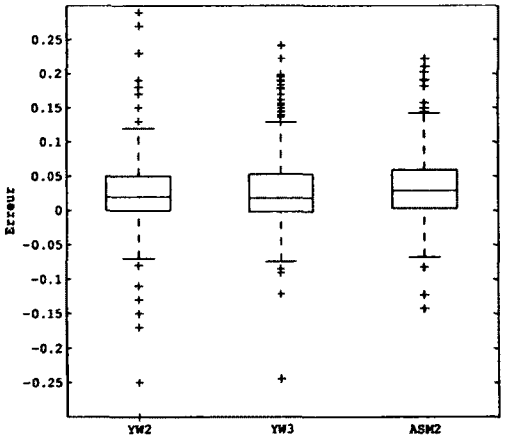


Figure 6.5 Diagramme à moustache de l'erreur d'alignement pour le système 8B

Tableau 6.6 Statistiques des erreurs obtenues avec le système 8B

Statistique	YW2	YW3	ASM2
Moyenne	0.047s	0.044s	0.073s
Écart-type	0.055s	0.069s	0.136s
Err. max.	0.390s	0.982s	0.782s
Err. min.	-0.350s	-0.245s	-0.143s

6.3 Analyse des résultats et discussion

6.3.1 Influence de l'algorithme de la DTW

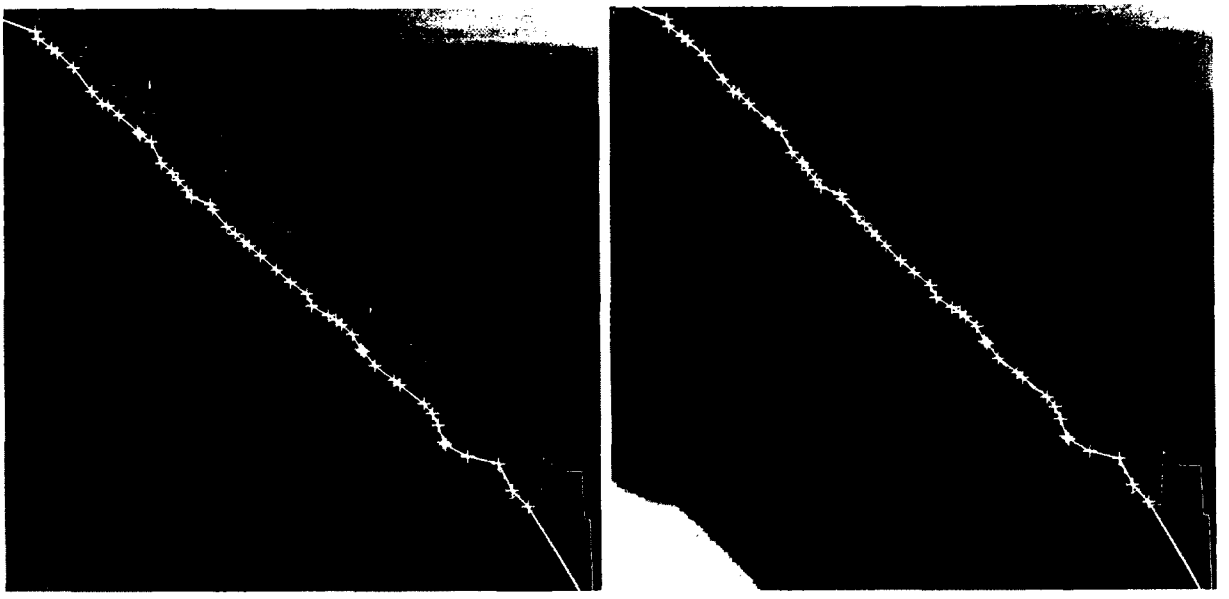
Il est important de noter qu'à paramètre T_E égal, la zone de recherche sera plus grande avec l'algorithme de Dixon, comme le montrent les figures 6.6a et 6.6b. Le trait en blanc est l'alignement obtenu.

(a) Dixon (b) Algorithme «B»

Figure 6.6 Comparaison des matrices de coûts cumulés obtenues pour différents algorithmes

Les figures 6.7a et 6.7b montrent un agrandissement des matrices de coûts cumulés pour l'algorithme de Dixon et l'algorithme «B». La matrice montre un écart relativement important entre l'alignement obtenu et l'alignement réalisé à la main (en blanc, avec des croix à chaque point d'alignement) et ce, même si les coûts cumulés semblent minimaux vis-à-vis l'alignement optimal. La raison pour laquelle l'alignement obtenu n'est pas plus près de l'optimal est que ces cellules ont été calculées après que l'alignement ait été déterminé, tel que décrit dans la section 3.2.1.

Les algorithmes «A» et «B» obtiennent des performances similaires, ce qui est prévisible lorsque l'algorithme d'alignement performe bien, puisque la différence entre les deux algorithmes réside dans la façon de déterminer le domaine sur lequel les matrices sont calculées.



