Reconnaissance d'accords à partir de signaux audio par l'utilisation de gabarits théoriques

Laurent Oudre

Thèse

présentée pour obtenir le grade de docteur

de TÉLÉCOM ParisTech

Spécialité : Signal et Images

Laurent OUDRE

Reconnaissance d'accords à partir de signaux audio par l'utilisation de gabarits théoriques

Template-based chord recognition from audio signals

Soutenue le 3 novembre 2010 devant le jury composé de

Laurent Daudet	Président
Dan Ellis	Rapporteurs
Sylvain Marchand	
Geoffroy Peeters	Examinateur
Yves Grenier	Directeurs de thèse
Cédric Févotte	

Pour me comprendre
Il faudrait savoir [...]
La résonance
De mes premiers accords.

M.B.

Remerciements

Mes premiers remerciements vont à Yves et Cédric, qui m'ont accompagné, soutenu, encadré... Je les remercie autant pour leur compétence scientifique que pour les avis et conseils éclairés qu'ils ont su me prodiguer tout au long de ces 3 années. Merci à eux de m'avoir fait confiance.

Merci aux membres de mon jury de thèse pour leur travail de lecture rigoureux, leurs remarques pertinentes et leur intêret pour mes travaux.

Je remercie toutes les personnes avec qui j'ai pu travailler au sein du département TSI. Un merci spécial aux membres du groupe AAO (mon groupe d'appartenance) et au groupe MMA (mon groupe de FIAP). Je ne donnerai pas ici une longue liste de prénoms, mais je citerai juste quelques collègues-amis (que les autres me pardonnent) : Aurélia, Tyze, Jérôme, Théo, Jean, Ismaël... Merci à Valentin de m'avoir mis en selle, pour sa patience et sa gentillesse.

Merci à Laurence et Fabrice pour leur aide dans tous mes déboires techniques et administratifs. Merci à Maryse et Clara pour leur sourire.

Je remercie mes parents et ma famille pour leur amour, leur soutien et leur confiance. Je remercie mes amis, tout simplement parce que je les aime (Thomas, Morgane, David, Hortense, Arthur, Edouard, Jean, Florence, Guitou, Xavier, Laurent, Jerôme...). Merci à Michaël pour la relecture de certains chapitres de ce manuscrit. Merci à toutes les personnes présentes lors de ma soutenance.

J'ai eu la chance immense, pendant ces 3 ans, de rencontrer un collègue qui est devenu un ami. Je crois qu'il n'est pas exagéré de dire que sans lui ma thèse n'aurait pas été ce qu'elle est. Il a su me soutenir, me relever et m'accompagner dans tous les moments difficiles (et dans les moments plus heureux, bien sûr). Merci à Thomas pour le pot de l'EDITE, pour les menus *Pestacle*, pour les pauses, pour *Chez Milou*, pour les parigottes, pour St Malo... Je n'oublierai rien.

Résumé en français

Résumé

Cette thèse s'inscrit dans le cadre du traitement du signal musical, en se focalisant plus particulièrement sur la transcription automatique de signaux audio en accords. En effet, depuis une dizaine d'années, de nombreux travaux visent à représenter les signaux musicaux de la façon la plus compacte et pertinente possible, par exemple dans un but d'indexation ou de recherche par similarité. La transcription en accords constitue une façon simple et robuste d'extraire l'information harmonique et rythmique des chansons et peut notamment être utilisée par les musiciens pour rejouer les morceaux.

Nous proposons deux approches pour la reconnaissance automatique d'accords à partir de signaux audio, qui offrent la particularité de se baser uniquement sur des gabarits d'accords théoriques, c'est à dire sur la définition des accords. En particulier, nos systèmes ne nécessitent ni connaissance particulière sur l'harmonie du morceau, ni apprentissage.

Notre première approche est déterministe, et repose sur l'utilisation conjointe de gabarits d'accords théoriques, de mesures d'ajustement et de post-traitement par filtrage. On extrait tout d'abord des vecteurs de chroma du signal musical, qui sont ensuite comparés aux gabarits d'accords grâce à plusieurs mesures d'ajustement. Le critère de reconnaissance ainsi formé est ensuite filtré, afin de prendre en compte l'aspect temporel de la tâche. L'accord finalement détecté sur chaque trame est celui minimisant le critère de reconnaissance. Cette méthode a notamment été présentée lors d'une évaluation internationale (MIREX 2009) et a obtenu des résultats très honorables.

Notre seconde approche est probabiliste, et réutilise certains éléments présents dans notre méthode déterministe. En faisant un parallèle entre les mesures d'ajustement utilisées dans l'approche déterministe et des modèles de probabilité, on peut définir un cadre probabiliste pour la reconnaissance d'accords. Dans ce cadre, les probabilités de chaque accord dans le morceau sont évaluées grâce à un algorithme Espérance-Maximisation (EM). Il en résulte la détection, pour chaque chanson, d'un vocabulaire d'accords adapté, qui permet l'obtention d'une meilleure transcription en accords. Cette méthode est comparée à de nombreux systèmes de l'état de l'art, grâce à plusieurs corpus et plusieurs métriques, qui permettent une évaluation complète des différents aspects de la tâche.

Chapitre 1

Introduction

Ce premier chapitre vise à introduire les principales notions musicales nécessaires à la bonne compréhension du présent document, mais aussi à préciser le contexte et les principales applications de ce travail de thèse.

1.1 Quelques notions musicales

1.1.1 Qu'est-ce qu'un accord ?

On définit un accord comme un ensemble de notes jouées simultanément. Même s'il peut théoriquement être composé de n'importe quelles notes, un accord forme en théorie musicale une véritable entité et n'est donc pas seulement un ensemble aléatoire de notes. Un accord peut être défini par trois notions (Harte et al. (2005); Benward & Saker (2003)) :

- *fondamentale* : la note à partir de laquelle l'accord est construit ;
- *type* : la structure harmonique de l'accord (qui donne aussi le nombre de notes composant l'accord) ;
- *renversement* : la relation de la basse avec les autres notes de l'accord.

La plupart des systèmes de reconnaissance d'accords ne cherchent pas à reconnaître le renversement de l'accord : dans la suite du manuscrit, on supposera donc que les seules informations voulues sont la fondamentale et le type. Par exemple, un accord de *do majeur* (couramment noté selon la terminologie anglo-saxonne : *C*) est défini par une fondamentale *do* et par un type *majeur*, indiquant que l'accord contient aussi la tierce majeure *mi* et la quinte *sol* (voir la Figure 1.1 pour un exemple de construction). Dans ce manuscrit, nous écrirons les accords avec la notation anglo-saxonne : les fondamentales (de do à si) sont représentées par des lettres (de C à B).

Il existe un grand nombre de types d'accords. Fujishima (1999), dans le premier système de reconnaissance d'accords, définit 27 types d'accords : il s'agit à l'heure actuelle de la publication considérant le plus grand nombre de types d'accords. La Figure 1.2 présente quelques exemples de types d'accords.

1.1.2 Qu'est-ce qu'un chroma ?

La perception d'une note peut être décomposée en deux notions : la *hauteur tonale*, représentant l'octave à laquelle la note appartient et le *chroma* (ou *classe de hauteur*) qui renseigne sur la relation de la note avec les autres notes de l'octave. Par exemple, la note *la4* (440 Hz)

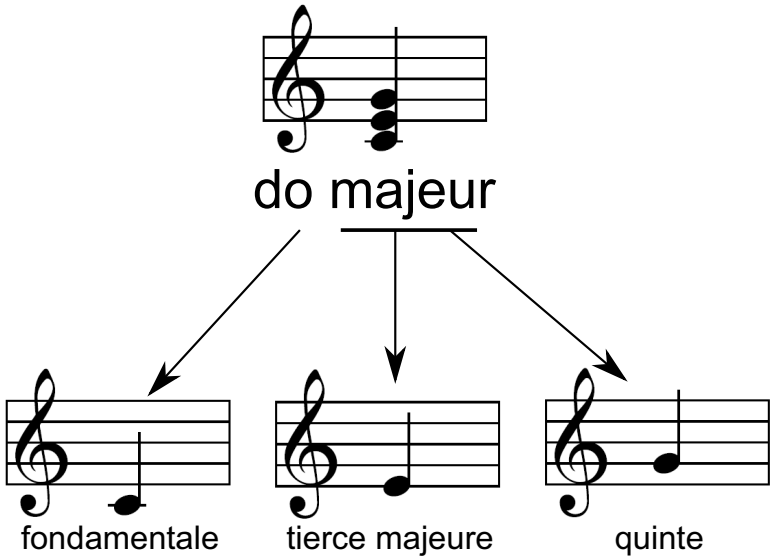


FIGURE 1.1 – Construction d'un accord de do majeur

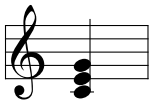
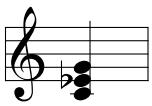
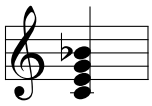
Nom de l'accord	Notation	Notes constituantes	Ecriture musicale
do majeur	C	do, mi, sol	
do mineur	Cmin	do, mi♭, sol	
do septième de dominante	C7	do, mi, sol, si♭	

FIGURE 1.2 – Exemples d'accords

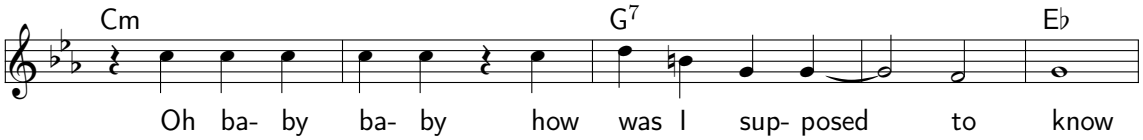


FIGURE 1.3 – Exemple de *lead sheet* (mélodie, paroles et accords) : *Baby one more time* de Britney Spears (1998). Jive Records.

```
Cm              G7                  E♭
Oh baby, baby How was I supposed to know
          Fm                  G
That something wasn't right here
```

FIGURE 1.4 – Exemple de transcription simple en accords (paroles et accords) : *Baby one more time* de Britney Spears (1998). Jive Records.

peut être décomposée en un numéro d'octave *4* et un chroma *la* (ou *A*). Les chromas peuvent être vus comme des classes d'équivalence d'octaves : la classe de hauteur *C* contient ainsi toutes les notes *do*, qu'importe leur octave. Nous verrons plus tard que ce concept de chroma est primordial dans le domaine de reconnaissance d'accords où l'on tente de détecter des notes indépendamment de leur octave.

1.2 Définition de la tâche de reconnaissance d'accords

1.2.1 Différentes transcriptions en accords

Une transcription en accords peut prendre plusieurs formes. Une *lead sheet* (voir Figure 1.3), est une partition présentant la progression des accords, la mélodie et les paroles. Elle peut être utilisée par des musiciens pour rejouer une chanson, par des joueurs de jazz pour improviser sur une progression d'accords ou comme description légale d'une chanson dans l'industrie musicale. La transcription en accords peut aussi consister seulement en une séquence d'accords avec les paroles, avec ou sans tablatures de guitare (voir Figure 1.4).

1.2.2 Principaux domaines d'application

La représentation d'une chanson en tant que séquence d'accords peut être utilisée dans de nombreuses applications en recherche d'information musicale (Music Information Retrieval (MIR) en anglais).

L'application la plus intuitive et évidente de la transcription en accords est la possibilité de rejouer une chanson en utilisant seulement les accords. Mais une transcription en accords peut aussi permettre d'accéder à des notions plus haut niveau, telles que la tonalité (Shenoy et al. (2004)), la structure (Maddage et al. (2004), Bello & Pickens (2005)) ou le rythme (Goto & Muraoka (1999)). Elle peut enfin être utilisée comme descripteur d'une chanson, permettant ainsi notamment des recherches par similarité (Lee (2006b), Bello (2007), Cheng et al. (2008)).

1.3 Plan du document

Ce document est organisé comme suit :

Chapitre 1 : Introduction

Ce chapitre présente le contexte et les motivations de ce manuscrit de thèse en introduisant certaines notions musicales et en décrivant les principales applications et problématiques de la tâche de reconnaissance d'accords.

Chapitre 2 : État de l'art

Ce chapitre donne un aperçu des principales méthodes de reconnaissance d'accords ainsi que des représentations fréquentielles utilisées en caractéristiques d'entrée.

Chapitre 3 : Corpus et évaluation

Avant de présenter nos systèmes de reconnaissance d'accords, nous décrirons dans ce chapitre les corpus utilisés ainsi que le protocole d'évaluation. En particulier, nous nous intéresserons aux métriques servant à évaluer les algorithmes, et nous verrons en quoi elles permettent de rendre compte des différentes facettes de la tâche de reconnaissance d'accords.

Chapitre 4 : Approche déterministe pour la reconnaissance d'accords

Ce chapitre introduit une méthode déterministe pour la reconnaissance d'accords utilisant des gabarits, en détaillant chacune des entités la composant. Une analyse des résultats obtenus sera présentée, ainsi qu'une comparaison avec l'état de l'art.

Chapitre 5 : Approche probabiliste pour la reconnaissance d'accords

Ce chapitre est une prolongation du chapitre précédent, définissant un cadre probabiliste pour la reconnaissance d'accords. Nous verrons en détail les liens existant entre ces deux approches, ainsi que les apports de ce nouveau contexte. Comme ce fut le cas pour l'approche probabiliste, on proposera une analyse des résultats et une comparaison avec l'état de l'art, en ayant recours à diverses métriques.

Chapitre 6 : Conclusion

Pour finir, ce chapitre résume les principales conclusions et contributions de ce manuscrit, et propose plusieurs pistes de travail.

Chapitre 2

État de l'art

La transcription en accords comporte dans la plupart des cas deux tâches successives : une extraction de caractéristiques contenant l'information musicale, puis une reconnaissance d'accords à partir de ces caractéristiques.

2.1 Phase 1 : Extraction des caractéristiques

Les caractéristiques utilisées dans la transcription en accords varient d'une méthode à l'autre, mais sont dans la plupart des cas des variantes des Pitch Class Profile (PCP) introduits par Fujishima (1999), dont le calcul repose sur la notion de *chroma*. Aussi appelés *vecteurs de chroma*, ces caractéristiques sont des vecteurs de dimension 12, chacune des composantes représentant l'énergie spectrale d'un demi-ton sur l'échelle chromatique, indépendamment de l'octave. La succession de ces vecteurs de chroma en fonction du temps est communément appelée *chromagramme*.

Leur mode de calcul repose soit sur la Transformée de Fourier à Court Terme (TFCT), auquel cas le choix des fenêtres d'analyse et des paramètres de recouvrement est crucial, soit sur la Transformée à Q Constant (TQC) (Brown (1991)). Les chromagrammes sont largement utilisés en traitement du signal musical, aussi bien pour la reconnaissance d'accords que pour l'extraction de tonalité (Peeters (2006), Gómez (2006a)).

Pour les caractéristiques calculées à partir d'une TFCT, citons notamment le *Pitch Pattern* de Leman (2000), la version améliorée des *Pitch Class Profiles* par Fujishima (2000), la *Pitch Class* de Chuan & Chew (2005), le *Chroma Summary Vector* de Izmirli (2005), le *Harmonic Pitch Class Profile* de Gómez (2006a), le *Enhanced Pitch Class Profile* de Lee (2006a), le *Chroma Vector* de Peeters (2006) et le *Chroma Profile* de Varewyck et al. (2008). Le mode de calcul de tous ces paramètres peut être vu comme une suite de modules présents ou pas selon les méthodes : détection des attaques, des silences, des pics, des transitions, algorithme de recherche de diapason, filtrage en pré- ou post-traitement, prise en compte des harmoniques etc...

La TQC a aussi été largement utilisée pour le calcul de ces vecteurs de chroma. On notera ainsi le *Constant Q profile* de Purwins et al. (2000), le *Pitch Profile* de Zhu et al. (2005), le *Chromagram* de Pauws (2004), le *Harmonic Pitch Class Profile* de Harte & Sandler (2005) etc... qui diffèrent principalement dans la façon dont la transformée à Q constant est traitée pour arriver à un vecteur de dimension 12 (algorithmes de diapason, sélection de pics préalables, post-traitement par lissage, bandes de fréquences considérées, etc...).

2.2 Phase 2 : Reconnaissance d'accords

Une fois calculé, le chromagramme va servir d'entrée à un système de reconnaissance d'accords, qui attribuera à chaque trame une étiquette d'accord.

Comment réaliser cette tâche ? La meilleure façon de répondre à cette question est de se demander : comment ferait un musicien pour transcrire un morceau en accords ? Un bon musicien possède deux qualités : une solide connaissance en théorie musicale, et des années de pratique. Lors du processus de transcription, il se demande : puis-je mettre cet accord étant donnée l'harmonie du morceau ? Ce changement d'accords est-il bien placé par rapport à la structure rythmique ? Mais il peut raisonner aussi de la façon suivante : l'accord que j'ai détecté correspond-il à ce que j'ai appris en transcrivant une grande quantité de morceaux ?

Ces deux approches (utilisation de théorie musicale ou d'apprentissage) sont effectivement les deux principales voies explorées par les chercheurs travaillant sur la reconnaissance d'accords. S'ajoutent à ces deux catégories, la reconnaissance d'accords par l'utilisation de gabarits (qui constitue notre approche), ou des méthodes hybrides utilisant à la fois la théorie musicale et l'apprentissage.

2.2.1 Méthodes utilisant des gabarits d'accords

La structure d'un accord étant entièrement définie par sa fondamentale et son type, il est aisé de créer des gabarits d'accords de dimension 12, qui reflètent la structure de l'accord en attribuant une amplitude théorique à chacun des demi-tons de la gamme chromatique. Le gabarit le plus simple, largement utilisé pour la reconnaissance d'accords, a une structure binaire : les notes constituant l'accord se voient attribuer une amplitude de 1, tandis que les notes absentes ont une amplitude théorique de 0.

La toute première méthode de reconnaissance d'accords proposée par Fujishima (1999) utilise justement ces gabarits binaires d'accords. 27 types d'accords sont testés et la transcription consiste soit en la minimisation de la distance euclidienne entre les PCP et les gabarits d'accords, soit en la maximisation d'un produit scalaire pondéré.

Un système similaire à celui de Fujishima est proposé par la suite par Harte & Sandler (2005), mais à partir d'un chromagramme plus élaboré, comprenant notamment une estimation de diapason. 4 types d'accords sont détectés (majeur, mineur, diminué et augmenté). La transcription en accords est réalisée par une maximisation de produit scalaire entre les trames du chromagramme et les gabarits d'accords.

2.2.2 Méthodes basées sur l'apprentissage

Les méthodes décrites ici utilisent seulement de l'apprentissage et des données annotées pour la reconnaissance d'accords.

Sheh & Ellis (2003) sont les premiers à utiliser de l'apprentissage pour la reconnaissance d'accords. Cette information est introduite dans un Modèle de Markov Caché (MMC). Un MMC est constitué par un certain nombre d'états cachés, une distribution initiale d'états, une distribution de transitions donnant les probabilités de passer d'un état à l'autre, ainsi qu'une distribution d'observation, donnant la vraisemblance d'un état précis pour une certaine donnée observée. Dans les systèmes de reconnaissance d'accords basés sur les MMC, chaque accord est représenté par un état caché, et les observations sont les trames du chromagramme. Étant donnés les paramètres du MMC, la reconnaissance d'accords consiste à rechercher la séquence d'états cachés (accords) la plus probable, qui puisse avoir généré la séquence d'observations

(chromagramme). Le modèle de Sheh & Ellis (2003) contient 147 états correspondant à 7 types d'accords (majeur, mineur, septième, majeur septième, mineur septième, augmenté et diminué) et à 21 fondamentales (12 demi-tons en distinguant les ♭ et les ♯). Tous les paramètres du MMC sont appris par un apprentissage semi supervisé grâce à un algorithme de type Espérance-Maximisation (EM).

Citons aussi notamment la méthode de Ryynänen & Klapuri (2008b), qui utilise un MMC considérant des modèles d'observation différents pour les notes aigues et les notes graves, et celle de Weller et al. (2009) qui repose sur l'entraînement de Support Vector Machines (SVMs) au lieu des MMCs couramment utilisés.

2.2.3 Méthodes basées sur la théorie musicale

Les méthodes suivantes n'utilisent pas d'apprentissage, mais de l'information musicale, qu'elle soit issue de règles d'harmonie ou de la détection conjointe de notions plus haut niveau (rythme, tonalité, etc...).

Bello & Pickens (2005) apportent des améliorations au système de Sheh & Ellis (2003) en réduisant le nombre d'états (24 accords majeurs et mineurs) et en permettant une initialisation des paramètres du MMC par de la théorie musicale, ce qui améliore significativement les performances. La distribution initiale des états et la matrice de transition entre états sont mises à jour par un algorithme EM, mais les distributions d'observation restent fixes, donnant ainsi à chaque accord une structure pré-déterminée. L'introduction de chromagrammes calés sur le rythme du morceau améliore aussi les performances du système.

Shenoy & Wang (2005) partent d'une première détection d'accords basée sur les vecteurs de chroma, et utilisent ensuite des informations haut niveau de rythme et de tonalité pour valider ou corriger des détections d'accords successives, tandis que Sailer & Rosenbauer (2006) construisent leur détection d'accords en attribuant des poids relatifs à l'amplitude, la durée et la tonalité. La méthode de Papadopoulos & Peeters (2008) est basée sur un MMC prenant en compte des informations de rythme, et notamment de mesure. Enfin, le système de Mauch et al. (2009) repose sur l'évaluation conjointe de nombreuses notions musicales (tonalité, position métrique, basse et structure) afin de produire des transcriptions en accords musicalement pertinentes.

2.2.4 Méthodes hybrides

Ces méthodes combinent les approches des deux catégories précédemment décrites, en se servant à la fois d'information musicale et de données annotées.

Yoshioka et al. (2004) réalisent une estimation conjointe des accords et de leurs frontières en sélectionnant des hypothèses basées sur des critères bas niveau tels que les vecteurs de chroma et les basses mais aussi sur des patrons de transitions entre accords. Ils utilisent en particulier des gabarits d'accords appris sur des bases de données annotées. Le système de Burgoyne & Saul (2005) améliore la méthode de Sheh & Ellis (2003) en introduisant un modèle d'observation harmonique plus complexe, permettant une estimation conjointe de la tonalité. Lee & Slaney (2008) se servent quant à eux d'une large base de données annotée générée à partir de morceaux MIDI afin de réaliser un apprentissage supervisé non pas sur un MMC, mais sur 24 MMCs dépendant de la tonalité. Ceci permet ainsi de produire une séquence d'accords, mais aussi la tonalité du morceau. Enfin, Khadkevich & Omologo (2009b) utilisent des modèles de langage inspirés des travaux sur le traitement de la parole pour la reconnaissance d'accords.

2.2.5 Bilan

La Table 2.1 présente un récapitulatif des avantages et inconvénients des quatre catégories de méthodes précédemment définies.

	+	-
Gabarits d'accords	– Pas besoin de données annotées – Indépendant du genre musical (en théorie) – Faible temps de calcul	– Produit parfois des résultats peu pertinents harmoniquement et rythmiquement – Produit parfois des transcriptions fragmentées
Apprentissage	– S'adapte bien aux données audio – Peut s'appliquer à des morceaux ne suivant pas forcément la théorie musicale – Pas d'a priori sur les résultats attendus	– Peut aussi capturer du bruit – Peut être dépendant du corpus de développement ou du genre musical – Temps de calcul élevé
Théorie musicale	– Prend en compte la structure multi-niveau de la musique (harmonie, rythme, etc...) – Pas besoin de données annotées – Estimation conjointe d'autres notions (tonalité, rythme, etc...)	– Peut être dépendant du genre musical – Produit parfois des résultats décevants sur les chansons ne suivant pas la théorie musicale
Hybride	voir méthodes par apprentissage et par théorie musicale	voir méthodes par apprentissage et par théorie musicale

TABLE 2.1 – Points forts et points faibles des différentes catégories de méthodes de reconnaissance d'accords.

Chapitre 3

Corpus et évaluation

Ce chapitre propose une description du protocole d'évaluation et des corpus utilisés pour l'analyse des performances et la comparaison des méthodes de reconnaissance d'accords. Afin de permettre une plus grande cohérence avec nos publications, nous avons choisi de conserver les noms de métriques en anglais.

Le cadre d'évaluation présenté dans ce chapitre est largement inspiré de celui défini lors des évaluations internationales Music Information Retrieval Evaluation eXchange (MIREX) 2008 [1] & 2009 [2]. La tâche proposée lors de ces évaluations est la transcription d'un fichier WAVE en une séquence d'accords, ainsi que le temps de début et de fin de ces accords. Le dictionnaire d'accords utilisé (ensemble des accords que les systèmes doivent détecter) comporte 25 étiquettes : 12 accords majeurs, 12 accords mineurs et une étiquette 'N' correspondant aux silences.

Ainsi, tous les accords présents dans les fichiers d'annotation sont ramenés aux seuls accords majeurs et mineurs selon les règles décrites dans le tableau suivant :

majeur	maj, dim, aug, maj7, 7, dim7, hdim7, maj6, 9, maj9, sus4, sus2
mineur	min, min7, minmaj7, min6, min9

TABLE 3.1 – Correspondance entre types d'accords utilisée lors de MIREX 2008 & 2009

3.1 Corpus

3.1.1 Beatles

Le premier corpus utilisé est constitué des 13 albums des Beatles (180 chansons, PCM 44100 Hz, 16 bits, monophonique). Les annotations pour ces 13 albums sont fournies par Harte et al. (2005). Dans ces fichiers d'annotation on trouve 17 types d'accords ainsi qu'un état 'sans accord' (N) correspondant aux silences ou au contenu non instrumental.

Les accords les plus présents dans le corpus sont les accords majeurs (63.89% de la durée totale), mineurs (16.19%), septièmes de dominante (7.17%) et les états 'sans accord' (4.50%).

1. http://www.music-ir.org/mirex/wiki/2008:Audio_Chord_Detection
2. http://www.music-ir.org/mirex/wiki/2009:Audio_Chord_Detection

3.1.2 Quaero

Le deuxième corpus d'évaluation est mis à notre disposition par le projet Quaero [3]. Il est composé de 20 chansons annotées par l'IRCAM (PCM 22050 Hz, 16 bits, monophonique) d'artistes divers (Pink Floyd, Queen, Buenavista Social Club, Dusty Springfield, Aerosmith, Shack, UB40, Fall Out Boy, Nelly Furtado, Justin Timberlake, Mariah Carey, Abba, Cher, Phil Collins, Santa Esmeralda, Sweet, FR David and Enya) et de genres musicaux différents (pop, rock, electro, salsa, disco,...).

L'intêret principal de ce corpus réside dans le fait qu'il n'a jamais été utilisé pour le développement des méthodes de reconnaissance d'accords, et qu'il permet donc une évaluation autonome et objective.

3.2 Évaluation

La tâche de reconnaissance d'accords est la fusion de plusieurs sous-tâches : une tâche de reconnaissance d'étiquettes à proprement parler (trouver pour chaque trame le bon accord), mais aussi une tâche de segmentation (trouver les bonnes frontières temporelles pour les accords). De plus, une bonne transcription en accords doit être compacte et utiliser un vocabulaire d'accords cohérent et le plus pertinent possible. Nous décrivons ici plusieurs métriques d'évaluation permettant de rendre compte de tous ces différents aspects.

Supposons que nous disposions d'un corpus $\mathcal{S}$ composé de S chansons. Dans les fichiers d'annotation, chaque chanson s est composée de T_s segments temporels $\mathcal{U}(s) = \{u_1(s), \ldots, u_{T_s}(s)\}$. Pour chaque segment $u_t(s)$, les annotations fournissent une étiquette d'accord $l_t(s)$.

On écrit $|u|$ la durée du segment u et $u \cap u'$ l'intersection des segments u et u'. La durée totale de la chanson s est donc, avec nos notations, $|s| = \sum_{t=1}^{T_s} |u_t(s)|$.

Avec notre méthode de reconnaissance d'accords, la chanson s est divisée en $\hat{T}_s$ segments, et on attribue à chacun de ces segments $\hat{u}_t(s)$ une étiquette d'accord $\hat{l}_t(s)$.

3.2.1 Métriques de performances

On définit l'Overlap Score (OS(s)) comme le rapport entre la somme des durées des accords bien détectés et la durée totale de la chanson, c'est à dire :

$$OS(s) = \frac{\sum_{t=1}^{T_s} \sum_{t'=1}^{\hat{T}_s} |u_t(s) \cap \hat{u}_{t'}(s)|_{l_t(s)=\hat{l}_{t'}(s)}}{|s|} \tag{3.1}$$

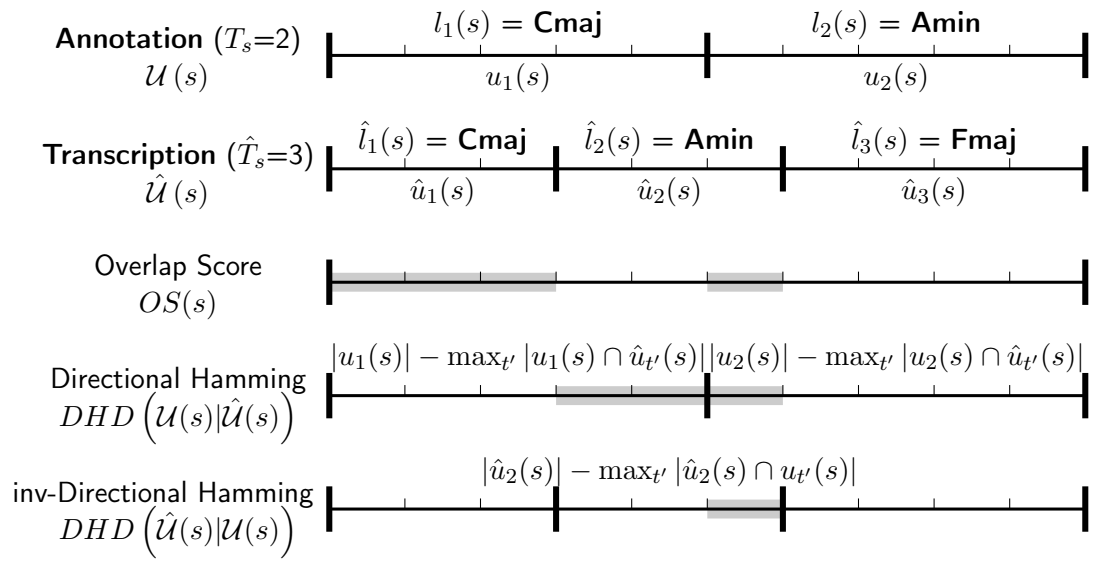
L'OS(s) est compris entre 0 et 1, et plus il est élevé, meilleures sont les performances.

L'Average Overlap Score (AOS), qui a notamment été utilisé lors de MIREX 2008, est la moyenne de tous les OS(s) du corpus :

$$AOS = \frac{1}{S} \sum_{s=1}^{S} OS(s) \tag{3.2}$$

On définit aussi l'Average Root Overlap Score (AROS) exactement de la même façon que l'AOS, mais évaluant cette fois-ci uniquement la détection de la fondamentale.

3. Quaero project : `http://www.quaero.org`



Overlap Score = $\frac{3+1}{10} = 0.4$

Hamming Distance = $\frac{1}{2} \times \left(\frac{2+1}{10} + \frac{1}{10}\right) = 0.2$

Reduced Chord Length = $\frac{\frac{3+3+4}{3}}{\frac{5+5}{2}} = \frac{2}{3}$

Reduced Chord Number = $\frac{3}{2}$

False Chord Label Number = 1

FIGURE 3.1 – Exemple de calcul des métriques suivantes : Overlap Score, Hamming Distance, Reduced Chord Length, Reduced Chord Number et False Chord Label Number.

3.2.2 Métriques de segmentation

Pour évaluer la qualité de la segmentation, des publications récentes (Mauch & Dixon (2010)) ont utilisé l'Hamming Distance (HD(s)) calculée à partir des Directional Hamming Divergences (DHDs) (Abdallah et al. (2005)). La DHD reflète l'inadéquation d'une segmentation par rapport à une autre. La DHD entre la segmentation annotée $\mathcal{U}(s)$ et celle issue de la transcription $\hat{\mathcal{U}}(s)$ est définie par :

$$DHD\left(\mathcal{U}(s)|\hat{\mathcal{U}}(s)\right) = \frac{\sum_{t=1}^{T_s} |u_t(s)| - \max_{t'} |u_t(s) \cap \hat{u}_{t'}(s)|}{|s|} \tag{3.3}$$

La DHD inverse est donc :

$$DHD\left(\hat{\mathcal{U}}(s)|\mathcal{U}(s)\right) = \frac{\sum_{t=1}^{\hat{T}_s} |\hat{u}_t(s)| - \max_{t'} |\hat{u}_t(s) \cap u_{t'}(s)|}{|s|} \tag{3.4}$$

Finalement, la HD(s) entre deux segmentations est définie comme la moyenne de ces deux DHDs :

$$HD\,(s) = \frac{DHD\left(\mathcal{U}(s)|\hat{\mathcal{U}}(s)\right) + DHD\left(\hat{\mathcal{U}}(s)|\mathcal{U}(s)\right)}{2} \tag{3.5}$$

La HD(s) reflète la dissimilarité de deux segmentations : elle prend des valeurs entre 0 et 1, et plus elle est faible, meilleure est la qualité de la segmentation. La moyenne de toutes les HD(s) du corpus est appelée Average Hamming Distance (AHD).

3.2.3 Métriques de fragmentation

La présence de nombreux accords fragmentés rend une transcription difficilement lisible et utilisable. Afin d'évaluer si une méthode produit des transcriptions fragmentées ou pas, nous introduisons une métrique appelée Average Chord Length (ACL). Définissons d'abord la Reduced Chord Length (RCL(s)), qui, pour une chanson s donnée, est le rapport entre la durée moyenne des accords sur la transcription et la durée moyenne des accords sur les annotations :

$$RCL(s) = \frac{\frac{1}{\hat{T}_s}\sum_{t=1}^{\hat{T}_s} |\hat{u}_t(s)|}{\frac{1}{T_s}\sum_{t=1}^{T_s} |u_t(s)|} \tag{3.6}$$

Cette métrique doit être la plus proche possible de 1 : en particulier, lorsqu'elle est inférieure à 1, cela signifie que la méthode a tendance à trop fragmenter le morceau. On définit naturellement l'ACL comme la moyenne de tous les RCL(s) du corpus.

3.2.4 Métriques de vocabulaire d'accords

Un autre indicateur de la qualité d'une transcription, est la compacité du vocabulaire d'accords utilisé pour la transcription. On définit le vocabulaire d'accords comme un sous ensemble du dictionnaire d'accords, qui contient toutes les étiquettes d'accords utiles pour transcrire une chanson donnée. Dans le cas où la chanson peut être rattachée à une tonalité donnée, il reflète le contenu tonal du morceau, mais nous voyons dans la notion de vocabulaire une entité plus générale. Par exemple, dans le cas de modulations, il est difficile de relier un morceau à une unique tonalité : le vocabulaire d'accords quant à lui peut contenir des accords de différentes

tonalités. La qualité du vocabulaire d'accords détecté est évalué par deux métriques : l'Average Chord Number (ACN) et l'Average False Chord Label Number (AFCLN).

Étant donnée une chanson s, nous définissons d'abord le Reduced Chord Number (RCN(s)) comme le rapport entre le nombre d'étiquettes d'accords différentes utilisées dans la transcription et celui des annotations. Cette métrique doit être le plus proche possible de 1 : quand elle est supérieure à 1, la transcription utilise un vocabulaire trop étendu. Finalement, l'ACN est la moyenne de tous les RCN(s) du corpus.

Pour chaque chanson s, le False Chord Label Number (FCLN(s)) est le nombre d'étiquettes d'accords qui n'appartiennent pas aux fichiers d'annotation. On définit l'AFCLN comme la moyenne de tous les FCLN(s) du corpus. Cette métrique doit être la plus petite possible : un AFCLN de 0 signifie que la méthode détecte exactement le bon vocabulaire.

Chapitre 4

Approche déterministe pour la reconnaissance d'accords

Publications associées

Oudre L., Grenier Y., Févotte C., "Chord recognition by fitting rescaled chroma vectors to chord templates", *submitted to* IEEE Transactions on Audio, Speech and Language Processing, 2009.

Oudre, L., Grenier, Y., Févotte, C. (2009). Chord recognition using measures of fit, chord templates and filtering methods. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New York, USA.

Oudre, L., Grenier, Y., Févotte, C. (2009). Template-based chord recognition : influence of the chord types. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. Kobe, Japan.

Nous décrivons dans ce chapitre une méthode déterministe de reconnaissance d'accords qui repose sur l'utilisation de gabarits d'accords.

4.1 Description de la méthode

4.1.1 Idée générale

Notre système de reconnaissance d'accords est basé sur l'idée intuitive que, étant donné un vecteur de chroma de dimension 12, les amplitudes des chromas présents dans l'accord joué doivent être plus grandes que celles des chromas absents. En introduisant des gabarits d'accords pour différents types d'accords et différentes notes fondamentales, l'accord présent sur une trame doit donc être celui dont le gabarit est le *plus proche* du vecteur de chroma, au sens d'une certaine mesure d'ajustement. Un paramètre d'échelle est introduit pour tenir compte des variations en amplitude, et finalement l'accord détecté est celui minimisant les mesures entre le vecteur de chroma mis à l'échelle et les gabarits d'accords.

Soit $\mathbf{C}$ le chromagramme, de dimension $12 \times N$ composé de N vecteurs de chroma successifs $\mathbf{c}_n$. Soit $\mathbf{w}_k$ le gabarit de dimension 12 définissant l'accord k. On veut trouver l'accord k dont le gabarit $\mathbf{w}_k$ est le *plus proche* de la trame de chromagramme $\mathbf{c}_n$ au sens d'une certaine

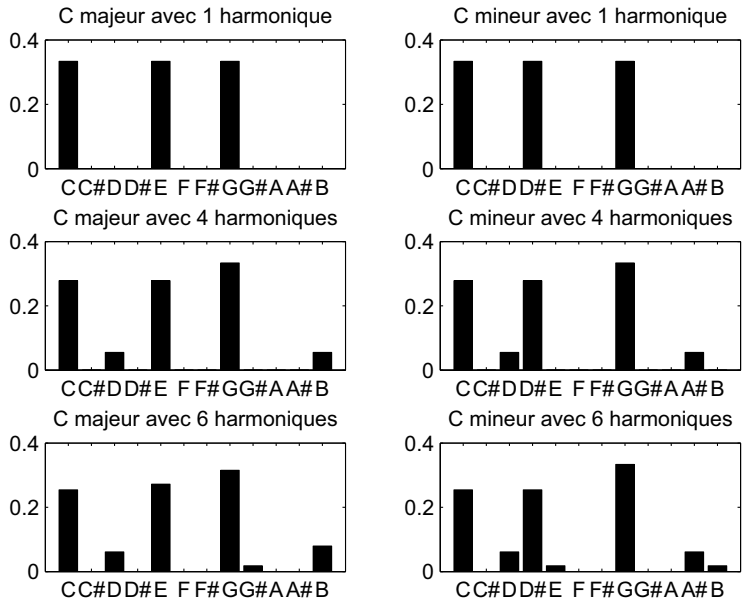


FIGURE 4.1 – Gabarits d'accords pour un accord de C majeur / C mineur avec 1, 4 et 6 harmoniques

mesure d'ajustement. On propose de mesurer la proximité entre le vecteur de chroma $\mathbf{c}_n$ et le gabarit $\mathbf{w}_k$ relativement à un paramètre $h_{k,n}$. Etant donné une mesure d'ajustement $D(\,.\,;\,.\,)$, un vecteur de chroma $\mathbf{c}_n$ et un gabarit d'accord $\mathbf{w}_k$, le paramètre d'échelle $h_{k,n}$ est calculé analytiquement pour minimiser la mesure d'ajustement entre $h\,\mathbf{c}_n$ and $\mathbf{w}_k$:

$$h_{k,n} = \underset{h}{\operatorname{argmin}}\; D\left(h\,\mathbf{c}_n; \mathbf{w}_k\right). \tag{4.1}$$

En pratique $h_{k,n}$ est calculé tel que :

$$\left[\frac{d\,D\left(h\,\mathbf{c}_n; \mathbf{w}_k\right)}{dh}\right]_{h=h_{k,n}} = 0. \tag{4.2}$$

On définit ensuite $d_{k,n}$:

$$d_{k,n} = D\left(h_{k,n}\,\mathbf{c}_n; \mathbf{w}_k\right). \tag{4.3}$$

L'accord détecté $\hat{k}_n$ pour la trame n est celui minimisant l'ensemble de valeurs $\{d_{k,n}\}_k$:

$$\hat{k}_n = \underset{k}{\operatorname{argmin}}\; d_{k,n}. \tag{4.4}$$

4.1.2 Gabarits d'accords

Nos gabarits d'accords sont des vecteurs de dimension 12, chaque composante représentant l'amplitude théorique de chaque chroma dans l'accord modélisé. Dans notre système, trois

structures d'accords sont définies. Des exemples pour les accords de *C majeur* et *C mineur* sont présentés sur la Figure 4.1.

- La **première structure d'accord** que nous avons utilisé consiste simplement en un modèle binaire : on donne une amplitude de 1 aux chromas présents dans l'accord que l'on veut modéliser et 0 aux chromas n'appartenant pas à l'accord. [1]
- La **deuxième structure d'accord** que nous utilisons s'inspire des travaux de Gómez (2006b), repris par Papadopoulos & Peeters (2007). Dans un chromagramme ou toute autre représentation spectrale du signal musical, on n'observe non pas l'intensité de chaque note, mais le mélange des intensités des différentes harmoniques de chaque note. Il est donc intéressant de prendre en compte les harmoniques pour chaque note de l'accord. On suppose une décroissance spectrale exponentielle pour l'amplitude des partielles, et on ajoute ainsi une amplitude de h^{i-1} pour la i^{th} harmonique de chaque note de l'accord. Le paramètre h est fixé empiriquement à 0.6. Notre deuxième structure d'accord prend seulement en compte les 4 premières harmoniques.
- La **troisième structure d'accord** est basée sur le même principe mais prend en compte les 6 premières harmoniques de chaque note de l'accord.

A partir de ces trois structures d'accords on peut construire des gabarits d'accords pour n'importe quel type (majeur, mineur, septième, etc...).

On supposera par convention que les gabarits d'accords sont normalisés de façon à ce que la somme des amplitudes vaille 1.

4.1.3 Mesures d'ajustement

Pour notre tâche de reconnaissance d'accords, plusieurs mesures populaires dans le domaine du traitement du signal sont considérées. La Table 4.1 donne les expressions de ces différentes mesures d'ajustement, celles des paramètres d'échelle calculés analytiquement à partir de l'Equation (4.2) ainsi que celles du critère de reconnaissance $d_{k,n}$.

- La **distance euclidienne** (*EUC*) est définie par :

$$D_{EUC}\left(\mathbf{x}|\mathbf{y}\right) = \sqrt{\sum_{m}\left(x_m - y_m\right)^2} \tag{4.5}$$

  et a notamment déjà été utilisée par Fujishima (1999) pour la tâche de reconnaissance d'accords.

- La **divergence d'Itakura-Saito** (Itakura & Saito (1968)) est définie par :

$$D_{IS}\left(\mathbf{x}|\mathbf{y}\right) = \sum_{m}\frac{x_m}{y_m} - \log\left(\frac{x_m}{y_m}\right) - 1 \tag{4.6}$$

  et fut présentée comme une mesure de proximité entre deux spectres : elle devint populaire dans la communauté du traitement de la parole dans les années 70. Ce n'est pas une distance : elle n'est en effet par symétrique. On peut donc la calculer de deux façons : $D\left(h_{k,n}\,\mathbf{c}_n|\mathbf{w}_k\right)$ définit la mesure *IS1*, tandis que $D\left(\mathbf{w}_k|h_{k,n}\,\mathbf{c}_n\right)$ définit *IS2* (voir Table 4.1 pour leur définition).

1. En pratique une très faible valeur est utilisée au lieu de 0, pour éviter les instabilités numériques qui pourrait apparaître avec certaines mesures.

– La **divergence de Kullback-Leibler** (Kullback & Leibler (1951)) mesure la dissimilarité entre deux distributions de probabilité. Elle fut très largement utilisée en particulier dans le domaine de la théorie de l'information et a donné lieu à de nombreuses variantes : on choisit ici d'utiliser la divergence de Kullback-Leibler généralisée, définie par :

$$D_{KL}\left(\mathbf{x}|\mathbf{y}\right) = \sum_{m} x_m \log\left(\frac{x_m}{y_m}\right) - x_m + y_m. \tag{4.7}$$

Tout comme la divergence d'Itakura-Saito, elle n'est pas symétrique et on peut donc introduire deux mesures, notées *KL1* et *KL2* (voir Table 4.1 pour leur définition).