(a) Dixon (b) Algorithme «B»

Figure 6.7 Agrandissement d'une section des matrices de coûts cumulés pour différents algorithmes

Problèmes encourus avec la paire YW1

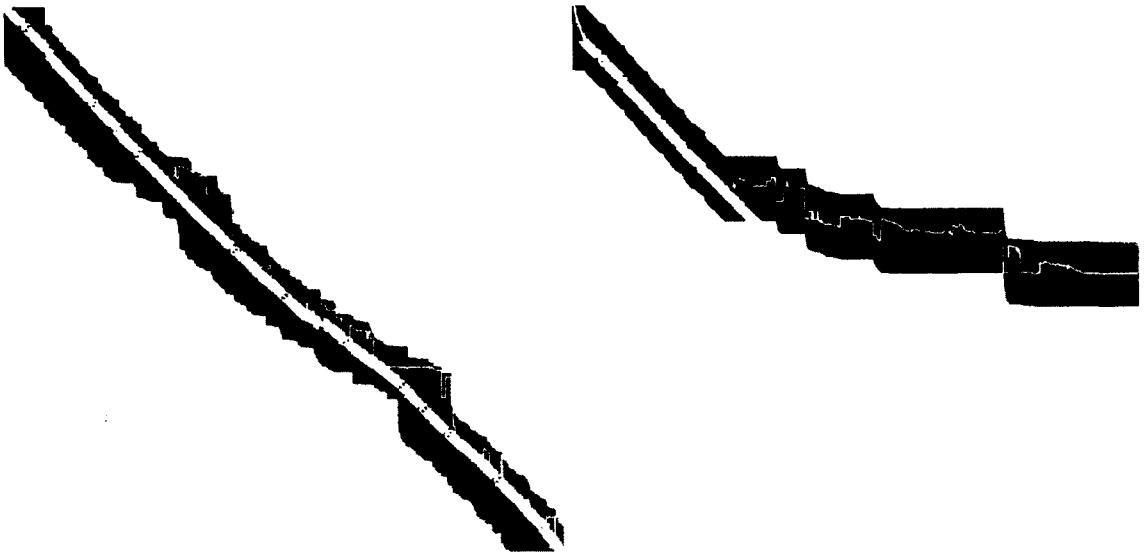
Les résultats présentés aux tableaux 6.4 et 6.5 ne montrent pas les résultats obtenus pour la paire de séquences YW1, qui a la particularité qu'une des deux interprétations est très accélérée et moins intelligible. La raison pour cette omission est que les résultats qui ont été obtenus pour cette paire sont décevants. Ceux-ci sont présentés dans le tableau 6.7.

Tableau 6.7 Résultats obtenus pour la paire YW1

Dés.	Analyses	Algo.	MSE	Métrique
8Z	$(1+\Delta)$PLP2+$(1+\Delta)$ME	Dixon	0.755	-0.87
8A	$(1+\Delta)$PLP2+$(1+\Delta)$ME	A	54.945	-0.7888
8B	$(1+\Delta)$PLP2+$(1+\Delta)$ME	B	54.792	-0.8402

Aucun des algorithmes n'a permis d'obtenir un alignement satisfaisant, mais l'algorithme de Dixon a pu éventuellement retrouver son chemin comme le montre la figure 6.8a, ce qui n'a pas été le cas avec l'algorithme «B» (figure 6.8b). Les valeurs de la métrique de performance obtenues avec cette paire de séquence sont largement moins bonnes que celles qui ont été obtenues pour avec les autres paires testées, ce qui explique les résultats d'alignement médiocre obtenus.

À la figure 6.9, un agrandissement des matrices de coûts obtenues près de la zone où l'algorithme «B» diverge est présenté. On peut remarquer que dans la partie droite de la



(a) Dixon (b) Algorithme «B»

Figure 6.8 Comparaison des alignements obtenus avec l'algorithme de Dixon et l'algorithme «B» pour la paire de séquences YW1

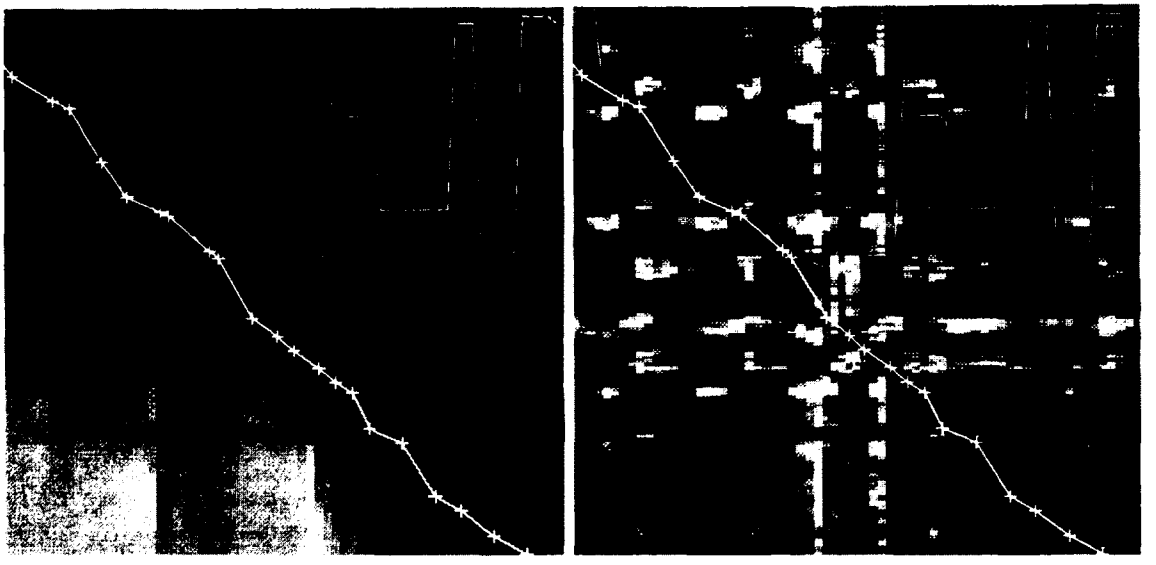
matrice, la matrice de coût cumulé se sépare en deux branches. Les coûts inférieurs de la branche du haut font en sorte que l'alignement diverge. Assez rapidement, le domaine de calcul de l'algorithme «B» ne couvre plus l'alignement de référence et l'algorithme n'arrive pas à recouvrer ce faux pas.

La zone de coût instantané élevé approximativement au centre de la figure 6.9b est la cause de l'affaiblissement de la branche du bas. En effet, il n'existe pas de chemin de faible coût permettant de faire le pont entre le «chemin» d'en haut à gauche et d'en bas à droite.

La matrice de coûts instantanés, sauf pour la portion centrale, laisse entrevoir le chemin qui devrait être emprunté par l'algorithme d'alignement. S'il était possible de moins pénaliser les cellules de droite, au bas de la matrice de coûts cumulés, il serait possible d'améliorer le résultat.

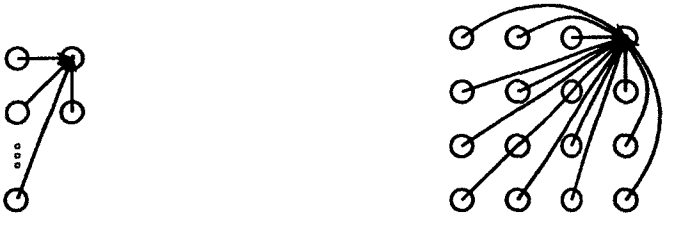
Quelques tentatives de modifications de l'algorithme ont été faites afin de permettre de faire des sauts qui permettraient d'éviter d'absorber tous les coûts instantanés nécessaires pour faire le pont entre les deux bouts de chemin. En modifiant les contraintes locales utilisées pour la DTW, il est possible d'autoriser des sauts de plus d'une cellule. Des essais ont été réalisés avec deux types de contraintes locales différentes. Celles-ci sont illustrées à la figure 6.10.

La contrainte *saut* permet d'effectuer un saut à partir de la cellule de coût cumulé minimum parmi les cellules de la colonne précédente. Il est évidemment nécessaire de pénaliser un



(a) Coûts cumulés (b) Coûts instantanés

Figure 6.9 Agrandissement des matrices de coûts instantanés et cumulés au point de divergence de l'algorithme «B», pour la paire YW1



(a) Contrainte *saut* (b) Contrainte *matrice étendue*

Figure 6.10 Contraintes locales testées afin d'améliorer les résultats obtenus avec la paire YW1

tel raccourci. Le facteur de pondération pour le saut a été posé égal à

$$w_s[i,j] = \frac{C[A[j-1],j-1]}{i+j} \cdot F_p(i - A[j]) + F_k(i - A[j]) + F_c c[i,j] \tag{6.3}$$

Pour certaines valeurs des coefficients F_p, F_k et F_c, il a été possible d'obtenir de très bons résultats avec la paire YW1, mais, pour ces mêmes valeurs, l'alignement des autres paires de séquences se mettait à diverger. Il n'a pas été possible de trouver des paramètres qui permettaient d'obtenir de bons résultats dans tous les cas.

La deuxième contrainte testée permet un déplacement à partir de toutes les cellules sur une grille de dimension 4×4 (en excluant la cellule calculée, évidemment). Afin d'éviter tout biais, les facteurs de pondération utilisés sont égaux à la distance de Manhattan (norme L_1) entre la cellule de départ et la cellule calculée. Les résultats obtenus sont intéressants parce qu'ils restent excellents dans tous les cas, mais malheureusement l'amélioration n'a pas été suffisante pour empêcher la divergence de l'algorithme «B» avec la paire YW1. Les résultats obtenus, pour le système 8B modifié, sont détaillés dans le tableau 6.8. Ces résultats montrent qu'en faisant une étude plus rigoureuse des différents types de contraintes locales, il serait possible d'améliorer de façon significative les résultats obtenus.

Tableau 6.8 Résultats obtenus avec la contrainte *matrice étendue*, pour le système 8B

Résultat	YW1	YW2	YW3	ASM2
MSE	38.021	0.009	0.014	0.016
Métrique	-0.941	-1.042	-1.002	-1.008

En analysant l'algorithme de DTW de Dixon, il a été possible de constater que la zone de recherche parvenait mieux à se recentrer sur l'alignement réel. Une des hypothèses avancées est que l'algorithme de Dixon fait la recherche du meilleur coût cumulé sur une rangée en plus d'une colonne. Lorsque le coût cumulé le plus faible n'est pas trouvé sur la dernière colonne calculée, autrement dit que les coûts cumulés des dernières colonnes deviennent prohibitifs, l'algorithme a une indication que la position présumée dans la séquence de référence est trop retardée. Une nouvelle rangée est alors calculée, ce qui a pour effet de recentrer la zone de recherche, dans la plupart des cas.

Cette hypothèse a été mise à profit en modifiant l'algorithme «B». La modification consiste à changer la façon dont est calculé l'incrément sur la position de la cellule centrale. À

l'équation 3.6, le terme $d[j]$ devient

$$d[j] = \begin{cases} d[j-1] + \frac{A[j]-P[j-1]}{D}, & \text{si } A[j] - P[j-1] \geq 0 \\ & \text{ou } C[P[j-1], R[j]] > C[A[j], j] \\ d[j-1] + \frac{j-R[j]}{D}, & \text{sinon} \end{cases} \tag{6.4}$$

$$R[j] = \arg\min_{x} C[P[j-1], x], \quad x = \{j, j-1, ..., j-T_E+1\} \ .$$

La modification implique que le nombre de cellules qui seront examinées afin de déterminer la position présumée ainsi que le déplacement du point central double. La figure 6.11 montre le résultat obtenu avec l'algorithme modifié (désigné «C»), pour la paire YW1. Les résultats sont essentiellement inchangés pour les autres paires de séquences testées comme en témoigne le tableau 6.9.