4.1.4 Post-traitement par filtrage

Jusqu'à présent, notre détection d'accords se fait trame par trame, sans tenir compte des résultats sur les trames adjacentes. En pratique, il est peu probable qu'un accord ne soit présent que sur une trame. De plus, l'information contenue dans les trames adjacentes peut aider à obtenir un résultat pertinent. Nous allons donc introduire un lissage qui agira non pas sur la séquence d'accords détectés, mais en amont, sur le critère de reconnaissance. Ce lissage vise à informer implicitement notre système de la durée acceptable d'un accord. Notons que certaines publications utilisaient déjà ce type de lissage sur le chromagramme (Fujishima (1999), Peeters (2006), Bello & Pickens (2005)) ou sur la séquence d'accords détectés (Bello & Pickens (2005)), mais son application directement sur le critère de reconnaissance est novatrice.

Supposons que l'on ait calculé comme précédemment les $d_{k,n}$ correspondant aux mesures entre les trames de chromagramme après changement d'échelle et les gabarits d'accords. Au lieu d'utiliser ces paramètres comme critères à minimiser, nous allons calculer des critères $\tilde{d}_{k,n}$ basés non plus uniquement sur les mesures concernant la trame n, mais sur les mesures concernant L trames centrées sur la trame n (L est donc impair). Ces critères $\tilde{d}_{k,n}$ vont ainsi être calculés à partir des mesures $d_{k,n}$ précédemment calculées sur les L trames adjacentes.

Dans notre système, deux types de filtrage vont être testés.

– Le **filtrage passe-bas** défini par :

$$\tilde{d}_{k,n} = \frac{1}{L} \sum_{n'=n-\frac{L-1}{2}}^{n'=n+\frac{L-1}{2}} d_{k,n'} \tag{4.8}$$

tend à lisser la séquence d'accords détectés et à refléter les changements d'accords à long terme.

– Le **filtrage médian** défini par :

$$\tilde{d}_{k,n} = \text{median}\left\{d_{k,n'}\right\}_{n-\frac{L-1}{2} \leq n' \leq n+\frac{L-1}{2}} \tag{4.9}$$

a été largement utilisé dans le traitement d'images et est particulièrement efficace pour la suppression des erreurs aléatoires.

Finalement, l'accord détecté $\hat{k}_n$ sur la trame n sera celui minimisant l'ensemble de valeurs $\left\{\tilde{d}_{k,n}\right\}_k$:

$$\hat{k}_n = \underset{k}{\text{argmin}}\ \tilde{d}_{k,n} \tag{4.10}$$

TABLE 4.1 – Présentation des mesures d'ajustement (les expressions supposent $\|\mathbf{p}_k\|_1 = 1$)

	Expression de $D\left(h_{k,n}\,\mathbf{c}_n;\mathbf{w}_k\right)$	Paramètre d'échelle $h_{k,n}$	Critère de reconnaissance $d_{k,n}$
EUC	$\sqrt{\sum_m \left(h_{k,n}\,c_{m,n} - w_{m,k}\right)^2}$	$\frac{\sum_m c_{m,n}\,w_{m,k}}{\sum_m c_{m,n}^2}$	$\sqrt{\sum_m w_{m,k}^2 - \frac{\left(\sum_m c_{m,n}\,w_{m,k}\right)^2}{\sum_m c_{m,n}^2}}$
IS1	$\sum_m \frac{h_{k,n}\,c_{m,n}}{w_{m,k}} - \log\left(\frac{h_{k,n}\,c_{m,n}}{w_{m,k}}\right) - 1$	$\frac{M}{\sum_m \frac{c_{m,n}}{w_{m,k}}}$	$M\log\left(\frac{1}{M}\sum_m \frac{c_{m,n}}{w_{m,k}}\right) - \sum_m \log\left(\frac{c_{m,n}}{w_{m,k}}\right)$
IS2	$\sum_m \frac{w_{m,k}}{h_{k,n}\,c_{m,n}} - \log\left(\frac{w_{m,k}}{h_{k,n}\,c_{m,n}}\right) - 1$	$\frac{1}{M}\sum_m \frac{w_{m,k}}{c_{m,n}}$	$M\log\left(\frac{1}{M}\sum_m \frac{w_{m,k}}{c_{m,n}}\right) - \sum_m \log\left(\frac{w_{m,k}}{c_{m,n}}\right)$
KL1	$\sum_m h_{k,n}\,c_{m,n}\log\left(\frac{h_{k,n}\,c_{m,n}}{w_{m,k}}\right) - h_{k,n}\,c_{m,n} + w_{m,k}$	$e^{-\sum_m c'_{m,n}\log\left(\frac{c_{m,n}}{w_{m,k}}\right)}$ avec $c'_{m,n} = \frac{c_{m,n}}{\|\mathbf{c}_n\|_1}$	$1 - e^{-\sum_m c'_{m,n}\log\left(\frac{c'_{m,n}}{w_{m,k}}\right)}$ avec $c'_{m,n} = \frac{c_{m,n}}{\|\mathbf{c}_n\|_1}$
KL2	$\sum_m w_{m,k}\log\left(\frac{w_{m,k}}{h_{k,n}\,c_{m,n}}\right) - w_{m,k} + h_{k,n}\,c_{m,n}$	$\frac{1}{\sum_m c_{m,n}}$	$\sum_m w_{m,k}\log\left(\frac{w_{m,k}}{c'_{m,n}}\right) - w_{m,k} + c'_{m,n}$ avec $c'_{m,n} = \frac{c_{m,n}}{\|\mathbf{c}_n\|_1}$

4.2 Tests

Avec les trois blocs constitutifs de notre approche (gabarits d'accords, mesures d'ajustement et post-traitement), nous pouvons construire un grand nombre de systèmes de reconnaissance d'accords. Nous avons à notre disposition :

- **5 mesures d'ajustement :** EUC, IS1, IS2, KL1 et KL2 ;
- **3 structures d'accords :** 1, 4 ou 6 harmoniques ;
- **2 types de filtrage :** passe-bas et médian ;
- **12 tailles de voisinage (lorsque l'on applique un post-traitement) :** de $L = 3$ à $L = 25$.

Cela donne $5 \times 3 \times 2 \times 12$ (avec filtrage) $+ 5 \times 3$ (sans filtrage) $= 375$ choix de paramètres qui peuvent être vus comme autant de systèmes de reconnaissance d'accords.

Notre but dans cette section est de sélectionner quel est le choix de paramètre le plus efficace et le plus robuste afin de ne conserver qu'un seul système.

4.2.1 Calcul du chromagramme

Le chromagramme est calculé de la même façon que celui de Bello & Pickens (2005). On utilise la transformée à Q constant introduite par Brown (1991), qui permet une analyse fréquentielle sur des bandes fréquentielles dont les fréquences centrales sont réparties de façon logarithmique. En effet, la fréquence centrale f_k de la k^{eme} bande fréquentielle est définie de la façon suivante :

$$f_k = 2^{\frac{k}{b}} f_{min} \tag{4.11}$$

où b représente le nombre de bandes fréquentielles par octave, et f_{min} la fréquence à partir de laquelle l'analyse commence.

Le signal est tout d'abord sous échantillonné à 5512.5 Hz, et l'on réalise la transformée à Q constant avec $b = 36$ soit 3 bins par demi-ton, entre les fréquences 73.42 Hz (D2) et 587.36 Hz (D5). La longueur de la fenêtre d'analyse est ainsi de 4096 échantillons (743 ms) et l'on fait des sauts de 512 échantillons (93 ms).

Grâce à la résolution de 36 bandes fréquentielles par octave, on peut utiliser un algorithme de diapason, comme celui proposé par Harte & Sandler (2005), dans une version simplifiée. Après une détection de pics dans le chromagramme, on calcule un facteur de correction afin de tenir compte du décalage de diapason. On réalise enfin un filtrage médian afin d'éliminer les transitions trop brutales.

Des précisions sur le calcul du chromagramme peuvent être trouvées dans Bello & Pickens (2005).

4.2.2 Premiers résultats

Nous avons calculé pour chacun des 375 systèmes l'AOS obtenu sur le corpus des Beatles en utilisant un dictionnaire de 24 accords (majeur et mineur).

Le tableau 4.2 présente les AOSs obtenus sur le corpus des Beatles. Les meilleurs résultats sont obtenus avec la mesure *KL2*, la structure d'accord à 4 harmoniques, et le filtrage médian avec $L = 15$ (2.04s), donnant un taux de reconnaissance de 71.8%. Dans la suite du document, nous appelerons cette méthode OGF1 *(maj-min)*.

	sans filtrage			filtrage passe-bas			filtrage médian		
	1 harm.	4 harm.	6 harm.	1 harm.	4 harm.	6 harm.	1 harm.	4 harm.	6 harm.
EUC	0.665	0.636	0.588	0.710	0.684	0.646	0.705	0.679	0.636
IS1	0.665	0.441	0.399	0.706	0.460	0.415	0.706	0.465	0.422
IS2	0.657	0.667	0.170	0.704	0.713	0.178	0.703	0.714	0.178
KL1	0.665	0.487	0.140	0.700	0.532	0.151	0.692	0.498	0.143
KL2	0.667	0.672	0.612	0.709	0.712	0.648	0.714	**0.718**	0.656

TABLE 4.2 – Average Overlap Scores obtenus sur les 13 albums des Beatles avec l'approche déterministe. Afin de ne pas alourdir ce tableau, nous avons seulement représenté pour chaque méthode de post-traitement les résultats correspondant à un choix optimal de L.

4.2.3 Introduction d'autres types d'accords

La simplicité de notre méthode permet d'introduire facilement des gabarits d'accords pour des accords autres que majeurs et mineurs. On étudie dans cette partie l'influence des types d'accords considérés sur les performances de notre système. Le choix de ces types d'accords est guidé par les statistiques sur le corpus présentés Section 3.1.1 : on introduira en priorité les accords les plus présents.

Dans le corpus des Beatles, les deux types d'accords les plus présents (à part majeur et mineur) sont les accords de septième de dominante *(7)* et les accords de mineur septième *(min7)*. Les résultats pour les accords majeurs, mineurs, septièmes de dominante, et mineurs septième sont présentés sur la tableau 4.3.

Types d'accords	AOS	Paramètres optimaux
maj-min	0.718	KL2, 4 harm, médian, L=15 (2.04s)
maj-min + 7	0.724	KL2, 1 harm, médian, L=17 (2.23s)
maj-min + 7 + min7	0.706	IS1, 1 harm, passe-bas, L=13 (1.86s)

TABLE 4.3 – Average Overlap Scores obtenus sur le corpus des Beatles avec les accords majeurs, mineurs, septièmes de dominante et mineurs septième.

Les meilleurs résultats sont obtenus en détectant les accords majeurs, mineurs et septièmes de dominante, avec la mesure *KL2*, la structure d'accord à 1 harmonique et le filtrage médian avec $L = 17$, donnant un taux de reconnaissance de 72.4%. Ce système particulier sera nommé OGF2 *(maj-min-7)* dans la suite du document.

Seule l'introduction des accords septièmes de dominante, très présents dans le corpus des Beatles, améliore les résultats. En effet, la structure des accords de mineur septième mène à des confusions entre l'accord mineur original et l'accord de relative majeure. Par exemple, un accord de *Cmin7* peut être confondu avec un accord de $E\flat$.

Les accords diminués et augmentés ont été pris en compte dans de nombreux systèmes de reconnaissance d'accords basés sur des gabarits d'accords (Fujishima (1999), Harte & Sandler (2005)). Bien que les accords diminués et augmentés sont très rares dans le corpus des Beatles (respectivement 0.62% et 0.38% de la durée totale), l'introduction de gabarits d'accords pour ces accords ne dégrade pas les résultats. On obtient en effet un taux de reconnaissance de 70.2% en considérant les accords majeurs, mineurs, diminués et augmentés, et de 72.4% en considérant les accords majeurs, mineurs, septièmes de dominante, diminués et augmentés.

L'introduction d'autres types d'accords (neuvième, majeur septième, quatrième suspendu,

etc...) n'améliore pas les résultats. Cela peut être expliqué soit par la structure des accords qui peut créer des confusions avec un autre type d'accord, ou par le faible nombre d'accords de ces types dans le corpus des Beatles. En effet, l'introduction d'un gabarit pour un nouveau type d'accord donne certes une meilleure détection pour les accords de ce type, mais produit aussi de nouvelles erreurs assimilables à des fausses détections. C'est pourquoi seuls les types d'accords très fréquents peuvent être introduits, assurant ainsi que l'amélioration causée par la meilleure reconnaissance de ces types d'accords est plus grande que la dégradation des résultats causés par les fausses détections.

4.2.4 Test sur le corpus Quaero

Nous avons pour le moment uniquement montré les résultats sur le corpus des Beatles, qui constitue le corpus de développement de nos systèmes. Nous pouvons ainsi nous demander si le choix de paramètres que nous avons fait sur le corpus est aussi valable pour d'autres artistes et genres musicaux. Nous avons donc lancé nos deux systèmes OGF1 *(maj-min)* et OGF2 *(maj-min-7)* sur le corpus Quaero décrit Section 3.1.2.

On obtient sur ce corpus un AOS de 0.706 avec OGF1 *(maj-min)* et de 0.682 avec OGF2 *(maj-min-7)*. On peut remarquer que ces AOSs sont assez proches de ceux obtenus sur le corpus des Beatles.

De plus, nos simulations montrent que le choix de paramètres fait sur le corpus des Beatles est aussi pertinent sur ce corpus. En effet, considérant les accords majeurs et mineurs, les paramètres optimaux donnent un AOS de 0.709 (contre 0.706 avec les paramètres de base). En considérant les accords majeurs, mineurs, et septièmes de dominante, les meilleurs paramètres obtiennent un score de 0.695 (contre 0.682 avec les paramètres classiques). Ceci laisse penser que nos systèmes ne sont pas fortement dépendants de leur corpus de développement et que le choix de paramètres fait sur le corpus des Beatles est valide sur d'autres morceaux.

4.3 Résultats

Les systèmes présentés dans ce chapitre furent développés en 2008-2009, lorsque les méthodes de l'état de l'art étaient celles proposés pour MIREX 2008. Nous ne comparerons nos méthodes qu'à celles disponibles à cette époque. Le chapitre suivant proposera une comparaison avec les méthodes de MIREX 2009.

De plus, puisqu'une évaluation avec un large spectre de métriques sera présentée dans le chapitre suivant, nous avons décidé de n'utiliser ici qu'une métrique : l'AOS.

Pour notre comparaison, nous avons choisi des méthodes qui ont participé à MIREX 2008, qui ont obtenu de bons résultats, qui n'utilisent pas d'apprentissage (ou sont pré-entraînés) et dont les auteurs ont accepté de nous laisser utiliser leur code. Nos systèmes sont maintenant comparés aux méthodes suivantes :

- **BP :** Bello & Pickens (2005) utilisent un MMC à 24 états avec des initialisations inspirées par de la théorie musicale, des distributions d'observation gaussiennes et un entraînement EM pour la distribution initiale d'états et la matrice de transitions entre états.
- **RK :** Ryynänen & Klapuri (2008a) utilisent un MMC à 24 états avec des distributions d'observation générées en comparant deux profils d'aigus et de graves avec des gabarits d'accords estimés. On utilise un entraînement EM pour la distribution initiale d'états et la matrice de transitions entre états.

	Corpus Beatles		Corpus Quaero	
	AOS	Temps de calcul	AOS	Temps de calcul
OGF1 *(maj-min)*	0.718	790s	0.706	95s
OGF2 *(maj-min-7)*	**0.724**	796s	0.682	97s
BP	0.707	1619s	0.699	261s
RK	0.705	2241s	**0.730**	350s
KO	0.663	1668s	0.503	255s
PVM	0.647	12402s	0.664	2684s

TABLE 4.4 – Comparaison avec l'état de l'art

– **KO :** Khadkevich & Omologo (2008) utilisent 24 MMCs : un pour chaque accord. Les distributions d'observation sont des mixtures de Gaussiennes et tous les paramètres sont entraînés par un algorithme EM.

– **PVM :** Pauwels et al. (2008) utilisent un cadre probabiliste pour les tâches conjointes de reconnaissance d'accords et de tonalité.

Les résultats sur le corpus des Beatles et le corpus Quaero sont présentés Table 4.4.

Tout d'abord, on peut remarquer que toutes les méthodes donnent des résultats assez proches sur le corpus des Beatles : il n'y a que 8% de différence entre les méthodes donnant les meilleurs et les moins bons résultats. Notre méthode OGF2 *(maj-min-7)* donne les meilleurs résultats, mais surtout, avec un temps de calcul très faible. En effet, notre méthode est deux fois plus rapide que la meilleure méthode de l'état de l'art (BP).

Il est intéressant de remarquer que les scores sur le corpus Quaero sont plus contrastés : en particulier KO donne de moins bons scores alors que RK obtient ici les meilleurs résultats. Nos méthodes, bien qu'elles ne donnent plus les meilleurs résultats, se comportent bien sur ce corpus : notre méthode OGF1 *(maj-min)* donne le deuxième meilleur AOS.

4.4 Conclusion

Résumons les principales contributions et conclusions de ce chapitre :

– L'utilisation de gabarits d'accords théoriques permet d'éviter la tâche fastidieuse d'annotation de données et d'introduire facilement de nombreux types d'accords. Nos expériences ont montré que l'introduction des harmoniques dans les gabarits d'accords n'améliorent pas significativement les résultats, et que la plupart du temps, des gabarits binaires suffisent à obtenir de bonnes performances.
– L'utilisation d'un post-traitement par filtrage sur le critère de reconnaissance est nouvelle et les résultats montrent qu'elle permet de prendre en compte la durée probable des accords dans une chanson.
– Nos méthodes se comportent bien sur d'autre corpus, qu'importe le genre, le style ou l'artiste. Le choix de paramètres fait sur le corpus des Beatles, est aussi valide sur d'autres corpus, ce qui tend à montrer que notre système n'est pas fortement biaisé par son corpus de développement.
– La simplicité de nos méthodes permet de conserver un temps de calcul faible.

Nos systèmes souffrent néanmoins des désavantages inhérents aux méthodes utilisant uniquement des gabarits d'accords. Par exemple, nous avons observé que nos méthodes tendent à

produire des transcriptions trop fragmentées, qui deviennent ainsi difficilement utilisables pour rejouer directement une chanson. De plus, elles ont aussi tendance à utiliser trop d'accords pour transcrire un morceau, ce qui produit des accords parasites, qui, encore une fois, nuisent à la lisibilité de la transcription.

Chapitre 5

Approche probabiliste pour la reconnaissance d'accords

Publications associées

Oudre L., Févotte C., Grenier Y., "Probabilistic template-based chord recognition", *accepted in* IEEE Transactions on Audio, Speech and Language Processing, 2010.

Oudre, L., Févotte, C., Grenier, Y. (2010). Probabilistic framework for template-based chord recognition. *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*. St Malo, France.

Nous avons présenté dans le chapitre 4 une méthode simple et efficace pour la reconnaissance d'accords. Néanmoins, nous avons observé que cette méthode, dans laquelle aucune information autre que la définition des accords n'était introduite, avait parfois des difficultés à percevoir certains aspects temporels. En particulier, les transcriptions produites par notre système déterministe étaient parfois trop fragmentées pour être directement utilisables pour rejouer des morceaux.

Nous proposons dans ce chapitre une approche qui se base sur notre système précédent, mais décrit un cadre probabiliste qui permet, en prenant en compte la distribution de probabilité des accords de la chanson, de produire des transcriptions plus claires, et ainsi de résoudre certains problèmes posés par notre approche déterministe.

5.1 Description de la méthode

Dans ce cadre probabiliste, les mesures servant à comparer les vecteurs de chroma et les gabarits d'accords sont vues comme des fonctions de vraisemblance, et les apparitions des accords sont des évènements probabilistes. En particulier, la probabilité d'apparition de chaque accord du dictionnaire dans la chanson est apprise directement à partir de la chanson, ce qui permet une estimation explicite du vocabulaire d'accords. Nous rappelons ici que nous appelons vocabulaire d'accords l'ensemble des accords présents dans une chanson (voir Section 3.2.4).

Dans notre approche déterministe, nous avions utilisé comme critère de reconnaissance le terme $d_{k,n} = D\left(h_{k,n}\,\mathbf{c}_n; \mathbf{w}_k\right)$. Avec cette formulation, $h_{k,n}$ était un paramètre d'échelle ou normalisation, qui s'appliquait au vecteur de chroma $\mathbf{c}_n$ afin de l'adapter au gabarit d'accord

Gaussien	$\mathcal{N}\left(x;\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Gamma	$\mathcal{G}\left(x;\alpha,\beta\right) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta x}$
Poisson	$\mathcal{P}\left(x;\lambda\right) = \frac{\lambda^x}{\Gamma(x+1)} e^{-\lambda}$

où Γ est la fonction Gamma.

TABLE 5.1 – Définitions des distributions de probabilité

$\mathbf{w}_k$. Dans ce chapitre, nous allons travailler avec le pendant de ce paramètre d'échelle, qui s'applique non plus au vecteur de chroma mais au gabarit d'accord. Ainsi, nous définissons $a_{k,n}$ comme étant un *paramètre d'amplitude* et l'hypothèse

$$h_{k,n}\,\mathbf{c}_n \approx \mathbf{w}_k \tag{5.1}$$

devient maintenant

$$\mathbf{c}_n \approx a_{k,n}\,\mathbf{w}_k. \tag{5.2}$$

Ainsi, le critère précédemment utilisé devient, avec nos nouvelles notations $d'_{k,n} = D\left(\mathbf{c}_n; a_{k,n}\,\mathbf{w}_k\right)$. Nous verrons par la suite que ces deux conceptions sont très proches et quasi équivalentes. Néanmoins, ces nouvelles notations seront plus pratiques pour la définition de notre système probabiliste.

5.1.1 Modèle génératif

Lorsque nous utilisions la distance euclidienne ou les divergences de Kullback-Leibler ou d'Itakura-Saito, le critère $D\left(\mathbf{c}_n; a_{k,n}\,\mathbf{w}_k\right)$ était en fait une log-vraisemblance. En effet, les mesures précédemment définies sous-tendent en fait respectivement des modèles de bruit d'observation additif Gaussien, Poisson, et multiplicatif Gamma (définis Table 5.1) et peuvent être reliées à des log-vraisemblances telles que :

$$-\log p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right) = \varphi_1\, D\left(\mathbf{c}_n|a_{k,n}\,\mathbf{w}_k\right) + \varphi_2, \tag{5.3}$$

où $p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right)$ est la probabilité du vecteur de chroma $\mathbf{c}_n$ (qui est maintenant une variable aléatoire), étant donnés le gabarit $\mathbf{w}_k$ et le paramètre d'amplitude $a_{k,n}$. φ_1 and φ_2 sont des constantes vis à vis de $a_{k,n}$ et $\mathbf{w}_k$. Les correspondances exactes entre chaque mesure et son modèle statistique équivalent sont données Table 5.2.

Introduisons la variable d'état discrète $\gamma_n \in [1,\ldots,K]$ qui indique quel accord est joué à la trame n ($\gamma_n = k$ si l'accord k est joué). Nous pouvons alors écrire :

$$p\left(\mathbf{c}_n|\gamma_n = k, a_{k,n}\right) = p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right). \tag{5.4}$$

Soit α_k la probabilité d'apparition de l'accord k dans la chanson. Nous avons donc :

$$P(\gamma_n = k) = \alpha_k, \tag{5.5}$$

Notre modèle génératif devient finalement :

$$p\left(\mathbf{c}_n|\boldsymbol{\alpha},\mathbf{a}_n\right) = \sum_{k=1}^{K} \alpha_k\, p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right). \tag{5.6}$$

Pour récapituler, selon notre modèle une trame de chromagramme $\mathbf{c}_n$ est générée en :

	Structure de bruit	Modèle d'observation $p(\mathbf{c}_n \vert a_{k,n}, \mathbf{w}_k)$	Log-vraissemblance $-\log\left(p\left(\mathbf{c}_n \vert a_{k,n}, \mathbf{w}_k\right)\right)$
Gaussien	Bruit additif Gaussien $\mathbf{c}_n = a_{k,n}\mathbf{w}_k + \epsilon$	$\prod_{m=1}^{M} \mathcal{N}\left(c_{m,n}\ ;\ a_{k,n}w_{m,k}, \sigma^2\right)$	$\frac{1}{2\sigma^2} d^2_{EUC}\left(\mathbf{c}_n ; a_{k,n}\mathbf{w}_k\right) + cst$
Gamma	Bruit multiplicatif Gamma $\mathbf{c}_n = \left(a_{k,n}\mathbf{w}_k\right) \cdot \epsilon$	$\prod_{m=1}^{M} \frac{1}{a_{k,n}w_{m,k}} \mathcal{G}\left(\frac{c_{m,n}}{a_{k,n}w_{m,k}}\ ;\ \beta, \beta\right)$	$\beta\, d_{IS}\left(\mathbf{c}_n \vert a_{k,n}\mathbf{w}_k\right) + cst$
Poisson	Bruit Poisson	$\prod_{m=1}^{M} \mathcal{P}\left(c_{m,n}\ ;\ a_{k,n}w_{m,k}\right)$	$d_{KL}\left(\mathbf{c}_n \vert a_{k,n}\mathbf{w}_k\right) + cst$

où $\mathcal{N}$, $\mathcal{G}$ et $\mathcal{P}$ sont les distributions de probabilité définies Table 5.1 et cst représente des termes constants vis à vis de $a_{k,n}\mathbf{w}_k$.

TABLE 5.2 – Correspondance entre mesures et modèles d'observation

	Paramètre d'échelle $h_{k,n}$	Paramètre d'amplitude $a_{k,n}$
EUC / Gaussien	$\frac{\sum_{m=1}^{M} c_{m,n}\, w_{m,k}}{\sum_{m=1}^{M} c_{m,n}^2}$	$\frac{\sum_{m=1}^{M} c_{m,n}\, w_{m,k}}{\sum_{m=1}^{M} w_{m,n}^2}$
IS1 / Gamma	$\frac{M}{\sum_{m=1}^{M} \frac{c_{m,n}}{w_{m,k}}}$	$\frac{1}{M} \sum_{m=1}^{M} \frac{c_{m,n}}{w_{m,k}}$
KL1 / Poisson	$e^{-\sum_{m=1}^{M} c'_{m,n} \log\left(\frac{c_{m,n}}{w_{m,k}}\right)}$	$\sum_{m=1}^{M} c_{m,n}$

TABLE 5.3 – Correspondances entre paramètres d'échelle et d'amplitude

1. choisissant aléatoirement un accord k avec la probabilité α_k,
2. mettant à l'échelle $\mathbf{w}_k$ avec le paramètre $a_{k,n}$ afin de prendre en compte les variations d'amplitude,
3. générant $\mathbf{c}_n$ selon le modèle de bruit voulu et $a_{k,n}\mathbf{w}_k$.

Les paramètres que nous devons estimer sont les probabilités d'accords $\boldsymbol{\alpha}$ et les paramètres d'amplitude $\mathbf{A} = \{a_{k,n}\}_{kn}$. Une fois ces paramètres estimés, la reconnaissance d'accords pour chaque trame n est réalisée en sélectionnant l'accord avec la plus grande probabilité à posteriori $\bar{\alpha}_{k,n} = p(\gamma_n = k|\mathbf{c}_n, \hat{\boldsymbol{\alpha}}, \hat{\mathbf{a}}_n)$:

$$\hat{\gamma}_n = \underset{k}{\operatorname{argmax}}\ \bar{\alpha}_{k,n}. \tag{5.7}$$

5.1.2 Algorithme Espérance-Maximisation (EM)

Nous ne détaillerons pas dans ce résumé les calculs menant à la formulation de l'algorithme d'estimation des paramètres $\boldsymbol{\alpha}$ et $\mathbf{A} = \{a_{k,n}\}_{kn}$. Notons juste que ces calculs montrent qu'il est possible de pré-calculer les paramètres d'amplitude $\mathbf{A} = \{a_{k,n}\}_{kn}$ selon les règles définies Table 5.3.

L'algorithme final est présenté ci dessous.

Algorithm 1: Algorithme EM pour la reconnaissance d'accords

Input: Chromagramme $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$, gabarits d'accords $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$
Output: Probabilités d'accords $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$

Initialiser $\boldsymbol{\alpha}$
Calculer les paramètres d'amplitude $\mathbf{H}$

for $i = 1 : n_{iter}$ **do**

$$\bar{\alpha}_{k,n}^{(i-1)} = \frac{p\big(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\big)\,\alpha_k^{(i-1)}}{\sum_{l=1}^{K} p\big(\mathbf{c}_n|a_{l,n},\mathbf{w}_l\big)\,\alpha_l^{(i-1)}} \qquad \texttt{// E-Step}$$

$$\alpha_k^{(i)} = \frac{\sum_{n=1}^{N} \bar{\alpha}_{k,n}^{(i-1)}}{\sum_{l=1}^{K}\sum_{n=1}^{N} \bar{\alpha}_{l,n}^{(i-1)}} \qquad \texttt{// M-Step}$$

5.1.3 Post-traitement par filtrage

Nous avons déjà vu que notre critère de reconnaissance est basé sur les probabilités à posteriori :

$$\hat{\gamma}_n = \underset{k}{\operatorname{argmax}} \{\bar{\alpha}_{k,n}\}_k. \tag{5.8}$$

Tout comme dans l'approche déterministe, cette détection trame-par-trame peut être améliorée en appliquant un filtrage *ad hod* sur ces probabilités, afin d'informer implicitement notre système de la durée probable d'un accord dans une chanson.

5.2 Tests

Cette section vise à déterminer les meilleurs paramètres pour nos systèmes afin d'adapter nos modèles à la tâche de reconnaissance d'accords et à avoir une première idée sur leurs performances.

5.2.1 Détermination des hyper-paramètres

Deux de nos trois modèles de bruit d'observation dépendent d'hyper-paramètres. En effet, le modèle Gaussien prend en compte un paramètre σ^2, tandis que le modèle Gamma utilise le paramètre β. Le rôle de ces hyper-paramètres est d'adapter le modèle de bruit au bruit réellement présent dans les vecteurs de chroma.

Avant de donner les premiers résultats, nous devons évaluer les meilleurs hyper-paramètres pour ces deux modèles de bruit d'observation. Nous proposons de lancer plusieurs tests à vide (sans filtrage) pour plusieurs valeurs de ces hyper-paramètres et de calculer l'AOS obtenu sur le corpus des Beatles. Le nombre d'itérations de l'algorithme EM est fixé à 200, vu que selon nos simulations, ce nombre est suffisant pour faire converger le critère de maximisation.

Les résultats de ces expériences montrent que l'ajustement de ces hyper-paramètres est crucial puisque, par exemple, des valeurs trop basses de σ^2 ou trop élevées de β donnent de mauvais résultats. Le choix optimal pour σ^2 est 0.04 et pour β est 3. Nous utiliserons ces valeurs dans le reste du chapitre.

5.2.2 Premiers résultats

Nous présentons Table 5.4 les AOS obtenus par nos systèmes sur le corpus des Beatles. Les meilleurs résultats sont obtenus avec le modèle de bruit d'observation Gamma et le filtrage passe-bas sur 15 trames.

	sans filtrage	passe-bas	médian
Gaussien	0.714	0.748	0.749
Gamma	0.730	**0.758**	**0.758**
Poisson	0.727	0.742	0.744

TABLE 5.4 – Average Overlap Scores obtenus sur les 13 albums des Beatles avec l'approche probabiliste. Afin de ne pas alourdir ce tableau, nous avons seulement représenté pour chaque méthode de post-traitement les résultats correspondant à un choix optimal de L.

Le modèle de bruit d'observation Gamma donne de meilleurs résultats que le modèle Gaussien ou Poisson. Néanmoins, tout comme il n'y avait pas de différences flagrantes entre les mesures *EUC*, *IS1* et *KL1*, il n'y a pas d'écart majeur entre les trois modèles de bruit d'observation.

Pour chaque modèle de bruit d'observation, nous définissons un système avec les paramètres optimaux suivants :

- **Bruit additif Gaussien** : $\sigma^2 = 0.04$ et filtrage médian sur 17 trames (2.23s) ;
- **Bruit multiplicatif Gamma** : $\beta = 3$ et filtrage passe-bas sur 15 trames (2.04s) ;
- **Bruit Poisson** : filtrage médian sur 13 trames (1.86s).

Ces trois méthodes seront respectivement appelées PCR/Gaussian, PCR/Gamma et PCR/Poisson.

5.3 Résultats

Nous proposons dans cette section de comparer nos méthode aux systèmes de l'état de l'art grâce aux différentes métriques définies dans le chapitre 3. Toutes les méthodes ont été testées avec leur implémentation originale et ont toutes participé à MIREX 2008 ou 2009.

MIREX 2008 :

- BP : Bello & Pickens (2005)
- RK : Ryynänen & Klapuri (2008a)
- PVM : Pauwels et al. (2008)

MIREX 2009 :

- KO1 & KO2 : Khadkevich & Omologo (2009a)
- DE : Ellis (2009)
- OGF1 *(maj-min)* & OGF2 *(maj-min-7)* : nos méthodes de référence

5.3.1 Corpus des Beatles

Les résultats obtenus par ces 11 systèmes sur le corpus des Beatles sont présentés Table 5.5.

Les scores visant à évaluer les performances, tels que l'AOS ou l'AROS montrent que nos nouvelles méthodes dépassent légèrement l'état de l'art : l'AOS obtenu avec PCR/Gamma est en effet 2% plus élevé que le meilleur score (DE).

L'introduction d'autres métriques d'évaluation permet de comparer les méthodes de reconnaissance d'accords selon plusieurs points de vue. En effet, les quatre autres métriques évaluent la segmentation, la fragmentation et la bonne détection du vocabulaire d'accords.

La segmentation, c'est à dire la détection des frontières temporelles des accords, est évaluée grâce à l'AHD (qui doit être aussi faible que possible). Nous remarquons que, excepté pour la méthode PVM, tous les AHD sont très proches (autour de 0.15). Soulignons que le meilleur score est obtenu ici avec PCR/Gaussian.

La fragmentation est évaluée grâce à l'ACL (qui doit être le plus proche possible de 1). Certaines méthodes semblent sous estimer la durée des accords (RK, KO1, KO2 & PCR/Poisson), mais la plupart d'entre elles ont tendance à trop fragmenter les transcriptions. Certaines méthodes (PVM, OGF1 *(maj-min)*) détectent même des durées d'accords en moyenne deux fois plus petites qu'elles ne le sont réellement. Notons d'ailleurs que l'amélioration entre nos méthodes déterministes et probabilistes est éloquente en ce qui concerne la fragmentation : les meilleurs résultats sont obtenus avec PCR/Poisson.

Une des contributions principales apportées par l'introduction du cadre probabiliste est l'évaluation explicite du vocabulaire d'accords. Voyons comment les métriques associées (l'ACN qui

	MIREX 2008			MIREX 2009					méthodes proposées		
	BP	RK	PVM	KO1	KO2	DE	OGF1	OGF2	Gaussian	Gamma	Poisson
Average Overlap Score (AOS)	0.707	0.705	0.648	0.722	0.734	0.738	0.714	0.724	0.749	**0.758**	0.744
Average Root Overlap Score (AROS)	0.740	0.763	0.680	0.754	0.761	0.772	0.775	0.783	0.785	**0.787**	0.775
Average Hamming Distance (AHD)	0.153	0.146	0.209	0.152	0.150	0.156	0.163	0.152	**0.146**	0.149	0.156
Average Chord Length (ACL)	0.941	1.074	0.422	1.169	1.168	0.890	0.552	0.717	0.872	0.920	**1.057**
Average Chord Number (ACN)	1.441	1.414	2.285	1.507	1.319	1.667	2.070	1.693	1.314	1.185	**1.012**
Average False Chord Label Number (AFCLN)	3.560	3.330	8.490	3.760	2.590	4.590	7.390	4.990	2.640	1.860	**1.060**
Temps de calcul	1619	2241	12402	6382[1]	6382[1]	1403	790	796	480	482	486

[1] Les méthodes KO1 et KO2 utilisent des ressources communes : nous présentons ici le temps nécessaire à lancer les deux algorithmes.

TABLE 5.5 – Comparaison avec l'état de l'art sur le corpus des Beatles

	MIREX 2008			MIREX 2009					méthodes proposées		
	BP	RK	PVM	KO1	KO2	DE	OGF1	OGF2	Gaussian	Gamma	Poisson
Average Overlap Score (AOS)	0.699	0.730	0.664	0.670	0.665	0.719	0.707	0.682	0.739	**0.773**	0.760
Average Root Overlap Score (AROS)	0.743	0.768	0.693	0.705	0.695	0.759	0.783	0.790	0.788	**0.803**	0.779
Average Hamming Distance (AHD)	0.142	**0.117**	0.175	0.153	0.156	0.127	0.142	0.137	0.131	0.124	0.130
Average Chord Length (ACL)	0.903	**1.021**	0.494	1.084	1.109	0.823	0.565	0.683	0.835	0.896	0.806
Average Chord Number (ACN)	1.559	1.516	2.323	1.549	1.351	1.906	2.297	1.970	1.529	1.336	**1.138**
Average False Chord Label Number (AFCLN)	3.650	3.250	7.850	3.600	2.550	5.300	7.700	5.850	3.150	2.150	**1.150**

TABLE 5.6 – Comparaison avec l'état de l'art sur le corpus Quaero

doit être proche de 1 et l'AFCLN qui doit être le plus bas possible) se comportent. Un premier constat est que toutes les méthodes semblent sur-évaluer le nombre d'accords constituant le vocabulaire d'accords. Les méthodes PVM et OGF1 *(maj-min)* sont particulièrement touchées par ce phénomène. Au contraire, nos trois méthodes probabilistes estiment avec précision le vocabulaire d'accords : elles obtiennent en effet les trois meilleurs scores.

Puisque nos méthodes n'utilisent toujours pas d'apprentissage, le temps de calcul est toujours aussi bas. Une légère optimisation du code leur permet même d'être encore plus rapide que les méthodes déterministes, et ainsi deux fois plus rapide que les autres méthodes de l'état de l'art.

5.3.2 Corpus Quaero

Nous avons déjà discuté dans le chapitre précédent de l'utilité de tester les méthodes de reconnaissance d'accords sur d'autres corpus que celui des Beatles, qui est déjà largement utilisé dans la communauté. Nous avons donc lancé tous les systèmes précédemment testés sur le corpus Quaero : les résultats obtenus sont présentés Table 5.6.

Un premier constat est que, excepté pour RK, PVM, PCR/Gamma et PCR/Poisson, toutes les méthodes obtiennent des AOSs plus faibles sur ce corpus que sur celui des Beatles. Néanmoins, nos méthodes probabilistes donnent les meilleurs résultats : en particulier, la méthode PCR/Gamma réussit même mieux que sur le corpus des Beatles.

En ce qui concerne la segmentation, la méthode RK donne les meilleurs résultats mais notre méthode PCR/Gamma donne le deuxième AHD. Remarquons que contrairement à ce qu'il se passait sur le corpus des Beatles, les AHDs sont maintenant plus espacés : cela est probablement du au fait que tous ces systèmes ont été conçus sur le corpus des Beatles et ont parfois des difficultés à adapter leurs modèles à d'autres types de données.

Une fois de plus, la plupart des méthodes ont tendance à sous-estimer la durée des accords : PCR/Gamma donne le quatrième meilleur ACL. Nous observons aussi que les ACLs obtenus par nos méthodes OGF1 *(maj-min)* et OGF2 *(maj-min-7)* sont décevants : cela confirme nos inquiétudes concernant la fragmentation excessive dans les méthodes utilisant des gabarits d'accords. Bien que nos méthodes probabilistes ne donnent pas les meilleurs scores, nous remarquons tout de même une nette amélioration par rapport à nos méthodes déterministes en ce qui concerne la fragmentation.

Pour finir, nous observons grâce aux ACNs et AFCLNs que nos méthodes probabilistes obtiennent à nouveau de bons résultats pour l'estimation du vocabulaire d'accords : les méthodes PCR/Gamma et PCR/Poisson obtiennent en effet les deux meilleurs scores.

Ces résultats tendent à calmer l'inquiétude concernant la possible dépendance avec le corpus des Beatles. En effet, nos méthodes probabilistes obtiennent de meilleurs résultats sur le corpus Quaero que sur celui des Beatles : ceci montre encore une fois que le choix des paramètres fait sur le corpus des Beatles est aussi pertinent sur d'autres styles ou genres musicaux.

5.4 Conclusion

Résumons les principales contributions et conclusions de ce chapitre :

– Nos nouvelles méthodes probabilistes ne se basent que sur la définition des accords : elles appartiennent donc à la catégorie des méthodes de reconnaissance d'accords par l'utilisation de gabarits. Ainsi, elles possèdent toutes les qualités inhérentes à ces systèmes, comme le fait qu'aucun apprentissage ou connaissance musicale approfondie n'est nécessaire à la transcription. Cela leur permet aussi de ne pas dépendre fortement du corpus

sur lequel elles ont été développés, ni du genre musical. Enfin, le temps de calcul reste très faible, ce qui est une caractéristique intéressante, en particulier dans des systèmes embarqués.

– L'introduction des probabilités d'accords dans le modèle permet une estimation pertinente du vocabulaire d'accords, et la production de transcriptions en accords plus claires et compactes. Les performances s'en trouvent ainsi améliorées, ainsi que la qualité des transcriptions obtenues.
– La bonne estimation du vocabulaire d'accords, combinée à l'utilisation d'un post-traitement par filtrage sur le critère de reconnaissance permet une bonne détection des frontières temporelles des accords, ainsi qu'une bonne segmentation.

Chapitre 6

Conclusion

6.1 Bref résumé des contributions

La principale contribution de ce travail de thèse est le développement de deux méthodes pour la reconnaissance d'accords à partir de signaux audio, utilisant seulement des gabarits d'accords théoriques. Ces deux approches, bien que plus simples que la plupart des méthodes de l'état de l'art, donnent des résultats très honorables.

Une autre contribution est l'évaluation de nos méthodes grâce à de nombreuses métriques, permettant ainsi une analyse complète des performances. Ajoutons à cela la comparaison avec de nombreuses méthodes de l'état de l'art, qui met en perspective les avantages et désavantages de nos approches.

Des résumés plus détaillés de nos deux approches sont présentés Sections 4.4 & 5.4.

6.2 Quelques pistes de travail

Nous proposons ici quelques pistes et perspectives de travail inspirées par les travaux décrits dans ce manuscrit.

Rythme et structure

Une des perspectives les plus évidentes pour l'amélioration de nos méthodes est l'introduction de connaissance musicale dans nos systèmes. En particulier, l'aspect temporel de la tâche est pour le moment pris en charge uniquement par le filtrage du critère de reconnaissance. Nous pouvons penser que l'introduction d'un modèle rythmique complet, par exemple intégré dans notre modèle probabiliste, pourrait mieux prendre en compte la structure d'une chanson, notamment en terme de mesures et pulsations.

Choix adaptatif des paramètres

Tous les paramètres de nos méthodes sont estimés par essai/erreur : nombre d'harmoniques, mesures d'ajustement/modèles de bruit, hyper-paramètres, méthodes de filtrage, etc... Nous pouvons penser qu'il serait intéressant d'intégrer le choix des paramètres dans nos algorithmes. En particulier, les hyper-paramètres de l'approche probabiliste pourraient être estimés à l'intérieur de l'algorithme EM en utilisant des méthodes numériques.

Chromagramme

Dans ce manuscrit, nous avons utilisé le chromagramme proposé par Bello & Pickens (2005). Néanmoins, des travaux récents (Mauch & Dixon (2010)) ont significativement amélioré leurs performances en changeant le mode de calcul de leurs vecteurs de chroma. Nous pouvons penser que ces nouvelles représentations pourraient aussi augmenter nos scores, sous réserve que leur structure s'adapte bien à notre approche par gabarits.

Manuscrit en anglais

Contents

List of Acronyms

ACL Average Chord Length. 58, 59, 109, 111, 112, 115

ACN Average Chord Number. 58, 59, 109, 111, 112, 115

AFCLN Average False Chord Label Number. 58, 59, 109, 111, 112

AHD Average Hamming Distance. 58, 59, 109, 111, 112, 115

AOS Average Overlap Score. 56, 59, 72–74, 78, 81–83, 85, 86, 102, 108–110, 112, 115

AROS Average Root Overlap Score. 56, 59, 109, 110

CQT Constant-Q Transform. 28, 31–34, 71

CRF Conditional Random Field. 36, 43

CTT Chord Type Templates. 37

DBN Dynamic Bayesian Network. 35, 41

DCR Deterministic Chord Recognition. 71–73, 75, 80, 82, 83, 85, 86, 88, 91, 96, 100–102, 104, 106–108, 112, 115, 116, 119, 121

DHD Directional Hamming Divergence. 56, 58

EM Expectation-Maximization algorithm. 38, 40, 85, 86, 99–102, 120

FCLN(s) False Chord Label Number. 59

FLM Factored Language Model. 45

HD(s) Hamming Distance. 56, 58, 59, 115, 116

HMM Hidden Markov Model. 35, 36, 38–45, 68, 85, 86

MIDI Musical Instrument Digital Interface. 19, 35, 36, 40, 44, 51, 54

MIR Music Information Retrieval. 18, 24, 48, 89, 121

MIREX Music Information Retrieval Evaluation eXchange. 24, 26, 48, 56, 59, 85, 88, 89, 108, 112, 120

OS(s) Overlap Score. 56, 59, 115, 116

PCA Principal Component Analysis. 75

PCP Pitch Class Profile. 28, 31, 32

PCR Probabilistic Chord Recognition. 101, 102, 104, 106–108, 110–112, 115, 117

RCL(s) Reduced Chord Length. 58, 59, 115, 116

RCN(s) Reduced Chord Number. 59, 115, 116

ROS(s) Root Overlap Score. 59

STFT Short-Time Fourier Transform. 28, 31–34

SVM Support Vector Machine. 35, 39

WAOS Weighted Average Overlap Score. 56, 59, 89

WAROS Weighted Average Root Overlap Score. 56, 59, 89

List of Figures

List of Tables

Notations

M	number of bins in the chroma vector (in practice $M = 12$)
N	number of frames
K	number of chord templates
$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$	$M \times N$ chromagram
$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$	$M \times K$ chord dictionary
$h_{k,n}$	scale parameter
$d_{k,n}$	recognition criterion
$a_{k,n}$	amplitude parameter
α_k	chord probability
$\gamma_{k,n}$	state variable
$\bar{\alpha}_{k,n}$	posterior probability of state variable $\gamma_{k,n}$

Abstract

This thesis is in line with the music signal processing field and focuses in particular on the automatic chord transcription from audio signals. Indeed, for the past ten years, numerous works have aimed at representing music signals in a compact and relevant way, for example for indexation or music similarity search. Chord transcription constitutes a simple and robust way of extracting harmonic and rhythmic information from songs and can notably be used by musicians to playback musical pieces.

We propose here two approaches for automatic chord recognition from audio signals, which are based only on theoretical chord templates, that is to say on the chord definitions. In particular, our systems neither need extensive music knowledge nor training.

Our first approach is deterministic and relies on the joint use of chord templates, measures of fit and post-processing filtering. We first extract from the signal a succession of chroma vectors, which are then compared to chord templates thanks to several measures of fit. The so defined recognition criterion is then filtered, so as to take into account the temporal aspect of the task. The detected chord for each frame is finally the one minimizing the recognition criterion. This method notably entered an international evaluation (MIREX 2009) and obtained very fair results.

Our second approach is probabilistic and builds on some components introduced in our deterministic method. By drawing a parallel between measures of fit and probability models, we can define a novel probabilistic framework for chord recognition. The probability of each chord in a song is learned from the song through an Expectation-Maximization (EM) algorithm. As a result, a relevant and sparse chord vocabulary is extracted for every song, which in turn leads to better chord transcriptions. This method is compared to numerous state-of-the-art systems, with several corpora and metrics, which allow a complete and multi-facet evaluation.

Chapter 1

Introduction

Contents

This first chapter aims at briefly introducing a number of musical notions that may be useful for the good understanding of this document. It also describes the context and motivations for this work, as well as the main applications of chord recognition. Finally, it presents and summarizes the content of the following chapters.

1.1 What is music ?

The definition of music varies much with the context. In physics, music is a phenomenon characterized by frequencies and times while in social sciences, it is the reflection of a given culture at a given time. For artists music is the result of a personal and creative process ; in music theory, however, it is a complex entity defined by several rules of harmony.

This work is in line with the field of music signal processing. Therefore, all musical notions will be defined as time-frequency events. We could go as far as saying that in such a context, a song is just a series of 0's and 1's we want to convert into a sequence of symbols. At the same time, all the research that has been conducted in Music Information Retrieval (MIR) tends to prove that a good understanding of music is often necessary in order to develop efficient algorithms. Thus it seems relevant to introduce some musical notions before describing our contributions.

The purpose of this section is to define these notions from the angle of signal processing as well as of music theory. We shall focus on entities such as notes, chords and keys, but also on how they combine to form music.

1.1.1 Notes

A piece of music is composed of several sequences of notes played by different instruments. A note played by an instrument results in a complex time-frequency event depending on numerous parameters such as pitch, instrument, vibrato, intensity, etc. Yet, in theory, it is often assumed that a note k can be characterized by one *fundamental frequency* f_k. These fundamental frequencies are not random: they are linked so that the notes sound good together. These notes can then group together in order to form *scales*. We can cite for example the *diatonic* scale, which is a 7-note musical scale comprising five whole steps and two half steps for each octave.

In the standard Western equal temperament, the steps of the scale form a geometric sequence, that is to say that the ratio between the fundamental frequencies of two consecutive notes is constant. In the particular case of the 12-tone equal temperament, in which the octave is divided into 12 equal parts, the ratio between two semitones is:

$$r = \sqrt[12]{2}. \tag{1.1}$$

The fundamental frequencies are therefore spaced linearly in log-frequency and thus the frequency of a note k can be defined as:

$$f_k = r^k f_0. \tag{1.2}$$

Equal temperament naturally defines the *chromatic* scale, which is a musical scale with twelve equally spaced pitches, each a semitone apart.

Different instruments must refer to the same frequency scale so as to play together. A reference frequency has been defined in order to tune instruments. This reference frequency

has changed with time, but now the most commonly used is $F_{ref} = 440Hz$, which corresponds to the concert A (or $A4$).

In Musical Instrument Digital Interface (MIDI) notation every note is represented with an integer number. The notes considered in this notation range from $C-1$ (note 0, 8.175 Hz) to $G9$ (note 127, 12557 Hz). The concert pitch $A4$ is therefore given the number 69. The frequency of a note k is thus easily determined by:

$$f_k = r^{k-69} F_{ref}. \tag{1.3}$$

As such, the perception of pitch is twofold (Krumhansl (1990)): the *height* corresponds to the octave the note belongs to and the *chroma* or *pitch class* indicates the relation of the note to other notes within an octave. The pitch class C is therefore the set $\{C_i\}_{i\in\mathbb{Z}}$ composed by all possible C's, in whatever octave position. In the 12-tone equal temperament, there are therefore 12 chromas (which stand for the 12 semitones of the chromatic scale). Each of these chromas is given an integer number which ranges from 0 to 11 (0 for C, 1 for $C\sharp/D\flat$, etc.).

The chroma m of a note k, can be calculated as:

$$m = 69 + 12\log_2\left(\frac{f_k}{F_{ref}}\right) \bmod 12. \tag{1.4}$$

For example, the note $C4$ can be analyzed as an octave number 4 and a chroma C or 0.

1.1.2 Chords

A chord is an aggregate of musical pitches sounded simultaneously. The name of a given chord depends on the number of notes within it. Chords can contain 2 (dyads), 3 (triads), 4 (tetrads), 5 (pentads), 6 (hexads) or more notes.

A chord can theoretically be composed of random notes ; in music theory, however, a chord forms an entity and is not just a set of notes. Indeed, a chord can be defined by three notions (Harte et al. (2005); Benward & Saker (2003)):

- *root*: the note upon which the chord is built;
- *type*: the harmonic structure of the chord (it also gives the number of notes in the chord);
- *inversion*: the relationship of the bass to the other notes in the chord.

Specifying these three characteristics is sufficient to define which notes are to be played in the chord, regardless of the octave. For instance, a $Cmajor$ chord is defined by a root note C and a type $major$ which indicates that the chord will also contain the major third and the perfect fifth, namely notes E and G (see Figure 1.1 for an example of chord construction).

There are many ways to write chords depending on time, music style or field of application. Some of them include wider notions such as key or harmony (see Section 1.1.3), while others only rely on chord definitions. Figure 1.2 presents some examples of notation (Harte et al. (2005)). In the following document, we will only use the typical popular music guitar style, which is widely used in standard songbooks.

There is a very large number of possible chord types. Fujishima (1999), in the first audio chord recognition system, defines 27 chord types. So far, it is the publication which has considered the greatest number of chord types. The definitions of the main chord types are provided in the Appendix.

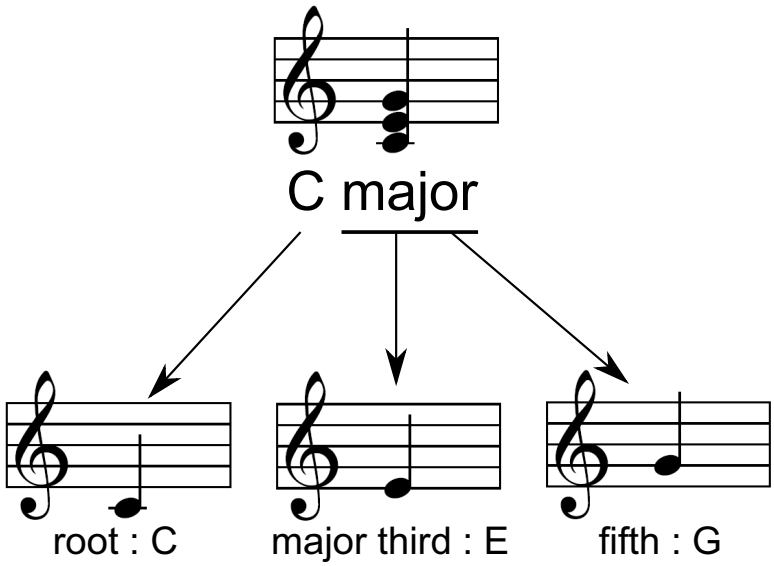


Figure 1.1: Construction of a C major chord

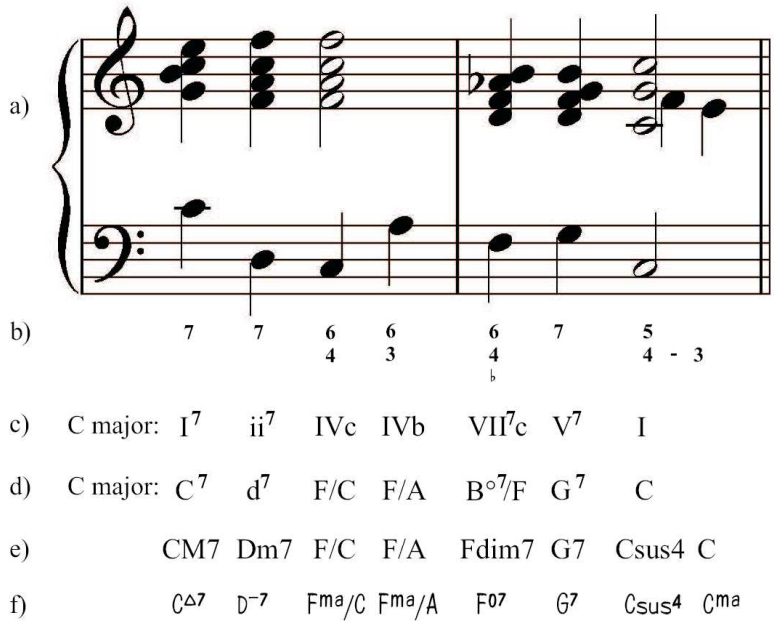


Figure 1.2: A short extract of music in C major with different harmony notations: a) Musical score b) Figured bass, c) Classical Roman numeral, d) Classical letter, e) Typical Popular music guitar style, f) Typical jazz notation (reproduced from Harte et al. (2005))

If one sticks to the definition of a chord as a set of notes played simultaneously, then a number of pieces of music can not be fully transcribed into chord sequences. Indeed, our chord recognition algorithms are expected to associate each musical event with a chord label. In fact, they function in the same way than the human ear in that they memorize and associate notes which are played in close succession. In our automatic transcribers, the concept of simultaneity is therefore given a loose definition.

1.1.3 Chords, key and harmony

The conception of chords as distinct entities (and not only as results of the superposition of melody lines) appeared in the 16th century, but the theorization of chords only began in the 17th century, with the emergence of *tonal harmony* (Baron (1973)). Harmony can be defined as the part of musical theory studying chords and how they can be used together for composition. The principle of tonal harmony originated in the observation that there is a very large number of note combinations which therefore can not all be studied. Which is why only a small number of chords are to be analyzed, because of their good auditive qualities. Music theorists of the 17th century such as Rameau (1722) attempted to build a coherent and universal harmony theory, relying both on mathematical principles (relations between the notes within the chord) and naturalism (study of the composition methods used until then). In particular, they underlined interesting note intervals for the construction of chords - the third or the fifth for example.

These principles were notably implemented by Bach (1722) ; only then did we introduce the notion of *key* as a new entity in musical construction. *"A key denote a system of relationships between a series of pitches (forming melodies and harmonies) having a* key tonic *(which is a note of the chromatic scale) as its most important (or stable) element"* (Gómez (2006a)). There are two basic *key modes*: major and minor ; each of them defines a musical scale. Figure 1.3 presents the notes making up the $Cmajor$ key and the $Aminor$ key. Within a key, chords are composed primarily of the notes within the musical scale defined by the key. The triad chords are built by choosing one note over two within this scale. Therefore, a key naturally defines 7 chords, whose root and type are entirely determined by the key tonic and mode. In this particular case these chords can be written with Roman numerals, which indicate the role of the chords within a given key. Table 1.1 shows the 7 chords defined by $Cmajor$ key and $Aminor$ key.

Nowadays, the notion of tonal harmony is still relevant in Western music genres such as rock, pop, jazz, electro, rap or hip hop. Nevertheless, the notion of key is now more blurred than it used to be and its definition is wider because of the use of more complex harmonic progressions or melodies. For instance, many pop songs now contain off-key chords. These new chord relationships also have their notations: Table 1.2 presents some of them for $Cmajor$ and $Cminor$ chords.

1.1.4 Circle of fifths and doubly-nested circle of fifths

The 12 notes of the chromatic scale define 12 major keys and 12 minor keys. The harmonic relationships between these keys can be displayed on a *circle of fifths* (see Figure 1.4). This circle represents all 24 keys: two adjacent keys on this circle have their tonic note spaced by a perfect fifth interval. Because of the particular consonance of the fifth interval, this circle expresses the harmonic proximity between keys.

Many researchers (Krumhansl & Kessler (1982), Chuan & Chew (2005), Purwins (2005), Harte et al. (2006)) have attempted to model key, chords or chroma relationships within a

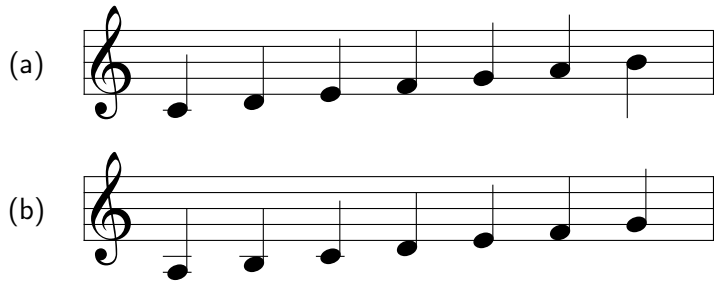
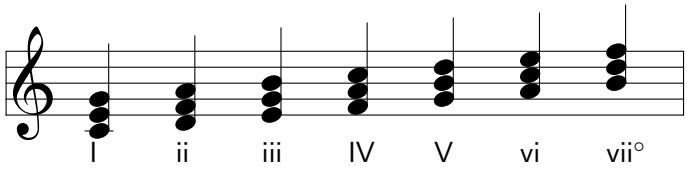


Figure 1.3: (a) The C major diatonic scale - (b) The A minor diatonic scale



Function	Classic Roman	Typical Popular
tonic	I	C
supertonic	ii	Dm
mediant	iii	Em
subdominant	IV	F
dominant	V	G
submediant	vi	Am
leading tone/subtonic	vii°	Bdim

Table 1.1: The 7 chords of the C major key

Reference chord	C	Cm
parallel	Cm	C
relative (submediant)	Am	A♭
mediant	Em	E♭
subdominant	F	Fm
dominant	G	Gm

Table 1.2: Particular relationships between chords: examples for C major and C minor chords

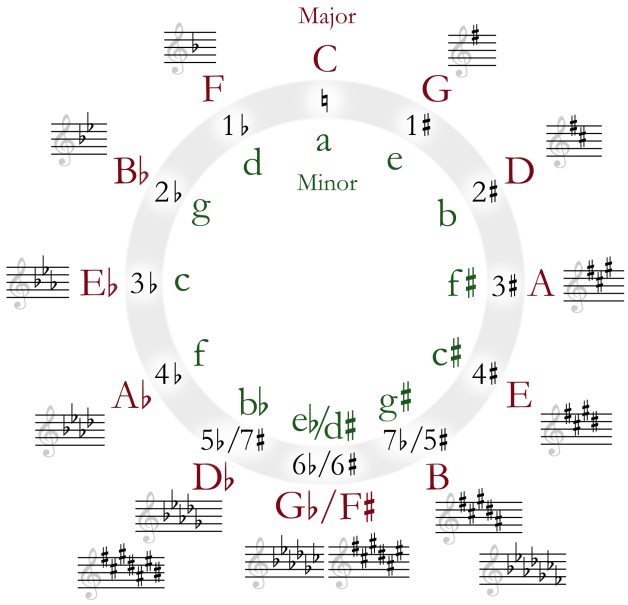


Figure 1.4: The circle of fifths representing the relationships between the 12 chromas in the pitch class space: major keys (in red), minor keys (in green), numbers of ♯ and ♭ and key signatures. (reproduced from `http://en.wikipedia.org/wiki/Circle_of_fifths`)

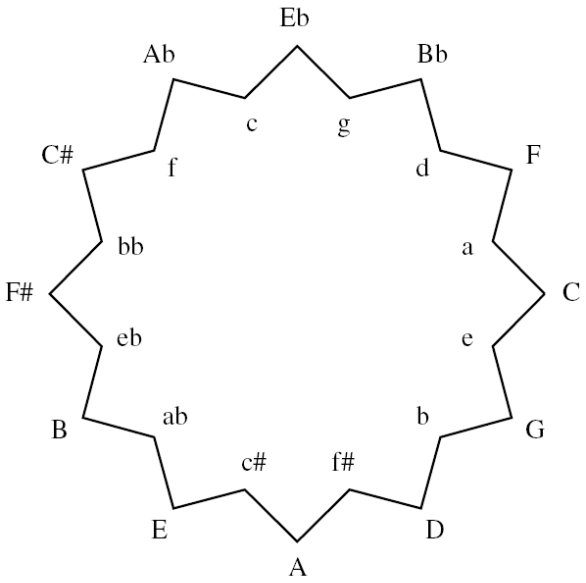


Figure 1.5: The doubly-nested circle of fifths representing the relationships between the 24 major/minor chords: major chords are represented by capital letters, while minor chords are in lowercase characters. (reproduced from Bello & Pickens (2005))

compact visual representation. Harmony has been represented as a spiral, a circle, or a multi-dimensional space. In particular, Bello & Pickens (2005) have built on the circle of fifths in order to propose a visual representation of chord relationships (see Figure 1.5). This *doubly-nested circle of fifths* represents the harmonic proximity between the 24 major and minor triads. In theory, the closer two chords are on this circle, the more consonant they are.

1.2 What is chord transcription?

Now that we have a more precise definition of chords, and of how they can be linked to key and harmony, let us answer the following questions: how can we use chords to describe a song and how can we use chord transcriptions in MIR?

1.2.1 Definition

A chord transcription can take several forms. A *lead sheet* (see Figure 1.6) is a score which gives the chord progression, the melody and the lyrics. Melody is written in modern Western music notation, lyrics are written as text below the staff and chord symbols are specified above it. Lead sheets can be used by musicians to playback song, by jazz players to improvise on the chord progression, or as legal song descriptions in the music industry.

Alternately, a chord transcription can consists in lyrics and chord symbols only, with or without guitar tablatures (see Figure 1.7). Most songbooks and online tablatures only display lyrics and chord symbols.

We can also cite the format used in Music Information Retrieval Evaluation eXchange (MIREX), formalized by Harte et al. (2005). It is composed of onset and offset times followed by the chord symbol (see Figure 1.8).

1.2.2 Main fields of application

We have seen that a chord transcription is a relevant representation of a musical piece, pop songs in particular, as it captures both the harmonic and rhythmic content of a song. This representation can be used in numerous MIR applications as a summary of the song.

The most intuitive and straightforward use of chord transcription is maybe the song playback. In particular, numerous songbooks offer chord transcriptions which are used by amateur or professional musicians just like scores. For example, musicians can sing the lyrics while accompanying themselves. Jazz musicians also use chord transcriptions as a map for improvisation.

Chord transcriptions can also be used in order to access higher-level information such as key (Shenoy et al. (2004)), rhythm (Goto & Muraoka (1999)), or structure (Maddage et al. (2004), Bello & Pickens (2005)). In the case of key, the estimation can be performed either jointly or separately. As far as rhythm is concerned, music theory rules or pop composition habits can help distinguishing downbeats by using the fact that some chord changes are more likely to occur on particular beat times. Finally, by grouping similar chord sequences, one can also analyze a song from a structure point of view and thus detect higher-level notions such as verse or chorus.

Chord information is also useful as a mid-level descriptor of a song, which allows to compare songs in order to find similar songs (browsing or recommendation tasks) or to retrieve cover songs (Lee (2006b), Bello (2007), Cheng et al. (2008)). It can also be used for music classification in genre, mood or emotion (Cheng et al. (2008)).

1.3 Outline of the thesis

The present document is organized as follows:

Chapter 1: Introduction

In this chapter we have presented the context and motivation of this dissertation by introducing a number of music notions and describing the main applications and issues of chord recognition task.

Chapter 2: State-of-the-art

This chapter aims at giving an overview of the state-of-the-art chord recognition methods. We will also focus on the input features of these systems and on how they can be defined and calculated. Chord recognition systems will be classified into several categories, and we will describe the main characteristics and principles of each of them.

Chapter 3: Corpus and evaluation

Before presenting our chord recognition methods, we will define the chord recognition task, and describe the corpora and metrics used to evaluate and compare systems. We will analyze our three corpora in details in order to understand their specificities. Particular emphasis will be laid on the metrics used for evaluation, which allow to assess the different facets of the chord recognition task. Finally, some statistic tests will be explained, which are very useful for the comparison between chord recognition methods.

Chapter 4: Deterministic template-based chord recognition

This chapter introduces our deterministic template-based chord recognition system by describing all its components. We will propose a complete and detailed analysis of the results obtained by our systems. We will also focus on the influence of parameters and on the understanding of performances. Our systems will then be tested on several corpora along with many state-of-the-art methods. A quantitative and qualitative analysis of the errors will finally be proposed.

Chapter 5: Probabilistic template-based chord recognition

This chapter extends the previous chapter by building a probabilistic framework from the deterministic method. We will explain how this novel approach can relate to the baseline system and detail the derivation of the final algorithm. An example will be thoroughly treated, so as to emphasize the main differences between the new methods and the previous ones. The new transcribers will finally be compared to many state-of-the-art methods on several corpora with a large number of metrics, allowing a complete analysis of the results.

Chapter 6: Conclusion

Finally, this chapter provides a summary of the conclusions and contributions of this work, and proposes some directions for future work.

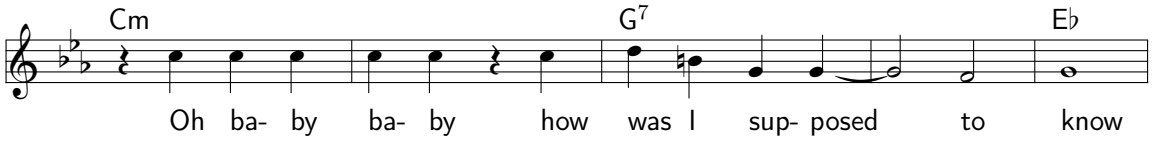


Figure 1.6: Example of lead sheet (melody, lyrics and chord symbols): *Baby one more time* by Britney Spears (1998). Jive Records.

```
Cm                G7                    Eb
Oh baby, baby How was I supposed to know
          Fm                  G
    That something wasn't right here
```

Figure 1.7: Example of simple chord transcription (lyrics and chord symbols): *Baby one more time* by Britney Spears (1998). Jive Records.

```
0.0000   1.2562   C:min
1.2562   1.8975   G:7
1.8975   2.6843   Eb
2.6843   3.4694   F:min
3.4694   3.9694   G
```

Figure 1.8: Example of MIREX-like chord transcription (times and chord symbols): *Baby one more time* by Britney Spears (1998). Jive Records.

Chapter 2

State-of-the-art

Contents

The audio chord transcription task can be broken down into two successive steps. In order to achieve good performances, it is crucial that both steps play their part. The first stage is a signal processing step, which consists in extracting from the signal some frequency domain features. This process should select the relevant harmonic and rhythmic information contained in the signal and avoid the capture of noise. The second stage is the chord recognition step itself, which transforms the calculated features into an output chord sequence.

These two steps are summarized on Figure 2.1.

2.1 Input features

For the past ten years, numerous techniques have been developed in order to extract the harmonic content of music signals. Most of the recent chord recognition methods use features called *chroma vectors*. This section aims at defining them and at explaining how they can be calculated.

2.1.1 Definition of chromagram

As seen in Chapter 1, the perception of pitch is based on two notions: height and chroma. Fujishima (1999) introduced the notion of Pitch Class Profile (PCP) whose calculation is based on a chroma representation of harmony. These features have been given many names in the literature (see Table 2.1 & 2.2) but for sake of clarity, we will call them *chroma vectors* in the present document. Chroma vectors are M-dimensional vectors, with M being a multiple of 12. When M is equal to 12, every component represents the spectral energy or salience of a semi-tone within the chromatic scale regardless of octave. When M is equal to 24, 36 or larger, every semi-tone is represented with several bins. This higher resolution allows to select the bins which best reflect the tuning of the song.

The succession of chroma vectors over time is called *chromagram*. The calculation is either performed on fixed-length or variable-length frames (depending for example on the tempo). Table 2.1 and Table 2.2 give a detailed description of existing chroma-based features, along with the main steps for their calculation. Examples of chromagram calculations can be found on Figure 2.2.

2.1.2 Main methods of calculation

Numerous methods have been developed in order to calculate features which capture well the musical information contained in audio signals. Our aim here is not to display an exhaustive description of all existing chromagram representations, but rather to give an overview of the most common ways to calculate them. The computation of these input features often relies on a time-frequency transform which is either the Short-Time Fourier Transform (STFT) or the Constant-Q Transform (CQT). We present here some principles for the calculation of chroma vectors with both these transforms.

Short-Time Fourier Transform (STFT) *(eg. Fujishima (1999))*

Let us consider a discrete signal x, sampled with sampling frequency F_s. Let us suppose that this signal is divided into N frames $x^{(n)}$ (with possibly overlap), each composed of N_f samples. The frames are separated with hop size N_h.

Hence, we can write:

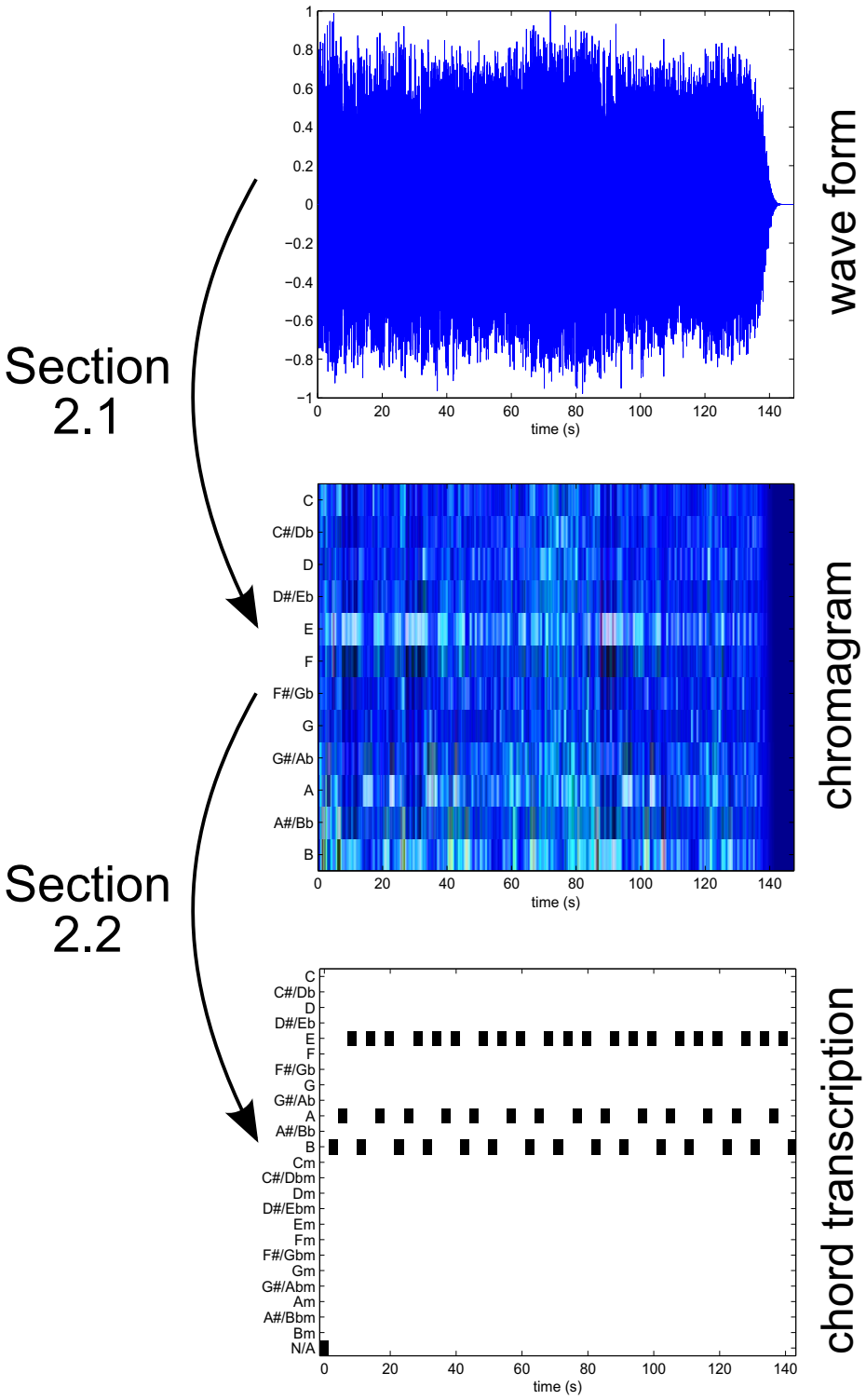


Figure 2.1: From audio signal to chord transcription.

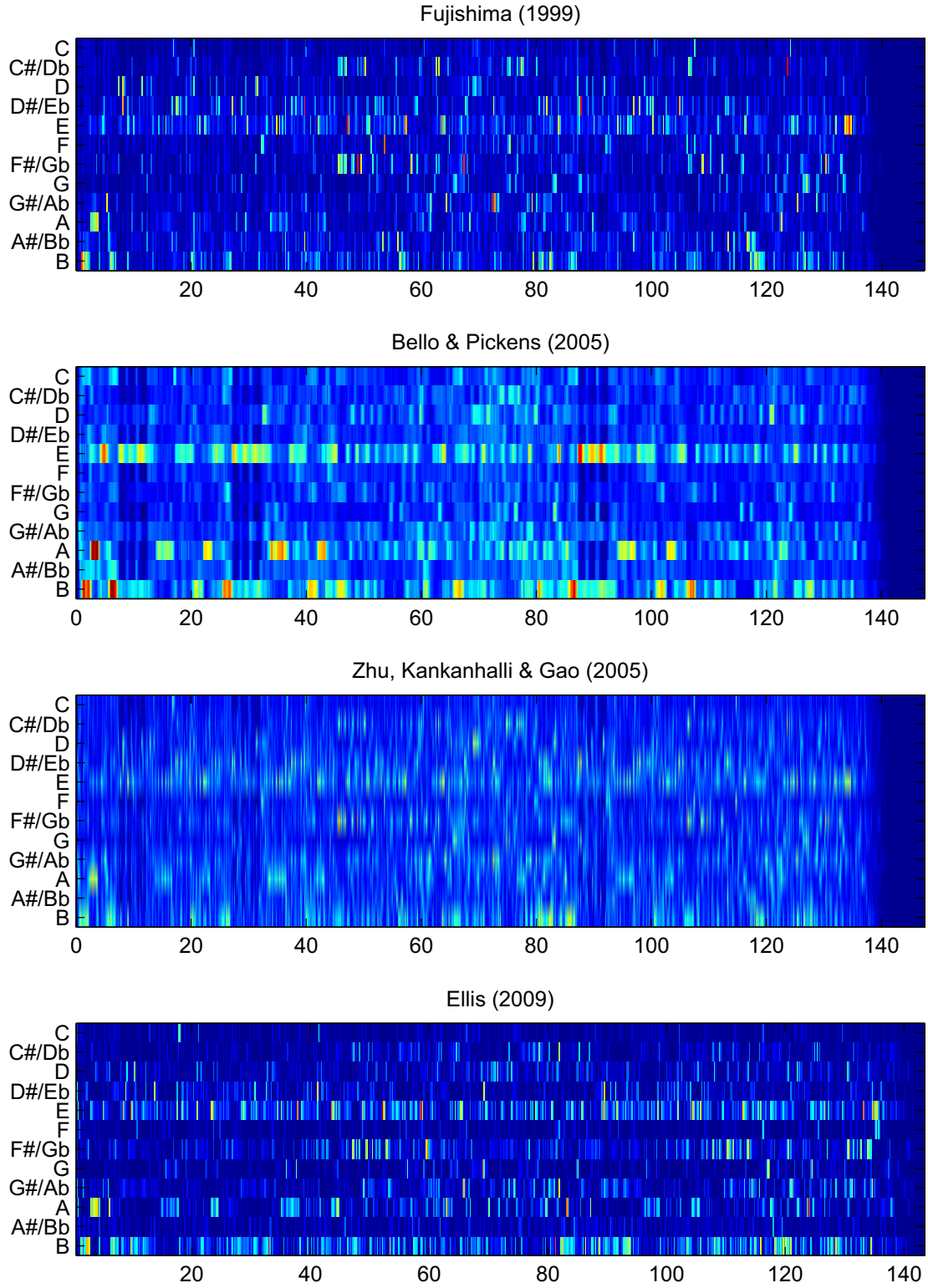


Figure 2.2: Examples of chromagram calculations on the Beatles song *Boys*. The x-axis is a time-scale (in seconds).

Table 2.1: Summary of the main chromagram calculation methods (part 1)

Authors	Name	Dimension	Frame length	Hop size	Transform	Frequency range	Processing	Application
Fujishima (1999)	Pitch Class Profile (PCP)	12	256 ms	256 ms	STFT	C2-??	smoothing, chord change sensing, non-linear scaling, elimination of less important regions, silence detection, attack detection...	chord
Purwins et al. (2000)	CQ Profile	36	N/A	N/A	CQT	N/A	none	key
Goto (2003)	chroma vector	12	256 ms	80 ms	STFT	130Hz-8000Hz	none	structure
Sheh & Ellis (2003)	Pitch Class Profile (PCP)	24	371.5 ms	100 ms	STFT	N/A	none	chord
Maddage et al. (2004)	Pitch Class Profile (PCP)	60	60 ms	30 ms	STFT	128Hz-8192Hz	none	chord, key, structure
Pauws (2004)	chromagram	12	100 ms	100 ms	STFT	A0-A6	spectral peak enhancement, amplitude weighting	key
Yoshioka et al. (2004)	chroma vectors	12 + bass	256 ms	80 ms	STFT	C3-B8	beat detection	chord, key
Bello & Pickens (2005)	chromagram	12	746 ms	92.9 ms	CQT	G2-5250Hz	tuning algorithm, low-pass filtering, beat detection	chord
Burgoyne & Saul (2005)	Pitch Class Profile (PCP)	12	250 ms	125 ms	STFT	C2-??	silence detection	chord, key
Cabral et al. (2005)	Pitch Class Profile (PCP)	12	N/A	N/A	STFT	N/A	weighting functions	chroma
Chuan & Chew (2005)	Pitch Class and Strength	12	N/A	371.5 ms	STFT	C2-B6	peak picking, weighting functions	key
Harte & Sandler (2005)	Harmonic Pitch Class Profile (HPCP)	12	746 ms	92.9 ms	CQT	G2-5250Hz	low-pass and median filtering, peak-picking, tuning algorithm	chord
Izmirli (2005)	chroma summary vectors	12	185.8 ms	371.5 ms	STFT	50-2000Hz	peak picking	key
Shenoy & Wang (2005)	chroma vectors	12	N/A	N/A	N/A	C2-B6	beat detection	chord, key, beat
Zhu et al. (2005)	Pitch Class Profile (PCP)	12	N/A	11.6 ms	CQT	A0-A7	tuning algorithm, note partial component extraction, consonance filtering	key
Gómez (2006a)	Harmonic Pitch Class Profile (HPCP)	12	92.9 ms	11.6 ms	STFT	100Hz-5000Hz	transient detection, peak-picking, tuning algorithm, harmonic weighting, spectral whitening, normalization	key
Lee (2006a)	Enhanced Pitch Class Profile (EPCP)	12	746 ms	92.9 ms	STFT	96Hz-5250Hz	Harmonic Product Spectrum (HPS), tuning algorithm, smoothing	chord
Peeters (2006)	semi-tone pitch class spectrum	12	371.5 ms	185.8 ms	STFT	100Hz-2000Hz	silence detector, sinusoidal analysis/re-synthesis, tuning algorithm, median filtering, sone-converting,...	key

Table 2.2: Summary of the main chromagram calculation methods (part 2)

Authors	Name	Dimension	Frame length	Hop size	Transform	Frequency range	Processing	Application
Burgoyne et al. (2007)	Pitch Class Profile (PCP)	12	185.8 ms	92.9 ms	STFT	C4-??	none	chord
Papadopoulos & Peeters (2007)	Harmonic Pitch Class Profile (HPCP)	12	480 ms	420 ms	STFT	60Hz-1000Hz	tuning algorithm, harmonic weighting, median filtering	chord
Zenz & Rauber (2007)	Pitch Class Profile (PCP)	12	46.4 ms	23.2 ms	Enhanced Autocorrelation (EAC)	N/A	none	chord, key, beat
Lee & Slaney (2008)	Pitch Class Profile (PCP) & Tonal Centroid	12 & 6	734 ms	185.8 ms	CQT	N/A	tuning algorithm	chord, key
Mauch & Dixon (2008)	Pitch Class Profile (PCP) & Bass Pitch Class Profile	12 & 12	734 ms	92.9 ms	CQT	A2-A6 & A1-A3	median filtering, tuning algorithm	chord
Papadopoulos & Peeters (2008)	Harmonic Pitch Class Profile (HPCP)	12	480 ms	420 ms	STFT	60Hz-1000Hz	tuning algorithm, harmonic weighting, median filtering, beat detection	chord
Pauwels et al. (2008)	chroma profile	12	150 ms	20 ms	STFT	100Hz-??	salient pitches detection, weighting functions, pitch candidates selection	chord
Ryynänen & Klapuri (2008a)	pitch saliences & accent signal	N/A	92.9 ms	23.2 ms	STFT	35Hz-1100Hz	weighting functions	chord
Sumi et al. (2008)	chroma vectors	12 + bass	256 ms	80 ms	STFT	C3-B8	beat detection	chord, key
Varewyck et al. (2008)	chroma profile	12	250 ms	N/A	STFT	100Hz-??	salient pitches detection, weighting functions, pitch candidates selection	chroma
Zhang & Gerhard (2008)	Pitch Class Profile (PCP)	12	500 ms	125 ms	STFT	N/A	none	guitar chord
Khadkevich & Omologo (2009b)	Pitch Class Profile (PCP)	12	185.8 ms	92.9 ms	STFT	100Hz-2000Hz	tuning algorithm, median filtering	chord
Mauch et al. (2009b)	treble chromagram & bass chromagram	12 & 12	371.5 ms	46.4 ms	STFT	G♯2-G♯6 & D♯1-G♯3	beat detection, median filtering, salient pitches detection, tuning algorithm, structure analysis	chord
Reed et al. (2009)	chroma vector	12	185.8 ms	92.9 ms	CQT	A0-C8	harmonic/percussive source separation, tuning algorithm, DFT chroma features	chord
Weil et al. (2009)	chroma vector	12	371.5 ms	46.4 ms	CQT	E2-D♯6	beat detection	chord, beat
Mauch & Dixon (2010)	treble chromagram & bass chromagram	12 & 13	185.8 ms	46.4 ms	STFT	D♯3-G♯5 & D♯1-G♯3	harmonic weighting, salient pitches detection, tuning algorithm, median filtering	chord

$$x^{(n)}[t] = x[(n-1)\,N_h + t]. \tag{2.1}$$

Let us consider a window w_{N_f} (Hanning, Hamming, etc.) also containing N_f samples. The Short-Time Fourier Transform (STFT) value for frame n and frequency bin k is:

$$X_{STFT}\,[k,n] = \sum_{t=0}^{N_f-1} w_{N_f}[t]\,x^{(n)}[t]\,e^{-\frac{j2\pi kt}{N_f}}. \tag{2.2}$$

Chroma vector $\mathbf{c}_n$ is calculated by summing the STFT bins corresponding to frequencies belonging to the same chroma:

$$c_{m,n} = \sum_{\substack{k\ such\ that \\ \mathrm{chroma}(k)=m}} |X_{STFT}\,[k,n]|^2, \tag{2.3}$$

where

$$\mathrm{chroma}(k) = \left\lfloor M \log_2\left(\frac{k}{N_f}\cdot\frac{F_s}{f_0}\right)\right\rfloor \bmod M. \tag{2.4}$$

f_0 is the frequency of the first bin in the chroma vector. Note that we use here the magnitude square of the STFT, but we could also use the magnitude, the log-magnitude, etc.

Constant-Q Transform (CQT) *(eg. Harte & Sandler (2005))*

In the STFT, frequency bins are linearly spaced, which might not be convenient for the analysis of music signals, where notes are logarithmically spaced. Furthermore, the frequency resolution should also be geometrically related to the frequency. In order to take these remarks into account, Brown (1991) proposed the Constant-Q Transform (CQT), which offers the property that the ratio Q of frequency to resolution is constant. The transform is completely determined by the lowest frequency about which information is desired (center frequency of the first bin) f_0, the number of bin per octave M and the number of octaves spanned by the spectrum B_{octave}. With an appropriate choice of parameters, the center frequencies of the bins can directly be linked to notes. In order to keep Q constant, the time resolution depends on the frequency. Hence, contrarily to the STFT, the window size N_f is now a function of frequency bin k: $N_f(k)$ decreases when frequency increases (see Table 2.3 for details). Fast and efficient methods exist for the calculation of the CQT (Brown & Puckette (1992), Blankertz (2001)).

The CQT value for frame n and frequency bin k is:

$$X_{CQT}\,[k,n] = \frac{1}{N_f(k)} \sum_{t=0}^{N_f(k)-1} w_{N_f(k)}[t]\,x^{(n)}[t]\,e^{-\frac{j2\pi kt}{N_f(k)}}. \tag{2.5}$$

In this case, chroma vector $\mathbf{c}_n$ is easily computed with:

$$c_{m,n} = \sum_{b=0}^{B_{octave}-1} |X_{CQT}\,[m+Mb,n]|\,. \tag{2.6}$$

A summary of the differences between the two transforms is displayed on Table 2.3.

	STFT	CQT
Center frequencies	$f_k = \frac{k}{N_f} F_s$ linear in k	$f_k = 2^{\frac{k}{M}} f_0$ exponential in k
Window size	N_f constant	$N_f(k) = Q\frac{F_s}{f_k}$ variable
Frequency resolution Δf	$\frac{F_s}{N_f}$ constant	$\frac{f_k}{Q}$ variable
Frequency to resolution ratio $\frac{f_k}{\Delta f}$	k variable	$Q = \frac{1}{2^{\frac{1}{M}}-1}$ constant

Table 2.3: Differences between STFT and CQT (Brown (1991))

Pre- and post-processing

Researchers have developed many efficient pre- and post-processing steps in the chromagram calculation so as to improve the front-end of their systems. Most of these processing stages tend to suppress noise or inharmonic content (such as silences, transients, drums, etc.), but also to adapt to issues inherent to real-life recordings (such as detuning, changing rhythms, etc.). The aim of this processing is often to produce clearer and sparser chromagrams, which better reflect the played notes. We shall not discuss these processing methods in details but Tables 2.1 & 2.2 give a summary of the main pre- and post-processing methods used in the chromagram calculation.

2.2 Chord recognition methods

Let us now suppose that the input features have been extracted from the audio signal. Our goal is to output a chord sequence from these features: this is the chord recognition phase.

Now the question is: how can we automatically recognize chords from audio features? The intuitive idea is to ask ourselves: how would a musician perform this task? A fine musician owns two qualities: knowledge about music theory and a lot of practice. When transcribing a musical piece into chords, a musician would ask himself: is this chord sequence possible given this particular harmony and given what I know about music theory? and is this chord sequence possible given what I have learned by transcribing other musical pieces?

These are actually the two main directions taken by researchers working on automatic chord recognition. We will therefore distinguish two types of methods: those driven by music theory knowledge and those based on training with audio data. Besides these two types of approaches we can also find template-based methods which use no a priori knowledge and other methods combining both a musical and training approach. A summary of these four categories are presented on Table 2.6.