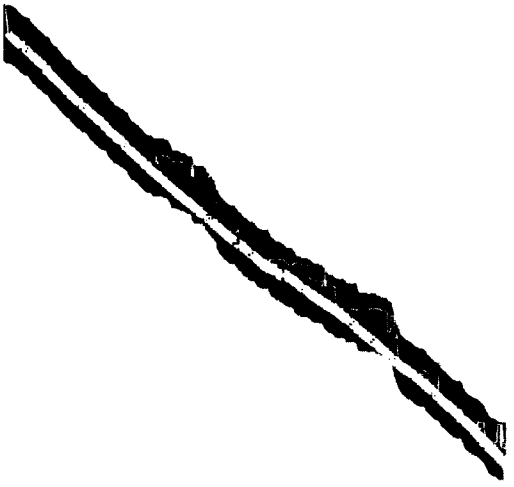


Figure 6.11 Matrice de coûts et alignement obtenus avec l'algorithme «C» et la paire YW1

Tableau 6.9 Résultats obtenus avec l'algorithme «C», pour le système 8B

Résultat	YW1	YW2	YW3	ASM2
MSE	1.145	0.005	0.007	0.024

6.3.2 Progression dans les phonèmes voisés étendus

Le système obtenu a été testé sur quelques paires de séquences qui n'ont pas été utilisées pour l'entraînement des fonctions de transformation et des poids relatifs. La figure 6.12 montre un exemple d'un alignement et de la matrice de coûts cumulés associée réalisé pour la paire d'interprétations CR1 (utilisée comme référence) et CF1 (utilisée comme séquence

de test). L'alignement manuel n'a pas été réalisé pour cette paire, ce qui fait en sorte qu'il n'est pas possible de fournir de résultats quantitatifs. L'algorithme n'a pas divergé et une inspection de l'alignement en quelques points révèle une bonne performance.

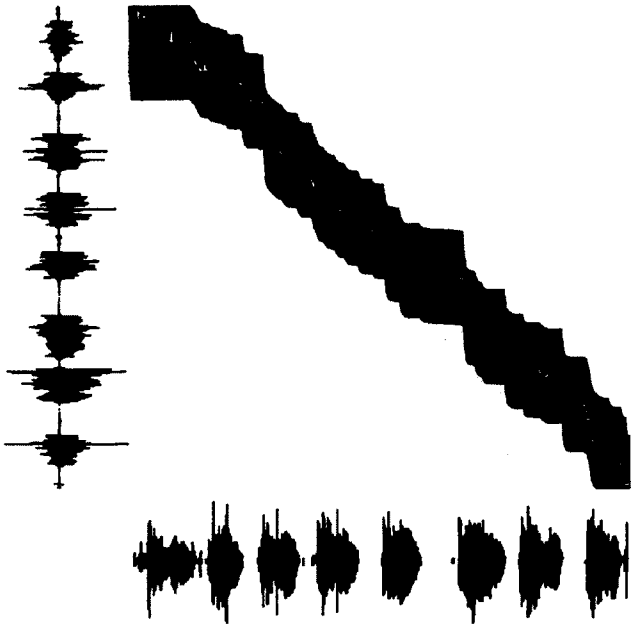


Figure 6.12 Alignement et matrice de coûts cumulés obtenus pour la paire CR1-CF1

L'alignement obtenu peut cependant sembler relativement saccadé : à plusieurs endroits on peut voir que la position estimée ne change pas pendant une période de temps étendue pour ensuite faire un saut important. Un agrandi d'une section montrant ceci est présenté à la figure 6.13. Celui-ci permet de bien observer la cause de ces sauts. En effet, le coût instantané dans ces sections, après transformation, est nul sur un grand domaine rectangulaire. Puisque le coût instantané reste nul, le coût cumulé est constant sur tout ce domaine. L'absence de variation dans cette zone fait en sorte qu'il est impossible de préciser davantage la position estimée. Le même phénomène est présent pendant les périodes de silence.

Ces zones de coûts nuls sont attribuables aux transformations appliquées sur les coûts instantanés. Dans le cas des zones de la figure 6.13, celles-ci surviennent vis-à-vis des segments où un même phonème est chanté et tenu pendant un certain temps. Dans ces sections, il est normal que les variations des paramètres spectraux et de l'énergie soient très limitées et donc, si les phonèmes correspondent d'une interprétation à l'autre, qu'on obtienne de très faibles valeurs de coûts qui seront annulées par les fonctions de transformation appliquées.

(a) Coûts instantanés (b) Coûts cumulés

Figure 6.13 Agrandissement de l'alignement et des matrices de coûts obtenus pour la paire CR1-CF1

Un tel problème nuit aux performances du système, évidemment, mais n'est pas nécessairement critique étant donné que le système l'alignement résultant indiquera quel est le phonème courant dans l'interprétation, permettant d'obtenir une cible pour l'algorithme de modification de la voix, qui sera généralement près de la cible idéale, sauf pour certaines fins de phrases où la hauteur (*pitch*) change à même un phonème.

Afin d'améliorer le suivi sur de telles zones, il serait possible de faire quelques modifications à l'algorithme d'alignement. Premièrement, il serait possible d'introduire un biais établi en fonction de la cadence calculée pour les dernières trames qui pondérerait légèrement chaque cellule de la colonne courante. Évidemment, si la cadence de l'interprétation est modifiée pendant une telle zone, il sera impossible de suivre celle-ci correctement, mais dans la plupart des cas, le résultat serait meilleur.