2.2.1 Template-based methods

Template-based chord recognition methods are based on the hypothesis that only the chord definition is necessary to extract chord labels from a music piece. In fact, neither training data or extensive music theory knowledge (such as harmony rules, chord transitions, key, rhythm,

Table 2.4: Summary of the main chord recognition methods (part 1)

	Authors	Model	Chord models	Other outputs	Chord dictionary	Corpus
Template-based	Fujishima (1999)	Euclidean distance, dot product	fixed and binary	none	324	keyboard sounds, CD recordings
	Harte & Sandler (2005)	dot product	fixed and binary	none	48 (major, minor, diminished and augmented)	28 Beatles songs
	Lee (2006a)	correlation	fixed and binary	none	24 (major and minor)	Bach Preludes
Training-based	Sheh & Ellis (2003)	HMM	learned	none	147 (major, minor, major seventh, minor seventh, dominant seventh, augmented and diminished)	20 Beatles songs
	Ryynänen & Klapuri (2008b)	HMM	learned	melody, bass line	24 (major and minor)	110 Beatles songs
	Weller et al. (2009)	SVMstruct	learned	none	24 (major and minor)	180 Beatles songs
Music-based	Bello & Pickens (2005)	HMM	fixed and binary	none	24 (major and minor)	28 Beatles songs
	Shenoy & Wang (2005)	rules	fixed and binary	key, rhythm	24 (major and minor)	30 songs
	Sailer & Rosenbauer (2006)	rules	fixed and binary	none	72 (major, minor, minor seventh, major seventh, suspended 4th and diminished chords)	3 resynthesized MIDI and 6 real audio inputs
	Papadopoulos & Peeters (2008)	HMM	fixed and taking into account harmonics	rhythm	24 (major and minor)	66 Beatles songs
	Pauwels et al. (2008)	probabilistic model	fixed and binary	key	24 (major and minor)	180 Beatles songs
	Mauch et al. (2009b)	DBN	fixed and binary	key, rhythm, structure	48 (major, minor, diminished and dominant)	125 Beatles songs
	Mauch & Dixon (2010)	DBN	fixed and binary	key, rhythm	108 (major, minor, major seventh, dominant seventh, major sixth, diminished, augmented, first and second major inversion)	176 Beatles songs

Table 2.5: Summary of the main chord recognition methods (part 2)

	Authors	Model	Chord models	Other outputs	Chord dictionary	Corpus
Hybrid	Maddage et al. (2004)	HMM	learned	key, rhythm, structure	48 (major, minor, diminished and augmented)	40 songs
	Yoshioka et al. (2004)	hypothesis search	learned	key	48 (major, minor, diminished and augmented)	7 songs
	Burgoyne & Saul (2005)	HMM	learned	none	24 (major and minor)	1 Mozart symphony
	Burgoyne et al. (2007)	HMM and CRF	learned	none	48 (major, minor, diminished and augmented)	20 Beatles songs
	Papadopoulos & Peeters (2007)	HMM	learned or fixed and taking into account harmonics	none	24 (major and minor)	110 Beatles songs
	Zenz & Rauber (2007)	rules	learned	key, beat	36 (major, minor and diminished)	35 songs
	Lee & Slaney (2008)	HMM	learned	key	24 or 36 (major, minor and diminished)	2 classical pieces (Bach, Haydn) and 28 Beatles songs
	Mauch & Dixon (2008)	HMM	learned	none	60 (major, minor, dominant, diminished and suspended)	175 Beatles songs
	Sumi et al. (2008)	hypothesis search	learned	key	48 (major, minor, diminished and suspended 4th)	150 Beatles songs
	Khadkevich & Omologo (2009b)	HMM and language models	learned	none	24 (major and minor)	180 Beatles songs
	Weil et al. (2009)	HMM	learned	key, beat, main melody	24 (major and minor)	278 resynthesized MIDI

	Music theory	Training data
Template-based	no	no
Training-based	no	yes
Music-driven	yes	no
Hybrid	yes	yes

Table 2.6: Our four categories of chord recognition methods

etc.) is used. Most of them define some fixed chord templates according to a user-defined chord dictionary and then evaluate the fit between these chord templates and chroma vectors in order to finally retrieve the chord sequence. A chord template is a 12-dimensional vector representing the 12 semi-tones (or *chroma*) of the chromatic scale. Each component of the pattern is given a theoretical amplitude according to the chord definition.

The first template-based system for chord recognition from audio signals was developed by Fujishima (1999). This method is the first one considering chords not only as sets of individual notes, but rather as entities whose structure is determined by one root and one type. In his approach, 27 chord types are considered, which gives a total of 324 chords. Each of them is modeled by a binary Chord Type Templates (CTT), with amplitudes of 1 for the chromas within the chord definition and 0 for other chromas. The chord detection is performed by first calculating scores for every root and chord type, then selecting the best score. The scores are computed from chroma vectors and hand-tuned variations of the original CTT. Two matching methods between chroma vectors and CTTs are tested: the Nearest Neighbor Method (Euclidean distance between chroma vector and hand-tuned CTT) and the Weighted Sum Method (dot product between chroma vector and hand-tuned CTT). The hand-tuning is done by trial-and-error and accounts for the chord type probability and the number of notes within the chord type. Two post-processing methods are introduced in order to take into account the temporal structure of the chord sequence. The first attempt is to smooth over the past chroma vectors to both reduce the noise and use the fact that a chord usually lasts for several frames. The second heuristic is to detect chord changes by monitoring the direction of the chroma vectors.

The algorithm is first tested on audio keyboard sounds produced with a Yamaha PSR-520, then on real audio files. Results on the electronic keyboard corpus are satisfactory: Nearest Neighbor Method recognizes every chord type without hand-tuning CTTs, while Weighted Sum Method correctly detects almost all chord types by hand-tuning CTTs. Results obtained on the real audio corpus are also good but in every case CTTs need to be hand-tuned. The introduction of the two post-processing methods improves the results.

Harte & Sandler (2005) use a very similar method to Fujishima's. The chromagram extraction is improved by applying a frequency tuning algorithm. They define binary chord templates for 4 chord types (major, minor, diminished and augmented) and then calculate a dot product between chroma vectors and chord templates. The temporal information is captured by applying low-pass filtering on the chromagram and median filtering to the detected chord sequence.

The algorithm is tested on 28 annotated songs by the Beatles. Results vary much with the album and the production style. Most of the errors are due to confusions between harmonically close chords.

Lee (2006a) also uses binary chord templates, this time for the 24 major/minor triads. He

What are Hidden Markov Models? (Rabiner (1989))

A Hidden Markov Model is a discrete statistical model in which the modeled system is assumed to be a Markov process with unobservable states.

An HMM is characterized by the following elements:

- the number of states in the model ;
- the number of distinct observation symbols per state (i.e. the discrete alphabet size) ;
- the state transition probability distribution (probability of switching from a state to another) ;
- the observation symbol probability distribution (probability of emitting an observation symbol being in a state) ;
- the initial state distribution (initial probability of being in a state).

In Hidden Markov Models, the state duration has inherently an exponential duration distribution.

introduces a new input feature called Enhanced Pitch Class Profile (EPCP) using the harmonic product spectrum. The chord recognition is then carried out by maximizing the correlation between chroma vectors and chord templates. Just like the two previously described methods, a post-processing smoothing is applied to chroma vectors in order to take into account the temporal context. The algorithm is then tested on real recording examples.

2.2.2 Training-based methods

The methods described in this section share one common hypothesis: only training is needed in order to perform good chord recognition. This means that all necessary information for detecting chords is already contained in the data: chord definitions, possible chord transitions, etc. These methods tend to completely fit systems to data without introducing much a priori music knowledge.

The first chord recognition method based on training was the one designed by Sheh & Ellis (2003). Their approach relies on Hidden Markov Models (HMMs)[1] which became famous in the speech processing community. Drawing a parallel between words and chord sequences, they adapted the framework used in speech processing for chord transcription.

In their system, each chord is modeled by one HMM state. Since the chroma vectors used for this method are 24-dimensional, each state is described by one 24-dimensional Gaussian defined by one mean vector and one diagonal covariance matrix. These parameters, along with the initial state distribution and the transition probability matrix, are all learned from data with an Expectation-Maximization algorithm (EM). The learning process is performed with real audio data. Only chord sequences are annotated: in particular, the chord boundaries are unknown. Since the training data is not completely annotated, we can refer to this method as partially supervised learning. The EM training outputs a set of state parameters (mean vectors and covariance matrices for all the defined chord states).

It seems intuitive that, for example, the models obtained for $Cmajor$ and $Emajor$ chords should be very close, given a rotation of the mean vectors and covariance matrices. Thus, learning all chord models seems unnecessary: the authors propose to only build one major and one minor model. These models are calculated by averaging all models across the chord roots:

[1] An introduction to HMMs is presented in the box *What are Hidden Markov Models?* (p 38)

the contribution of each chord is weighted by its occurrence. For instance, all major chords are used to create one major chord model. This avoids overfitting (as the learning process is less influenced by the chord distribution in the training corpus) and increases the size of the training set of every individual chord.

The final frame-to-frame chord sequence is obtained by decoding the HMM with a Viterbi algorithm. Two tasks can be performed: forced alignment (which consists in recovering the chord boundaries giving the chord sequence) and recognition (where no knowledge is introduced except for the learned HMM parameters).

147 chords are defined, which correspond to 7 chord types (major, minor, major seventh, minor seventh, dominant seventh, augmented and diminished) and 21 roots (chromatic scale with differences between ♯ and ♭). Results show that the use of averaged models improves the performances. Unsurprisingly, the recognition scores are lower than the forced alignment scores.

In Ryynänen & Klapuri (2008b)'s system, two types of chord templates are learned from annotated audio data: one for high-register notes and one for low-register notes. They use a 24-states HMM: each chord is modeled by one state, described by a Gaussian mixture. Just like Sheh & Ellis (2003), data is mapped so that, for example, all major chords can be used for the calculation of the major chord profile. All other HMM parameters are learned from annotated data and the observation distributions take into account both the high and low chord profiles.

The method is tested on a corpus composed of 110 Beatles songs, giving results comparable to those of Bello & Pickens (2005).

Weller et al. (2009) propose a machine learning chord transcription method using a large margin structured prediction approach (SVMstruct) instead of the commonly used HMM. Results show that if training is performed with enough data, learned models can give good performances for the chord transcription task.

2.2.3 Music-driven methods

The methods described in this section only rely on music theory, in order to build models which can produce musically relevant chord sequences. They often use external notions such as key, harmony, bass or rhythm to perform either only chord recognition or joint recognition of all these notions. In particular, none of these methods is using explicit training.

The system proposed by Bello & Pickens (2005) was one of the first chord recognition systems which explicitly used music theory. Just like Sheh & Ellis (2003), this system is based on HMMs. Each of the 24 major and minor chords is modeled by one state in the HMM. Like in every HMM system, 3 probability distributions should be defined:

- The initial state distribution is supposed to be uniform (all the chords are initially equiprobable).
- The observation model assumes a 12-dimensional Gaussian distribution. The mean vector is initialized as a binary chord template and the covariance matrix is initialized with values inspired by music theory. Three covariance matrix initializations are tested: diagonal (like Sheh & Ellis (2003)), weighted diagonal (with empirical values reflecting musical knowledge), or non-diagonal (inspired by musicology and cognitive results). The mean vectors and covariance matrices are defined for major and minor chords and then shifted for all the 24 chords.

- The state transition matrix initialization reflects the fact that, according to music theory, some chord transitions are more likely to occur than others. The authors draw their inspiration from the doubly-nested circle of fifths (see Section 1.1.4). The closest two chords are on this circle, the higher the probability of switching from one state to another is.

All the HMM parameters can then selectively be tuned to data thanks to an EM algorithm. A Viterbi algorithm is used for the final detection.

This method is evaluated on 28 songs by the Beatles. Many variants are tested, such as beat-synchronous chromagrams, fixed or EM-updated probability distribution, random or music-driven initializations, etc. Results show that the introduction of musical knowledge in the initializations significantly improves the performances of the method. Interestingly, the results are better when the observation distribution parameters stay fixed, which tends to prove that chord definitions do not necessarily have to be learned.

Other musical information can be used: Shenoy & Wang (2005) present a framework for the analysis of audio signals in term of key, chords and rhythmic structure. This system can be seen as a rule-based system: music theory is converted into rules, which are successively applied so as to give more-and-more relevant results.

This hierarchic system builds on the key detection method presented by Shenoy et al. (2004). This key detection method begins with a chord recognition phase. For every frame, each chord is given a weight according to the fit between its corresponding binary chord template and the 4 larger values of the chroma vector. This first chord transcription is then used to estimate the key, by explicitly using music theory, in particular the definitions of keys and scales. Building on this key estimation system, the chord transcription method consists in three successive post-processing steps of the key and the chord sequence output by the key detection method. The first improvement step aims at rejecting chords which do not fit the key, and at testing the relevance of chords with the adjacent frames. A second step extracts the rhythmic structure. This information is then used in a third phase for correcting the chord sequence by checking that all chords in a measure are the same.

The system is tested on 30 pop songs and only major and minor chords are detected. Results show that every post-processing step allows to significantly improve the scores, that is to say that the key and rhythm information are useful in the chord detection process.

Sailer & Rosenbauer (2006) also use the notion of key in order to achieve a better chord recognition. The first step of their algorithm is a key detection phase performed by Krumhansl (1990)'s and Temperley (2001)'s method.

The input features used in this method are sparse: only a few note candidates are present on a frame. From these note candidates, all possible chords are formed and the most plausible chord path among all these chord candidates is calculated through a rule-based process. In order to retrieve the most plausible solution, three criteria are used: the sum of the amplitudes of the chromas within the chord, the chord duration and the fitness to key. These criteria tend to favor loud and long-lasting chords belonging to the evaluated key. Each of these three criteria is evaluated with a number of points. The final chord path is the one obtaining the largest number of points.

The system is evaluated on two corpus: one containing resynthesized MIDI and one containing real pop audio signals. The chord dictionary is composed of major, minor, minor seventh, major seventh, suspended 4th and diminished chords.

Papadopoulos & Peeters (2008) describe a beat-dependent chord recognition system. A beat estimation algorithm is used (Peeters (2007)) in order to extract the tactus and tatum positions[2]. An ergodic HMM is built: each of the states represents one chord label and one metric position. The system detects major and minor chords, and the chord recognition is either performed on the tactus level (4-beats meter) or the tatum level (8-beats meter): the number of states is therefore either $24 \times 4 = 96$ or $24 \times 8 = 192$.

The initial state distribution is supposed to be uniform. The observation probabilities are built by considering both chords and metric positions. The chord observation probabilities are obtained by computing the correlation between chroma vectors and pre-defined chord templates, while the metric position probabilities are supposed to be uniform. The state transition matrix takes into account both chord transitions and metric positions, and is built with musically inspired values. The final path is decoded with a Viterbi algorithm.

The system is evaluated on 66 Beatles songs: results show that taking into account the metric position improves the performances.

Pauwels et al. (2008) define a probabilistic framework for chord recognition using the key information. They define an acoustic model where each component of the chroma vector is modeled by a single-sided Gaussian either centered on 0 or $\frac{1}{3}$ depending on whether the chroma is present or not. The chord-key transition model is derived from the harmonic distance introduced by Lerdahl (2001). The transcription is finally carried out by using a Dynamic Programming Search.

Mauch et al. (2009b) and Mauch & Dixon (2010) also use key, along with metric position (rhythm) and bass information for their chord recognition systems.

Two input features are introduced: one 12-dimensional treble chroma vector and one 13-dimensional bass chroma vector containing one extra 'no bass' component. These features are all beat-synchronous. Instead of HMMs, which have already widely been used for chord recognition, the authors choose to base their method on Dynamic Bayesian Networks (DBNs). Their DBN is defined with 4 labels (metric position, key, chord and bass) and 2 observations (treble and bass chromagrams). All the relationships between labels are modeled with empirically defined and hand-tuned probabilities:

- The current metric position depends on the previous one.
- The current key depends on the previous one.
- The current chord depends on the previous one, on the current key and on the current metric position. It is observed with the treble chroma vector.
- The current bass depends on the current chord and on the previous chord. It is observed with the bass chroma vector.

In this system, 109 chords are considered (major, minor, major seventh, dominant seventh, major sixth, diminished, augmented, inverted major and no chords), each of them modeled by a fixed 12-dimensional Gaussian distribution. The algorithm is evaluated on the Beatles corpus: results show that the method is significantly better than the state-of-the-art.

Mauch et al. (2009b) propose to also use the structure information: the system extracts similar segments in the audio file, and the chromagram information is averaged on these segments in order to produce more robust chord transcriptions.

[2] *"The tactus, or beat level is a moderate metrical level which corresponds to the foot tapping rate. The tatum level corresponds to the shortest durational values in music that are still more than accidentally encountered"* (Papadopoulos & Peeters (2008))

2.2.4 Hybrid methods

The methods presented here combine aspects of the two previously described categories: they use both musical information (see 2.2.3) and training (see 2.2.2). We shall refer to them as hybrid methods.

Just like Sheh & Ellis (2003), Maddage et al. (2004)'s system is composed of several continuous density HMMs. Each model is composed of 5 states: 3 Gaussian mixtures whose parameters are all learned from the data, an entry and an exit state. The final path is calculated through a Viterbi algorithm. 4 chord types are detected by the system: major, minor, diminished and augmented, leading to 48 HMMs. The detected chords are then processed with a rule-based method, which detects the local key and corrects the chords which do not belong to the key. Some theoretical rhythmic information are also introduced so as to favor some chord transition times.

The system is tested by cross validation: for every turn, 30 songs are used for training and 10 for testing. In addition to this training data, chord samples from real or synthetic instrument are also used. Results show that taking into account musical information such as key or rhythmic structure in addition of training allows to improve the performances of this chord recognition system.

Yoshioka et al. (2004) propose a method for joint recognition of key, chord symbols and chord boundaries. The method assumes that the key does not change through the piece and that it is always major. Three parameters are to be output by the system: chord symbols, chord boundaries and the key symbol. These 3 parameters are concurrently calculated by a rule-based system which evaluates all possible hypothesis with music- or training-based criteria. A hypothesis-search algorithm is applied in order to generate and select the most plausible hypothesis. Three input features are used for the derivation of the hypothesis: chroma vectors, chord progression patterns and bass sound information.

The evaluated criteria are:

- the fitting of chroma vectors to chord models learned from annotated audio data ;
- the fitting of chord progression to theoretical chord progressions from music theory and to the current detected key ;
- the fitting of bass sound to current chord.

This system is synchronized on beat times thanks to the beat detection method proposed by Goto (2001) and is evaluated on 7 one-minute pieces of music. The training data used for the computation of the chord models is composed of 2592 synthesized chord samples and of the 6 one-minute pieces that are not used for the test. The system detects major, minor, diminished and augmented chords. Results show in particular that the introduction of music-inspired chord progression patterns improves the performances.

The method proposed by Burgoyne & Saul (2005) relies on Sheh & Ellis (2003)'s method but introduces some musical information by defining a more complex harmonic model.

For every key, the 24 chords within the chord dictionary (major and minor) are divided into 4 harmonic groups, depending on their fit to key. This harmonic space is then used to build an harmonic HMM with 24 states. The observation distribution is assumed to be Dirichlet. The transition matrix is built with 5 empirical parameters: the probability of staying in the same harmonic group, and the 4 probabilities of staying in the same chord state inside every harmonic

group. The Dirichlet distribution parameters are learned from annotated audio data composed of 5 Mozart symphonies.

The method is tested on one Mozart symphony: most of the errors are caused by the harmonic proximity of chords.

Burgoyne et al. (2007) propose a comparison of several training-based approaches for chord recognition, in particular HMMs and Conditional Random Fields (CRFs). In a CRF, the hidden state not only depends on the current observation but also on the complete sequence of observations. The decoding process is comparable to Viterbi's but needs aligned and annotated data.

Three systems are compared:

- HMMs where each chord has its own learned model (Sheh & Ellis (2003)): training with annotated but not necessarily aligned data ;
- HMMs where each chord is modeled by one state (Bello & Pickens (2005) (see 2.2.3): training with annotated and aligned data ;
- CRFs trained with annotated and aligned data.

In the HMM systems, the observation distribution is assumed to be a Gaussian mixture: several numbers of Gaussians are tested. The observation distribution used for the CRF system is either Gaussian, Dirichlet, or a combination of these distributions.

4 types of chords are detected: major, minor, augmented and diminished. Just like Sheh & Ellis (2003), several models are to be merged but this time, they introduce some music knowledge since the averaging principle is based on key. The real key is indeed supposed to be known for the training songs: the chroma vectors can therefore be rotated so as all the songs are in *C major* key. This mapping enables the training to be independent of the key.

The methods are tested on a 20 Beatles songs corpus: the best results are obtained with HMM systems where each chord has its own learned model.

Papadopoulos & Peeters (2007) propose a large-scale study of HMM-based chord recognition systems, in particular those of Sheh & Ellis (2003) and Bello & Pickens (2005). Taking a 24-state HMM detecting major and minor chords as a baseline system, many variants are tested:

- 3 observation distributions: one randomly initialized Gaussian model with all parameters learned from annotated audio data, one musically-inspired Gaussian model taking into account one or more harmonics of the notes within the chord (see Gómez (2006b)), and one non-Gaussian model based on the correlations between chroma vectors and fixed chord templates (taking into account one or more harmonics) ;
- 4 state transition matrices: one music-based (Bello & Pickens (2005)), one inspired by cognitive studies (Noland & Sandler (2006)), one trained through an EM algorithm (Sheh & Ellis (2003)) and one trained directly from the annotated database.

The final chord path is determined with a Viterbi algorithm. The method is evaluated on 110 Beatles songs. Results show that the untrained state transition matrix give better performances and that the correlation observation model with 6 harmonics outperforms the other tested observation distributions.

Zenz & Rauber (2007) use musical information such as key and beat for the chord transcription. For each beat-synchronous frame, several probable chords are detected. The selection is

done by calculating a distance between chroma vectors and reference chroma vectors learned on audio data. Three chord types are tested: major, minor and diminished. The chord transcription is performed from the list of possible chords for every timespan. A chord change penalty is applied so as to favor long-lasting chords and a chord sequence filtering is introduced so as to favor chords belonging to the detected key.

The method is evaluated on 35 songs of various styles and genres. Step-by-step results show that all the introduced post-processing information allows to increase the accuracy rates.

Lee & Slaney (2008) describe a joint chord and key estimation system based on training. They avoid the fastidious task of data annotation by using wave files generated from automatically annotated MIDI files.

The MIDI files are first annotated thanks to a symbolic chord transcription method. Then, WAVE files are generated from these MIDI files. This easily obtained annotated audio data can be used for training.

Two systems are defined. One is composed of 24 key-dependent HMMs, each of them containing 24 states (one per major and minor chord) and trained with annotated pop data (Beatles). The other one is defined by 24 key-dependent HMM each containing 36 states (major, minor and diminished) and trained with annotated classical data. In both these systems, each chord is defined by one state and modeled by a Gaussian observation distribution. The state transition matrix is supposed to be diagonal.

In order to efficiently train the models, they use the protocol introduced by Sheh & Ellis (2003): only one model per chord type is trained, by rotating chroma vectors. Likewise, only one major and one minor key-dependent HMM are trained, by rotating the state transition matrices. The two systems are finally decoded with a Viterbi algorithm which outputs the most likely chord path and key-model for every song.

The system is tested on two Beatles albums and two classical pieces. Results show that the introduction of key in the system allows to improve the results and that the transcription is sensitive to the type of data the models have been trained with.

Mauch & Dixon (2008) propose to model the sonorities a chord is made up by developing a method inspired by word models used in speech processing. These sonorities, or *subchords*, are modeled with a 3-component Gaussian mixture whose parameters are learned from annotated data. A subchord score function is calculated, which describes the probability of a subchord given an observation. Bass information can also be used in the score function computation. The most likely subchord sequence is used as an input for a HMM.

In this HMM, each chord is modeled by 3 states, in order to get round the assumption of exponential time distribution. 60 chords are detected (major, minor, dominant, diminished and suspended). All the HMM parameters are learned from annotated data.

The system is evaluated with 175 Beatles songs, divided in 5 groups: 4 groups are used for training and 1 for testing. Results are good despite the fact that the used chord dictionary is rather large, and the fragmentation is also reduced thanks to the 3-state model which assumes a Gamma and not an exponential temporal distribution.

The method introduced by Sumi et al. (2008) builds on Yoshioka et al. (2004)'s one. Just like Yoshioka et al. (2004), their aim is to provide a system allowing to detect chord labels, chord boundaries and key in musical pieces. The chord transcription is performed by optimizing a probabilistic function depending on 3 notions: acoustic features (chromagram), bass pitch probabilities and chord transition probabilities.

The acoustic features score function is based on several fully-trained Gaussian mixtures. One model is build for each chord type (major, minor, diminished and suspended 4th). The bass pitch probabilities are determined by using a score function relying on learned values, expressing the statistical probability of appearance of each bass pitch in each of the 48 chords. Finally, the chord transition probabilities score function is built from trained 2-gram models depending on key. The final transcription is carried out with a hypothesis search.

The algorithm is tested on a corpus composed of 150 Beatles songs. All these songs are used for training of the chord transition probabilities model, while only 120 of these songs are used for training of the acoustic features and bass pitch probabilities score functions. Results show that taking into account other types of information such as key or bass allows to improve the performances of the system.

The system proposed by Khadkevich & Omologo (2009b) builds on language models previously used in speech processing. They introduce statistical information on chord progressions for bigrams (two successive chords), 3-grams or more generally N-grams.

The first step of the process is the construction of a key-dependent HMMs, where each chord is represented by 3 states. The first and last states do not model any observation. This 3-states construction interestingly allows to avoid the exponential duration model assumed in classical HMMs. Observation distributions are supposed to be 512-dimensional Gaussian mixtures, while covariance matrices are supposed to be diagonal. 24 chords are modeled in the system (major and minor chords). Just like other key-dependent HMMs, only two models are trained, one for major keys and one minor keys, by rotating state transition matrices. The HMM is trained with annotated data: the key information is given by Peeters (2006)'s method. A penalty is introduced so as to favor long-lasting chords.

A Viterbi algorithm produces a lattice which is then used as an input for a language model. This language model describes the chord progressions by calculating from annotated data the probabilities of appearance of every N-gram. In order to incorporate rhythmic structure and to use the mutual dependency between rhythm and harmony, Factored Language Models (FLMs) are introduced, which allow to define a chord as a set of factors (namely the chord label and the chord duration). All the FLM parameters are learned from annotated data.

The method is tested on the Beatles corpus. Results show that the introduction of language models enhances the performances but that the use of FLM only brings a slight improvement.

Weil et al. (2009) use an HMM composed of 24 states, each of them representing a chord (major and minor). All the HMM parameters are learned from beat-synchronous annotated data. The model is then decoded with a Viterbi algorithm. From these first chord sequence, the measure grid is evaluated by assuming that the probability of chord change depends on the position in the measure. This measure grid is then used to compute a refined chord sequence, by introducing the measure information in the state transition matrix.

The system is evaluated on a 278 files database composed of resynthesized MIDI files.

2.2.5 Summary

Table 2.7 presents the main weak and strong points of the four previously presented approaches.

	+	-
Template-based	• No need for annotated data • Theoretically independent from music style • Low-computational time	• Can produce harmonically and rhythmically irrelevant results • Can output fragmented transcriptions
Training-based	• Adapts well to real audio data • Can apply to songs which do not necessarily follow music theory • No a priori on the expected results	• Can capture noise, transients, etc. • Can be dependent on development corpus or on music style • High computational time
Music-driven	• Takes into account the multi-level character of music (harmony, rhythm, structure) • No need for annotated data • Joint estimation of external outputs (key, beats, etc.)	• Can be dependent of music genre • Can produce disappointing results on songs which do not follow music theory
Hybrid	cf training-based and music-driven methods	cf training-based and music-driven methods

Table 2.7: Weak and strong points of the four categories of chord recognition methods

Chapter 3

Corpus and evaluation

Contents

This chapter proposes a complete description of the protocol and the corpora used in this manuscript for the evaluation and comparison of chord recognition methods.

The evaluation framework presented in this chapter is largely inspired by the one defined within the MIREX 2008[1] & 2009[2]. Downie (2008) describes MIREX *"as the community-based framework for the formal evaluation of Music Information Retrieval (MIR) systems and algorithms"*. This large-scale evaluation includes many tasks and has been performed since 2005. The task we are interested in is the Audio Chord Detection task. It consists in transcribing a WAVE file into a sequence of chord labels with their respective on and off time. Until now, the chord dictionary used in MIREX is composed of 25 labels: 12 major chords, 12 minor chords, and a "no chord" state written 'N', which corresponds to silences or untuned material.

Because of this rather small output chord dictionary, all the chord types present in the annotated ground-truth files are mapped into the major and minor types according to the rules described in Table 3.1. The general rule for the chord mapping used in MIREX is that all explicitly minor chords are mapped to the minor type, while all the others are mapped to the major type.

major	major, diminished, augmented, major seventh, dominant seventh, diminished seventh, half diminished seventh, major sixth, dominant ninth, major ninth, suspended fourth, suspended second
minor	minor, minor seventh, minor major seventh, minor sixth, minor ninth

Table 3.1: Chord types mapping used in MIREX 2008 & 2009

3.1 Presentation of the corpora

3.1.1 Corpus 1 : Beatles

Our first evaluation database is made of the 13 Beatles albums (180 songs, PCM 44100 Hz, 16 bits, mono). This database is in particular the one used in MIREX 2008 & 2009 for the Audio Chord Detection task. The evaluation is performed thanks to the chord annotations kindly provided by Harte et al. (2005). The alignment between annotations and WAVE files is performed with the algorithm provided by Christopher Harte. In these annotation files, 17 types of chords and one 'no chord' label (N) are present (see Table 3.1). The distribution of chord types in the corpus (before the mapping to major and minor) is presented on Figure 3.1 while the one obtained after mapping is displayed on Figure 3.2. We can see that before mapping, the most common chord types are major, minor, dominant seventh, 'no chord' states, minor seventh and ninth. Any other chord type represents less than 1% of the total duration. After mapping, the vast majority of the chords is major: we shall see in Section 4.2.5.1 that this property influences the results we obtain on this corpus.

This corpus has been widely used since its release, and is now the reference corpus for the task. All the recent chord recognition systems have used this corpus either for development or for training. Indeed the distribution of chord labels after mapping is rather spread (see Figure

[1] `http://www.music-ir.org/mirex/wiki/2008:Audio_Chord_Detection`
[2] `http://www.music-ir.org/mirex/wiki/2009:Audio_Chord_Detection`

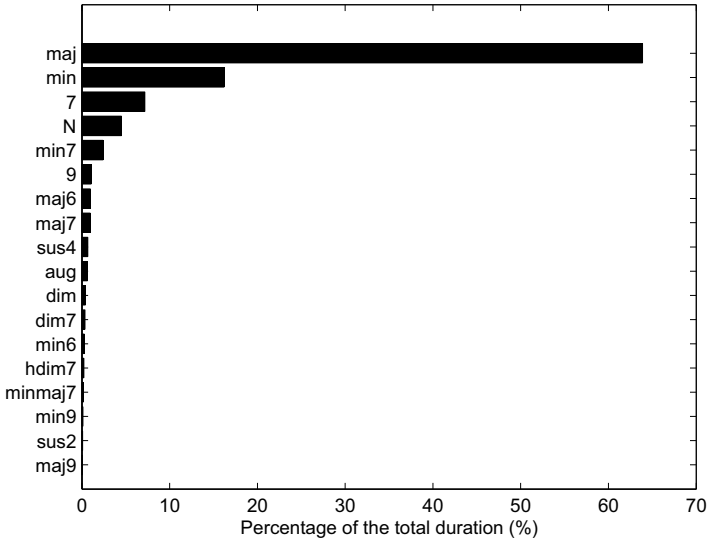


Figure 3.1: Distribution of chord types in the Beatles corpus before mapping (as percentage of the total duration)

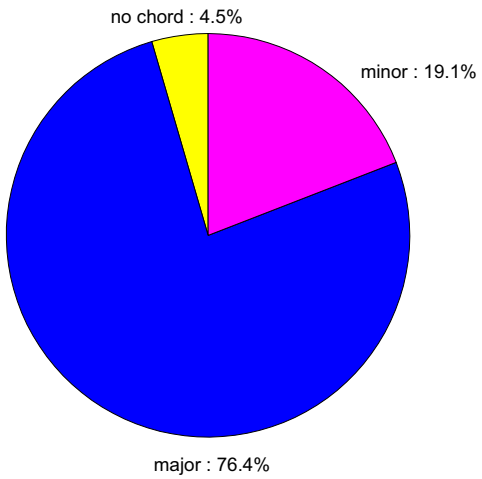


Figure 3.2: Distribution of chord types in the Beatles corpus after mapping (as percentage of the total duration)

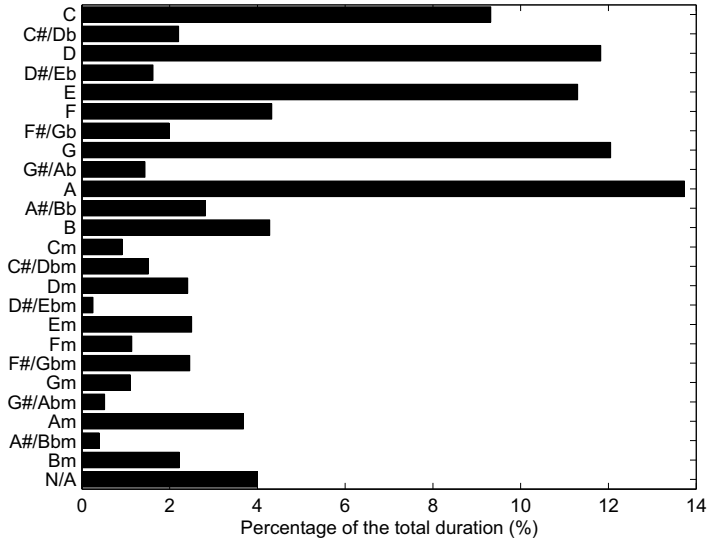


Figure 3.3: Distribution of the 25 chord labels in the Beatles corpus after mapping (as percentage of the total duration)

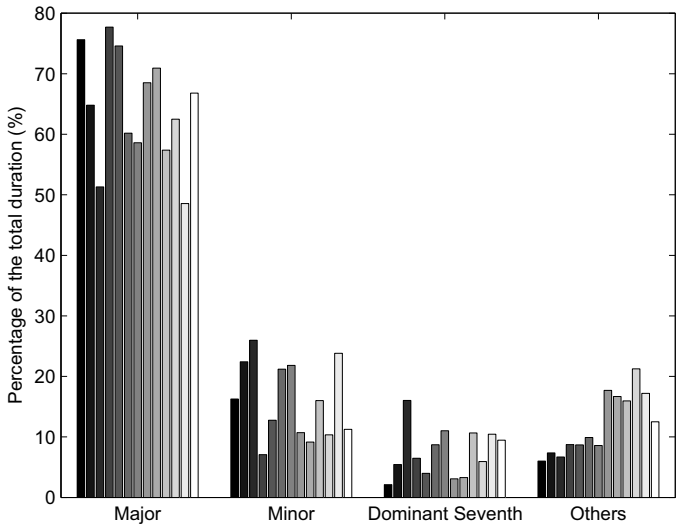


Figure 3.4: Distribution of chord types in the 13 Beatles albums before mapping (in chronological order)

3.3) and enables to efficiently train systems. Nevertheless, we notice that there is a large number of major chords, and that the two chord types are therefore not equiprobable.

The 13 Beatles albums have been released between 1963 and 1970, which is a rather short time period. This is also a time where the recording conditions were not always optimal: several songs are very noisy or detuned, in particular in the first albums. Indeed, in some songs, the tuning is really low (eg. *Lovely Rita*, *Wild Honey Pie*,...) or even changes within the song (eg. *Strawberry Fields Forever*,...). Some songs also contain untuned material such as spoken voice, non-harmonic instruments or experimental noises (applause, screams, car noise,...) (eg. Revolution 9). This corpus can be referred to as pop-rock or rock music, although the Beatles have experimented various genres and styles, especially in their last albums. For instance, they have used instruments coming from other cultures than the proper rock culture. The complexity of the chord vocabulary also changes within the albums: Figure 3.4 displays the distribution of chord types in the albums. We can see that the number of major, minor and dominant seventh chords varies much with the album. Yet, the last six albums clearly contain more chord types (other than major, minor and dominant seventh) than the first seven ones. The corpus is therefore tricky as it is coherent (in genre and in time) but still owns many complex exceptions (complex chord progressions, unusual instruments, bad recordings conditions).

More information can be found in the very complete and interesting musicology analysis (harmony, recording conditions, instruments, etc...) of the 180 Beatles songs by Alan W. Pollack[3].

3.1.2 Corpus 2 : MIDI

Our second evaluation database is composed of 12 songs from various artists in different genres (blues, country, pop and rock): see Table 3.2 for the song titles. The audio files (PCM 44100 Hz, 16 bits, mono) are synthesized from MIDI files[4] using the free software Timidity ++[5]. Timidity ++ is a software synthesizer which can generate realistic audio data from MIDI files using a sample-based synthesis method. We have manually annotated the songs: 5 types of chords are present (maj, min, 7, sus2, sus4) as well as the 'no chord' label (N). The distribution of chord types in the MIDI corpus before mapping is displayed on Figure 3.5, and the one after mapping on Figure 3.6. We see that for the most part the chords of this corpus are major: the gap between major and minor chords is even larger than on the Beatles corpus.

This corpus is rather small and consequently all the chord labels are not represented (see Figure 3.7). Furthermore, since this corpus is artificially synthesized, the characteristics of the instruments (timbre, onset and offset models,...) do not exactly reflect the problematic of real audio recordings. In particular, the recording conditions are necessarily perfect as well as the tuning. Nevertheless, this corpus allows to experiment chord recognition system on various genres of music and to build home-made examples by controlling the instruments present in the song.

3.1.3 Corpus 3 : Quaero

The third corpus was provided to us by the Quaero project[6]. It consists of 20 real audio songs annotated by IRCAM (PCM 22050 Hz, 16 bits, mono) from various artists (see Table 3.3) and

[3] `http://www.icce.rug.nl/~soundscapes/DATABASES/AWP/awp-notes_on.shtml`
[4] The MIDI files were obtained on `http://www.mididb.com`
[5] The software is freely downloadable on `http://timidity.sourceforge.net`
[6] Quaero project : `http://www.quaero.org`

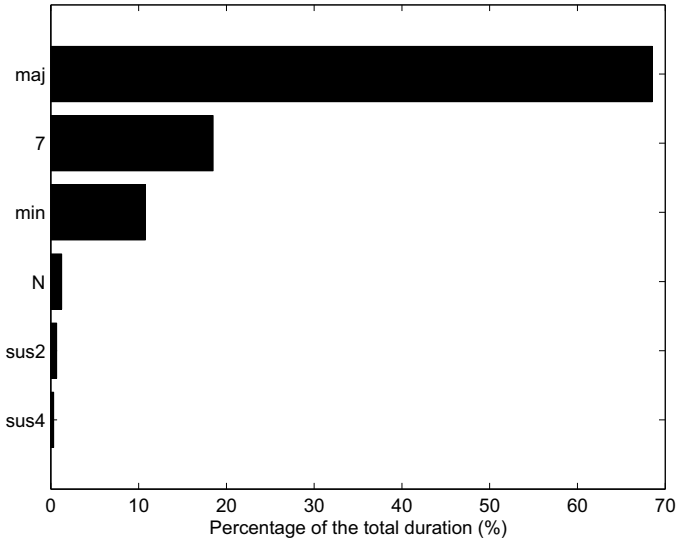


Figure 3.5: Distribution of chord types in the MIDI corpus before mapping (as percentage of the total duration)

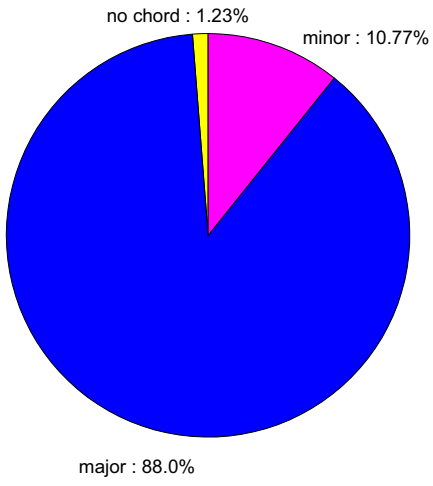


Figure 3.6: Distribution of chord types in the MIDI corpus after mapping (as percentage of the total duration)

Genre	Title	Artist
Country	Ring of fire	Johnny Cash
	Tennessee waltz	Roy Acuff
	Stand by your man	Tammy Wynette
Pop	Dancing queen	ABBA
	I drove all night	Cyndi Lauper
	Born to make you happy	Britney Spears
Blues	Blues stay away from me	The Delmore Brothers
	Boom, boom, boom	John Lee Hooker
	Keep it to yourself	Sonny Boy Williamson
Rock	Twist and shout	The Beatles
	Let it be	The Beatles
	Help !	The Beatles

Table 3.2: Description of the MIDI corpus

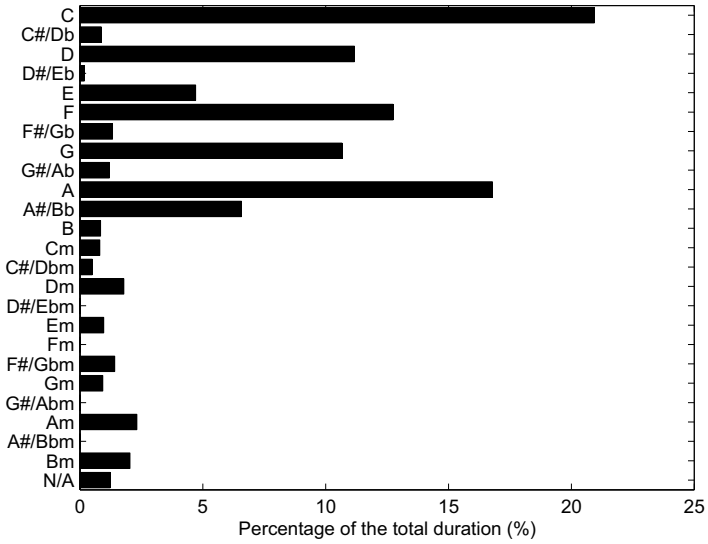


Figure 3.7: Distribution of the 25 chord labels in the MIDI corpus after mapping

various genres (pop, rock, electro, salsa, disco,...). The annotation files only contain major and minor chords (see Figure 3.8). This corpus, just like the MIDI one, has a rather small size: all the 25 chord labels are therefore not present (see Figure 3.9).

Interestingly, this corpus does not contain any Beatles song, so there is no overlap with the other corpora. Furthermore, it also contains more minor chords than the two previously described corpora. This corpus has never been used as development corpus, so in spite of its small size, it constitutes a good test corpus allowing to objectively compare chord recognition methods.

Title	**Artist**
Breathe	Pink Floyd
Brain Damage	Pink Floyd
I'm in love with my car	Queen
Chan chan	Buenavista Social Club
De camino a la vereda	Buenavista Social Club
Son of a preacher man	Dusty Springfield
Cryin	Aerosmith
Pull together	Shack
Kingston town	UB40
This ain't a scene, it's an arms race	Fall Out Boy
Say it right	Nelly Furtado
...Comes around	Justin Timberlake
Touch my body	Mariah Carey
Waterloo	ABBA
Believe	Cher
Another day in paradise	Phil Collins
Don't let me be misunderstood	Santa Esmeralda
Fox on the run	Sweet
Words	FR David
Orinoco flow	Enya

Table 3.3: Description of the Quaero corpus

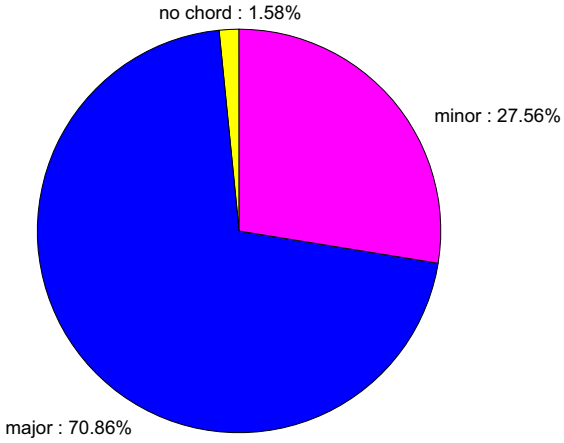


Figure 3.8: Distribution of chord types in the Quaero corpus (as percentage of the total duration)

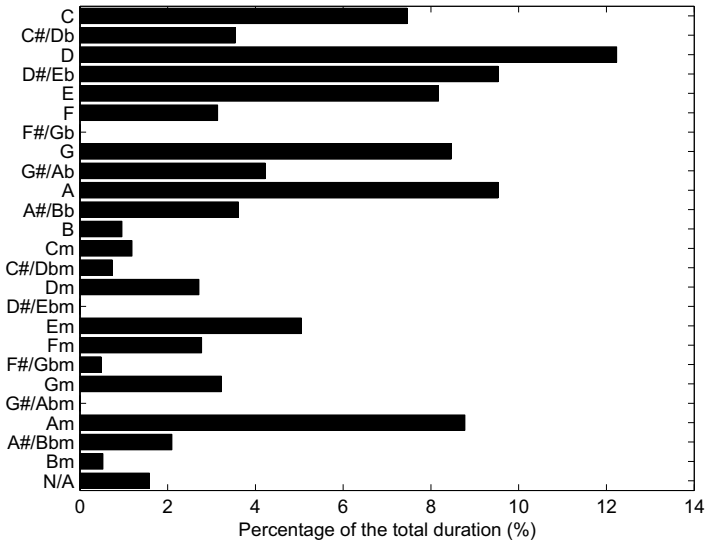


Figure 3.9: Distribution of the 25 chord labels in the Quaero corpus (as percentage of the total duration)

3.2 Evaluation

The chord transcription task is actually the fusion of two subtasks: a recognition task (find the correct label for each frame) and a segmentation task (find the correct chord boundaries). Also a good transcription is supposed to be compact and to use a sparse chord vocabulary. We list here some metrics in order to evaluate not only the quality of transcription but also the accuracy of segmentation and of chord vocabulary, as described below.

Let $\mathcal{S}$ be our corpus composed of S songs. In the annotation files, each song s is segmented with T_s temporal segments $\mathcal{U}(s) = \{u_1(s), \ldots, u_{T_s}(s)\}$. These segments have variable lengths and each of them stands for one chord. For each segment $u_t(s)$, the annotation files provide a chord label $l_t(s)$.

Let us denote $|u|$ the duration of segment u and $u \cap u'$ the intersection of segments u and u'. The total length of the song s is, with our notations, $|s| = \sum_{t=1}^{T_s} |u_t(s)|$.

With our transcription method, each song s is divided into $\hat{T}_s$ segments, and every segment $\hat{u}_t(s)$ is given a chord label $\hat{l}_t(s)$.

3.2.1 Recognition metrics

Our primary goal is to evaluate the accuracy of the chord labels attributed by our method. Overlap Score (OS(s)) is defined as the ratio between the length of the correctly analyzed chords and the total length of the song, i.e.,

$$OS(s) = \frac{\sum_{t=1}^{T_s} \sum_{t'=1}^{\hat{T}_s} |u_t(s) \cap \hat{u}_{t'}(s)|_{l_t(s)=\hat{l}_{t'}(s)}}{|s|} \tag{3.1}$$

This OS(s) ranges from 0 to 1. The higher the score is, the better the recognition accuracy is.

The Average Overlap Score (AOS), which has been used for MIREX 2008, is the mean of all the OS(s) of the corpus:

$$AOS = \frac{1}{S} \sum_{s=1}^{S} OS(s) \tag{3.2}$$

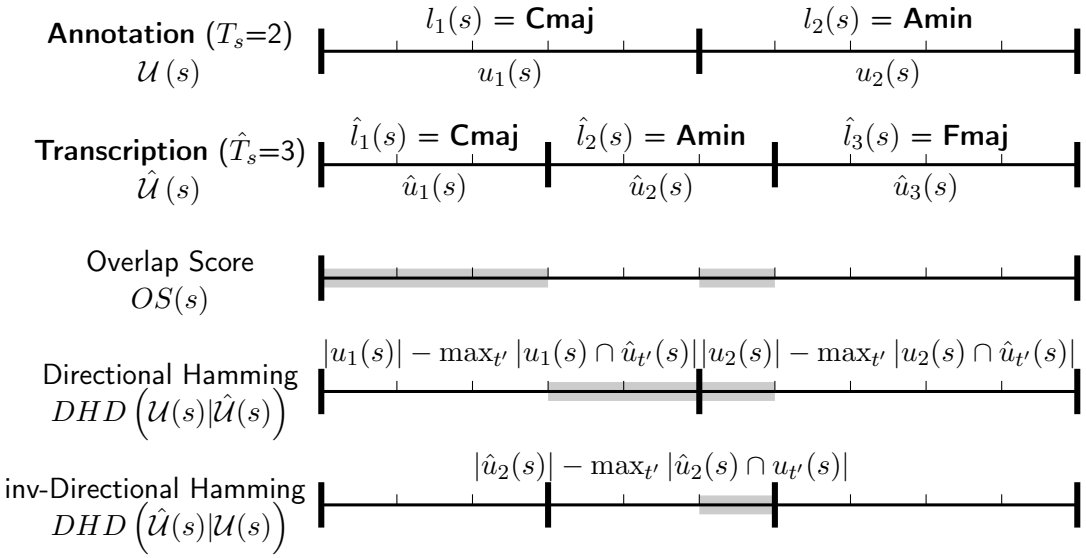
Another score, called Weighted Average Overlap Score (WAOS), can be defined as a weighted mean of all the OS(s) of the corpus. Every score is weighted by its respective song length. This score can be seen as the OS(s) obtained if all the songs of the corpus were put together in order to form one song. This score has been used for MIREX 2009:

$$WAOS = \frac{\sum_{s=1}^{S} |s| \times OS(s)}{\sum_{s=1}^{S} |s|} \tag{3.3}$$

The chord recognition task can be seen as the joint recognition of chord root and chord type. Two other metrics can also be defined: the Average Root Overlap Score (AROS) and the Weighted Average Root Overlap Score (WAROS), which are defined just like the AOS and WAOS, but only assess root detection.

3.2.2 Segmentation metric

In order to evaluate the quality of the segmentation, recent chord publications (Mauch & Dixon (2010)) have used the Hamming Distance (HD(s)) calculated from the Directional Hamming



$$\text{Overlap Score} = \frac{3+1}{10} = 0.4$$

$$\text{Hamming Distance} = \frac{1}{2} \times \left(\frac{2+1}{10} + \frac{1}{10}\right) = 0.2$$

$$\text{Reduced Chord Length} = \frac{\frac{3+3+4}{3}}{\frac{5+5}{2}} = \frac{2}{3}$$

$$\text{Reduced Chord Number} = \frac{3}{2}$$

$$\text{False Chord Label Number} = 1$$

Figure 3.10: Example of calculation of Overlap Score, Hamming Distance, Reduced Chord Length, Reduced Chord Number and False Chord Label Number. The figure uses a discrete clock for purpose of illustration but in reality the time scale is continuous up to the sample period.

Divergence (DHD) (Abdallah et al. (2005)). The DHD reflects the unfitness of one segmentation to another. The DHD between the annotation segmentation $\mathcal{U}(s)$ and the transcription segmentation $\hat{\mathcal{U}}(s)$ is defined as:

$$DHD\left(\mathcal{U}(s)|\hat{\mathcal{U}}(s)\right) = \frac{\sum_{t=1}^{T_s} |u_t(s)| - \max_{t'} |u_t(s) \cap \hat{u}_{t'}(s)|}{|s|} \tag{3.4}$$

The inverse DHD is defined as:

$$DHD\left(\hat{\mathcal{U}}(s)|\mathcal{U}(s)\right) = \frac{\sum_{t=1}^{\hat{T}_s} |\hat{u}_t(s)| - \max_{t'} |\hat{u}_t(s) \cap u_{t'}(s)|}{|s|} \tag{3.5}$$

Finally, the HD(s) between the two segmentations is defined as the mean of the two directional Hamming divergence:

$$HD\left(s\right) = \frac{DHD\left(\mathcal{U}(s)|\hat{\mathcal{U}}(s)\right) + DHD\left(\hat{\mathcal{U}}(s)|\mathcal{U}(s)\right)}{2} \tag{3.6}$$

The HD(s) tends to reflect the dissimilarity of two segmentations: this metric takes values between 0 and 1 and the lower the value, the better the quality of the segmentation. In particular, a value of 0 is obtained when both segmentations are exactly the same. The mean of all the HD(s) of the corpus is called Average Hamming Distance (AHD).

3.2.3 Fragmentation metric

A chord transcription is expected to display "compactness". Indeed, the presence of numerous fragmented chords can lead to noisy and hardly understandable transcriptions. In order to evaluate whether a chord recognition method produces fragmented transcriptions or not, we propose to introduce a metric called Average Chord Length (ACL). We first define for a song s, the Reduced Chord Length (RCL(s)) as the ratio between the experimental average chord duration and the ground truth one. That is to say:

$$RCL(s) = \frac{\frac{1}{\hat{T}_s}\sum_{t=1}^{\hat{T}_s} |\hat{u}_t(s)|}{\frac{1}{T_s}\sum_{t=1}^{T_s} |u_t(s)|} \tag{3.7}$$

Note that this score can also be defined as the ratio between T_s and $\hat{T}_s$. This metric should be as close as possible to 1: when lower than 1, the transcriber tends to overfragment the piece. The ACL is finally the mean of all the RCL(s) of the corpus.

3.2.4 Chord vocabulary metrics

Another indicator of the quality of a chord transcription is the compactness of the chord vocabulary used for the transcription. For each song, we define the chord vocabulary as the subset of the chord dictionary containing all the chord labels useful to transcribe the song. In the case where the song can relate to a particular key, it reflects by extension the tonal context of the piece, but we see in chord vocabulary a more general notion. For example, in case of modulations, the chord vocabulary can contain chords from various keys. We define two metrics for assessing the correctness of the estimated chord vocabulary: the Average Chord Number (ACN) and the Average False Chord Label Number (AFCLN).

Context	Song score	Corpus score	Range	Criterion
Recognition	Overlap Score (OS(s))	Average Overlap Score (AOS) Weighted Average Overlap Score (WAOS)	$[0;1]$	as high as possible
	Root Overlap Score (ROS(s))	Average Root Overlap Score (AROS) Weighted Average Root Overlap Score (WAROS)		
Segmentation	Hamming Distance (HD(s))	Average Hamming Distance (AHD)	$[0;1]$	as low as possible
Fragmentation	Reduced Chord Length (RCL(s))	Average Chord Length (ACL)	$[0;\infty[$	as close to 1 as possible
Chord vocabulary	Reduced Chord Number (RCN(s))	Average Chord Number (ACN)	$[0;\infty[$	as close to 1 as possible
	False Chord Label Number (FCLN(s))	Average False Chord Label Number (AFCLN)	$[0;\infty[$	as low as possible

Table 3.4: Summary of the evaluation metrics

Given a song s, we define the Reduced Chord Number (RCN(s)) as the ratio between the number of different chord labels used for the transcription and the one used in the ground truth annotation. This metric should be close to 1. When greater than 1, the transcription uses a too wide chord vocabulary. The RCN(s) is the mean of all the RCN(s) of the corpus.

For each song s, the False Chord Label Number (FCLN(s)) is the number of chord labels that do not belong to the annotation files. Then, the AFCLN is the mean of all the FCLN(s) of the corpus. It should be as low as possible: an AFCLN of 0 would indicate that the method always detects the good chord vocabulary.

An example of calculation for all these metrics is displayed on Figure 3.10. The main properties of these metrics are also summarized on Table 3.4.

3.3 Significant differences between chord recognition systems

The scores we calculate in order to evaluate the performances of chord recognition methods can be very close, and therefore not give a precise idea whether a method is really better than another. In order to indicate whether there are significant differences between chord recognition methods, a statistical test was used in MIREX 2008 & 2009. This non-parametric statistical test, called Friedman's test (Friedman (1937)), allows to detect the differences in treatments (for us, chord recognition methods) across multiple blocks (songs of the corpus). One of the main advantages of this test among other variance tests is that it does not assume a normal distribution of the data. Furthermore, since this test is only based on rank statistics, it reduces the influence of the songs. The test outputs a p-value, which, when low enough, suggests that there is at least one method significantly different from the others. More details on the Friedman's test can be found in the box *What is Friedman's test?* (p 60).

In order to find out which are the methods significantly different from others, some post-hoc tests can be run. The one used in MIREX 2008 & 2009 is the Tukey-Kramer's test (Tukey (1953), Kramer (1956)). This is a single-step multiple comparison procedure used to find which mean values are significantly different from one another. In our case, the test compares the average rank of every treatment (for us, chord recognition method) to the average rank of every other treatment. More details on the Tukey-Kramer's test can be found in the box *What is Tukey-Kramer's test?* (p 60).

What is Friedman's test? (Friedman (1937))

Let $\mathbf{X} = \{x_{s,i}\}$ be the $S \times I$ matrix containing all the scores obtained by I chord recognition systems on a S-song corpus.

Friedman's test of significance compares column effects (influence of the chord recognition methods) in this matrix and is computed in 3 steps:

1. Calculate the ranks of the methods for each song: let $r_{s,i}$ be the rank of chord recognition method i for song s. When two methods get the same score on a song, the assigned rank is the average of the ranks that would have been assigned without ties.
2. Calculate the following values:
   - $\bar{r}_{\cdot,i} = \frac{1}{S}\sum_{s=1}^{S} r_{s,i}$: average rank on all songs for the chord recognition method i.
   - $\bar{r} = \frac{I+1}{2}$: average rank on all songs and all chord recognition methods.
   - $Q = \frac{12S}{I(I+1)}\sum_{i=1}^{I}\left(\bar{r}_{\cdot,i} - \bar{r}\right)^2$: dispersion between the $\bar{r}_{\cdot,i}$ around $\bar{r}$.
3. When S or I is large, the probability distribution of Q can be approximated by a chi-square distribution ($Q \approx \chi^2_{I-1}$). In this case the p-value is given by the probability $P\left(\chi^2_{I-1} \geq Q\right)$. If the p-value is low (usually when $p < 0.05$ or 0.01), it suggests that at least one chord recognition method is significantly different than the others, and appropriate post-hoc multiple comparisons tests can be performed.

What is Tukey-Kramer's test? (Tukey (1953))

Let suppose that a Friedman's test has been performed. We here compare the average rank on all songs $\bar{r}_{\cdot,i}$ two by two and determine whether they are significantly different of not.

The formula for the Turkey-Kramer's test q between chord recognition methods i and j is:

$$q = \frac{|\bar{r}_{\cdot,i} - \bar{r}_{\cdot,j}|}{\hat{\sigma}}$$

where $\hat{\sigma} = \sqrt{\frac{I(I+1)}{12S}}$.

This value is compared to a critical value $q_{critical}$ resulting from the *studentized range distribution* and depending on the desired confidence interval $1-\alpha$ (eg. $\alpha = 0.05$ means a confidence interval of 95%). The studentized range distribution has been tabulated and appears in many textbooks on statistics.

Finally, if

$$q > q_{critical}$$

then the two chord recognition methods i and j are significantly different.

Chapter 4

Deterministic template-based chord recognition

Contents

As seen in Chapter 2, chord recognition methods can be classified into four main categories: template-based, training-based, music-driven and hybrid. In this chapter, we propose to describe a deterministic template-based chord recognition approach, which only uses chord definitions to perform the transcription. It deterministically outputs one chord label from each input chromagram frame.

4.1 Description of the approach

4.1.1 General idea

Let $\mathbf{C}$ denote the chromagram, with dimensions $M \times N$ (in practice $M = 12$) composed of N successive chroma vectors $\mathbf{c}_n$. Let $\mathbf{W}$ be our $12 \times K$ chord dictionary, composed of K 12-dimensional chord templates $\mathbf{w}_k$. We define the chord dictionary as the set of all user-defined chord candidates. We want to find the chord k whose template $\mathbf{w}_k$ is the *closest* to chromagram frame $\mathbf{c}_n$ according to a specific measure of fit. We propose to measure the fit of chroma vector $\mathbf{c}_n$ to template $\mathbf{w}_k$ up to a scale parameter $h_{k,n}$. Given a measure of fit $D(\,.\,;\,.\,)$, a chroma vector $\mathbf{c}_n$ and a chord template $\mathbf{w}_k$, the scale parameter $h_{k,n}$ is analytically calculated so as to minimize the measure between $h\,\mathbf{c}_n$ and $\mathbf{w}_k$:

$$h_{k,n} = \underset{h}{\operatorname{argmin}}\ D\left(h\,\mathbf{c}_n; \mathbf{w}_k\right). \tag{4.1}$$

In practice $h_{k,n}$ is calculated such that:

$$\left[\frac{d\,D\left(h\,\mathbf{c}_n; \mathbf{w}_k\right)}{dh}\right]_{h=h_{k,n}} = 0. \tag{4.2}$$

We then define $d_{k,n}$ as:

$$d_{k,n} = D\left(h_{k,n}\,\mathbf{c}_n; \mathbf{w}_k\right). \tag{4.3}$$

The detected chord $\hat{k}_n$ for frame n is then the one minimizing the set $\{d_{k,n}\}_k$:

$$\hat{k}_n = \underset{k}{\operatorname{argmin}}\ d_{k,n}. \tag{4.4}$$

4.1.2 Chord templates

The first block making up our chord recognition system consists of chord templates. Each of them represents one of the chords within the chord dictionary.

4.1.2.1 Binary templates

Chord templates are 12-dimensional vectors in which each component represents the theoretical amplitude of one chroma within the chord. As seen in Chapter 2, these chord templates can either be learned from audio data (Sheh & Ellis (2003), Lee & Slaney (2008), Ryynänen & Klapuri (2008a)) or predetermined (Fujishima (1999), Pardo & Birmingham (2002), Bello & Pickens (2005), Harte & Sandler (2005), Lee (2006a), Papadopoulos & Peeters (2007)).

In the present manuscript, we chose only to use predetermined and fixed chord templates. Indeed, music theory already provides clear chord definitions which therefore do not necessarily have to be learned. Furthermore, learned chord models could also capture unwanted information

from chroma vectors. In fact, chroma vectors often contain noise or transients, which can produce blurred or unusable chord templates. Finally, the use of fixed chord models allows to skip the time-consuming learning phase and the long process of data annotation.

The most intuitive and simple chord model is a simple binary mask: an amplitude of 1 is given to the chromas within the chord and an amplitude of 0 is given to the other chromas.[1] For example in a $Cmajor$ chord an amplitude of 1 is given to chromas C, E and G while the other chromas have an amplitude of 0. Examples for $Cmajor$ and $Cminor$ chords are displayed on Figure 4.1.

With this principle, we can build chord templates for all types of chords (major, minor, dominant seventh, diminished, augmented,...) and for all root notes, by rotating the chord model. By convention in our system, the chord templates are normalized so that the sum of the amplitudes is 1 but any other normalization could be employed.

4.1.2.2 Harmonic-dependent templates

Chromagrams are supposed to extract the harmonic content of a music piece. That is to say that when an A is played, we expect to see one high intensity for the A chroma and low intensities for the other chromas. In practice, most of the pitched sounds are complex waveforms consisting of several components (called *partials* or *harmonics*). In this case, the frequency of each of these components is a multiple of the lowest frequency called the *fundamental frequency* (Gómez (2006a)). Thus, the information contained in a chromagram or any other spectral representation not only captures the intensity of every note but rather a blend of intensities for the harmonics of every note.

Figure 4.3 illustrates this phenomenon by displaying both the spectrogram and the chromagram obtained with a Cello playing an $A1$. We can see on the spectrogram that many harmonics are played besides the fundamental frequency, causing some chromas other than A to appear on the chromagram.

The two chord models defined in this section are inspired from the work of Gómez (2006b) and Papadopoulos & Peeters (2007). They build chord templates which take into account not only the chord notes, but also the harmonics of every note within the chord. An exponentially decreasing spectral profile is assumed for the partial amplitudes (see Figure 4.4). An amplitude of s^{i-1} is added for the i^{th} harmonic of every note within the chord. The parameter s is empirically set to 0.6 by Gómez (2006b) and Papadopoulos & Peeters (2007). This model is supposed to capture the average contribution of harmonics whatever instrument is playing. Table 4.1 presents the derivation of the model for an A note.

We build under this principle two chord models: one with 4 harmonics and one with 6 harmonics. In both cases, the chord model is obtained by summing the contributions of the harmonics of the chord notes. Examples for C major and C minor chords are displayed on Figure 4.2.

4.1.3 Measures of fit

The second block in our chord recognition method is composed of measures of fit, whose role is to estimate the fit between chroma vectors and chord templates.

[1] In practice a small value is used instead of 0, to avoid numerical instabilities that may arise with some measures of fit.

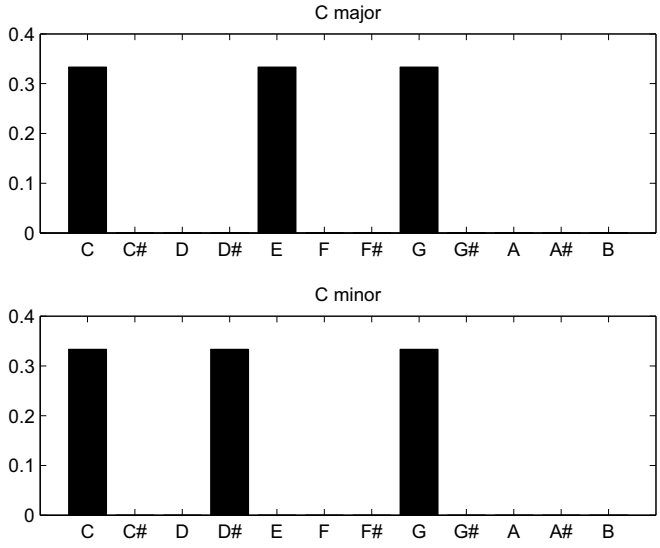


Figure 4.1: Binary chord templates for C major and C minor.

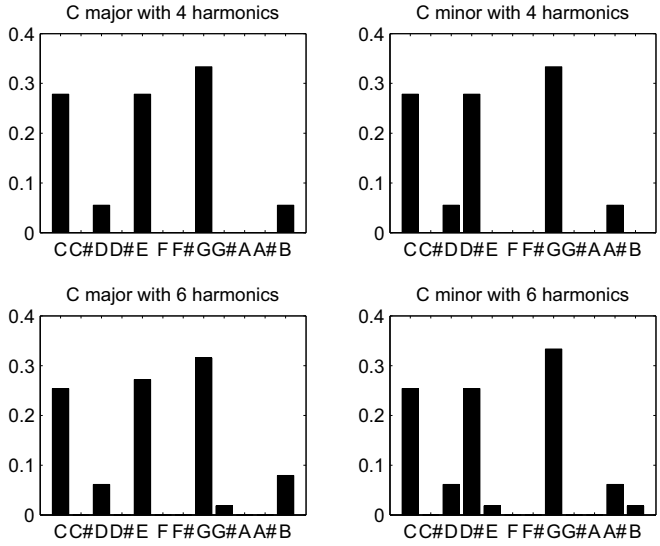


Figure 4.2: Chord templates for C major and C minor with 4 and 6 harmonics.

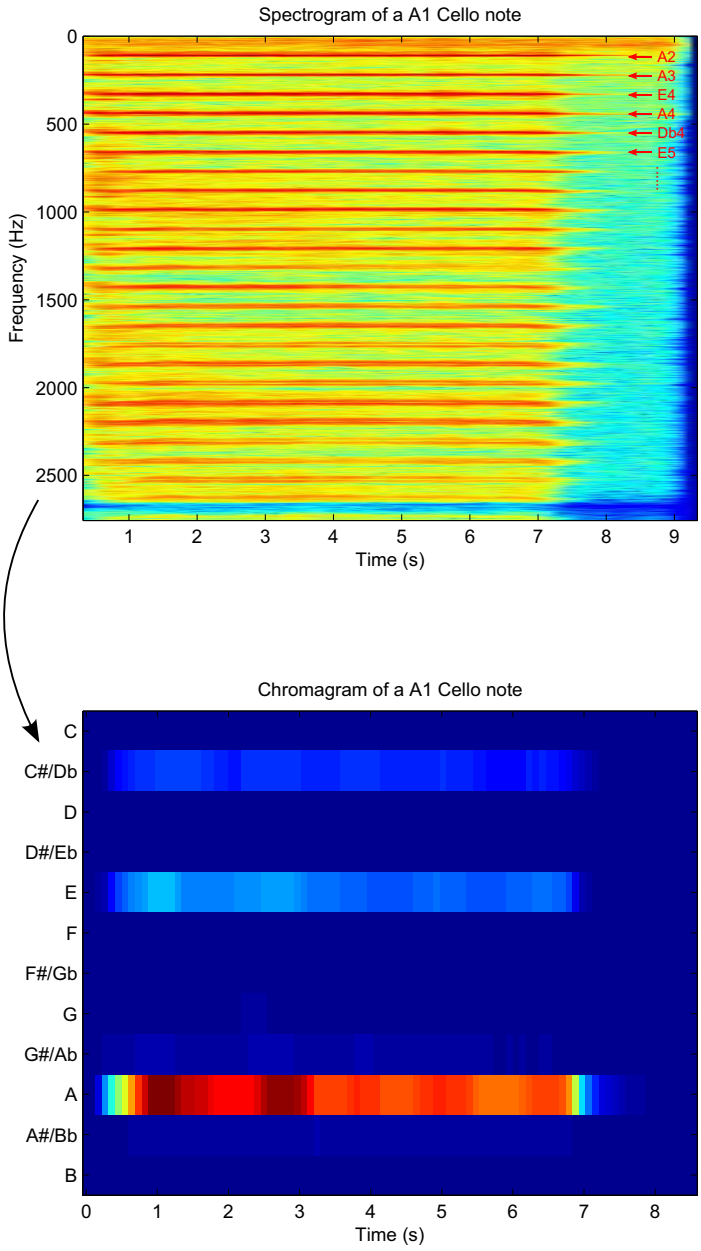


Figure 4.3: Spectrogram and chromagram of a A1 Cello note.

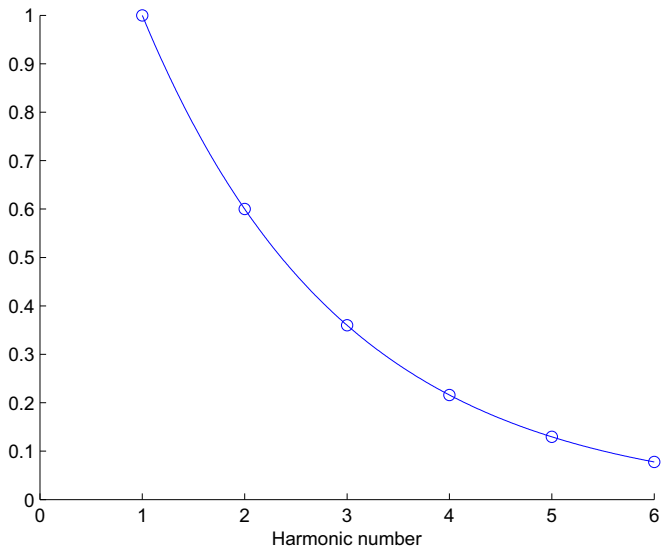


Figure 4.4: Contribution of the harmonics in Gómez (2006b)'s model (see Table 4.1).

Number	Frequency	Contribution	Chroma
1	f	1	A
2	$2f$	0.6	A
3	$3f$	0.6^2	E
4	$4f$	0.6^3	A
5	$5f$	0.6^4	$D\flat$
6	$6f$	0.6^5	A

Table 4.1: Frequencies and contributions of the 6 first harmonics for a A note in Gómez (2006b)'s model.

4.1.3.1 Definitions

For our recognition task we consider several measures of fit, which are popular in the field of signal processing. Table 4.3 gives the expressions of these different measures, along with the scale parameter analytically calculated from Equation (4.2) and the final expression of the recognition criterion $d_{k,n}$.

The well-known **Euclidean distance** (*EUC*) defined by

$$D_{EUC}\left(\mathbf{x}|\mathbf{y}\right) = \sqrt{\sum_m \left(x_m - y_m\right)^2} \tag{4.5}$$

has already been used by Fujishima (1999) for the chord recognition task.

The **Itakura-Saito divergence** (Itakura & Saito (1968)) defined by

$$D_{IS}\left(\mathbf{x}|\mathbf{y}\right) = \sum_m \frac{x_m}{y_m} - \log\left(\frac{x_m}{y_m}\right) - 1 \tag{4.6}$$

was presented as a measure of the goodness of fit between two spectra and became popular in the speech community during the seventies. This is not a distance, since it is not symmetrical. It can therefore be calculated in two ways: $D\left(h_{k,n}\,\mathbf{c}_n|\mathbf{w}_k\right)$ will define *IS1*, while $D\left(\mathbf{w}_k|h_{k,n}\,\mathbf{c}_n\right)$ will define *IS2*.

The **Kullback-Leibler divergence** (Kullback & Leibler (1951)) measures the dissimilarity between two probability distributions. It has been widely used in particular in information theory and has given rise to many variants. In the present paper we use the generalized Kullback-Leibler divergence defined by

$$D_{KL}\left(\mathbf{x}|\mathbf{y}\right) = \sum_m x_m \log\left(\frac{x_m}{y_m}\right) - x_m + y_m. \tag{4.7}$$

Just like Itakura-Saito divergence, the generalized Kullback-Leibler divergence is not symmetrical, so that we can introduce two measures of fit: $D\left(h_{k,n}\,\mathbf{c}_n|\mathbf{w}_k\right)$ *(KL1)* and $D\left(\mathbf{w}_k|h_{k,n}\,\mathbf{c}_n\right)$ *(KL2)*.

4.1.3.2 Interpretation of the asymmetric measures of fit

While the Euclidean distance had already been used for the chord recognition task, the use of Itakura-Saito and Kullback-Leibler divergences is innovative. The non-symmetry of these divergences allows to define two variants (IS1 & IS2 and KL1 & KL2). This section aims to investigate the properties of these two variants and interpret them in our chord recognition context.

Figure 4.5 displays plots of the scalar versions of our measures of fit on $[0,1]\times[0,1]$. We see that the terms $D_{IS}(x|y)$ and $D_{KL}(x|y)$ take high values when x is close to 1 and y is close to 0. The *IS1* and *KL1* measures of fit being just sums of 12 of these terms, we can deduce that a high value of *IS1* or *KL1* would be obtained if, for at least one of the chromas, the first term $h_{k,n}\,c_{m,n}$ is way larger that the $w_{m,k}$ term. That is to say if the chroma does not belong to the chord template but is present in the chromagram frame. This means that the *IS1* and *KL1* measures of fit reject in priority chords whose null chromas are nevertheless present in the chroma vector.

By looking at the terms $D_{IS}(y|x)$ and $D_{KL}(y|x)$, we notice that they take high values when x is close to 0 and y is close to 1. Therefore, a high value of *IS2* or *KL2* is obtained when, for at least one of the chromas, $h_{k,n}\,c_{m,n}$ is way lower that $w_{m,k}$. That is to say if the chroma is present in the chord template but not in the chromagram frame. This means that the *IS2* and *KL2* measures of fit reject in priority chords whose notes are not all present in the chroma vector.

Toy examples

Let us check these assumptions on a very simple toy example. Let us suppose that we want to find a C major chord in a chromagram frame $\mathbf{x}$. The chord template can be written $\mathbf{y} = [1,\epsilon,\epsilon,\epsilon,1,\epsilon,\epsilon,1,\epsilon,\epsilon,\epsilon,\epsilon]$ with ϵ being a very small value used to avoid numerical instabilities.[2]

- Case 1: The chromagram frame is exactly the C major chord template

$$\mathbf{x} = [1,\epsilon,\epsilon,\epsilon,1,\epsilon,\epsilon,1,\epsilon,\epsilon,\epsilon,\epsilon]$$

[2] For example in MATLAB the lowest possible value is approximately 10^{-16}.

- Case 2 (extra note): The chromagram frame is a C major chord, with an extra D

$$\mathbf{x} = [1, \epsilon, 1, \epsilon, 1, \epsilon, \epsilon, 1, \epsilon, \epsilon, \epsilon, \epsilon]$$

- Case 3 (missing note): The chromagram frame is a C5 chord (only C and G)

$$\mathbf{x} = [1, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, 1, \epsilon, \epsilon, \epsilon, \epsilon]$$

	EUC	IS1	IS2	KL1	KL2
Case 1	0	0	0	0	0
Case 2	$1-\epsilon$ ~ 1	$\frac{1}{\epsilon}+\log(\epsilon)-1$ $\sim \frac{1}{\epsilon}$	$\epsilon-\log(\epsilon)-1$ $\sim -\log\epsilon$	$\epsilon-\log(\epsilon)-1$ $\sim -\log\epsilon$	$\epsilon\log(\epsilon)-\epsilon+1$ ~ 1
Case 3	$1-\epsilon$ ~ 1	$\epsilon-\log(\epsilon)-1$ $\sim -\log\epsilon$	$\frac{1}{\epsilon}+\log(\epsilon)-1$ $\sim \frac{1}{\epsilon}$	$\epsilon\log(\epsilon)-\epsilon+1$ ~ 1	$\epsilon-\log(\epsilon)-1$ $\sim -\log\epsilon$

Table 4.2: Expressions of the measures of fit on toy examples along with their equivalent when $\epsilon \to 0$.

Table 4.2 shows the expressions of the measures of fit calculated for each of these toy examples along with their equivalent when $\epsilon \to 0$. We observe that in Case 2, for a very small value of ϵ, the IS1 measure of fit tends to be very high, *IS2* and *KL1* are finite, and *KL2* is close to 1. This indicates that *IS1* strongly reacts to the presence of parasite notes in the chroma vector, while it does not make a big difference for *KL2*. On the contrary, in Case 3, the *IS2* measure of fit is really sensitive to the fact that all the notes within the chord template can be found in the chroma vector, while *KL1* is not too affected.

4.1.3.3 Scale parameters

As seen in Section 4.1.1, we can analytically calculate the scale parameters $h_{k,n}$ with Equation (4.2). Table 4.3 displays the expressions of these scale parameters for every measure of fit.

4.1.4 Post-processing filtering

So far, our chord detection is performed frame-by-frame without taking into account the results on adjacent frames. In practice, it is rather unlikely for a chord to last only one frame. Furthermore, the information contained in the adjacent frames can help decision (Shenoy et al. (2004)): it is one of the main advantages of the methods using HMM, where the introduction of transition probabilities naturally leads to a smoothing effect. Nevertheless, HMM-based methods assume an exponentially temporal distribution, which does not suit well the rhythmic structure of pop songs (Mauch & Dixon (2008)). We therefore propose, as a third block for our method, to use an *ad hoc* filtering process which implicitly informs the system of the expected chord duration. The post-processing filtering is applied upstream to the calculated measures. Note that the use of this filtering process is innovative, since it had been previously applied to chromagrams (Fujishima (1999), Peeters (2006), Bello & Pickens (2005)) or detected chord sequences (Bello & Pickens (2005)), but never to the recognition criterion itself.

We introduce new criteria $\tilde{d}_{k,n}$ based on L successive values centered on frame n (L is then odd). These $\tilde{d}_{k,n}$ are calculated from the $d_{k,n}$ previously described on the L adjacent frames, as shown below. In our system two types of filtering are tested.

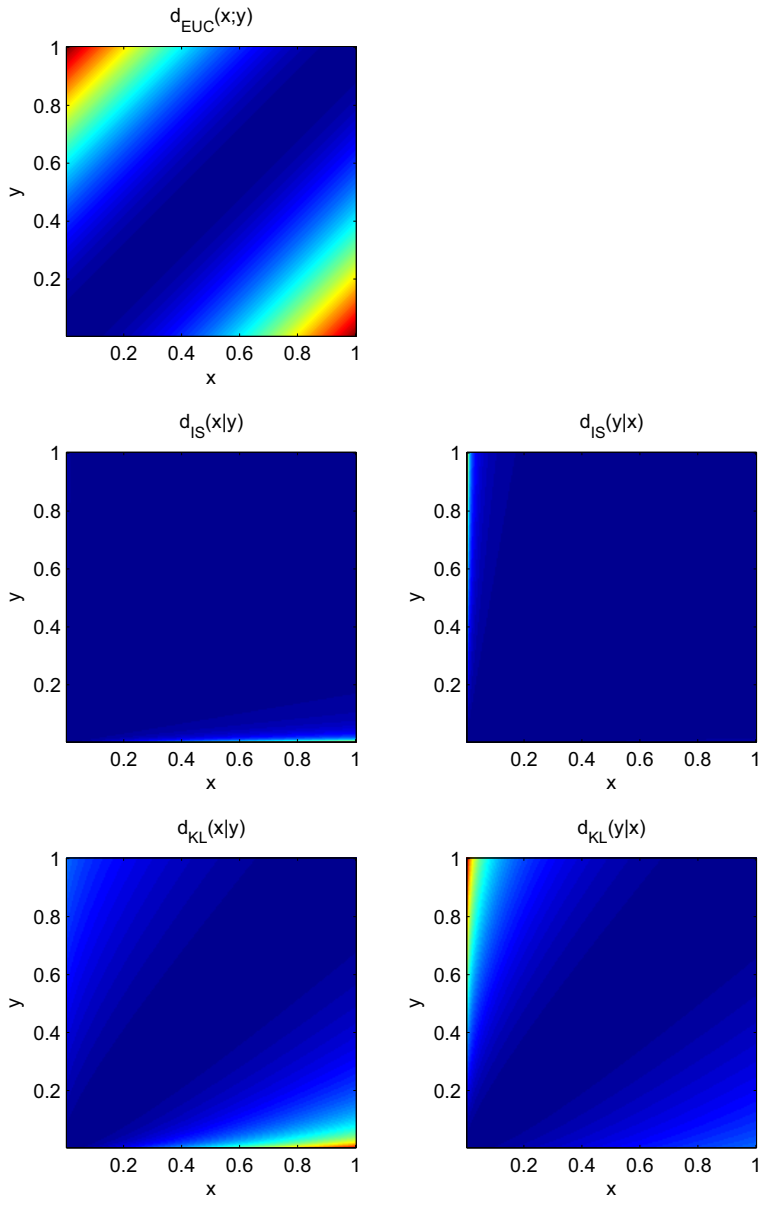


Figure 4.5: Plots of the considered measures of fit.

	Expression of $D\left(h_{k,n}\,\mathbf{c}_n;\mathbf{w}_k\right)$	Scale parameter $h_{k,n}$	Recognition criterion $d_{k,n}$
EUC	$\sqrt{\sum_m \left(h_{k,n}\,c_{m,n} - w_{m,k}\right)^2}$	$\frac{\sum_m c_{m,n}\,w_{m,k}}{\sum_m c_{m,n}^2}$	$\sqrt{\sum_m w_{m,k}^2 - \frac{\left(\sum_m c_{m,n}\,w_{m,k}\right)^2}{\sum_m c_{m,n}^2}}$
IS1	$\sum_m \frac{h_{k,n}\,c_{m,n}}{w_{m,k}} - \log\left(\frac{h_{k,n}\,c_{m,n}}{w_{m,k}}\right) - 1$	$\frac{M}{\sum_m \frac{c_{m,n}}{w_{m,k}}}$	$M\log\left(\frac{1}{M}\sum_m \frac{c_{m,n}}{w_{m,k}}\right) - \sum_m \log\left(\frac{c_{m,n}}{w_{m,k}}\right)$
IS2	$\sum_m \frac{w_{m,k}}{h_{k,n}\,c_{m,n}} - \log\left(\frac{w_{m,k}}{h_{k,n}\,c_{m,n}}\right) - 1$	$\frac{1}{M}\sum_m \frac{w_{m,k}}{c_{m,n}}$	$M\log\left(\frac{1}{M}\sum_m \frac{w_{m,k}}{c_{m,n}}\right) - \sum_m \log\left(\frac{w_{m,k}}{c_{m,n}}\right)$
KL1	$\sum_m h_{k,n}\,c_{m,n}\log\left(\frac{h_{k,n}\,c_{m,n}}{w_{m,k}}\right) - h_{k,n}\,c_{m,n} + w_{m,k}$	$e^{-\sum_m c'_{m,n}\log\left(\frac{c_{m,n}}{w_{m,k}}\right)}$ with $c'_{m,n} = \frac{c_{m,n}}{\lVert\mathbf{c}_n\rVert_1}$	$1 - e^{-\sum_m c'_{m,n}\log\left(\frac{c'_{m,n}}{w_{m,k}}\right)}$ with $c'_{m,n} = \frac{c_{m,n}}{\lVert\mathbf{c}_n\rVert_1}$
KL2	$\sum_m w_{m,k}\log\left(\frac{w_{m,k}}{h_{k,n}\,c_{m,n}}\right) - w_{m,k} + h_{k,n}\,c_{m,n}$	$\frac{1}{\sum_m c_{m,n}}$	$\sum_m w_{m,k}\log\left(\frac{w_{m,k}}{c'_{m,n}}\right) - w_{m,k} + c'_{m,n}$ with $c'_{m,n} = \frac{c_{m,n}}{\lVert\mathbf{c}_n\rVert_1}$

Table 4.3: Presentation of the measures of fit (the expressions assume $\lVert\mathbf{p}_k\rVert_1 = 1$).

The **low-pass filtering** defined by

$$\tilde{d}_{k,n} = \frac{1}{L} \sum_{n'=n-\frac{L-1}{2}}^{n+\frac{L-1}{2}} d_{k,n'} \tag{4.8}$$

tends to smooth the output chord sequence and to reflect the long-term trend in the chord changes.

The **median filtering** defined by

$$\tilde{d}_{k,n} = \text{median} \left\{ d_{k,n'} \right\}_{n-\frac{L-1}{2} \leq n' \leq n+\frac{L-1}{2}} \tag{4.9}$$

has been widely used in image processing and is particularly efficient to correct random errors. Furthermore, this filtering has the property of respecting transitions.

In every case, the detected chord $\hat{k}_n$ on frame n is the one that minimizes the set of values $\left\{ \tilde{d}_{k,n} \right\}_k$:

$$\hat{k}_n = \underset{k}{\text{argmin}} \ \tilde{d}_{k,n} \tag{4.10}$$

4.2 Experiments

With the 3 previously described blocks, we can build a large number of chord transcribers by choosing each time one measure of fit, one chord model, one post-processing filtering process and one neighborhood size. We have introduced:

- **5 measures of fit:** EUC, IS1, IS2, KL1 and KL2 ;
- **3 chord models:** 1, 4 or 6 harmonics ;
- **2 types of filtering:** low-pass and median filtering ;
- **12 neighborhood sizes (when post-processing is applied):** from $L = 3$ to $L = 25$.

This gives $5 \times 3 \times 2 \times 12$ (filtering) + 5×3 (no filtering) $= 375$ parameter sets which can be seen as as many chord recognition systems. We shall refer to these deterministic chord recognition methods as **DCR** methods.

4.2.1 Chromagram computation

Based on preliminary experiments, we chose among three types of chromagram (Bello & Pickens (2005), Peeters (2006), Zhu et al. (2005)), the one proposed by Bello & Pickens (2005), which appeared to give the best results for our chord transcription task. This chromagram is computed with the CQT (Brown (1991)) allowing a frequency analysis on bins centered on logarithmically spaced frequencies (see Section 2.1.2).

The signal is first downsampled to 5512.5 Hz and the CQT is calculated with $b = 36$ (3 bins per semi-tone), between frequencies 73.42 Hz (D2) and 587.36 Hz (D5). These parameters lead to a window length of 4096 samples (743 ms) and the hop size is set to 512 samples (93 ms).

Thanks to the 36 bins per octave resolution, a tuning algorithm (Harte & Sandler (2005)) can be used. After a peak detection in the chromagram, a correction factor is calculated so as

	no filtering			low-pass filtering			median filtering		
	1 harm.	4 harm.	6 harm.	1 harm.	4 harm.	6 harm.	1 harm.	4 harm.	6 harm.
EUC	0.665	0.636	0.588	0.710	0.684	0.646	0.705	0.679	0.636
IS1	0.665	0.441	0.399	0.706	0.460	0.415	0.706	0.465	0.422
IS2	0.657	0.667	0.170	0.704	0.713	0.178	0.703	0.714	0.178
KL1	0.665	0.487	0.140	0.700	0.532	0.151	0.692	0.498	0.143
KL2	0.667	0.672	0.612	0.709	0.712	0.648	0.714	**0.718**	0.656

Table 4.4: Average Overlap Scores obtained on the Beatles corpus. For sake of conciseness, we only display for each post-processing method the results for the optimal choice of L.

to take into account the detuning. A median filtering is finally applied in order to eliminate too sharp transitions.

Their implementation also performs a silence ('no chord') detection using an empirically set threshold on the energy of the chroma vectors. Some details about the calculation of the chromagram can be found in Bello & Pickens (2005). We used the code kindly provided by the authors.

4.2.2 First results

In order to investigate the differences between our 375 DCR systems, we have chosen to test them with one metric and one corpus. We calculate for each system the AOS obtained on the Beatles corpus by detecting major and minor chords.

Table 4.4 gives the overall AOS obtained for every chord recognition system. The best average result is obtained with *KL2*, the 4 harmonics chord model and a median filtering with $L = 15$ (2.04s) giving a recognition rate of 71.8%. In the following section, we shall refer to this method as **OGF1** ***(maj-min)***.

Interestingly, we notice that for the *EUC*, *IS1* and *KL1* measures of fit, the results worsen when we increase the number of harmonics. We propose here two explanations for these results. In the particular cases of *IS1* and *KL1*, this can be explained by the fact that they both contain a logarithm component which is sensitive to the zeros within chord templates. We have seen in Section 4.1.3.2 that these measures of fit categorize chords by comparing every null chroma within the chord template to its value in the chroma vector. Since chord models with high number of harmonics contain less null chromas (see Figure 4.2), the discrimination between chords is harder, which results in worse scores. A more general explanation for this phenomenon can be found by relating back to the notion of chord template itself. As such, a chromagram frame is very hard to characterize: it is supposed to only contain the notes played by the instruments, but in reality it also captures noise or drums. Furthermore, it also depends on instrument timbres and on the relative amplitudes of the played notes. The question is: what kind of templates should we use in order to only capture useful information from these chroma vectors? The exponential model introduced in harmonic-dependent templates is supposed to better suit the reality of music, but it also degrades the chord templates, by making less clear what they are supposed to capture. By introducing notes which are not explicitly present in the chord, the templates may detect notes which are actually due to noise, drums, or even melody. Indeed, our results show that the only cases where the harmonic chord templates perform well

are the *IS2* and *KL2* with 4 harmonics and still the differences between these scores and those obtained with only one harmonic are very small.