Deuxièmement, il serait possible d'intégrer d'autres paramètres au système. En intégrant un paramètre représentant la variation de hauteur, par exemple, il serait possible d'avoir un meilleur suivi pendant ces zones problématiques, si l'interprète veut bien collaborer. Évidemment, si celui-ci reste monotone, l'ajout d'un tel paramètre n'améliorerait probablement pas le suivi.

Finalement, en modifiant les fonctions de transformation appliquées aux coûts instantanés, il serait possible d'éviter qu'une portion aussi importante des coûts soit complètement

annulée. Afin d'accomplir ceci, le modèle des fonctions de transformations utilisé pourrait être modifié en ajoutant des segments ou en paramétrant la valeur de sortie à la fin du premier segment de la fonction de transformation, par exemple. De cette façon, il serait possible de conserver une pente très faible qui permettrait une différenciation des valeurs faibles. Cependant, en augmentant le nombre de paramètres, le problème d'optimisation serait complexifié, ce qui pourrait affecter le rendement de celui-ci.

L'alignement manuel effectué pour les paires de séquences utilisées pour l'entraînement (paire désignée ASM2) compte des points principalement aux débuts et à la fin des mots ainsi qu'aux changements de phonèmes. Aucun point n'a été placé à mi-chemin pendant un phonème voisé étendu. Il aurait été désirable d'en ajouter quelques-uns pour ainsi faire en sorte que le processus d'optimisation n'annule pas tous les coûts dans ces zones.

6.3.3 Mesures de distance utilisées

La distance utilisée pour le calcul du coût avant transformation, pour les analyses spectrales, est un simple écart quadratique avec, dans certains cas, une pondération par le logarithme de l'indice. Cette mesure de distance n'est pas du tout optimale étant donné qu'elle ne maximise pas la séparabilité des différents phonèmes chantés. Les paramètres statistiques de chacun des coefficients des paramètres vectoriels ne sont pas uniformes. À titre d'exemple, les statistiques de chacun des coefficients directs C1 à C7 de l'analyse PLP2 pour le signal de référence de la paire ASM2 sont présentées au tableau 6.10.

Tableau 6.10 Statistiques de chacun des coefficients de l'analyse PLP2 pour le signal de référence de la paire ASM2

Statistique	C1	C2	C3	C4	C5	C6	C7
Moyenne	$-3.1 \cdot 10^{-9}$	$-5.5 \cdot 10^{-9}$	$-4.7 \cdot 10^{-11}$	$-1.3 \cdot 10^{-6}$	$-2.7 \cdot 10^{-9}$	$-1.6 \cdot 10^{-11}$	$2.9 \cdot 10^{-14}$
Écart type	$1.5 \cdot 10^{-8}$	$1.8 \cdot 10^{-8}$	$1.6 \cdot 10^{-10}$	$6.0 \cdot 10^{-6}$	$1.1 \cdot 10^{-8}$	$8.1 \cdot 10^{-11}$	$4.3 \cdot 10^{-12}$

De plus, il aurait été intéressant d'avoir disposé de signaux de voix chantée segmentés selon les phonèmes prononcés. Des bases de données contenant une telle segmentation sont disponibles pour la voix parlée, mais il n'existe pas, pour le moment, une telle base de données pour la voix chantée. Avec ces données, il serait possible de regrouper les vecteurs obtenus avec les différentes analyses considérées par classe (par phonème) et de tenter de maximiser la distance entre les différentes classes et de minimiser la distance intra-classe. Une mesure de distance calibrée de la sorte aurait pu améliorer de beaucoup l'efficacité des matrices de coûts obtenues et ainsi faciliter le travail de la DTW.

Un effort de segmentation selon les phonèmes prononcé de séquences de chant a été entrepris pendant les travaux ayant mené à ce mémoire, mais l'effort nécessaire s'est révélé être beaucoup trop important pour qu'une base de données de taille suffisante puisse être construite.

6.3.4 Paramètres spectraux utilisés

Une autre façon d'améliorer les résultats obtenus serait de considérer d'autres paramètres spectraux. Trois types différents ont été évalués, mais il existe une multitude d'analyses spectrales applicables à la voix ; certaines d'entre elles permettent d'obtenir de meilleurs résultats pour des applications connexes, telle la reconnaissance de parole automatique.

[Mporas *et al.*, 2007] présente une comparaison d'une multitude d'analyses spectrales évaluées dans le cadre d'un système de reconnaissance de la parole dont plusieurs sont des variantes sur le modèle des MFCC et des LFCC. Certaines variantes remplacent la TFD par une transformée en ondelettes discrète et d'autres proposent des modifications au type de banc de filtre utilisé.

Parmi les analyses évaluées, les MFCC et l'analyse PLP obtiennent les moins bons résultats, les meilleurs ayant été obtenus par une analyse utilisant une décomposition en paquets d'ondelettes nommée *Subband Based Cepstral Coefficients (SBC)*. L'amélioration de 20% à 30% qu'ils observent pour la reconnaissance de mots par rapport aux analyses PLP et MFCC laisse présager une amélioration significative des matrices de coûts obtenues dans le cadre d'un système d'alignement.

6.4 Sommaire

En résumé, une méthode d'assemblage permettant d'obtenir une matrice de coût instantanée combinant l'information de plusieurs matrices de coûts associées à différents paramètres, basée sur l'optimisation des poids et des paramètres de fonctions de transformation a été décrite. Plusieurs résultats ont été présentés pour différents jeux de paramètres parmi lesquels l'analyse PLP2 a obtenu les meilleurs résultats. Différents algorithmes de DTW ont été comparés et des améliorations ont été mises en oeuvre afin de permettre d'éviter la divergence de l'algorithme dans certains cas plus difficiles. Le système obtenu offre des performances acceptables, avec une erreur d'alignement moyenne de l'ordre de 50ms, pour la plupart des paires évaluées. De plus, des pistes d'amélioration qui

permettraient d'augmenter sensiblement la performance de l'algorithme ont été identifiées, parmi lesquelles figurent :

- une étude plus formelle des divers types de contraintes locales,
- l'utilisation d'un estimateur de cadence pour les zones de coûts nuls,
- l'utilisation d'une mesure de distance optimisée pour maximiser la distance interphonème
- l'intégration de nouveaux paramètres permettant le suivi pendant les zones de coûts nuls, la hauteur, par exemple,
- l'étude d'autres paramètres spectraux, dont entre autres l'analyse SBC.

CHAPITRE 7

CONCLUSION

Dans le cadre des travaux ayant mené à ce mémoire, le problème de l'alignement d'une séquence audio de voix chantée avec une référence de même nature a été abordé. Le système développé, comportant principalement trois sections, soit l'algorithme d'alignement, l'extraction des paramètres des signaux et les transformations appliquées aux matrices de coûts, a été décrit de façon détaillée.

L'algorithme d'alignement proposé est une variante de l'algorithme de la DTW ordinaire qui est se prête bien à l'utilisation de vecteurs de paramètres peu sous-échantillonnés, par rapport aux techniques basées sur les modèles de Markov cachés. Une adaptation simple de la DTW qui permet d'obtenir une complexité d'ordre $\mathcal{O}(1)$, à chaque nouvelle trame, permettant ainsi une utilisation en temps réel et évitant toute limite sur la longueur des séquences alignées, a été proposée. Cette adaptation a été comparée à une adaptation proposée par [Dixon et Widmer, 2005] et a permis d'obtenir de meilleurs résultats, dans la plupart des cas.

L'algorithme de la DTW prend, à son entrée, des séquences de vecteurs de paramètres. Pour l'application visée, les paramètres les plus importants sont les paramètres spectraux, qui sont d'extrême importance pour la cognition de la voix. Plusieurs paramètres ont été étudiés, implémentés et évalués, dont les coefficients MFCC, les coefficients WDCTC et les coefficients PLP. Une analyse de l'effet des différents paramètres de ces algorithmes a été effectuée et il a été possible de déterminer que l'analyse PLP produisait les meilleurs résultats pour la voix chantée. De plus, l'exercice a été réalisé pour un paramètre non spectral, soit le niveau d'énergie.

Les vecteurs de paramètres obtenus sont comparés à l'aide d'une fonction de coût qui mesure la dissemblance des vecteurs obtenus. L'ensemble des coûts obtenus forme une matrice de coûts instantanés. Il est possible de modifier ces matrices afin que les coûts obtenus sur le chemin d'alignement idéal soient minimaux par rapport au reste en appliquant une fonction de transformation non linéaire aux coûts obtenus. La fonction de transformation utilisée est une fonction linéaire par morceaux dont les paramètres sont déterminés par optimisation en boucle fermée, en utilisant une fonction-objectif qui mesure l'efficacité de la matrice de coûts cumulés obtenue. La méthode d'optimisation utilisée est

une recherche directe par motif. De plus, la fonction de transformation utilisée limite le domaine du coût de sortie à l'intervalle $[0, 1]$, ce qui permet d'utiliser des poids représentant l'importance relative de chacun des paramètres pour créer une matrice de coûts combinée.

L'assemblage du système complet nécessite de paramétrer les poids relatifs des paramètres considérés ainsi que les fonctions de transformations utilisées pour chacun des paramètres. Tous ces paramètres sont optimisés de façon simultanée avec la méthode de recherche directe par motif. Plusieurs jeux de paramètres ont été évalués, combinant des paramètres spectraux ainsi que le paramètre d'énergie. Les meilleurs résultats ont été obtenus avec le jeu PLP2, en intégrant les coefficients directs de l'analyse PLP et le niveau d'énergie ainsi que les dérivées estimées des coefficients obtenus avec ces deux analyses. Les adaptations de la DTW proposées ont été comparées en utilisant le jeu de paramètres choisi, montrant la supériorité de l'algorithme proposé «B». Pour les signaux testés, l'écart moyen sur l'alignement est de 55 ms avec un écart-type de 86.5 ms.

L'analyse des résultats obtenus permet d'identifier quelques lacunes de l'algorithme qui seraient à améliorer. Une modification de l'algorithme de la DTW permet d'améliorer les caractéristiques de convergence pour les séquences difficiles à aligner. Les contraintes locales utilisées par la DTW pourraient également être modifiées afin d'améliorer les matrices de coûts cumulés obtenus. Dans les zones de phonème voisé étendues ou les silences, l'absence de différenciation fait en sorte que l'algorithme n'est pas en mesure de mettre à jour son alignement. Un estimateur de cadence pourrait être intégré à l'algorithme afin d'améliorer l'alignement obtenu. De nouveaux paramètres pourraient également être combinés au jeu de paramètre choisi pour améliorer l'alignement obtenu en intégrant de nouvelles informations à la matrice de coûts obtenue.

Il serait également intéressant de considérer d'autres types de paramètres spectraux dont les performances ont été mesurées supérieures à l'analyse PLP pour la reconnaissance de la voix, dont l'analyse SBC. De plus, en constituant une base de données de voix chantée segmentée par phonème, il serait possible de créer une fonction de distance qui maximiserait la distance inter-classe et qui permettrait d'obtenir des matrices de coûts qui seraient plus faciles à aligner.