A pathological situation appears when using the Itakura-Saito divergences *IS1* and *IS2* with the 6 harmonics chord model. Indeed, we observe that the use of *IS2* with the 6 harmonics chord model leads to a systematic detection of minor chords, while the *IS1* measure with 6 harmonics chord model only detects major chords. In the case of the *IS1* the loss in scores is less noticeable, because of the high number of major chords in the Beatles corpus. We believe that the explanation of this phenomena lies in the structure of the 6 harmonics chord model. Indeed, the 6 harmonics chord model gives a different number of null components for the major and minor chords: we can see on Figure 4.2 that the major chord model has 6 null components while the minor chord has 5 null components. The recognition criterion associated to the *IS2* has the property that given a chroma vector, the more zeros in the chord template $\mathbf{w}_k$, the larger the value of the criterion. This measure of fit will therefore always give larger values for the chord models having more null components, that is to say the major chords, which leads to a systematic detection of only minor chords. The same phenomenon can be observed for the *IS1* measure of fit, this time with a systematic detection of major chords.

Both low-pass filtering and median filtering give good results: the low-pass filtering tends to smooth the chord sequence while the median filtering reduces the random errors. In most cases the optimal value of L lies between 13 and 19 which corresponds, with our window parameters, to a length of approximately 2 seconds.

Some songs give disappointing results ($<$0.100) with all parameter sets: it is often due either to a strong detuning which is too large to be corrected by the tuning algorithm present in the chromagram computation *(eg. Wild Honey Pie, Lovely Rita)*, or to untuned material such as spoken voice, non-harmonic instruments or experimental noises (applause, screams, car noise, etc.) *(eg. Revolution 9)*.

4.2.3 Influence of the parameters

We have defined 375 DCR systems, each of them composed of one different parameter set. We propose in this section to investigate the influence of these parameters on performances.

Influence of measure of fit

Let us first investigate the influence of the chosen measure of fit. With our protocol, each measure of fit defines 75 chord transcribers and consequently 75 Beatles AOS. On Table 4.5 we display, for each measure of fit, the maximum, minimum, mean and standard deviation of the corresponding set of 75 AOS.

The measures give close scores but differ in robustness. The most efficient and robust measures are *EUC* and *KL2*, as their standard deviation is low. This is probably due to the reaction of other measures of fit to the introduction of harmonics (see Section 4.2.2). On the contrary, for these two measures of fit, the standard deviation is low, which means that other parameters (harmonic number, filtering methods, neighborhood size) do not have a strong influence on the results and thus that the choice of the measure is here decisive.

Influence of harmonics

On Table 4.6 is presented an analysis of the 375 AOS according to the number of considered harmonics.

	Max	Min	Mean	Std. dev.
EUC	0.710	0.588	0.662	0.032
IS1	0.706	0.397	0.519	0.124
IS2	0.714	0.170	0.520	0.246
KL1	0.700	0.137	0.442	0.223
KL2	0.718	0.608	0.677	0.032

Table 4.5: Influence of measure of fit on the Average Overlap Scores obtained on the Beatles corpus. For each measure of fit, we present statistics on the scores obtained by the 75 so-defined chord transcribers.

	Max	Min	Mean	Std. dev.
1 harm	0.714	0.628	0.688	0.016
4 harm	0.718	0.436	0.604	0.106
6 harm	0.656	0.137	0.399	0.212

Table 4.6: Influence of chord model on the Average Overlap Scores obtained on the Beatles corpus. For each chord model, we present statistics on the scores obtained by the 125 so-defined chord transcribers.

The single harmonic and 4 harmonics chord models both give good results but only the single harmonic model gives robust performances. More generally, the introduction of higher harmonics leads to more variable results. This can be explained by the reaction of *IS1*, *IS2* and *KL1* to the introduction of harmonics, already described in Section 4.2.2.

Influence of filtering method

For each filtering method, we calculate the AOS improvement on the corresponding no-filtering method. Table 4.7 proposes some statistics on the improvements brought by the introduction of post-processing filtering.

	Max	Min	Mean	Std. dev.
low-pass	9.76%	-5.55%	4.37%	2.51%
median	8.05%	-2.81%	3.78%	2.28%

Table 4.7: Influence of post-processing filtering on the Average Overlap Scores obtained on the Beatles corpus. For each filtering method, we present statistics on the score improvements obtained by the 180 so-defined chord transcribers on the 15 corresponding no-filtering methods.

We have previously seen in Table 4.4 that acceptable results could be obtained by using either low-pass or median filtering. We can see here that indeed, filtering allows to improve the performances of the systems. According to these results, low-pass filtering seems to be more efficient but also less robust than median filtering. Interestingly, we have noticed that systems in which the introduction of post-processing does not improve the results correspond to pathological cases described in Section 4.2.2.

As far as neighborhood size is concerned, we notice that similar sizes give similar scores (see

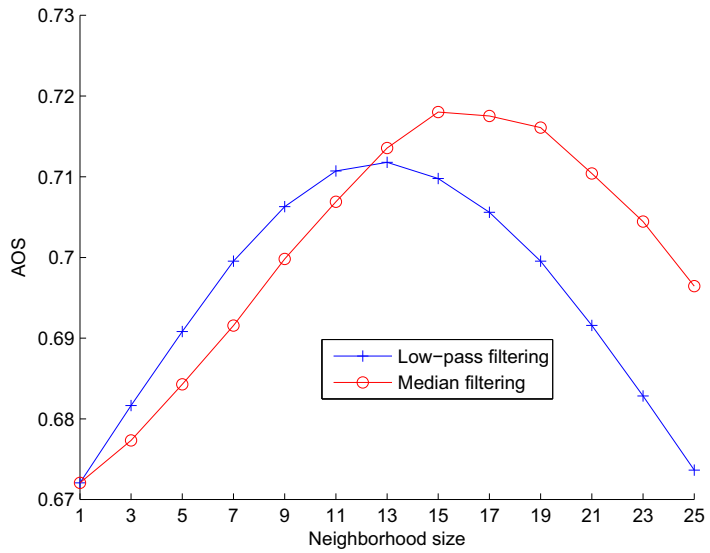


Figure 4.6: Influence of neighborhood size on the Average Overlap Scores obtained on the Beatles corpus with the KL2 measure of fit and the 4 harmonics chord model.

Figure 4.6). We see that even if an optimal size does exist, by choosing close neighborhood size, one does not change much the scores.

Joint influence of parameters

The analysis of all scores and parameters is complex due to the large number of parameter sets (375) and songs (180). In order to get a 2-D representation of all these DCR systems, we use the Principal Component Analysis (PCA) (Pearson (1901)). PCA is a method which consists in transforming some possibly correlated variables into a smaller number of uncorrelated variables. These new variables are called *principal components* or *axes*. It combines a geometrical approach (as it represents the variables in a new geometric space) and a statistical approach (as it looks for the independent axes which best explain the variability or variance of the data). More details on the PCA can be found in the box *What is Principal Component Analysis?* (p 76).

In our case, PCA is used in order to describe and visualize the results obtained with all our parameter choices. Figures 4.7 & 4.8 display the PCA computed from the Overlap Scores table: each of the 375 points within the graphic represents one DCR system. Colors allow to identify and label each group of points.

The first remark is that systems seem to group together in constellations: by looking at labels, we notice that the constellations contain systems composed of same measure of fit and chord model, and therefore only having different post-processing. The choice of the measure of fit and chord model seems to be more determinant than the choice of the post-processing method.

Also, we can see that 8 groups of DCR systems seem to be pretty close: the 5 groups of systems using the single harmonic chord model, and the EUC, IS2 and KL2 systems with 4 harmonics. The differences between these DCR systems does not seem to be really important. Contrarily, the introduction of harmonics seem to spread the constellations: it is true in particular for the pathological cases described in Section 4.2.2.

What is Principal Component Analysis? (Pearson (1901))

Let $\mathbf{X}$ be the $S \times I$ matrix containing all the scores. I stands for the number of chord recognition systems, and S for the number of songs within the corpus. Each observed chord recognition system is thus defined by S variables. As such, it is impossible to visualize all the scores on the same plot: we therefore want to reduce the number of variables from S to 2 or 3.

Each row $\mathbf{x}_s^t$ (song) of the matrix can be characterized by:

- a mean $\mu_s = \frac{1}{I}\sum_{i=1}^{I} x_{s,i}$;
- a standard deviation $\sigma_s = \sqrt{\frac{1}{I}\sum_{i=1}^{I}(x_{s,i} - \mu_s)^2}$.

We know that songs influence the scores contained in the matrix: some songs might be harder to transcribe and therefore only get bad scores. In order to take this into account, the matrix $\mathbf{X}$ is transformed into a matrix $\tilde{\mathbf{X}}$ where each row of the matrix is centered and standardized:

$$\tilde{\mathbf{x}}_s = \frac{\mathbf{x}_s - \mu_s}{\sigma_s}$$

. The $S \times S$ correlation matrix $\mathbf{V}$ is then calculated as $\mathbf{V} = \frac{1}{I-1}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathbf{t}}$. It is a square, symmetrical and real matrix: it can therefore be diagonalized. We can find two matrix $\mathbf{P}$ and $\mathbf{D}$ such as:

- $\mathbf{P}^{-1}\mathbf{V}\mathbf{P} = \mathbf{D}$.
- $\mathbf{D}$ is a $S \times S$ diagonal matrix containing the eigenvalues of matrix $\mathbf{V}$ in decreasing order.
- $\mathbf{P}$ is a $S \times S$ matrix containing the eigenvectors of matrix $\mathbf{V}$.

The projection of matrix $\mathbf{X}$ on the new uncorrelated variables is:

$$\mathbf{Y} = \tilde{\mathbf{X}}^t\mathbf{P}$$

.

Let suppose that we want to represent the data on M dimensions, with $M < S$. Then, the cumulative energy content g for the M^{th} eigenvector is the sum of the energy content across all the eigenvalues from 1 through M:

$$g = \sum_{i=1}^{M} d_{i,i}$$

. In practice we want g to be high enough so that the graphical display is relevant enough. When M is chosen, $\mathbf{y}^i = [y_{i,1}, \ldots, y_{i,M}]$ represents the coordinates of the i^{th} chord recognition system in the new space.

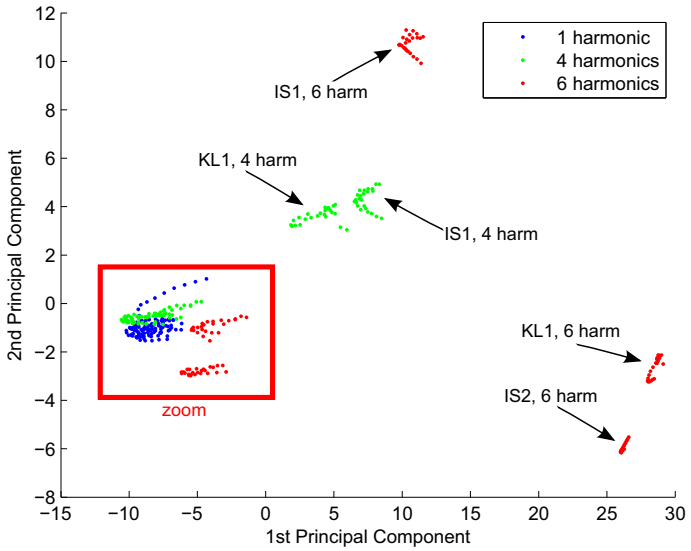


Figure 4.7: Principal Component Analysis of the Average Overlap Scores obtained on the Beatles corpus.

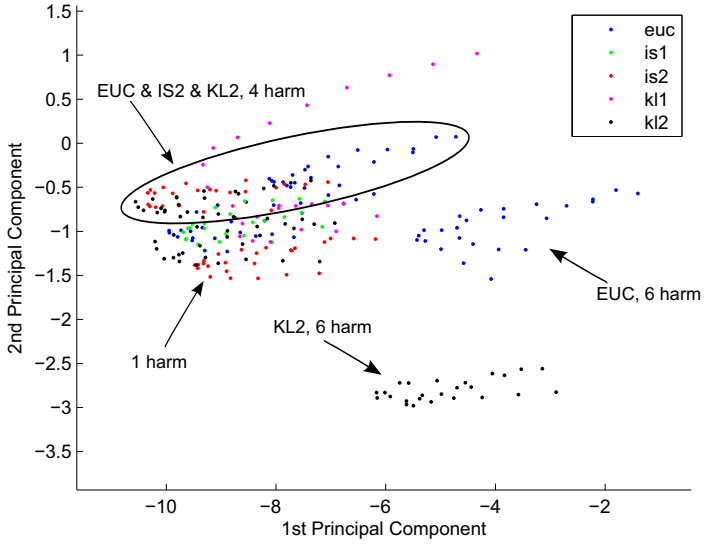


Figure 4.8: Zoom on the Principal Component Analysis of the Average Overlap Scores obtained on the Beatles corpus.

4.2.4 Study of the recognition criterion

In order to understand how our OGF1 *(maj-min)* method works, we propose here to study an example: we chose the Beatles song *Eight days a week* from the album *Beatles for Sale*. Figure 4.9 displays both the distance matrix containing the recognition criteria $d_{k,n}$ and the final chord transcription. Note in passing that the transcription is satisfactory since we reach a 87% recognition score. We can see on the top figure that the dark blue zones on the distance matrix (low values) correspond most of the time to the chord present in annotation files (in pink).

Another noticeable thing is that on a given frame, several chords have a dark blue or blue color on the distance matrix. For example, by carefully looking at the $Bminor$ chord present at approximately 63s, we can see that 3 chords seem to have a very low distance value: $Bmajor$, $Gmajor$ and $Dmajor$. We know that the chord recognition system, as described before, picks for every frame the chord minimizing the set of criteria. Nevertheless, it seems that this choice is not that obvious since several chords have a low criterion value. Thus, the chord recognition system chooses between several potential *chord candidates*. We can ask ourselves whether these candidates are relevant or not: for example, what would happen if we chose the second or the third smallest instead of the smallest?

Table 4.8 shows the AOS obtained on the Beatles corpus by choosing for every frame the i^{th} candidate, that is to say, the chord having the i^{th} lowest criterion value. The *silences* columns represents the scores obtained by only detecting silences. The theoretical AOS obtained by considering i candidates is the sum of the AOSs obtained by considering the i first candidates and the silences. We can see that for example, by selecting 3 candidates we get an AOS of 0.883 and 0.925 with 5 candidates. This shows that these *chord candidates* are not random and that they are actually relevant. Also, the AOS seems to quickly decrease with the candidate rank. This means that when our system is wrong on one frame, the correct chord is most of the time the second or the third candidate.

silences	1st	2nd	3th	4th	5th
0.018	0.700	0.116	0.049	0.026	0.016

Table 4.8: Average Overlap Scores obtained on the Beatles corpus by choosing for every frame the chord having the i^{th} lowest criterion value.

Let us look carefully at top plot of Figure 4.9, and more specifically at the area around 63s corresponding to a $Bminor$ chord. We have brought out 3 chords with low criterion values: $Bmajor$, $Gmajor$ and $Dmajor$. Interestingly, when referring to the definitions of these chords, we notice that all these chords have two notes in common with the detected chord $Bminor$. They are respectively the *parallel*, *submediant* and *mediant* chords (see Chapter 1 for a reminder of the names given to chord relationships). Is this situation due to the characteristics of the song or is this phenomenon common? Do first candidates have particular relationships with the detected chord?

Table 4.9 presents some statistics on the relationships between chord candidates and detected chords. We can see that, at least for the second and third candidate, the relationship with the detected chord often belongs to the 3 types of relationships we have brought out from our example. In particular, one interesting result is that the second candidate is in almost half of the time the parallel of the first candidate. Yet, this rule becomes less true when considering higher candidates.

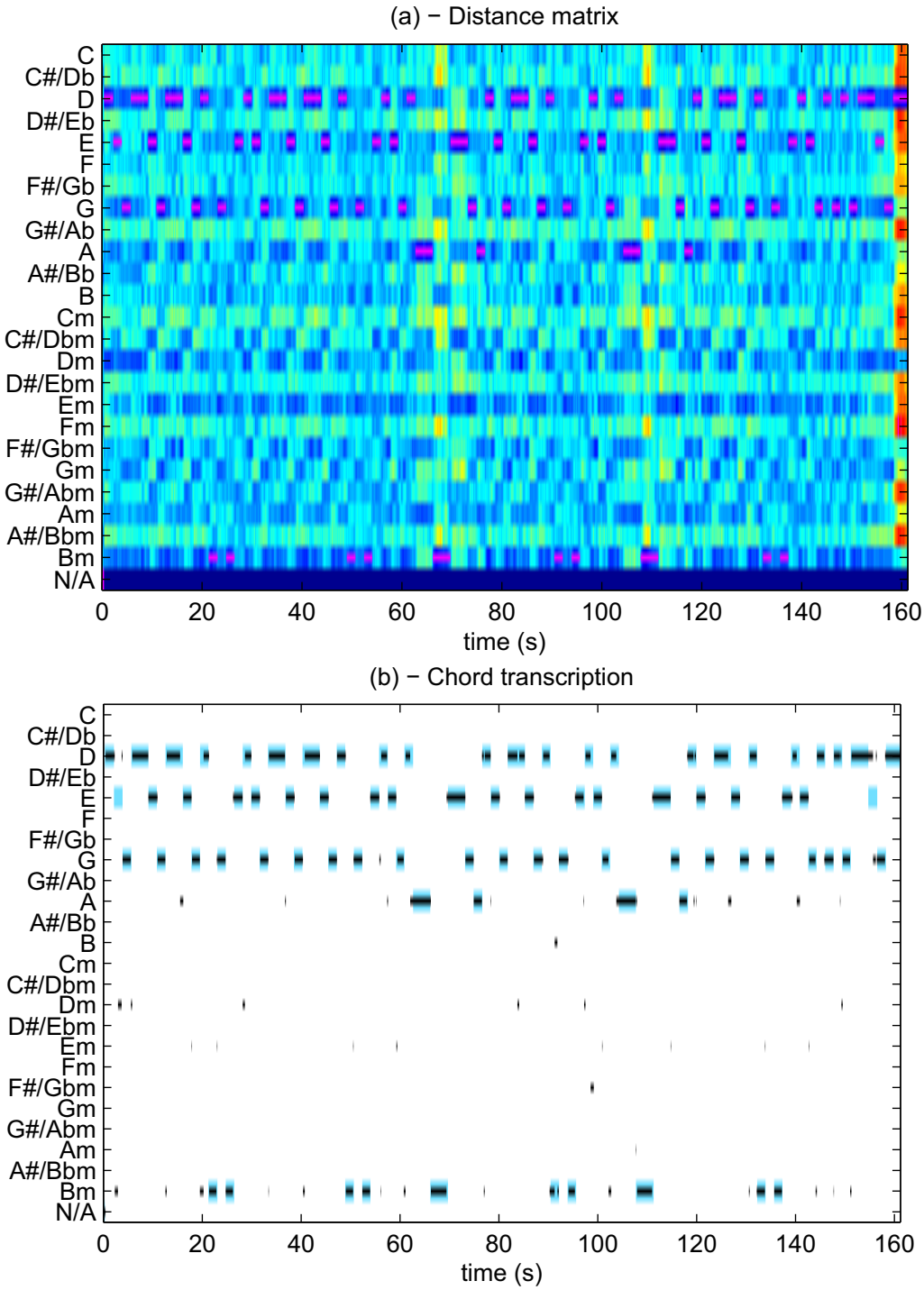


Figure 4.9: Recognition criteria and chord transcription of the Beatles song *Eight days a week* with the OGF1 *maj-min* chord recognition system. (a) - The ground true transcription is represented in pink. (b) - The estimated chord labels are in black while the ground-truth chord annotation is in light blue.

	2nd	3th	4th	5th
parallel	45.3	19.9	14.7	7.9
relative (submediant)	22.1	25.3	15	12.4
mediant	15.3	17.1	17.1	12.2
others	30.8	37.7	53.2	67.5

Table 4.9: Statistics on the relationship between chord candidates and detected chords on the Beatles corpus (in percentage).

4.2.5 Additional experiments

In this section, we propose to describe some experiments which were conducted during the development of our DCR systems. While some of them did not improve the results, they help to understand what makes the DCR methods work.

4.2.5.1 Introduction of extra chord types

The simplicity of our approach allows to easily introduce chord templates for chord types other than major and minor. We study here the influence of chord types on the performances of our DCR systems. The choice of these chord types is guided by the statistics on the corpus previously presented in Section 3.1.1. We introduce in priority the most common chords types of the corpus. Note that, according to our evaluation framework, once the chords have been detected with their appropriate model, they are then mapped to the major or minor type following the rules already used for the annotation files (described in Table 3.1).

Results

In the Beatles corpus, the two most common chord types other than major and minor are dominant seventh *(7)* and minor seventh *(min7)* chords. The results obtained by considering major, minor, dominant seventh and minor seventh chords are presented in Table 4.10.

Chord types	AOS	Optimal parameters
maj-min	0.718	KL2, 4 harm, median, L=15 (2.04s)
maj-min + 7	0.724	KL2, 1 harm, median, L=17 (2.23s)
maj-min + 7 + min7	0.706	IS1, 1 harm, low-pass, L=13 (1.86s)

Table 4.10: Average Overlap Scores obtained on the Beatles corpus with major, minor, dominant seventh and minor seventh chords.

The best results are obtained by detecting major, minor and dominant seventh chords, with the *KL2* measure of fit, the single harmonic chord model and the median filtering with $L = 17$ (2.23s) giving a recognition rate of 72.4%. We shall later refer to this chord recognition system as **OGF2** ***(maj-min-7)***. Only the introduction of dominant seventh chords, which are very common in the Beatles corpus, enhances the results. The introduction of minor seventh chords, which are less common, degrades the results. Indeed, the structure of a minor seventh chord (for example $Cminor7$) leads to confusion between the actual minor chord and its relative major chord ($Eb major$ in our example).

Augmented and diminished chords have been considered in several template-based chord recognition systems (Fujishima (1999),Harte & Sandler (2005)). Interestingly, while the augmented and diminished chords are very rare in the Beatles corpus (respectively 0.62% and 0.38% of the total length), the introduction of augmented and diminished chords does not degrade the results. We obtain an AOS of 0.702 by considering major, minor, augmented and diminished chords and of 0.724 by taking into account major, minor, dominant seventh, augmented and diminished chords.

The introduction of other chord types (ninth, major seventh, sus4, etc.) does not improve the results. This can be explained either by some chord structures which can lead to confusions with other chord types or by the low number of chords of these types in the Beatles corpus. Indeed, the introduction of a new chord type within our system might help to detect chords of this type but it also leads to new errors such as false detections. Therefore only frequent chords types should be introduced, ensuring that the enhancement caused by the better recognition of these chord types is larger than the degradations caused by false detections.

Study of the maj-min-7 method

We have previously seen that the introduction of dominant seventh chords allows to improve the results. According to our evaluation protocol, when a dominant seventh chord is detected, it is mapped to the associated major chord. Therefore, it is difficult to figure out whether the detection of dominant seventh chords improves the results because of the proper detection of dominant seventh chords or just because it increases the number of detected major chords, which are very common in the Beatles corpus (see Section 3.1.1 for a description of this corpus).

In order to investigate this, let us ask ourselves: when a dominant seventh is detected with our OGF2 *(maj-min-7)* method, what is the corresponding chord in the annotation files? Results show that when our OGF2 *(maj-min-7)* detects a chord as dominant seventh, it is right in only 9.2% of the cases. Most of the time (53.8%) the ground-truth chord is in reality the associated major chord. The dominant seventh template therefore helps the detection of major chords. It means that often, when our OGF2 *(maj-min-7)* gives better results that our OGF1 *(maj-min)* method, it is due to the detection of major chords which would not have been detected with the OGF1 *(maj-min)* method. Indeed, the percentage of major chords obtained is 70% with OGF1 *(maj-min)* and 85% with OGF2 *(maj-min-7)*, while it is 76.4% in the Beatles corpus.

This assumption is confirmed when looking at the album statistics. Indeed, we have seen in Section 3.1.1 that the last six albums of the Beatles contain many chords which are neither major, minor or dominant seventh. Yet, results show that the improvement of our OGF2 *(maj-min-7)* on our OGF1 *(maj-min)* method is particularly important on these albums. This can be caused by the fact that our OGF2 *(maj-min-7)* can detect as major these chords which are neither major, minor or dominant, but will later be mapped into major during the evaluation.

4.2.5.2 Introduction of beat information

The filtering process we have been applying so far uses a fixed and predetermined length. It seems interesting to introduce beat information in our system, so as to better take into account the rhythmic structure of music. We see two ways of doing it: either in the chromagram computation or in the post-processing applied to the recognition criteria. For our tests, we used the beat-detection algorithm provided by Davies & Plumbley (2007). The baseline system used here is the OGF1 *(maj-min)* system, using KL2 and the 4 harmonics chord model.

- **Experiment 1:** The first way to take into account the beat information is to compute a beat-synchronous chromagram. That is to say averaging the chromagram over the number of frames representing a beat time (with either low-pass or median filtering). This process has already been used for example by Bello & Pickens (2005). Since some beat-detection algorithm misevaluate the tactus and tatum times, we also propose to average the chromagram on 2 beat times. In this case, we need to distinguish two cases, depending if the averaging begins on the first or the second beat. The position of the first beat can be seen as a *beat phase*: we shall therefore test these two beat phases. Note that in this experiment, no post-processing filtering is applied on the recognition criteria, since all the beat information is supposed to be contained in the chromagram.
- **Experiment 2:** The second way to integrate the beat information is to filter the recognition criteria (either with the low-pass or the median filtering method) with a neighborhood size equal to the beat time. Just like in the first experiment, we shall also try to average the criteria on 2 beat times.
- **Experiment 3:** Our third proposal is to combine the use of beat-synchronous chromagrams with our classical post-processing filtering.

Table 4.11 presents the results of Experiments 1 & 2. When the chromagram or the recognition criterion is averaged on 2 beat times, there are 2 beat phases: the *optimal* column displays the score obtained by choosing for every song the optimal beat phase. We can see on this table that the introduction of beat information, either on the chromagram or on the distance matrix, always degrades the results. As far as Experiment 3 is concerned, the best AOS is 0.700, which is lower than the result obtained without using beat information. We believe that these disappointing results are probably due to the fact that the beat detection does not take into account the distinction between on-beats and off-beats. Indeed, the chord change tend to occur mainly on the on-beats and not on every beat. Averaging either the chromagram or the recognition criteria on every beat does not really capture the rhythmic information. Also, the averaging process removes some of the redundancy which was necessary to perform an accurate transcription.

Beat number		1	2		
Beat phase		N/A	1	2	optimal
Experiment 1: Chromagram	low-pass	0.685	0.651	0.672	0.705
	median	0.674	0.642	0.659	0.696
Experiment 2: Distance matrix	low-pass	0.687	0.660	0.677	0.712
	median	0.681	0.652	0.668	0.703

Table 4.11: Average Overlap Scores obtained on the Beatles corpus by taking into account beat information.

4.2.5.3 MIDI & Quaero corpora: influence of music genre

Until now, we only displayed results for the Beatles corpus. Despite the fact that the Beatles have experimented various genres and music styles, there is still a possibility that the scores are biased. In particular, we can wonder whether the system parameters, which were chosen for one particular corpus, can still apply to other corpora. Furthermore, it is the occasion to figure out if our DCR systems are style-dependent and if they work properly on other music

		maj-min		maj-min-7	
		Default	Optimal	Default	Optimal
Country	Ring of fire	0.844	0.918	0.848	0.924
	Tennessee waltz	0.941	0.955	0.949	0.955
	Stand by your man	0.895	0.909	0.902	0.911
Pop	Dancing queen	0.786	0.804	0.728	0.782
	I drove all night	0.870	0.891	0.856	0.889
	Born to make you happy	0.867	0.892	0.861	0.892
Blues	Blues stay away from me	0.630	0.791	0.854	0.912
	Boom, boom, boom	0.839	0.903	0.876	0.913
	Keep it to yourself	0.771	0.909	0.907	0.928
Rock	Twist and shout	0.827	0.892	0.850	0.901
	Let it be	0.835	0.876	0.876	0.880
	Help !	0.918	0.920	0.899	0.918
Total		**0.835**	**0.888**	**0.867**	**0.900**

Table 4.12: Overlap Scores for the 12 songs of the MIDI corpus.

types. We have therefore run our two DCR systems (OGF1 *(maj-min)* and OGF2 *(maj-min-7)*) on two new corpora containing various types of music (described in Chapter 3).

Table 4.12 shows the Overlap Scores obtained on the 12 songs of the MIDI corpus while Table 4.13 displays the scores on the Quaero corpus. On the MIDI table, besides the results obtained with the default parameters, we also displayed the results with the optimal parameters in order to evaluate the fitness of our default parameters.

Let us first focus on the MIDI results. The first thing we can observe is that the scores obtained with the default parameters are rather close to the optimal ones. This shows that the parameters we have deduced from the Beatles corpus can be used in a more general context. We can also see that the scores are all creditable. This can surely be explained by the fact that we work here with resynthesized wave files and not real audio. These audio files are indeed generated with instrument patterns which contain less noise and untuned material than real instrument recordings. Also, genre does not seem to have an influence on the scores. Nevertheless, the scores obtained on country songs are particularly large, but it is probably due to the very simple chord structures of these songs (mainly alternation of 3 chords).

As far as the Quaero results are concerned, we can see that the AOS obtained on this corpus are actually very close to those obtained on the Beatles corpus. Although they are variations in the scores, they seem to depend on the song rather than on the style: for example, the Justin Timberlake's and Mariah Carey's songs, which both can be classified as R&B have very different scores. Our systems do not seem to be influenced by the genre or style of the song but rather by the inherent difficulty of each song to be transcribed. Just like the MIDI corpus, this new corpus also enables to investigate the relevance of the parameters determined on the Beatles corpus. For the OGF1 *(maj-min)* method, the AOS obtained with the optimal parameters is 0.709 (against 0.706 with the default parameters) and for the OGF2 *(maj-min-7)*,

Song	Artist	maj-min	maj-min-7
Breathe	Pink Floyd	0.834	0.782
Brain Damage	Pink Floyd	0.828	0.939
I'm in love with my car	Queen	0.513	0.622
Chan chan	Buenavista Social Club	0.601	0.574
De camino a la vereda	Buenavista Social Club	0.732	0.832
Son of a preacher man	Dusty Springfield	0.929	0.948
Cryin	Aerosmith	0.774	0.788
Pull together	Shack	0.618	0.498
Kingston town	UB40	0.771	0.634
This ain't a scene, it's an arms race	Fall Out Boy	0.594	0.528
Say it right	Nelly Furtado	0.660	0.443
...Comes around	Justin Timberlake	0.688	0.648
Touch my body	Mariah Carey	0.358	0.239
Waterloo	ABBA	0.723	0.804
Believe	Cher	0.696	0.604
Another day in paradise	Phil Collins	0.666	0.581
Don't let me be misunderstood	Santa Esmeralda	0.676	0.653
Fox on the run	Sweet	0.763	0.808
Words	FR David	0.863	0.866
Orinoco flow	Enya	0.860	0.851
Total		**0.706**	**0.682**

Table 4.13: Overlap Scores for the 20 songs of the Quaero corpus.

the optimal score is 0.695 (while it is 0.682 with the default parameters). We can draw the same conclusion than on the MIDI corpus: the parameter choice made on the Beatles corpus is relevant on other corpora. This tends to show that our systems are not strongly overfitted with the Beatles corpus.

4.2.5.4 MIDI corpus: influence of the removal of drums

Our DCR methods strongly rely on chromagram, which is a harmonic music representation. We can therefore think that inharmonic components, such as drums, tend to add noise to the chromagram, which can lead to errors in the chord detection.

Working with audio data computed from MIDI files gives us the chance to synthesize them without the percussive parts. Indeed, the software Timidity ++ allows to mute one channel (instrument) for the wave-synthesis of the MIDI file.

The same simulations have been performed with these drum-free audio files. The removal of the percussions does not improve significantly the Overlap Scores. Indeed, the average score improvement is only 0.8% as well with the OGF1 *(maj-min)* system than with the OGF2 *(maj-min-7)*. We believe that the noise contained in the chromagram, which leads to errors, is not only due to drums but also, for example, to the melody itself, since it does not only play notes contained in the chord template.

4.3 Comparison with the state-of-the-art

The timeline of this PhD work has been based on the MIREX evaluations calendar. In this manuscript, we will display the results according to this sequential principle. As a consequence, the results obtained by our two approaches (deterministic and probabilistic) will be presented separately. The systems presented in this chapter have been developed in 2008-2009 when the state-of-the-art methods were the ones proposed for MIREX 2008. We will only compare our DCR methods to chord recognition systems available at this stage. A comparison of our systems to MIREX 2009 methods will be presented in the next chapter.

Furthermore, since an evaluation with a large set of metrics will be proposed in the next chapter, we chose only to display one evaluation metric in this chapter: the Average Overlap Score (AOS).

4.3.1 State-of-the-art

For our comparison, we have chosen methods among the top methods proposed in MIREX 2008, which did not contain any explicit training step (most of them are actually pre-trained on the Beatles corpus) and whose authors allowed us to use their original implementations. More details on these methods can be found in Chapter 2.

- **BP**: Bello & Pickens (2005) use 24-states HMM with musically inspired initializations, Gaussian observation probability distributions and EM-training for the initial state distribution and the state transition matrix.
- **RK**: Ryynänen & Klapuri (2008a) use 24-states HMM with observation probability distributions computed by comparing low and high-register profiles with some trained chord profiles. EM-training is used for the initial state distribution and the state transition matrix.

	Beatles corpus		Quaero corpus	
	AOS	Time	AOS	Time
OGF1 *(maj-min)*	0.718	790s	0.706	95s
OGF2 *(maj-min-7)*	**0.724**	796s	0.682	97s
BP	0.707	1619s	0.699	261s
RK	0.705	2241s	**0.730**	350s
KO	0.663	1668s	0.503	255s
PVM	0.647	12402s	0.664	2684s

Table 4.14: Comparison with the state-of-the-art on the Beatles and Quaero corpora.

- **KO**: Khadkevich & Omologo (2008) use 24 HMMs: one for every chord. The observation probability distributions are Gaussian mixtures and all the parameters are trained through EM.
- **PVM**: Pauwels et al. (2008) use a probabilistic framework derived from Lerdahl's tonal distance metric for the joint tasks of chords and key recognition.

We tested the methods on the Beatles and Quaero corpora. Results of this comparison with the state-of-the-art are presented on Table 4.14.

First of all, it is noticeable that all the methods give rather close results on the Beatles corpus: there is only a 8% difference between the methods giving the best and worse results. This is likely to be partially due to the fact that this database is the largest annotated database available and is commonly used by all researchers working on chord recognition, either to tune some parameters, train their methods or simply to test if their method works. In particular, the RK & KO methods are both pre-trained on the Beatles corpus. We can observe that our two DCR methods give the best AOS on this corpus. Yet, since all the scores are close, we propose to perform a Friedman and a Tukey-Krammer test (see Section 3.3 for details), in order to figure out whether there are significant differences between the tested chord recognition methods. Results are presented on Figure 4.10: it appears that our DCR method OGF1 *(maj-min)* is significantly better from two other tested methods (KO & PVM), while our OGF2 *(maj-min-7)* method is significantly better than three methods (RK, KO & PVM).

Interestingly, results on the Quaero corpus are more contrasted: in particular KO gives lower scores while RK obtain here the best results. This result is actually surprising, since both these methods use models which are trained on Beatles data and yet they respond really differently to the new corpus. It is interesting to see that our DCR methods, while not giving the best scores anymore, still performs well on this corpus: our OGF1 *(maj-min)* method indeed gives the second best result on this corpus. These good performances show that our parameters choice, while undeniably being optimal for the Beatles corpus, also fits well other genres or styles of music.

Our DCR methods are also characterized by a very low computational time. They are indeed twice as fast as the best state-of-the-art method (Bello and Pickens).

4.3.2 Analysis of the errors

In most chord transcription systems, the errors are often caused by the harmonic proximity or the structural similarity (common notes) between the real chord and the wrongly detected chord.

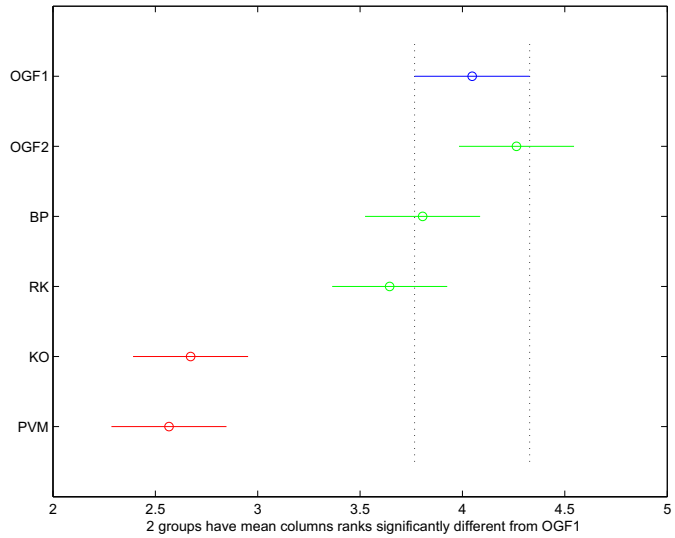


Figure 4.10: Tukey-Kramer's test performed on Overlap Scores calculated on the Beatles corpus. On this figure, the x-axis shows the average rank of each chord recognition method (previously denoted as $\bar{r}_{\cdot,i}$) along with its confidence interval. Ranks are presented in ascending order and two mean ranks are significantly different if their confidence intervals are disjoint (see Section 3.3).

Harmonic proximity

Errors can be caused by the harmonic proximity between the original and the detected chord, that is to say chords which have a particular relationship involving harmony, key, etc. In Section 1.1.4, we have presented the doubly nested circle of fifths which represents the major chords (capital letters), the minor chords (lower-case letters) and their harmonic relationships. The distance linking two chords on this doubly nested circle of fifths is an indication of their harmonic proximity. Given a major or minor chord, the 4 closest chords on this circle are the relative (submediant), mediant, subdominant and dominant (see Table 1.2 for a reminder on the names given to the chord relationships).

Structural similarity

As seen in Section 4.2.4, two chords are also likely to be mistaken one for another when they *look alike*, that is to say, when they share notes (especially in template-based systems). Given a major or minor chord, there are 3 chords which have 2 notes in common with this chord: the parallel minor/major, the relative minor/major (or submediant) and the mediant chord. Note that these two last chords are also harmonically close to the original chord. These types of errors are very likely to occur in template-based methods, which does not extract harmonic structure information from the music piece.

We have therefore brought out 5 potential sources of errors among the 23 possible ones (i.e., the 23 other wrong candidates for one reference chord). Some of these errors seem specific to the template-based systems, while the others seem to apply to every chord recognition system.

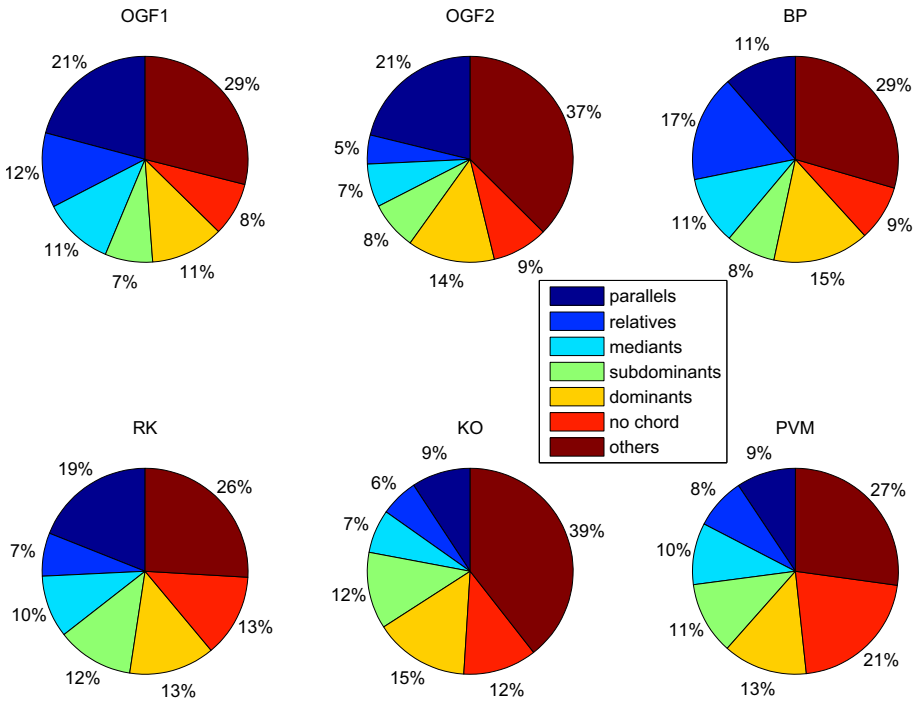


Figure 4.11: Error distribution on the Beatles corpus (as a percentage of the total number of errors).

Figure 4.11 displays the distribution of these error types as a percentage of the total number of errors for every evaluated method. Errors due to the bad detection of the 'no chord' states are represented with the 'no chord' label.

The main sources of errors correspond to the situations previously described and to the errors caused by silences ('no chord'). Actually, in most methods, the 5 types of errors previously considered (over the 23 possible ones) represent more than half of the errors. Note that for example, the PVM system is clearly mostly penalized by the wrong detection of these 'no chord' states.

As expected, our DCR methods have the largest parallel errors percentage (with the exception of the RK method), which is probably due to the fact that we work with chord templates and therefore do not detect any harmonic features such as key, harmony, chord vocabulary, etc. Furthermore, for our DCR methods (and the RK method), the percentages of errors due to structural similarity and harmonic proximity are the same, while for all other tested methods, the proportion of harmonic proximity errors is larger. It is actually very surprising that RK has an error distribution very close to our OGF1 *(maj-min)* method, while being very different and based on opposed principles.

Among interesting results, one can notice that the introduction of the dominant seventh chords clearly reduces the proportion of the errors due to relative (submediant) and mediant (-11%).

4.3.3 MIREX 2009

Our DCR methods have taken part in the MIREX 2009 evaluation in the Audio Chord Detection task for pre-trained systems. The evaluation corpus was not only composed of Beatles songs

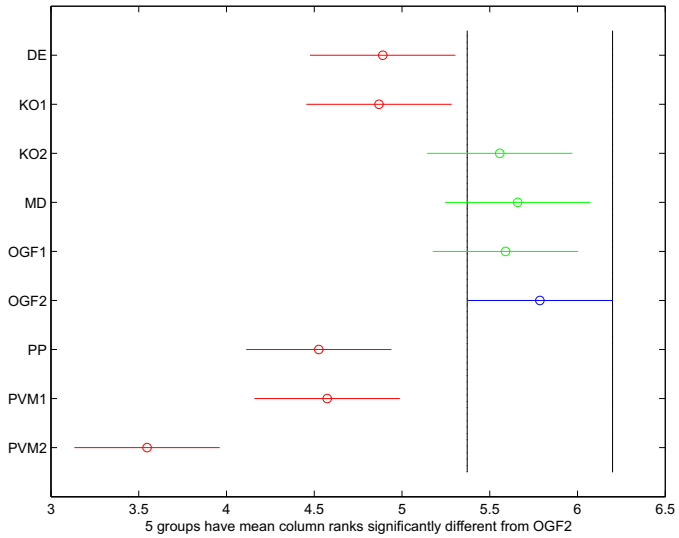


Figure 4.12: Tukey-Kramer's test performed on Overlap Scores of MIREX 2009. On this figure, the x-axis shows the average rank of each chord recognition method (previously denoted as $\bar{r}_{\cdot,i}$) along with its confidence interval. Ranks are presented in ascending order and two mean ranks are significantly different if their confidence intervals are disjoint (see Section 3.3).

but also of 36 songs from Queen and Zweieck. The metrics used for evaluation were the WAOS and the WAROS. Tables 4.15 & 4.16 give the results published for MIREX 2009.[3] While once again the scores are very close, our OGF2 *(maj-min-7)* and OGF1 *(maj-min)* methods gave respectively the second and fourth best WAOS for the major-minor chord recognition task and reached the first and second places for the root detection task (WAROS).

As always when scores are very close, it is interesting to perform a significant difference test. Figure 4.12 displays the results of the Friedman's and Tukey-Kramer's test: it shows that our OGF2 *(maj-min-7)* method is significantly better than 5 other tested chord recognition methods (DE, KO1, PP, PVM1 & PVM2). As far as our OGF1 *(maj-min)* method is concerned, the test shows that it is significantly different from 3 methods (PP, PVM1 & PVM2).