L'algorithme développé, visant une application dans un système de karaoké, risque d'être utilisé dans des environnements bruyants. Il aurait été intéressant de mesurer l'influence du bruits sur les performance de l'algorithme global et de la robustesse des différents paramètres spectraux évalués en présence de bruit.

Finalement, aucun cadre d'évaluation commun n'existe pour les algorithmes d'alignement de la voix chantée. Un tel cadre d'évaluation comprendrait des séquences audio de chant ainsi que des segmentations ou des alignements manuels, ce qui permettrait de comparer les algorithmes développés de façon objective avec ceux répertoriés dans la littérature. Un tel cadre permettrait de faire de significatifs progrès sur le problème de l'alignement de la voix chantée.

LISTE DES RÉFÉRENCES

Beaudette, D. (2010). *Suivi de chansons par reconnaissance automatique de parole et alignement temporel.* Mémoire de maîtrise, Université de Sherbrooke.

Bloch, J. J. et Dannenberg, R. B. (1985). Real-time computer accompaniment of keyboard performances. Dans *Proceedings of 1985 the International Computer Music Conference.* International Computer Music Association, Vancouver, Canada, p. 279–290.

Bonada, J., Cano, P., Loscos, A. et Serra, X. (2000). Voice morphing system for impersonating in karaoke applications. Dans *Proceedings of the 1999 International Computer Music Conference.* International Computer Music Association, Beijing, China.

Bullock, J. (2007). Libxtract : A lightweight library for audio feature extraction. Dans *Proceedings of the 2007 International Computer Music Conference.* International Computer Music Association.

Cano, P., Loscos, A. et Bonada, J. (1999). Scoreperformance matching using HMMs. Dans *Proceedings of the 1999 International Computer Music Conference.* International Computer Music Association, Beijing, China, p. 441–444.

Cho, N. I. et Mitra, S. K. (2000). Warped discrete cosine transform and its application in image compression. *Institute of Electrical and Electronics Engineers Transactions on Circuits and Systems for Video Technology*, volume 10, numéro 8, p. 1364–1373.

Cont, A. (2006). Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical HMMs. Dans *Acoustics and Speech Signal Processing 2006 Proceedings, Institute of Electrical and Electronics Engineers International Conference on.* volume 5. Institute of Electrical and Electronics Engineers, Toulouse, France, p. V.

Cont, A. et Schwarz, D. (2006). Score following at ircam. Dans *Music Information Retrieval Evaluation eXchange (MIREX) Score Following Contest.* International Society for Music Information Retrieval, Illinois, USA.

Dannenberg, R. B. (1984). An on-line algorithm for real-time accompaniment. Dans *Proceedings of the 1984 International Computer Music Conference.* International Computer Music Association, Paris, France, p. 193–198.

Davis, S. B. et Mermelstein, P. (1990). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Readings in speech recognition*, p. 65–74.

Dixon, S. (2005). Live tracking of musical performances using on-line time warping. Dans *Proceedings of the 8th International Conference on Digital Audio Effects.* Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones, Madrid, Espagne, p. 92–97.

Dixon, S. et Widmer, G. (2005). Match, a music alignment tool chest. Dans *Proceedings of the 6th International Conference of Music Information Retrieval.* University of London, Londres, Angleterre, p. 492–497.

Frigo, M. et Johnson, S. G. (1998). FFTW : an adaptive software architecture for the FFT. Dans *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 Institute of Electrical and Electrical Engineers International Conference on.* volume 3. Institute of Electrical and Electronics Engineers, p. 1381–1384.

Gagnon, B., Brunet, C.-A. et Lefebvre, R. (2007). A high level musical score alignment technique based on fuzzy logic and DTW. Dans *Proceedings of the 123rd Audio Engineering Society Convention.* Audio Engineering Society, New York.

Grubb, L. et Dannenberg, R. B. (1997). A stochastic method of tracking a vocal performer. Dans *Proceedings of the 1997 International Computer Music Conference.* International Computer Music Conference, Thessaloniki, Grèce, p. 301–308.

Grubb, L. et Dannenberg, R. B. (1998). Enhanced vocal performance tracking using multiple information sources. Dans *Proceedings of the International Computer Music Conferнce.* International Computer Music Association, Ann Arbor, USA, p. 37–44.

Hassanein, H. et Rudko, M. (1984). On the use of discrete cosine transform in cepstral analysis. *Institute of Electrical and Electronics Engineers Transactions on Acoustics, Speech and Signal Processing*, volume 32, numéro 4, p. 922–925.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, volume 87, numéro 4, p. 1738–1752.

Hooke, R. et Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery (JACM)*, volume 8, numéro 2, p. 212–229.

Inoue, W., Hashimoto, S. et Ohteru, S. (1993). A computer music system for human singing. Dans *Proceedings of the 1993 International Computer Music Conference.* International Computer Music Association, Waseda, Japon, p. 150–153.

Inoue, W., Hashimoto, S. et Ohteru, S. (1994). Adaptive karaoke System—Human singing accompaniment based on speech recognition. Dans *Proceedings of the 1994 International Computer Music Conference.* International Computer Music Association, Skovgaardsgade, Danemark, p. 70–77.

Juang, B. et Rabiner, L. (1993). *Fundamentals of Speech Recognition*, 1re édition. Prentice Hall.