One very important question about such evaluations concerns the influence of training. Indeed, on this task, the trained systems (i.e. systems trained on $\sim \frac{2}{3}$ of the test corpus and tested on $\sim \frac{1}{3}$) gave better scores that pre-trained systems (i.e. systems either pre-trained or not using training): the best trained system (Weller et al. (2009)) obtained a WAOS of 0.742 and a WAROS of 0.777. Although one can draw the conclusion than trained methods are better, we can see that it is not that simple: indeed, our OGF2 *(maj-min-7)* method, which belongs to the pre-trained category, gives the same WAROS and is not using explicit training on Beatles data. Another interesting thing about this evaluation is the introduction of non-Beatles data on the corpus. The release of Christopher Harte's annotation files is clearly a major contribution for MIR researchers that has enabled and helped the development of chord transcriptions methods. Nevertheless, since all chord recognition systems use this database as a development corpus, either to tune some parameters, train their methods or simply to test if

[3] Results by Harte & Sandler (2009) were not included since they are only partly available on MIREX website, and Rocher et al. (2009)'s systems were not included because of some errors in algorithm runs.

	Beatles	Others	Total
Ellis (2009) (DE)	0.710	0.645	0.697
Khadkevich & Omologo (2009a) (KO1)	0.718	0.620	0.697
Khadkevich & Omologo (2009a) (KO2)	0.730	0.626	0.708
Mauch et al. (2009a) (MD)	0.720	0.683	0.712
Oudre et al. (2009a) (OGF1)	0.717	0.665	0.706
Oudre et al. (2009b) (OGF2)	0.729	0.646	0.711
Papadopoulos & Peeters (2009) (PP)	0.691	0.609	0.673
Pauwels et al. (2009) (PVM1)	0.687	0.666	0.682
Pauwels et al. (2009) (PVM2)	0.661	0.629	0.654

Table 4.15: MIREX 2009 Results: Weighted Average Overlap Scores.

	Beatles	Others	Total
Ellis (2009) (DE)	0.742	0.688	0.731
Khadkevich & Omologo (2009a) (KO1)	0.748	0.685	0.734
Khadkevich & Omologo (2009a) (KO2)	0.754	0.690	0.741
Mauch et al. (2009a) (MD)	0.755	0.721	0.748
Oudre et al. (2009a) (OGF1)	0.778	0.739	0.770
Oudre et al. (2009b) (OGF2)	0.789	0.732	0.777
Papadopoulos & Peeters (2009) (PP)	0.728	0.670	0.715
Pauwels et al. (2009) (PVM1)	0.713	0.700	0.711
Pauwels et al. (2009) (PVM2)	0.705	0.673	0.698

Table 4.16: MIREX 2009 Results: Weighted Average Root Overlap Scores.

their methods work, it is really difficult to know in what extent these methods are overfitted with this database. To our knowledge, there is no recent chord recognition system which has not used this database for development: the overfitting concern therefore applies to every tested method. For example, by comparing scores obtained on the Beatles corpus and other data, we can clearly see this problem. Despite the fact that most of these methods are not explicitly using training, we notice that the scores obtained for the non-Beatles corpus are lower than those obtained on the Beatles corpus (average loss of 6.4%). This is a very tough problem since they are no ways to prove whether the good scores are due to some overfitting or to the good performances of the systems. Only the introduction of new and large annotated chord databases shall adress this issue. Despite this unsolvable problem, the only thing we can say is that, according to these results, our systems do not seem to overfit more than other tested chord recognition systems.

4.4 Conclusion and discussion

In Chapter 2, we have separated the chord recognition methods in four main categories: template-based, training-based, music-driven and hybrid methods. Since 2006, no special attention had been turned to template-based methods, since much more sophisticated and efficient systems had been developed, taking into account either training, musical knowledge or both. The major contribution described in this chapter is the building of a complete and new template-based chord recognition system, which can honorably compare with the state-of-the-art while being simpler and more straightforward.

Let us summary the main contributions and conclusions of this chapter:

- The use of pre-determined chord templates allows to avoid the fastidious task of annotating data and also to easily introduce chord types other than major and minor. Our experiments have shown that the introduction of harmonics in the chord templates does not significantly improve the results, and that, most of the time, binary chord templates are enough to give good results.
- The use of post-processing filtering on the recognition criterion is innovative and the results obtained by using this filtering clearly proves that it enables to take into account the expected chord lengths, sometimes in a better way than HMM systems, which inherently assume an exponential time distribution.
- Our methods perform well on other corpora, no matter what genre, style or artist. The parameter choice, while being optimal for the Beatles corpus, is also relevant for other types of music. Therefore, independently of the known problem of overfitting which can apply to every recent chord recognition system since the release of Christopher Harte's annotation files, our DCR systems do not seem to be too dependent on the corpus it has been developed on.
- The simplicity of our methods allows to be very low time-consuming. This characteristic can be useful in particular for embedded systems having low resources or very limited battery life.

Yet, our systems also suffer from the disadvantages inherent to template-based methods. For example, we have seen that our methods tend to produce a large number of major-minor confusions (see Section 4.3.2). Also, while post-processing methods inform our systems about

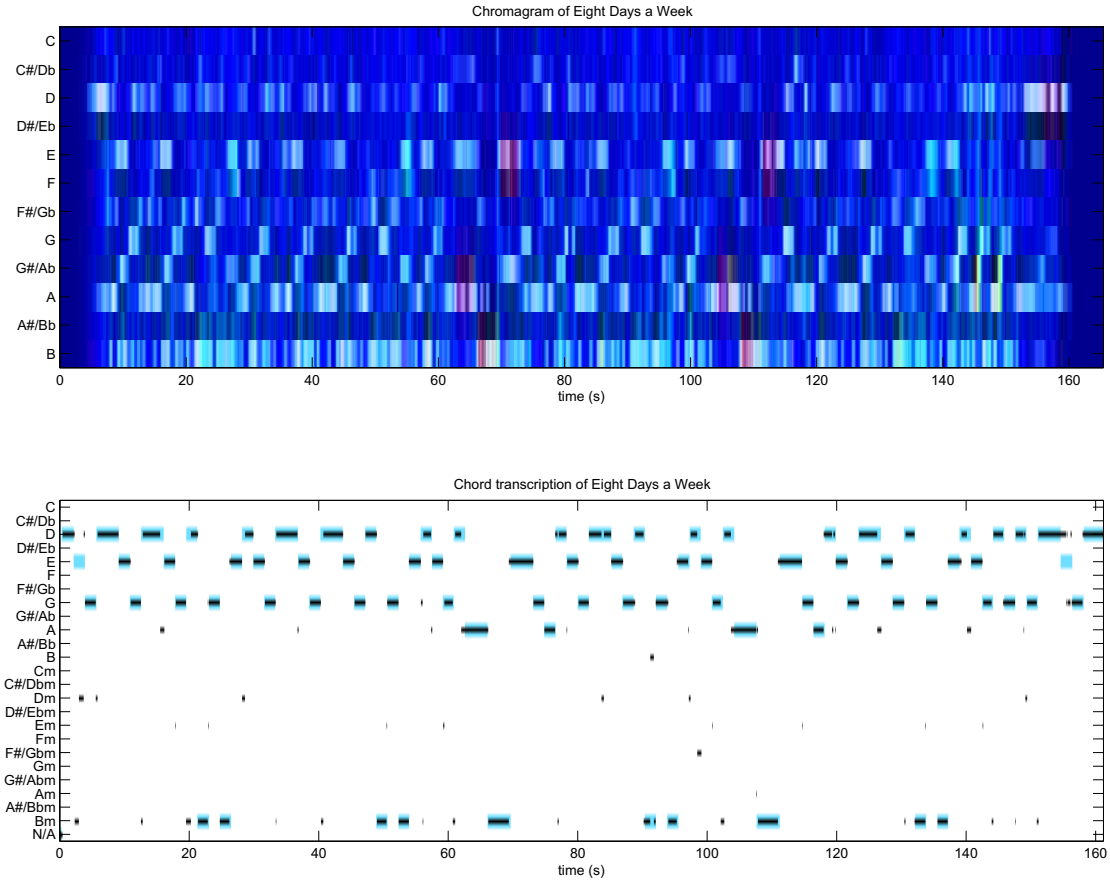


Figure 4.13: (Chromagram and chord transcription of the Beatles song *Eight days a week* with the OGF1 *(maj-min)* chord recognition system. On the bottom, the estimated chord labels are in black while the ground-truth chord annotation is in light blue.

the expected chord length, we still notice that our transcriptions are sometimes fragmented. While these fragmented chords, thanks to their short duration, do not penalize much the recognition scores, they degrade the transcription by making it less clear and readable. Finally, we also notice that our methods tend to overestimate the number of chord labels needed for transcription: this also results in hardly understandable chord transcriptions. We have displayed on Figure 4.13 one example of chord transcription output by our system. We can observe on this transcription the three phenomena previously described (major-minor confusions, fragmented chords, overestimation of the number of different chords).

These issues are addressed in the next chapter, which describes a probabilistic framework for chord recognition based on a number of components presented in this chapter.

Related publications

Oudre L., Grenier Y., Févotte C., "Chord recognition by fitting rescaled chroma vectors to chord templates", *submitted to* IEEE Transactions on Audio, Speech and Language Processing, 2009.

Oudre, L., Grenier, Y., Févotte, C. (2009). Chord recognition using measures of fit, chord templates and filtering methods. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* New York, USA.

Oudre, L., Grenier, Y., Févotte, C. (2009). Template-based chord recognition: influence of the chord types. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR).* Kobe, Japan.

Chapter 5

Probabilistic template-based chord recognition

Contents

In chapter 4, we have presented a deterministic template-based chord recognition approach. We have also seen that our DCR methods, which do not introduce any other information than the chord definition, sometimes have difficulties to capture long-term variations in chord sequences, as well as producing clear and compact chord transcriptions. In particular, since no rhythm or harmony knowledge is used, the DCR methods can give good frame-to-frame results but it is often difficult to directly use the algorithm output to playback a song.

The approach presented in this chapter builds on the template-based methods described in the previous chapter but gives a probabilistic framework by modeling the chord probability distribution of the song. The introduction of these chord probabilities aims at producing sparser chord transcriptions, and thus attempts to address the weaknesses of the deterministic approach.

5.1 Description of the approach

In this section we describe a novel probabilistic template-based chord recognition system. Our approach builds on the DCR systems described in Chapter 4, but where the measures of fit are turned into likelihood functions and where the chord occurrences in the chromagram are treated as probabilistic events. In particular, the probability of each chord is learned from the song, and this will be shown to *sparsify* the chord vocabulary (elimination of spurious chords), which in turn greatly improves transcription accuracy. Remember that for each song the vocabulary is a subset of the user-defined chord dictionary containing all the chords played in the song (see Section 3.2.4).

In the deterministic model, we used as recognition criterion the term $d_{k,n} = D\left(h_{k,n}\,\mathbf{c}_n; \mathbf{w}_k\right)$. With this formulation, $h_{k,n}$ was a scale parameter applied to chromagram frame $\mathbf{c}_n$ in order to fit it to chord template $\mathbf{w}_k$. In this chapter, we shall define an *amplitude parameter* $a_{k,n}$, as the counterpart of the scale parameter, except that it is now applied on the chord template. As a result, the assumption

$$h_{k,n}\,\mathbf{c}_n \approx \mathbf{w}_k \tag{5.1}$$

now becomes

$$\mathbf{c}_n \approx a_{k,n}\,\mathbf{w}_k. \tag{5.2}$$

Choice (5.1) was motivated by previous template-based works that would normalize the chromagram priori to recognition, while choice (5.2) describes a generative model. Note that we could have performed the whole work presented in Chapter 4 with this notation, only by introducing a new recognition criterion $d'_{k,n} = D\left(\mathbf{c}_n; a_{k,n}\,\mathbf{w}_k\right)$.

5.1.1 Generative model

When the Euclidean distance, the Kullback-Leibler (KL) divergence or the Itakura-Saito (IS) divergence is used as the measure of fit, the criterion $D\left(\mathbf{c}_n; a_{k,n}\,\mathbf{w}_k\right)$ defined in Chapter 4 is actually a log-likelihood in disguise. Indeed, these measures of fit respectively underlie Gaussian additive, Poisson and Gamma multiplicative observation noise models (defined in Table 5.1) and they may be linked to a log-likelihood such that

$$-\log p\left(\mathbf{c}_n | a_{k,n}, \mathbf{w}_k\right) = \varphi_1\, D\left(\mathbf{c}_n | a_{k,n}\,\mathbf{w}_k\right) + \varphi_2, \tag{5.3}$$

where $p\left(\mathbf{c}_n | a_{k,n}, \mathbf{w}_k\right)$ is the probability of chroma vector $\mathbf{c}_n$ (now treated as a random variable) given chord template $\mathbf{w}_k$ (a fixed deterministic parameter) and scale $a_{k,n}$ (treated as an

Gaussian	$\mathcal{N}\left(x;\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Gamma	$\mathcal{G}\left(x;\alpha,\beta\right) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta x}$
Poisson	$\mathcal{P}\left(x;\lambda\right) = \frac{\lambda^x}{\Gamma(x+1)} e^{-\lambda}$

where Γ is the Gamma function.

Table 5.1: Definitions of the probability distributions.

unknown deterministic parameter), and where φ_1 and φ_2 are constants w.r.t. $a_{k,n}$ and $\mathbf{w}_k$. The exact correspondences between each measure of fit and its equivalent statistical observation noise model are given in Table 5.2. Note that with this formulation, we can only build correspondences with *KL1* and *IS1*, since *KL2* and *IS2* do not define a proper probabilistic model.

The distribution $p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right)$ represents the probability of observing $\mathbf{c}_n$ *given* that the chord played at frame n is the k^{th} one, i.e., the one modeled by template $\mathbf{w}_k$. Let us introduce the discrete state variable $\gamma_n \in [1,\ldots,K]$ which indicates which chord is played at frame n, i.e, $\gamma_n = k$ if chord k is played at frame n. Hence, we may write

$$p\left(\mathbf{c}_n|\gamma_n = k, a_{k,n}\right) = p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right). \tag{5.4}$$

We are slightly abusing notations here as $\mathbf{w}_k$ should also appear on the left-hand side of Equation (5.4), but as this is a fixed parameter as opposed to a parameter to be estimated, we will drop it from the notations. Now let us denote by α_k the probability of occurrence of chord k in the song. Hence we have

$$P(\gamma_n = k) = \alpha_k, \tag{5.5}$$

where we assume that the frames are independent. Let us introduce the vector variables $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_K]^T$ (vector of all chord probabilities) and $\mathbf{a}_n = [a_{1,n},\ldots,a_{K,n}]$ (vector of all scale parameters at frame n). Averaging over all possible states (chords), the statistical generative model of the chromagram defined by Equations (5.4) and (5.5) may be written more concisely as

$$p\left(\mathbf{c}_n|\boldsymbol{\alpha},\mathbf{a}_n\right) = \sum_{k=1}^{K} \alpha_k\, p\left(\mathbf{c}_n|a_{k,n},\mathbf{w}_k\right), \tag{5.6}$$

which defines a *mixture model*.

To recap, given a dictionary of chords $\mathbf{W}$ with occurrence probabilities $\boldsymbol{\alpha}$, under our model, a chromagram frame $\mathbf{c}_n$ is generated by:

1. randomly choosing chord k with probability α_k,
2. scaling $\mathbf{w}_k$ with parameter $a_{k,n}$ (to account for amplitude variations),
3. generating $\mathbf{c}_n$ according to the assumed noise model and $a_{k,n}\mathbf{w}_k$.

The only parameters to be estimated in our model are the chord probabilities $\boldsymbol{\alpha}$ and the set of amplitude coefficients $\mathbf{A} = \{a_{k,n}\}_{kn}$. Given estimates of these parameters, chord recognition

	Noise structure	Observation noise model $p(\mathbf{c}_n \vert a_{k,n}, \mathbf{w}_k)$	Log-likelihood $-\log(p(\mathbf{c}_n \vert a_{k,n}, \mathbf{w}_k))$
Gaussian	Additive Gaussian noise $\mathbf{c}_n = a_{k,n}\mathbf{w}_k + \epsilon$	$\prod_{m=1}^{M} \mathcal{N}\left(c_{m,n}\ ;\ a_{k,n}w_{m,k}, \sigma^2\right)$	$\frac{1}{2\sigma^2} d^2_{EUC}\left(\mathbf{c}_n; a_{k,n}\mathbf{w}_k\right) + cst$
Gamma	Multiplicative Gamma noise $\mathbf{c}_n = (a_{k,n}\mathbf{w}_k) \cdot \epsilon$	$\prod_{m=1}^{M} \frac{1}{a_{k,n}w_{m,k}} \mathcal{G}\left(\frac{c_{m,n}}{a_{k,n}w_{m,k}}\ ;\ \beta, \beta\right)$	$\beta\, d_{IS}\left(\mathbf{c}_n \vert a_{k,n}\mathbf{w}_k\right) + cst$
Poisson	Poisson noise	$\prod_{m=1}^{M} \mathcal{P}\left(c_{m,n}\ ;\ a_{k,n}w_{m,k}\right)$	$d_{KL}\left(\mathbf{c}_n \vert a_{k,n}\mathbf{w}_k\right) + cst$

where $\mathcal{N}$, $\mathcal{G}$ and $\mathcal{P}$ are the probability distributions defined in Table 5.1 and cst denotes terms constant w.r.t. $a_{k,n}\mathbf{w}_k$.

Table 5.2: Correspondences between the Euclidean distance, KL and IS divergences and their equivalent statistical observation noise model.

What is EM-algorithm? (Dempster et al. (1977))

Let us assume that we have:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$: observations
- $\mathbf{y} = (y_1, y_2, \ldots, y_n)$: missing data
- $\mathbf{z} = ((x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n))$: complete data

Let us suppose that the observations $\mathbf{x}$ depend on parameters $\boldsymbol{\theta}$. We want to maximize $p(\mathbf{x}; \boldsymbol{\theta})$ but it not feasible: we will therefore complete the observations $\mathbf{x}$ with $\mathbf{y}$ in order to obtain $\mathbf{z}$. There, we can easily maximize $p(\mathbf{z}; \boldsymbol{\theta})$.

The EM-algorithm is an iterative process which successively update parameters $\boldsymbol{\theta}^{(k)}$.

The algorithm works in two steps:

- **Step 1 (Expectation)**: We calculate the function

$$Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}\right) = \int_{\mathbf{y}} \underbrace{\log p\left(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}\right)}_{\substack{\text{complete data} \\ \text{likelihood}}} \underbrace{p\left(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}^{(k)}\right)}_{\substack{\text{missing data} \\ \text{posterior}}} d\mathbf{y}$$

- **Step 2 (Maximization)**: We maximize this function way relative to θ

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}\right)$$

may be performed at every frame n by selecting the chord with largest posterior probability, i.e,

$$\hat{\gamma}_n = \underset{k}{\operatorname{argmax}}\, p(\gamma_n = k | \mathbf{c}_n, \hat{\boldsymbol{\alpha}}, \hat{\mathbf{a}}_n). \tag{5.7}$$

5.1.2 Expectation-Maximization (EM) algorithm

We describe here an EM-algorithm for maximum likelihood estimation of parameters $\boldsymbol{\alpha}$ and $\mathbf{A}$. More details on the EM-algorithm can be found in the box *What is EM algorithm?* (p 99).

Let us denote $\boldsymbol{\Theta} = (\boldsymbol{\alpha}, \mathbf{A})$ the set of parameters. Our task is to maximize the following objective function

$$\log p\left(\mathbf{C} | \boldsymbol{\Theta}\right) = \sum_n \log p\left(\mathbf{c}_n | \boldsymbol{\alpha}, \mathbf{a}_n\right), \tag{5.8}$$

which may routinely be done with an EM-algorithm using the set of chord state variables as missing data, which we denote $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_N]$. The EM-algorithm involves computing (E-step) and maximizing (M-step) the following functional

$$Q\left(\boldsymbol{\Theta} | \boldsymbol{\Theta}'\right) = \sum_{\boldsymbol{\gamma}} \log p\left(\mathbf{C}, \boldsymbol{\gamma} | \boldsymbol{\Theta}\right) p\left(\boldsymbol{\gamma} | \mathbf{C}, \boldsymbol{\Theta}'\right) \tag{5.9}$$

where $\log p\left(\mathbf{C}, \boldsymbol{\gamma} | \boldsymbol{\Theta}\right)$ is referred to as the *complete data likelihood* and $p\left(\boldsymbol{\gamma} | \mathbf{C}, \boldsymbol{\Theta}'\right)$ is the *missing data posterior*. Each of the two EM steps is described next.

E-Step

Under the frame independence assumption, the functional (5.9) can be written as

$$Q\left(\mathbf{\Theta}|\mathbf{\Theta'}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} \log p\left(\mathbf{c}_n, \gamma_n = k|\mathbf{\Theta}\right) p\left(\gamma_n = k|\mathbf{c}_n, \mathbf{\Theta'}\right). \tag{5.10}$$

Let us denote $\bar{\alpha}_{k,n}$ the posterior probability of state variable $\gamma_{k,n}$ (the notation is chosen in analogy with the notation chosen for its prior probability α_k), i.e.,

$$\begin{aligned} \bar{\alpha}_{k,n} &= p\left(\gamma_n = k|\mathbf{c}_n, \mathbf{\Theta}\right) & (5.11)\\ &= \frac{\alpha_k\, p\left(\mathbf{c}_n|\gamma_n = k, \mathbf{\Theta}\right)}{\sum_{l=1}^{K} \alpha_l\, p\left(\mathbf{c}_n|\gamma_n = l, \mathbf{\Theta}\right)}, & (5.12)\end{aligned}$$

where the second equation comes naturally from the application of Bayes theorem and by the fact that the probabilities sum to 1. In the following, we denote by $\bar{\alpha}'_{k,n}$ the posterior state probabilities conditioned on parameter $\mathbf{\Theta}'$. Hence, by expanding the complete data likelihood as

$$\log p\left(\mathbf{c}_n, \gamma_n = k|\mathbf{\Theta}\right) = \log p\left(\mathbf{c}_n|a_{k,n}, \mathbf{w}_k\right) + \log \alpha_k, \tag{5.13}$$

the E-step amounts to evaluating the EM functional as

$$Q\left(\mathbf{\Theta}|\mathbf{\Theta'}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} \left[\log p\left(\mathbf{c}_n|a_{k,n}, \mathbf{w}_k\right) + \log \alpha_k\right] \bar{\alpha}'_{k,n}, \tag{5.14}$$

which we recall is to be maximized w.r.t to $\mathbf{\Theta} = (\boldsymbol{\alpha}, \mathbf{A})$ and subject to $\sum_{k=1}^{K} \alpha_k = 1$.

M-step

The derivative of $Q\left(\mathbf{\Theta}|\mathbf{\Theta'}\right)$ w.r.t to $a_{k,n}$ writes

$$\nabla_{a_{k,n}} Q\left(\mathbf{\Theta}|\mathbf{\Theta'}\right) = \bar{\alpha}'_{k,n} \nabla_{a_{k,n}} \log p(\mathbf{c}_n|a_{k,n}, \mathbf{w}_k), \tag{5.15}$$

so that updating $a_{k,n}$ amounts to solving

$$\nabla_{a_{k,n}} \log p(\mathbf{c}_n|a_{k,n}, \mathbf{w}_k) = 0, \tag{5.16}$$

which does not involve the current parameter estimate $\mathbf{\Theta}'$. Therefore, the parameter $\mathbf{A}$ can be precomputed and does not need to be updated during the EM iterations. Note that the estimation $\mathbf{A}$ is equivalent to that of Equation (4.2) in the deterministic approach. Table 5.3 presents the correspondences between the scale parameters used in the DCR methods and the amplitude parameters introduced in the probabilistic framework.

Regarding the optimization of parameter $\boldsymbol{\alpha}$, the sum constraint can routinely be handled with the introduction of a Lagrangian term, leading to the following update

$$\alpha_k = \frac{\sum_{n=1}^{N} \bar{\alpha}'_{k,n}}{\sum_{l=1}^{K}\sum_{n=1}^{N} \bar{\alpha}'_{l,n}}. \tag{5.17}$$

The resulting EM algorithm is summarized below.

	Scale parameter $h_{k,n}$	Amplitude parameter $a_{k,n}$
EUC / Gaussian	$\frac{\sum_{m=1}^{M} c_{m,n}\, w_{m,k}}{\sum_{m=1}^{M} c_{m,n}^2}$	$\frac{\sum_{m=1}^{M} c_{m,n}\, w_{m,k}}{\sum_{m=1}^{M} w_{m,n}^2}$
IS1 / Gamma	$\frac{M}{\sum_{m=1}^{M} \frac{c_{m,n}}{w_{m,k}}}$	$\frac{1}{M} \sum_{m=1}^{M} \frac{c_{m,n}}{w_{m,k}}$
KL1 / Poisson	$e^{-\sum_{m=1}^{M} c'_{m,n} \log\left(\frac{c_{m,n}}{w_{m,k}}\right)}$	$\sum_{m=1}^{M} c_{m,n}$

Table 5.3: Correspondences between scale and amplitude parameters.

Algorithm 1: EM algorithm for probabilistic template-based chord recognition.

Input: Chromagram data $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$, chord templates $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$
Output: Chord probabilities $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$

Initialize $\boldsymbol{\alpha}$
Compute scale parameters $\mathbf{A}$ as of Eq. (5.16)

for $i = 1 : n_{iter}$ **do**

$$\bar{\alpha}_{k,n}^{(i-1)} = \frac{p\big(\mathbf{c}_n | a_{k,n}, \mathbf{w}_k\big)\, \alpha_k^{(i-1)}}{\sum_{l=1}^{K} p\big(\mathbf{c}_n | a_{l,n}, \mathbf{w}_l\big)\, \alpha_l^{(i-1)}} \qquad \texttt{// E-Step}$$

$$\alpha_k^{(i)} = \frac{\sum_{n=1}^{N} \bar{\alpha}_{k,n}^{(i-1)}}{\sum_{l=1}^{K} \sum_{n=1}^{N} \bar{\alpha}_{l,n}^{(i-1)}} \qquad \texttt{// M-Step}$$

5.1.3 Post-processing filtering

As already discussed in Section 5.1.1, our chord recognition criterion is based on the frame-by-frame maximum state posterior probability, i.e.,

$$\hat{\gamma}_n = \underset{k}{\operatorname{argmax}}\ \bar{\alpha}_{k,n}. \tag{5.18}$$

Note that the state posterior probabilities are readily available from within the EM algorithm. Just like in the DCR methods, this frame-by-frame chord recognition system can be improved by taking into account the long-term trend in the chord changes. We therefore propose to use an *ad hoc* filtering process that implicitly informs the system of the expected chord duration. The post-processing filtering is performed on the state posterior probabilities $\bar{\alpha}_{k,n}$ and not on the chromagram (Fujishima (1999), Bello & Pickens (2005), Peeters (2006)) or on the detected chord sequence (Bello & Pickens (2005)).

5.2 Experiments

In this section, we propose to test our new probabilistic chord recognition methods (we shall refer to them as **PCR** methods). Note that we have a more limited parameter choice than in the DCR methods. Indeed, we have seen in the previous chapter that the introduction of higher harmonics in chord templates were not really useful: we shall therefore not extensively test it. As a result, here are the parameters we shall test:

- **3 observation noise models:** Gaussian, Gamma and Poisson ;
- **2 types of filtering:** low-pass and median filtering ;
- **12 neighborhood sizes (when post-processing is applied):** from $L = 3$ to $L = 25$.

This gives $3 \times 2 \times 12$ (filtering) $+ 3$ (no filtering) $= 75$ parameter sets.

5.2.1 Determination of the hyper-parameters

We notice on Table 5.2 that two observation noise distributions depend on hyper-parameters. Indeed, the additive Gaussian noise model introduces σ^2 while the multiplicative Gamma noise model includes a β parameter. The role of these hyper-parameters is to fit the noise distribution to the real noise present in chroma vectors.

Consequently, before giving some first results, we need to evaluate the best hyper-parameters for these two observation noise models. We propose to run some no-load tests (without filtering) for several hyper-parameter values and calculate the AOS obtained on the Beatles corpus. The vector $\boldsymbol{\alpha}$ is initialized with random values and the number of iterations for the EM-algorithm is set to 200 iterations, which appears sufficient to convergence.

Figures 5.1 and 5.2 present the results of these experiments. We clearly see that the adjustment of these hyper-parameters is crucial since, for example, low values of σ^2 and high values of β give very bad performances. The optimal choice for σ^2 is 0.04 and for β is 3. We shall use these values for the rest of this chapter.

5.2.2 First results

Table 5.4 presents the AOS obtained by the 75 probabilistic chord transcribers on the Beatles corpus. The best results are obtained with the multiplicative Gamma observation noise model and low-pass filtering applied to 15 frames.

	no filtering	low-pass	median
Gaussian	0.714	0.748	0.749
Gamma	0.730	**0.758**	**0.758**
Poisson	0.727	0.742	0.744

Table 5.4: Average Overlap Scores on the Beatles corpus obtained with the PCR methods. For sake of conciseness, we only display for each post-processing method the results obtained with the optimal choice of L.

The multiplicative Gamma observation noise model gives better results than the additive Gaussian and Poisson observation noise models. Nevertheless, just like the differences between the *EUC*, *IS1* and *KL1* measures of fit were small, there is also no major gap between the scores obtained with the 3 observation noise models.

As far as post-processing filtering is concerned, we can draw the same conclusions as for the DCR methods. We notice that once again, the introduction of filtering clearly enhances the system performances and that there does not seem to be a big difference between low-pass and median filtering. The influence of neighborhood size is similar to the one observed for the DCR systems. As shown on Figure 5.3, the scores obtained for close neighborhood sizes are also very close.

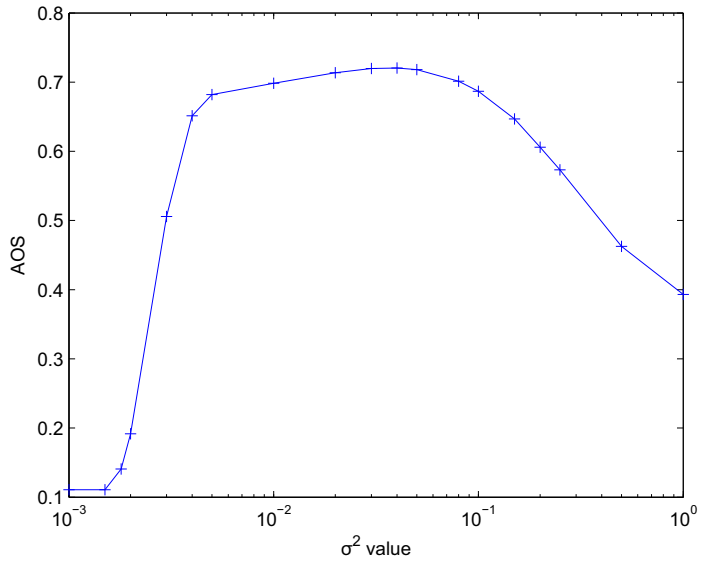


Figure 5.1: Influence of the value of hyper-parameter σ^2 within the additive Gaussian noise model on the Average Overlap Scores calculated on the Beatles corpus.

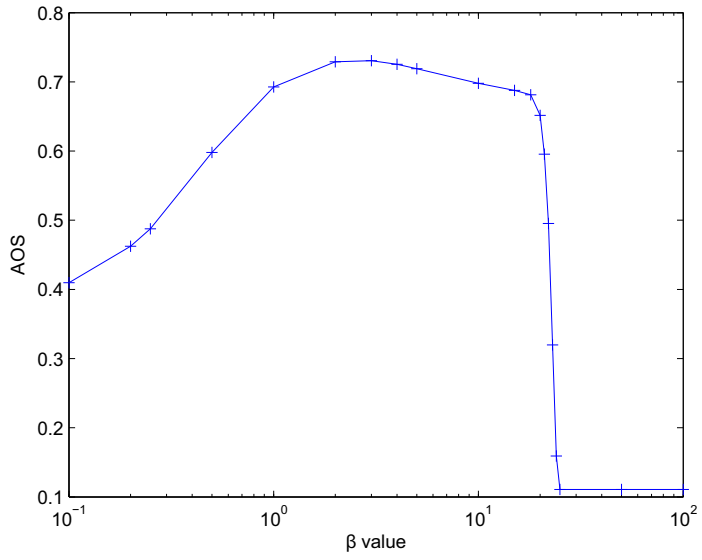


Figure 5.2: Influence of the value of hyper-parameter β within the multiplicative Gamma noise model on the Average Overlap Scores calculated on the Beatles corpus.

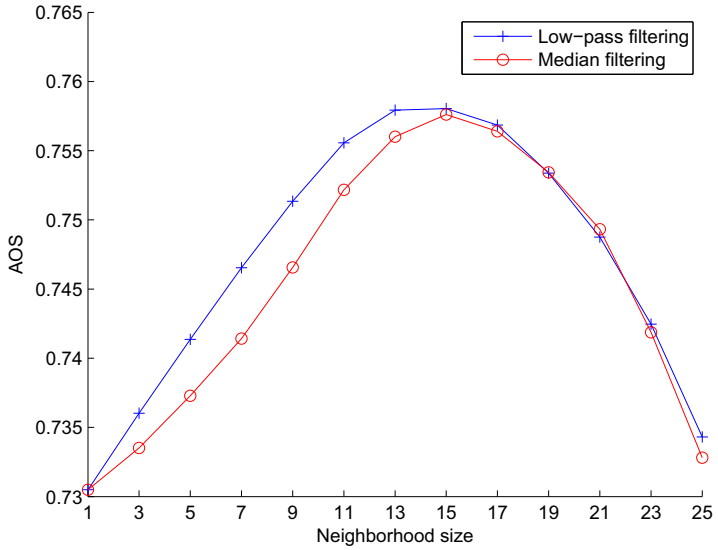


Figure 5.3: Influence of neighborhood size on the Average Overlap Score obtained on the Beatles corpus with the multiplicative Gamma observation noise model.

For each observation noise model, we can define a chord transcriber with the following optimal parameters:

- **Gaussian additive noise model**: $\sigma^2 = 0.02$ and median filtering on 17 frames (2.23s);
- **Gamma multiplicative noise model**: $\beta = 3$ and low-pass filtering on 15 frames (2.04s);
- **Poisson noise**: median filtering on 13 frames (1.86s).

We will respectively refer to the method based on the above models as **PCR/Gaussian**, **PCR/Gamma**, **PCR/Poisson**.

5.2.3 Discussion on one example

We propose here to investigate the differences between the DCR methods and the PCR methods on one example: the Beatles song *Run for your life* from the album *Rubber Soul*. This song has been chosen because it is characteristic of the improvements brought by our new method. The tested DCR method is the one referred in the previous chapter as OGF1 *maj-min*. The PCR method used here is the PCR/Gamma one.

Figure 5.4 presents the two chord transcriptions of the song *Run for your life*. The estimated chord labels are in black while the ground-truth chord annotation is in light blue. The first observation is that the PCR transcription seems to be more accurate than the DCR one. Indeed, the Overlap Scores for the song are respectively 0.733 and 0.919, which shows a clear improvement. By carefully looking at both transcriptions, we can find three explanations for this good result:

- The PCR method detects chords boundaries more accurately than the DCR method.
- The PCR method detects longer chords while the DCR method gives very segmented results.

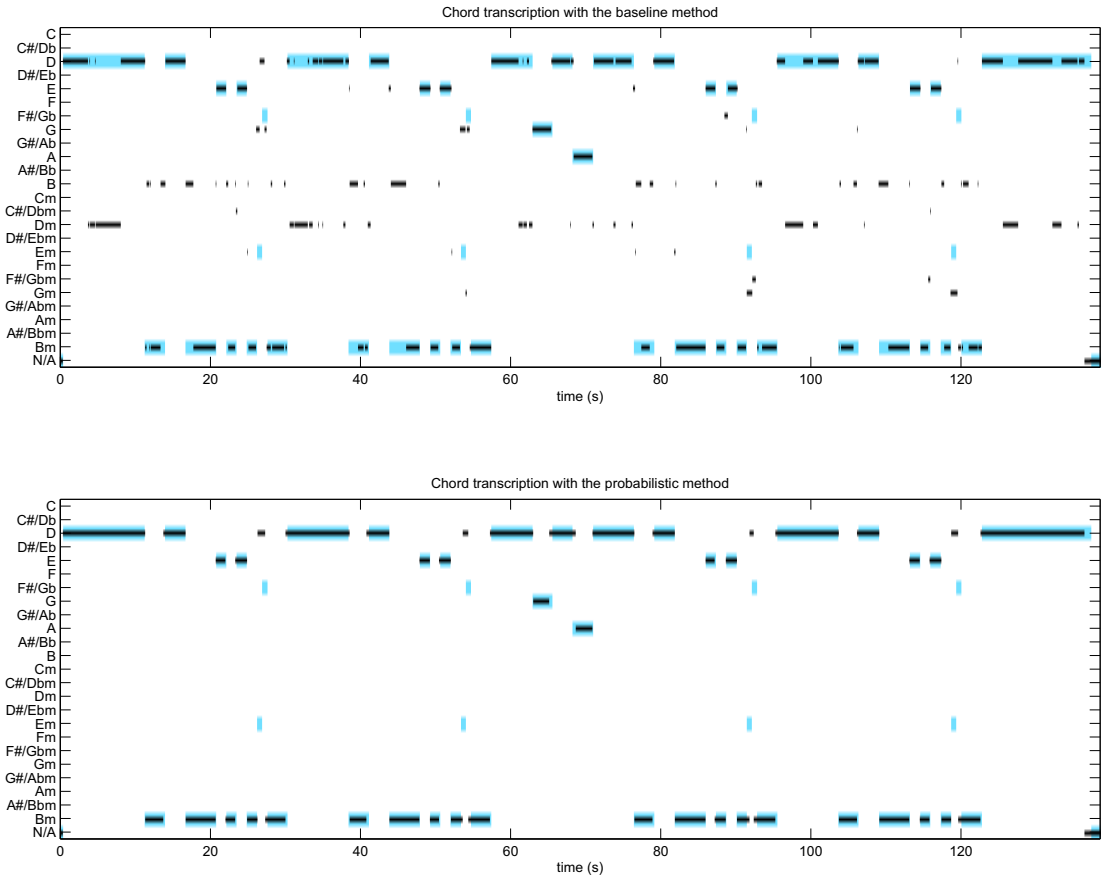


Figure 5.4: Examples of chord transcriptions of the Beatles song *Run for your life*. The estimated chord labels are in black while the ground-truth chord annotation is in light blue. At the top is represented the OGF1 *maj-min* method and at the bottom the PCR/Gamma method.

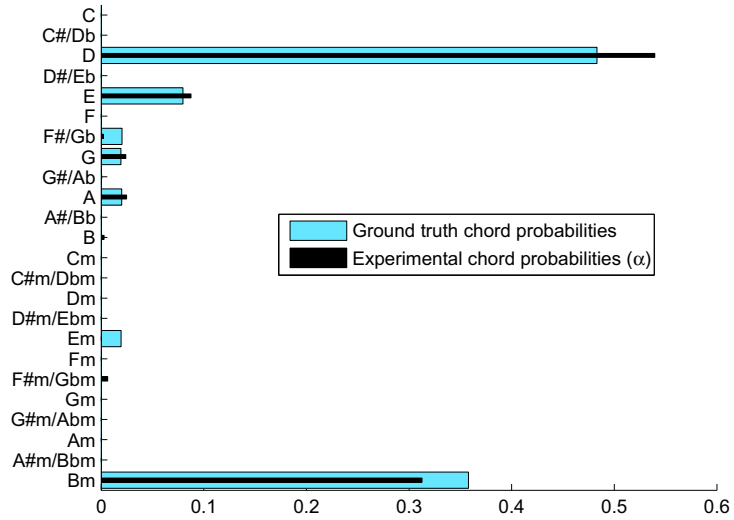


Figure 5.5: Ground-truth and experimental chord probability distributions on the Beatles song *Run for your life*.

- The chord vocabulary detected by the PCR method is sparser than the one used by the DCR method, preventing some major-minor confusions.

Indeed, the metrics calculated for this song confirm these assumptions:

- The Hamming Distances are respectively equal to 0.206 and 0.060, which reflects the fact that the segmentation provided by the PCR method is very similar to the one described in the annotation files.
- The Reduced Chord Lengths are respectively 0.345 and 1.085, which shows that the PCR method better evaluates the chord length.
- The chord vocabulary used by the PCR method is smaller than the DCR method's one: the Reduced Chord Numbers for the two methods are 1.75 and 0.75. Since the second value is closer to 1 than the first one, the number of chords used by the PCR method is the most accurate. The calculation of the False Chord Label Numbers confirms this: they are respectively equal to 6 and 0, which means that the second transcription does not use any chord labels that were not present in the annotation files.

Our PCR method can easily capture the chord vocabulary thanks to parameter $\boldsymbol{\alpha}$ that models, for each song, the chord probability distribution. Indeed, Figure 5.5 compares the values of the $\boldsymbol{\alpha}$ vector, which can be seen as the chord probability distribution, and the normalized chord length histogram of the annotation file, which represents the ground-truth chord probability distribution. We see that they are both close, which confirms our assumptions concerning the good estimation of the chord vocabulary.

5.2.4 Comparison with the DCR methods

Now that we have studied a specific example, let us investigate the main differences between the DCR and PCR methods on the whole Beatles corpus. Table 5.5 presents a comparison between the scores obtained by the DCR and the PCR methods.

	no filtering	low-pass	median
EUC	0.665	0.710	0.705
Gaussian	0.714	0.748	0.749
IS1	0.665	0.706	0.706
Gamma	0.730	0.758	0.758
KL1	0.665	0.700	0.692
Poisson	0.727	0.742	0.744

Table 5.5: Comparison between the Beatles Average Overlap Scores obtained with the DCR methods and those obtained with the PCR methods. For sake of conciseness, we only display for each post-processing method the results obtained with the optimal choice of L.

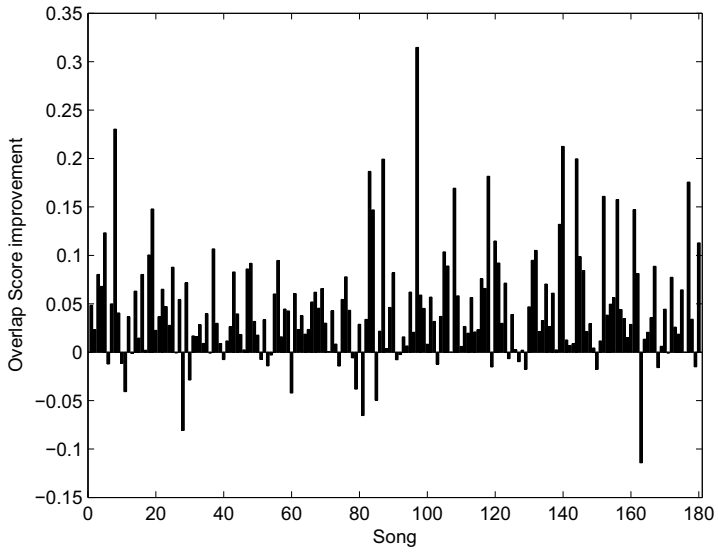


Figure 5.6: Overlap Score differences between PCR/Gamma and OGF1 *(maj-min)* methods on the 180 songs of the Beatles corpus.

We notice that the good performances of the PCR/Gamma method can not be explained by the particular efficiency of the *IS1* measure of fit. There does not seem to be an obvious correlation between the scores obtained by the PCR methods and their correspondent DCR method. Thus, we can think that the good scores obtained by the PCR/Gamma method are rather due to the good fit of this observation distribution to the noise present in our chroma vectors.

Figure 5.6 displays the Overlap Score improvement of the PCR/Gamma method on the OGF1 *maj-min* method for the 180 songs of the Beatles corpus. We see that the PCR/Gamma method outperforms the OGF1 *maj-min* method on a large number of songs (148 over 180). Note that the songs where the Overlap Score degradation exceeds 0.05 already obtained low scores with the OGF1 *maj-min* method. In addition, the average improvement (0.06) is higher than the average degradation (-0.02).

5.2.5 Additional experiments

In this section, we present the results of the experiments already run for the DCR methods (introduction of higher harmonics in chord models, introduction of chord types other than major and minor, and introduction of beat information).

In the results presented in this chapter, we did not introduce chord models with higher number of harmonics. Indeed, the *EUC*, *IS1* and *KL1* measures of fit did not respond well to these 4 or 6 harmonics chord templates, so we had no reason to think that it would work for the PCR systems. Indeed, additional experiments show that in every case, the introduction of harmonics degrades the results. Just like in the deterministic approach, the largest degradations occur with the Poisson observation model (*KL1* in the deterministic approach). The possible explanations are presented in Section 4.2.2.

The introduction of other chord types does not improve the results. Indeed our PCR methods detect the chord vocabulary by evaluating the chord probability distribution. Thus, when a chord is only present for a small number of frames, its probability is likely to be close to 0. As a result, this causes the chord not to be detected by our system. Since most of the Beatles songs only contain a small number of chords other than major and minor, they might not be detected as part of the chord vocabulary. Results show that when we introduce chord types other than major and minor, they are never detected by the system, which only outputs major and minor chords.