Kaprykowsky, H. et Rodet, X. (2006). Globally optimal short-time dynamic time warping application to score to audio alignment. Dans *Acoustics and Speech Signal Processing 2006 Proceedings, Institute of Electrical and Electronics Engineers International Conference on.* volume 5. Institute of Electrical and Electronics Engineers, Toulouse, France, p. V.

Kolda, T. G., Lewis, R. M. et Torczon, V. (2003). Optimization by direct search : New perspectives on some classical and modern methods. *Society for Industrial and Applied Mathematics Review*, volume 45, numéro 3, p. 385–482.

Lago, N. P. et Kon, F. (2004). The quest for low latency. Dans *Proceedings of the 2004 International Computer Music Conference*. International Computer Music Association, p. 33–36.

Loscos, A., Cano, P. et Bonada, J. (1999). Low-delay singing voice alignment to text. Dans *Proceedings of the 1999 International Computer Music Conference*. International Computer Music Association, Beijing, China, p. 437–440.

Makhoul, J. (1975). Spectral linear prediction : Properties and applications. *Acoustics, Speech and Signal Processing, Institute of Electrical and Electronics Engineers Transactions on*, volume 23, numéro 3, p. 283–296.

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, volume 116, p. 91–103.

Mporas, I., Ganchev, T., Siafarikas, M. et Fakotakis, N. (2007). Comparison of speech features on the speech recognition task. *Journal of Computer Science*, volume 3, numéro 8, p. 608–616.

Muller, M. et Appelt, D. (2008). Path-constrained partial music synchronization. Dans *Acoustics, Speech and Signal Processing, 2008 Institute of Electrical and Electronics Engineers International Conference on*. Institute of Electrical and Electronics Engineers (IEEE), p. 65–68.

Muller, M., Mattes, H. et Kurth, F. (2006). An efficient multiscale approach to audio synchronization. Dans *Proceedings of the 7th International Society for Music Information Retrieval*. International Society for Music Information Retrieval, Victoria, Canada, p. 192–197.

Muralishankar, R. et Ramakrishnan, A. G. (2005). Pseudo complex cepstrum using discrete cosine transform. *International Journal of Speech Technology*, volume 8, numéro 2, p. 181–191.

Muralishankar, R., Sangwan, A. et O'Shaughnessy, D. (2005). Warped discrete cosine transform cepstrum : A new feature for speech processing. Dans *Proceedings of the 13th European Signal Processing Conference*. Curran Associates, Inc., p. 285–288.

Oppenheim, A. et Schafer, R. (1968). Homomorphic analysis of speech. *Institute of Electrical and Electronics Engineers Transactions on Audio and Electroacoustics*, volume 16, numéro 2, p. 221–226.

Orio, N., Lemouton, S. et Schwarz, D. (2003). Score following : state of the art and new developments. Dans *Proceedings of the 2003 conference on New interfaces for musical expression*. National University of Singapore, Montreal, Quebec, Canada, p. 36–41.

Orio, N. et Schwarz, D. (2001). Alignment of monophonic and polyphonic music to a score. Dans *Proceedings of the 2001 International Computer Music Conference.* International Computer Music Association, Havana, Cuba, p. 155–158.

Pellegrini, T. et Duée, R. (2003). *Suivi de voix parlée grâce aux modèles de Markov cachés.* Rapport de stage de diplôme d'études approfondies, Institut de Recherche et Coordination Accoustique/Musique.

Prahallad, K., Sudhakar, V., Ranganatham, V., Bharat, K. M. et Debashish, S. R. (2006). Significance of formants from difference spectrum for speaker identification. Dans *Ninth International Conference on Spoken Language Processing.* International Speech Communication Association.

Puckette, M. (1995). Score following using the sung voice. Dans *Proceedings of the 1995 Internation Computer Music Conference.* International Computer Music Association, Banff, Canada, p. 175–178.

Puckette, M. et Lippe, C. (1992). Score following in practice. Dans *Proceedings of the 1992 International Computer Music Conference.* International Computer Music Association, San Jose, USA, p. 182–185.

Raphael, C. (1999). Automatic segmentation of acoustic musical signals using hidden markov models. *Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers Transactions on*, volume 21, numéro 4, p. 360–370.

Sangwan, A., Muralishankar, R. et O'Shaughnessy, D. (2005). Performance analysis of the warped discrete cosine transform cepstrum with MFCC using different classifiers. Dans *2005 Institute of Electrical and Electronics Engineers Workshop on Machine Learning for Signal Processing.* Institute of Electrical and Electronics Engineers, Mystic, CT, USA, p. 99–104.

Schroeder, M. R. (1977). Recognition of complex acoustic signals. *Life Sciences Research Report*, volume 5, numéro 324, p. 130.

Vercoe, B. (1984). The synthetic performer in the context of live performance. Dans *Proceedings of the 1984 International Computer Music Conference.* International Computer Music Association, Paris, France, p. 199–200.

Vercoe, B. et Puckette, M. (1985). Synthetic rehearsal : Training the synthetic performer. Dans *Proceedings of 1985 International Computer Music Conference.* International Computer Music Association, Vancouver, Canada, p. 275–278.

Xuan, Z., Yining, C., Jia, L. et Runsheng, L. (2002). Feature selection in mandarin large vocabulary continuous speech recognition. Dans *Signal Processing, Proceedings of the 6th International Conference on.* volume 1. Institute of Electrical and Electronics Engineers, p. 508–511.

Zheng, F., Wu, W. et Fang, D. (1997). A log-index weighted cepstral distance measure for speech recognition. *Journal of Computer Science and Technology*, volume 12, numéro 2, p. 177–184.

Zheng, F., Zhang, G. et Song, Z. (2001). Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*, volume 16, numéro 6, p. 582–589.