Finally, just like for the DCR systems, the introduction of beat times degrades the results. Simulations show that the loss on the AOS is variable but lies between 0.01 and 0.06. Explanations of this results are the same than the ones presented for the DCR methods (see Section 4.2.5.2).

5.3 Comparison with the state-of-the-art

In this section, our PCR methods are compared with some state-of-the-art systems according to the different metrics defined in Chapter 3. These methods have all been tested with their original implementations and have all participated in MIREX 2008 or 2009 (described in Section 4.3.1 & 4.3.3).

MIREX 2008:

- BP: Bello & Pickens (2005)
- RK: Ryynänen & Klapuri (2008a)
- PVM: Pauwels et al. (2008)

MIREX 2009:

- KO1 & KO2: Khadkevich & Omologo (2009a)
- DE: Ellis (2009)
- OGF1 & OGF2: our DCR methods

5.3.1 Beatles corpus

Table 5.6 presents the results obtained by these 11 chord recognition methods on the Beatles corpus.

	MIREX 2008			MIREX 2009					proposed methods		
	BP	RK	PVM	KO1	KO2	DE	OGF1	OGF2	Gaussian	Gamma	Poisson
Average Overlap Score (AOS)	0.707	0.705	0.648	0.722	0.734	0.738	0.714	0.724	0.749	**0.758**	0.744
Average Root Overlap Score (AROS)	0.740	0.763	0.680	0.754	0.761	0.772	0.775	0.783	0.785	**0.787**	0.775
Average Hamming Distance (AHD)	0.153	0.146	0.209	0.152	0.150	0.156	0.163	0.152	**0.146**	0.149	0.156
Average Chord Length (ACL)	0.941	1.074	0.422	1.169	1.168	0.890	0.552	0.717	0.872	0.920	**1.057**
Average Chord Number (ACN)	1.441	1.414	2.285	1.507	1.319	1.667	2.070	1.693	1.314	1.185	**1.012**
Average False Chord Label Number (AFCLN)	3.560	3.330	8.490	3.760	2.590	4.590	7.390	4.990	2.640	1.860	**1.060**
Run time	1619	2241	12402	6382[1]	6382[1]	1403	790	796	480	482	486

[1]The KO1 and KO2 methods shared some scratch data: here is presented the time for running both algorithms.

Table 5.6: Comparison with the state-of-the-art on the Beatles corpus.

	MIREX 2008			MIREX 2009					proposed methods		
	BP	RK	PVM	KO1	KO2	DE	OGF1	OGF2	Gaussian	Gamma	Poisson
Average Overlap Score (AOS)	0.699	0.730	0.664	0.670	0.665	0.719	0.707	0.682	0.739	**0.773**	0.760
Average Root Overlap Score (AROS)	0.743	0.768	0.693	0.705	0.695	0.759	0.783	0.790	0.788	**0.803**	0.779
Average Hamming Distance (AHD)	0.142	**0.117**	0.175	0.153	0.156	0.127	0.142	0.137	0.131	0.124	0.130
Average Chord Length (ACL)	0.903	**1.021**	0.494	1.084	1.109	0.823	0.565	0.683	0.835	0.896	0.806
Average Chord Number (ACN)	1.559	1.516	2.323	1.549	1.351	1.906	2.297	1.970	1.529	1.336	**1.138**
Average False Chord Label Number (AFCLN)	3.650	3.250	7.850	3.600	2.550	5.300	7.700	5.850	3.150	2.150	**1.150**

Table 5.7: Comparison with the state-of-the-art on the Quaero corpus.

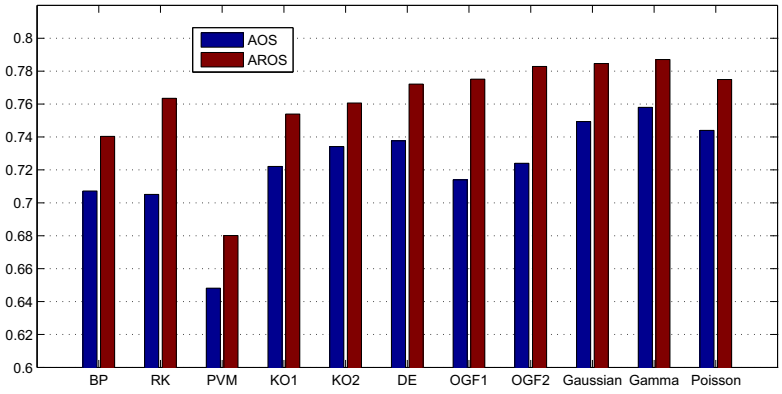


Figure 5.7: Average Overlap Scores on the Beatles corpus.

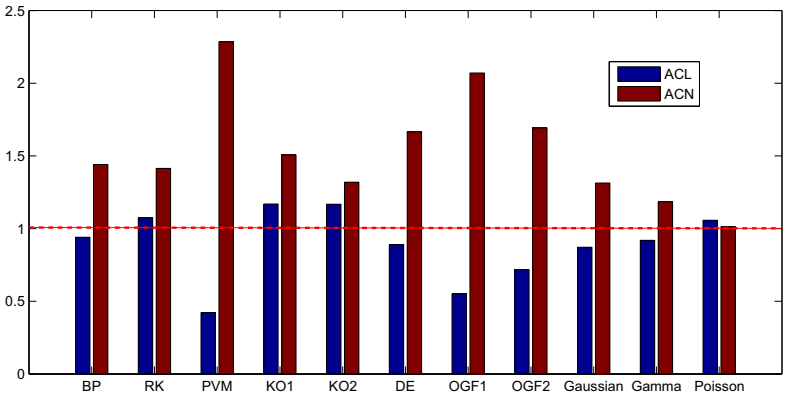


Figure 5.8: Average Chord Lengths and Average Chord Numbers on the Beatles corpus.

Recognition accuracy scores such as AOS or AROS (see Figure 5.7 for a graphical display of the scores) show that our PCR methods slightly outperform the state-of-the art: the AOS we obtain with the PCR/Gamma method is indeed 2% larger than the best score (DE). Since all the scores are close, it is interesting to figure out whether the methods are significantly different from each other. Results of the Tukey-Kramer's test are displayed on Figure 5.9. It shows that the improvement brought by our new PCR methods is significant. Indeed our PCR/Gamma method is significantly better than all the other tested methods except for the PCR/Gaussian one. The PCR/Gaussian method performs significantly better than all the methods except KO2 and all the PCR methods. Finally, the PCR/Poisson method is significantly better than BP, RK, PVM and OGF1 *(maj-min)*. This shows that the introduction of probabilistic framework has helped to achieve better performances, and that this enhancement is significantly shown on the calculated Overlap Scores.

The introduction of other evaluation metrics allows to compare chord recognition methods according to several criteria. Indeed, the 4 other metrics tend to evaluate the segmentation,

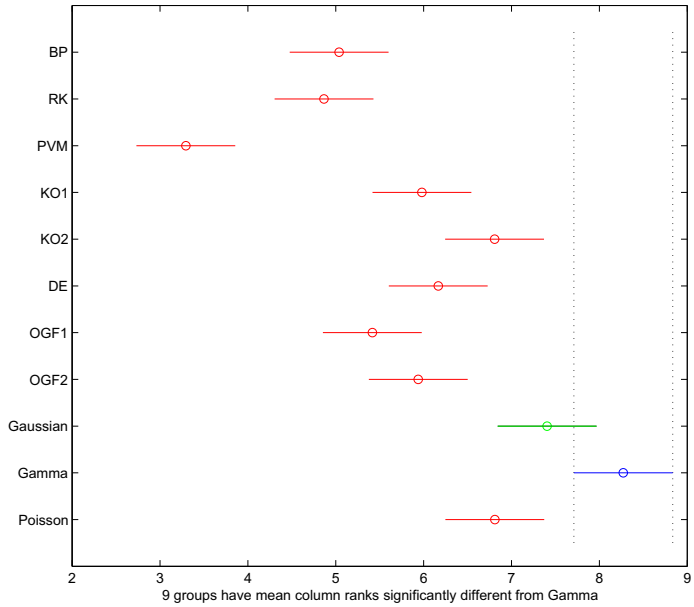


Figure 5.9: Tukey-Kramer's test performed on Overlap Scores calculated on the Beatles corpus. On this figure, the x-axis shows the average rank of each chord recognition method (previously denoted as $\bar{r}_{\cdot,i}$) along with its confidence interval. Ranks are presented in ascending order and two mean ranks are significantly different if their confidence intervals are disjoint (see Section 3.3).

the fragmentation and the good detection of the chord vocabulary.

The segmentation, that is to say the detection of chord boundaries, is evaluated thanks to the AHD (that should be as low as possible). We notice that, except for the PVM method, all the AHD are very close (around 0.15). Indeed, statistical tests are rather inconclusive: a Tukey-Kramer's test shows that except for the PVM method, they are no strong differences between the chord recognition methods. For example, the method obtaining the best AHD (PCR/Gaussian), is only significantly different from PVM, OGF1 *(maj-min)* & DE.

The fragmentation is evaluated thanks to the ACL (which should be as close to 1 as possible). Some methods seem to slightly underestimate the chord length (RK, KO1, KO2 & PCR/Poisson), but most of them tend to over-fragment the chords (see Figure 5.8). Some methods (PVM, OGF1 *(maj-min)*) even detect chords with half their real duration. On the contrary, our PCR methods seem to avoid this fragmentation effect: the best results are obtained with the PCR/Poisson method.

One of the main contributions of the probabilistic framework is the explicit evaluation of the song chord vocabulary. The accurate detection of this chord vocabulary is described by two metrics: the ACN (that should be as close to 1 as possible) and the AFCLN (that should be as low as possible). We notice that all methods seem to over-evaluate the number of chords (see Figure 5.8). The two methods PVM and OGF1 *(maj-min)* seem to be particularly penalized by this phenomenon. Our PCR methods, on the contrary, reliably evaluate the chord vocabulary: they obtain the 3 best scores. AFCLN scores confirm these results: the introduction of chord probabilities allows to capture the chord vocabulary in every song.

Figure 5.10 presents the error distribution for all 11 tested chord recognition systems. We conduct here the same study that has already be done in Section 4.3.2. We particularly focus here on the 3 new MIREX methods (KO1, KO2 & DE) and our 3 PCR systems (PCR/Gaussian, PCR/Gamma & PCR/Poisson). We notice that these 6 methods have a very similar error distribution. In particular, the parallel and relatives errors, which were very common with OGF1 *(maj-min)*, OGF2 *(maj-min-7)*, BP & RK, do not seem to be that important in the new MIREX 2009 methods. Furthermore, when comparing the error distributions of our PCR methods with those of the DCR systems, we see that the number of parallel errors has been significantly reduced (21% to 14% or 12%). It is interesting since these errors were really related to the template-based character of our methods, in which chords were likely to be mistaken one for another when they had notes in common. This phenomenon can be explained by the capacity of our PCR systems to efficiently evaluate the chord vocabulary, leading to a lower number of major-minor confusions.

Our PCR methods still do not need any training: the computational time should therefore be very low. Indeed, Table 5.6 presents the run time of these previously described state-of-the-art methods on the Beatles corpus. Thanks to some code optimization, our PCR methods perform even faster than the DCR methods, and are therefore twice as fast as other state-of-the-art methods.

5.3.2 Quaero corpus

We have already discussed the necessity of testing the chord recognition methods on other corpora that the popular Beatles corpus. We have therefore run all the tested systems on the Quaero corpus: results are displayed on Table 5.7.

The first observation is that except for RK, PVM, PCR/Gamma and PCR/Poisson, all the methods get lower AOS on this corpus than on the Beatles data (see Figure 5.11). Once again, our PCR methods give the best results: in particular, the PCR/Gamma method performs even better than on the Beatles corpus. Although the small number of songs in the corpus does not allow to perform a real significant difference test, the AOS obtained by the PCR/Gamma method is 4% higher than the best state-of-the-art method (RK), which is a rather large difference when looking at the scores.

As far as segmentation is concerned, the RK method gives the best results but nevertheless our PCR/Gamma method gives the second best AHD result. We notice that, contrarily to what happened on the Beatles, the AHD are rather spread: it is probably due to the fact that most of the tested systems were designed on the Beatles corpus and can therefore sometimes have difficulties to apply their model to other data.

Once again, most of the chord recognition methods tend to underestimate the chord length (see Figure 5.12): however our PCR/Gamma method gives the fourth best ACL score. We observe that OGF1 *(maj-min)* and OGF2 *(maj-min-7)* give disappointing ACL: this confirms the fragmentation concern about template-based methods. Although our PCR methods do not get the best scores, we still can see that an important improvement has been brought on the DCR methods concerning fragmentation.

Finally, we observe with the ACN and AFCLN metrics that our PCR methods still outperform other state-of-the-art methods on the chord vocabulary estimation (see Figure 5.12). Indeed our PCR/Poisson and PCR/Gamma methods get the two best ACN and AFCLN on this corpus.

An important point is that these results tend to answer the overfitting concern related to the Beatles corpus, since our PCR methods even achieve better performances on the Quaero corpus than on the Beatles one. In addition, music genre and style do not seem to influence

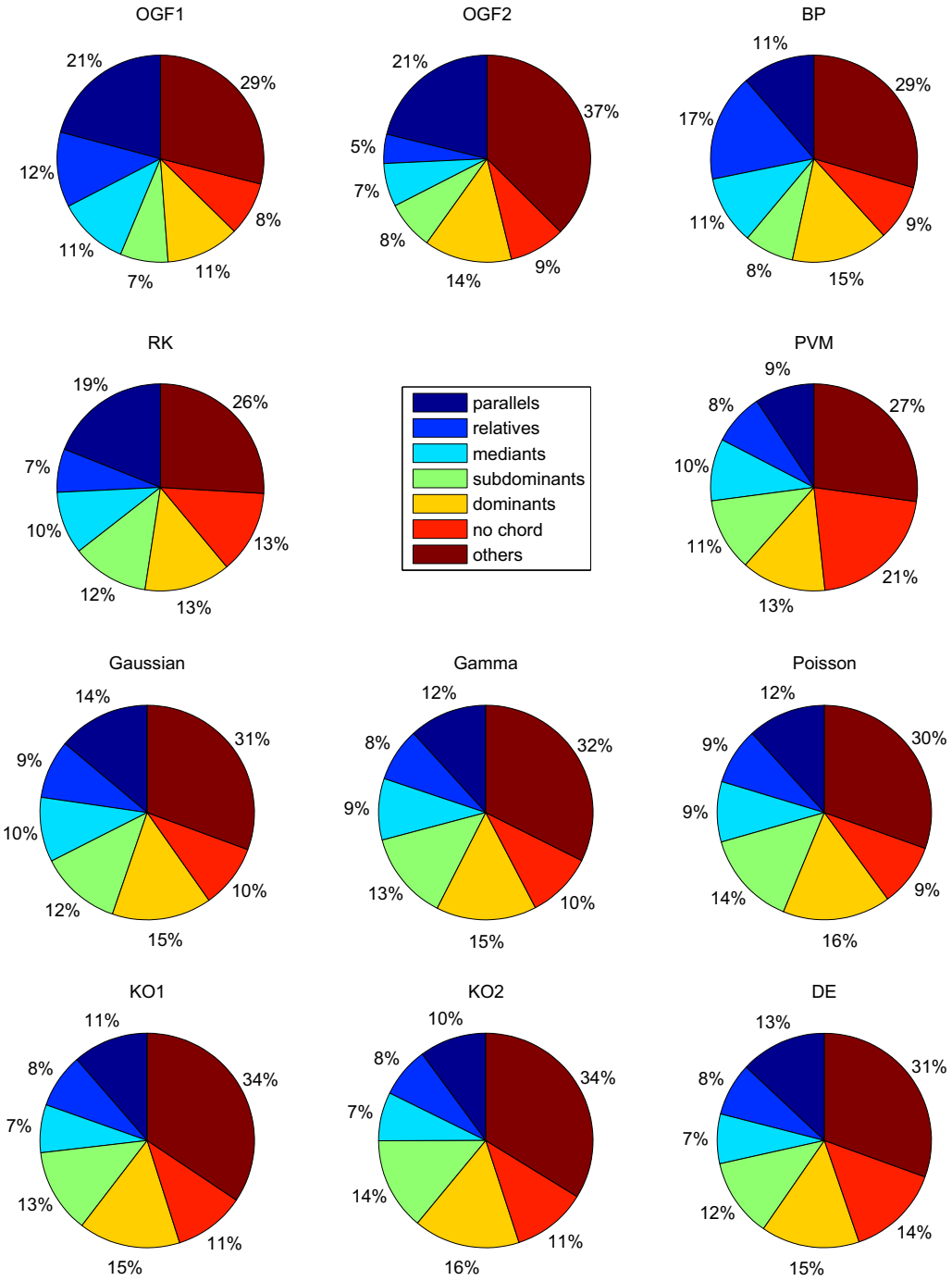


Figure 5.10: Error distribution on the Beatles corpus (as a percentage of the total number of errors).

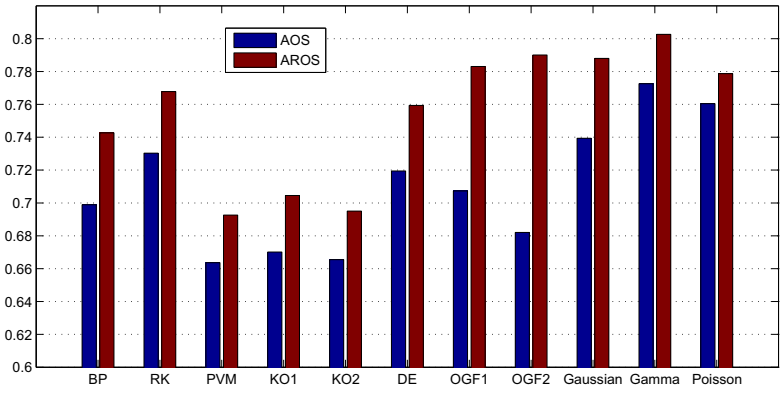


Figure 5.11: Average Overlap Scores on the Quaero corpus.

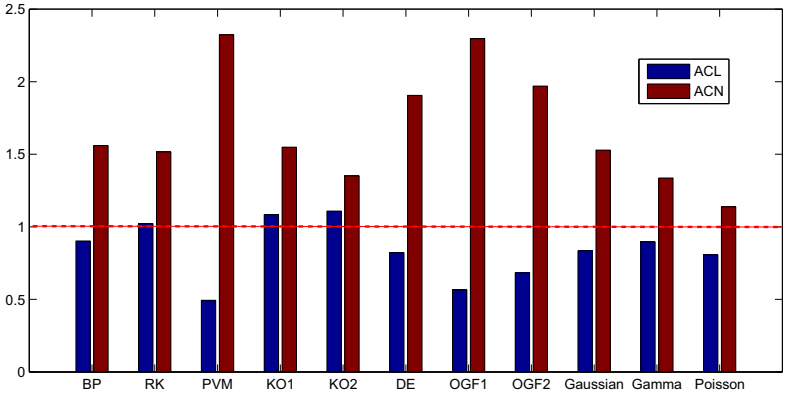


Figure 5.12: Average Chord Lengths and Average Chord Numbers on the Quaero corpus.

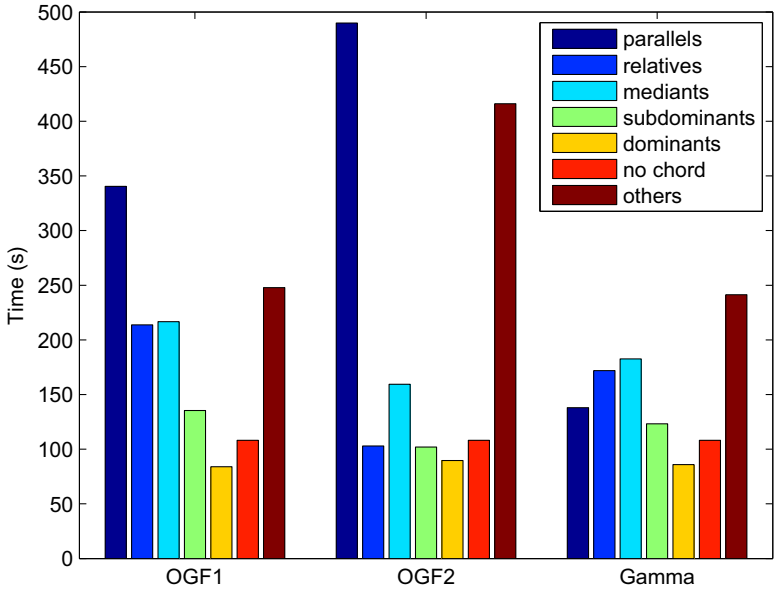


Figure 5.13: Error distribution of the OGF1 *(maj-min)*, OGF2 *(maj-min-7)* and PCR/Gamma methods on the Quaero corpus.

our PCR systems, as all the calculated scores for the Beatles corpus and the Quaero corpus are close.

We shall end this study of the Quaero results with some considerations about the error distribution. Figure 5.13 presents the error distribution for our two DCR methods and our PCR/Gamma method. Just like in the Beatles corpus, we notice that the number of major-minor confusions has been considerably reduced by the introduction of the probabilistic framework. Also, relative and mediant errors are less common in the PCR/Gamma method than in the OGF1 *(maj-min)* method, but more than in the OGF2 *(maj-min-7)*. The rest of the error distribution is very similar to the OGF1 *(maj-min)* one.

5.3.3 Analysis of the correlations between evaluation metrics

We have evaluated our algorithms according to 4 different types of metrics. Each of them allows to estimate one facet of the chord transcription task. Yet, it seems intuitive to think that all these notions might somehow be connected. In order to investigate this, we propose to calculate for every previously tested chord transcriber and every song of the Beatles corpus, a song-version of the AOS, AHD, ACL and ACN. By calculating the correlations between the $11 \times 180 = 1980$ scores obtained for each characteristic (recognition, segmentation, fragmentation and chord vocabulary) we should be able to learn more about the links between these different aspects of the chord transcription task.

As far as recognition accuracy is concerned, we can logically use the Overlap Score OS(s) as a song-version of the AOS. Likewise, we shall use the individual Hamming Distance HD(s) as a segmentation score. The Reduced Chord Length RCL(s) and Reduced Chord Number RCN(s) are not monotonic with performances: we therefore use respectively $|1 - RCL(s)|$ and $|1 - RCN(s)|$ as fragmentation and chord vocabulary score.

For each pair of scores we calculate the Pearson's linear correlation coefficient defined by:

$$r = \left| \frac{\sum_{i=1}^{N_{exp} \times S} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{I \times S} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{I \times S} (y_i - \bar{y})^2}} \right| \tag{5.19}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the score lists, $\bar{x}$ is the mean of all scores, I is the number of chord recognition systems and S the number of songs contained in the corpus. The correlation r ranges from 0 to 1. The higher r, the more correlated the metrics. Table 5.8 presents the results of this experiment.

	segmentation HD(s)	fragmentation RCL(s)	chord vocabulary RCN(s)
recognition OS(s)	0.685	0.253	0.292
segmentation HD(s)		0.454	0.388
fragmentation RCL(s)			0.156

Table 5.8: Correlation between the recognition, segmentation, fragmentation and chord vocabulary metrics.

We notice that the recognition and segmentation metrics are highly correlated. Figure 5.14 confirms this assumption: we clearly see that the better the segmentation, the more accurate the recognition (and vice versa). This observation seems to be less true for other metrics. Still, correlations show that a link between segmentation and fragmentation does exist, while not being as strong as the one with recognition accuracy. In Section 5.3.1 and 5.3.2, we have seen that one of the main improvements brought by our probabilistic framework on the DCR methods is the limitation of the fragmentation issue. Since no new chord boundaries detection algorithm has been introduced in this framework, we can thus think that the better segmentation scores are due to the deletion of fragmented chords. In the same way, according to correlation scores, the good evaluation of the chord vocabulary also helps to get better segmentation.

5.4 Conclusion and discussion

In Chapter 4, we described some simple and efficient deterministic template-based chord recognition systems. As it appears, the fact that the systems only relied on chord definitions and did not detect the relevant chord vocabulary, could lead to fragmented chord transcriptions. The major contribution of this chapter is the building of a probabilistic framework from this deterministic approach, which jointly evaluates the chord probability distribution and the chord sequence. This results in more relevant and compact chord transcriptions.

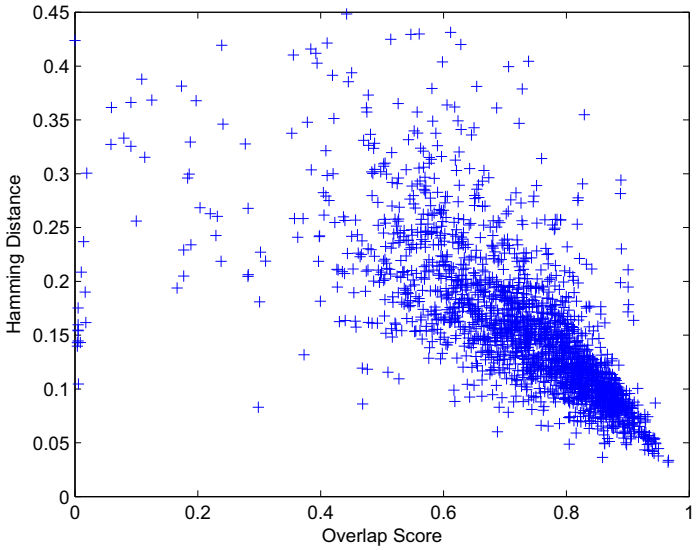


Figure 5.14: Links between Overlap Scores and Hamming Distance calculated on the Beatles corpus with the 11 tested chord recognition systems.

Let us summary the main contributions and conclusions of this chapter:

- The PCR methods still rely only on chord definitions and are therefore explicit template-based systems. As such, they have all the advantages of template-based methods, such as the fact than no training or extensive music knowledge is needed. Thus our PCR methods are neither strongly dependent on the development corpus nor genre or style-specific. Also, the computational time is kept very low which is an interesting characteristic in particular for embedded systems.
- The introduction of chord probabilities in the model allows to accurately evaluate the chord vocabulary, and therefore to produce more accurate, clear and compact chord transcriptions. The fragmentation phenomenon is clearly reduced and many major-minor confusions are corrected.
- The accurate evaluation of chord vocabulary combined with post-processing filtering methods enables to achieve a satisfactory chord boundaries detection and thus a good segmentation.

Related publications

Oudre L., Févotte C., Grenier Y., "Probabilistic template-based chord recognition", *accepted in* IEEE Transactions on Audio, Speech and Language Processing, 2010.

Oudre, L., Févotte, C., Grenier, Y. (2010). Probabilistic framework for template-based chord recognition. *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*. St Malo, France.

Chapter 6

Conclusion

In this last chapter, we propose to summarize the main contributions presented in this manuscript, and to give a number of directions for future work.

6.1 Summary of the contributions

In Chapter 2, we have classified chord recognition methods into four categories: template-based, training-based, music-driven and hybrid. Historically, the first chord transcription methods were template-based. This can be explained by the fact that it is probably the most intuitive way of detecting chords in a chromagram. However, these methods were not given much attention since the development of more complex methods, incorporating music theory elements or requiring annotated data. Indeed, recent chord recognition systems tend more-and-more to take into account numerous adjacent notions such as key, rhythm, structure, etc. Yet, template-based methods have many advantages:

- They do not need training or annotated data.
- They do not require extensive music theory knowledge.
- They do not depend on a particular corpus or music genre.
- They have a low computational time.

The main contribution presented in this thesis is the development of two novel template-based chord recognition methods, whose performances compare well to complex state-of-the-art methods, while being simpler and more straightforward.

In Chapter 4, we have described a deterministic approach to chord recognition. This approach is based on the joint use of chord templates, measures of fit and post-processing filtering. Several tests have been run in order to understand the influence of these parameters. We have shown that the introduction of harmonics in the chord templates does not significantly improve results. Also, we have demonstrated that the introduction of filtering on the recognition criterion clearly enhances the performances and allows to capture the long-term trend in the chord changes. Finally, a comparison with a number of state-of-the-art methods has been performed, showing that our DCR methods compare favorably to state-of-the-art and that the parameter choice done on the Beatles corpus can also apply to other corpora.

In Chapter 5, we have proposed a probabilistic approach for chord recognition. This approach, while building on components already used in the DCR methods, offers a novel statistical framework that explicitly models chord occurrences in songs as probabilistic events. The

evaluation of the chord probabilities allows to extract a relevant and sparse chord vocabulary for every song. This chord vocabulary was used in order to address some existing weaknesses of the deterministic system, in particular concerning segmentation and harmonic relevance of the transcriptions, leading to better performances. The main effect of the introduction of the chord probabilities in the model is the elimination of most spurious chords detected by our previous methods (i.e., the probability of chords absent from the song goes to zero). This leads to more compact and readable chord transcriptions while improving the detection scores. A detailed comparison with the state-of-the-art has been presented, showing that our PCR methods outperform the state-of-the-art, especially concerning the evaluation of the chord vocabulary.

Another important contribution of this thesis is the description of various metrics for the evaluation of chord recognition methods. These metrics allow to assess different aspects of the chord recognition task and to give a more precise description of the system performances. Also, the detailed comparison between numerous state-of-the-art methods allows to understand the current issues of the chord recognition task.

6.2 Future work and perspectives

The relative simplicity of our approach and the satisfactory results obtained by our systems allows to think of many perspectives for future work.

Rhythm and structure

One of the most straightforward enhancements we can think of is the introduction of music knowledge in our systems. Note that this would transform our template-based approach into music-driven approach. Until now, the temporal evolution of the musical piece is partially captured by the post-processing filtering applied on the recognition criterion. Yet, it is clear that this processing does not take into account the real rhythmic structure of music, but rather implicitly informs the system of the expected chord duration. One idea was to introduce the beat information: we already tested this heuristic and obtained disappointing results. However, we believe that more complex rhythmic information such as measures or downbeats could improve the performances of our systems. An interesting solution would be to integrate notions such as tempo or measure in the probabilistic framework proposed in Chapter 5, which could lead a better understanding of the rhythmic aspect of music. Taking into account rhythm can also allow to access to higher-level notions such as structure (chorus and verse), which in turn, can be used to improve chord transcription.

Adaptive parameter choice

All the parameters in our methods are estimated through trial-and-error: number of harmonics, measures of fit/observation models, hyper-parameters, filtering method, neighborhood sizes, etc. It would be interesting to integrate the parameter choice into our algorithms, for example by using model selection heuristics. In particular, the hyper-parameters in the probabilistic approach could be estimated within the EM-algorithm. Although this can be easily done for the PCR/Gaussian method, the automatic estimation of the PCR/Gamma hyper-parameter is not trivial but could be envisaged with numerical methods.

Additional outputs: chord types, lead sheets,...

This document only assesses chord recognition systems with a major and minor chord dictionary (following MIREX 2008, 2009 & 2010). We could however think of outputting other

chord types such as dominant seventh, diminished or augmented in order to produce more precise chord transcriptions. Indeed, we can theoretically consider any chord type by introducing appropriate chord templates in our systems. Yet we have seen by introducing dominant seventh templates in our DCR systems that this task may not be as easy as it seems, since complex chord templates sometimes do not respond as expected. Our methods could also be combined with principal melody extraction systems in order to produce complete lead sheets (Weil et al. (2009)). More generally, we think that in the future chord transcribers will be expected to produce more precise and complete chord transcriptions, which would pretty much look like hand-made transcriptions.

Chromagram

For our simulations we used the chromagram proposed by Bello & Pickens (2005). Nevertheless, recent works (Mauch & Dixon (2010)) have obtained large score improvements by changing the front-end of their chord recognition systems. It seems that the production of clearer and sparser chromagrams helps to significantly enhance the chord transcriber. Although it is not sure that our template-based methods would respond well to sparse chromagrams (because for example of numerical instabilities), it would be interesting to figure out whether this new input features can improve our performances.

From chord vocabulary to key

In Chapter 3 and 5, we have introduced the notion of chord vocabulary. We have indeed shown that taking into account the chord vocabulary allows to produce better chord transcriptions. As described in Chapter 1, the notion of key in popular modern music is sometimes blurred or hard to define as it is not rare to find off-key chords in pop songs. For example, the chord progression $Cmajor - Fminor - Gmajor$ does not theoretically correspond to any key and yet can be found in popular songs. However, the notion of chord vocabulary allows to take into account both off-key chords and modulations. Therefore, we think that chord vocabulary can be a novel and relevant description of the harmony of a song. Indeed, this notion is wider than key, and can be relevant for any type of music. The vector of chord probabilities estimated in our probabilistic approach reflects the *harmonic profile* of the song and may be of interest for applications such as key estimation or can serve as a descriptor for MIR tasks.

Index

List of Publications

Journal papers

Oudre L., Févotte C., Grenier Y., "Probabilistic template-based chord recognition", *accepted in* IEEE Transactions on Audio, Speech and Language Processing, 2010.

Oudre L., Grenier Y., Févotte C., "Chord recognition by fitting rescaled chroma vectors to chord templates", *submitted to* IEEE Transactions on Audio, Speech and Language Processing, 2009.

Conference papers

Oudre, L., Févotte, C., Grenier, Y. (2010). Probabilistic framework for template-based chord recognition. *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP).* St Malo, France.

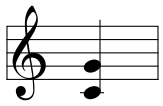
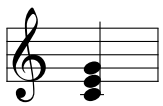
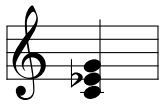
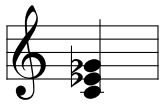
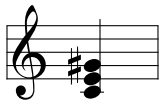
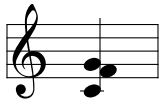
Rocher T., Robine M., Hanna P., Oudre L. (2010). Concurrent Estimation of Chords and Keys from Audio. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR).* Utrecht, Netherlands.

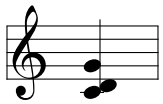
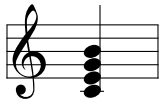
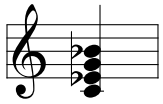
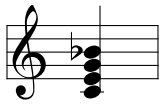
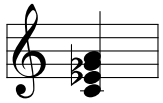
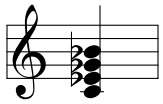
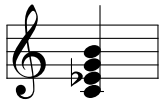
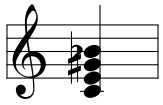
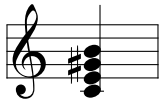
Oudre, L., Grenier, Y., Févotte, C. (2009). Chord recognition using measures of fit, chord templates and filtering methods. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* New York, USA.

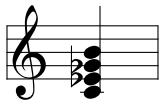
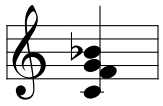
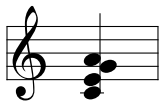
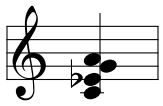
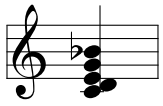
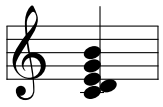
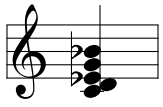
Oudre, L., Grenier, Y., Févotte, C. (2009). Template-based chord recognition : influence of the chord types. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR).* Kobe, Japan.

Appendix

Description of the main chord types

	Chord name	Notational forms	Composition	Definition
Dyads	single	C5	C, G	
Tryads	major	C, CM, Cma, Cmaj, CΔ	C, E, G	
	minor	Cm, Cmi, Cmin, C-	C, E$\flat$, G	
	diminished	Cdim, Cm($\flat$5), Cmin($\flat$5), C-($\flat$5), C$^\circ$	C, E$\flat$, G$\flat$	
	augmented	Caug, C$^+$, C+	C, E, G$\sharp$	
	suspended fourth	Csus4	C, F, G	

	Chord name	Notational forms	Composition	Definition
Tryads	suspended 2nd	Csus2	C, D, G	
Tetrads	major seventh	Cmaj7, CM7, C$^{\Delta 7}$, C^{j7}, C+7	C, E, G, B	
	minor seventh	Cmin7, Cm7, C-^{7}, C^{-7}	C, E♭, G, B♭	
	dominant seventh	C7, C^{7}, Cdom7	C, E, G, B♭	
	diminished seventh	Cdim^{7}, C$^{\circ}$7	C, E♭, G♭, A	
	half diminished seventh	Chdim7, C$^{\emptyset}$7, Cm7$^{\flat 5}$, C-7$^{(\flat 5)}$	C, E♭, G♭, B♭	
	minor major seventh	Cminmaj7, Cm(Maj7), C-$^{(j7)}$, Cm♯7, C-$^{\Delta}$7, C-maj7, Cm^{M7}, Cm^{maj7}, C-^{M7}	C, E♭, G, B	
	augmented seventh	Caug7, C+7, C7+, C7+5, C7♯5	C, E, G♯, B♭	
	augmented major seventh	C+(Maj7), CMaj7+5, CMaj7♯5, C+^{j7}, C$_{\Delta}$+7, Caug^{maj7}, C$^{\Delta}$+	C, E, G♯, B	

	Chord name	Notational forms	Composition	Definition
Tetrads	diminished major seventh	CdimM7, C°M7	C, E♭, G♭, B	
	seventh suspended fourth	C7^{sus4}, Csus4/7	C, F, G, B♭	
	major sixth	C6, Cmaj6	C, E, G, A	
	minor sixth	Cm6, Cmin6	C, E♭, G, A	
Pentads	dominant ninth	C9	C, E, G, B♭, D	
	major ninth	C^{M9}, Cmaj9, C$^{\Delta 9}$	C, E, G, B, D	
	minor ninth	Cm9, Cmin9, C-^{9}	C, E♭, G, B♭, D	

Bibliography

Abdallah, S., Noland, K., Sandler, M., Casey, M., & Rhodes, C. (2005). Theory and evaluation of a Bayesian music structure extractor. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 420–425). London, UK.

Bach, J. (1722). *Das wohltemperierte Klavier*.

Baron, M. (1973). *Précis pratique d'harmonie*. Brault et Bouthillier.

Bello, J. (2007). Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*. Vienna, Austria.

Bello, J., & Pickens, J. (2005). A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 304–311). London, UK.

Benward, B., & Saker, M. (2003). *Music in theory and practice*, vol. 1. McGraw-Hill Humanities/Social Sciences/Languages.

Blankertz, B. (2001). The Constant Q Transform. `http://ida.first.fhg.de/publications/drafts/Bla_constQ.pdf`.

Brown, J. (1991). Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, *89*(1), 425–434.

Brown, J., & Puckette, M. (1992). An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, *92*(5), 2698–2701.

Burgoyne, J., Pugin, L., Kereliuk, C., & Fujinaga, I. (2007). A cross-validated study of modelling strategies for automatic chord recognition in audio. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 251–254). Vienna, Austria.

Burgoyne, J., & Saul, L. (2005). Learning harmonic relationships in digital audio with Dirichlet-based hidden markov models. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 438–443). London, UK.

Cabral, G., Briot, J., & Pachet, F. (2005). Impact of distance in pitch class profile computation. In *Proceedings of the Brazilian Symposium on Computer Music*, (pp. 319–324). Belo Horizonte, Brazil.

Cheng, H., Yang, Y., Lin, Y., Liao, I., & Chen, H. (2008). Automatic chord recognition for music classification and retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, (pp. 1505–1508). Hannover, Germany.

Chuan, C., & Chew, E. (2005). Polyphonic audio key finding using the spiral array CEG algorithm. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, (pp. 21–24). Amsterdam, Netherlands.

Davies, M., & Plumbley, M. (2007). Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, *15*(3), 1009–1020.

Dempster, A., Laird, N., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological).*, *19*(1), 1–38.

Downie, J. (2008). The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, *29*(4), 247–255.

Ellis, D. (2009). The 2009 Labrosa pretrained audio chord recognition system. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/DE.pdf`.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701.

Fujishima, T. (1999). Realtime chord recognition of musical sound: a system using Common Lisp Music. In *Proceedings of the International Computer Music Conference (ICMC)*, (pp. 464–467). Beijing, China.

Gómez, E. (2006a). *Tonal description of music audio signals*. Ph.D. thesis, Universitat Pompeu Fabra.

Gómez, E. (2006b). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, *18*(3), 294–304.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, *30*(2), 159–171.

Goto, M. (2003). A chorus-section detecting method for musical audio signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (pp. 437–440). Hong Kong, China.

Goto, M., & Muraoka, Y. (1999). Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. *Journal of Speech Communication*, *27*, 311–335.

Harte, C., & Sandler, M. (2005). Automatic chord identification using a quantised chromagram. In *Proceedings of the Audio Engineering Society Convention*. Barcelona, Spain.

Harte, C., & Sandler, M. (2009). Automatic chord recognition using quantised chroma and harmonic change separation. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/harte_mirex09.pdf`.

Harte, C., Sandler, M., Abdallah, S., & Gomez, E. (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 66–71). London, UK.

Harte, C., Sandler, M., & Gasser, M. (2006). Detecting harmonic change in musical audio. In *Proceedings of the ACM Workshop on Audio and Music Computing Multimedia*, (pp. 21–26). Santa Barbara, CA.

Itakura, F., & Saito, S. (1968). Analysis synthesis telephony based on the maximum likelihood method. In *Proceedings of the International Congress on Acoustics*, (pp. 17–20). Tokyo, Japan.

Izmirli, O. (2005). Template based key finding from audio. In *Proceedings of the International Computer Music Conference (ICMC)*. Barcelona, Spain.

Khadkevich, M., & Omologo, M. (2008). Mirex audio chord detection. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2008/khadkevich_omologo_final.pdf`.

Khadkevich, M., & Omologo, M. (2009a). Improved automatic chord recognition. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/KO.pdf`.

Khadkevich, M., & Omologo, M. (2009b). Use of hidden markov models and factored language models for automatic chord recognition. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, (pp. 561–566). Kobe, Japan.

Kramer (1956). Extension of multiple range tests to group means with unequal number of replications. *Biometrics*, *12*(3), 307–310.

Krumhansl, C. (1990). *Cognitive Foundations of Musical Pitch*. Oxford University Press, USA.

Krumhansl, C., & Kessler, E. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial map of musical keys. *Psychological Review*, *89*(4), 334–368.

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*(1), 79–86.

Lee, K. (2006a). Automatic chord recognition from audio using enhanced pitch class profile. In *Proceedings of the International Computer Music Conference (ICMC)*. New Orleans, USA.

Lee, K. (2006b). Identifying cover songs from audio using harmonic representation. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2006/CS_lee.pdf`.

Lee, K., & Slaney, M. (2008). Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech and Language Processing*, *16*(2), 291–301.

Lerdahl, F. (2001). *Tonal pitch space*. Oxford University Press.

Maddage, N., Xu, C., Kankanhalli, M., & Shao, X. (2004). Content-based music structure analysis with applications to music semantics understanding. In *Proceedings of the ACM international conference on Multimedia*, (pp. 112–119). New York, USA.

Mauch, M., & Dixon, S. (2008). A discrete mixture model for chord labelling. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 45–50). Philadelphia, USA.

Mauch, M., & Dixon, S. (2010). Simultaneous estimation of chords and musical context from audio. *accepted in IEEE Transactions on Audio, Speech and Language Processing*.

Mauch, M., Noland, K., & Dixon, S. (2009a). Mirex submissions for audio chord detection (no training) and structural segmentation. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/ACD_SS_mauch.pdf`.

Mauch, M., Noland, K., & Dixon, S. (2009b). Using musical structure to enhance automatic chord transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, (pp. 231–236). Kobe, Japan.

Noland, K., & Sandler, M. (2006). Key estimation using a hidden Markov model. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 121–126). Victoria, Canada.

Oudre, L., Grenier, Y., & Févotte, C. (2009a). Mirex chord recognition system. system 1 : major and minor chords. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/OGF1.pdf`.

Oudre, L., Grenier, Y., & Févotte, C. (2009b). Mirex chord recognition system. system 2 : major, minor and dominant chords. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/OGF2.pdf`.

Papadopoulos, H., & Peeters, G. (2007). Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, (pp. 53–60). Bordeaux, France.

Papadopoulos, H., & Peeters, G. (2008). Simultaneous estimation of chord progression and downbeats from an audio file. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (pp. 121–124). Las Vegas, USA.

Papadopoulos, H., & Peeters, G. (2009). Joint estimation of chord and downbeats. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/PPupdated.pdf`.

Pardo, B., & Birmingham, W. (2002). Algorithms for chordal analysis. *Computer Music Journal*, *26*(2), 27–49.

Pauwels, J., Varewyck, M., & Martens, J.-P. (2008). Audio chord extraction using a probabilistic model. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2008/mirex2008-audio_chord_detection-ghent_university-johan_pauwels.pdf`.

Pauwels, J., Varewyck, M., & Martens, J.-P. (2009). Audio chord extraction using a probabilistic model. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/PVM.pdf`.

Pauws, S. (2004). Musical key extraction from audio. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 96–99). Barcelona, Spain.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, *2*(6), 559–572.

Peeters, G. (2006). Musical key estimation of audio signal based on hidden Markov modeling of chroma vectors. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, (pp. 127–131). Montreal, Canada.

Peeters, G. (2007). Template-based estimation of time-varying tempo. *EURASIP Journal on Advances in Signal Processing*, *2007*(8), 158–171.

Purwins, H. (2005). *Profiles of pitch classes circularity of relative pitch and key - experiments, models, computational music analysis, and perspectives*. Ph.D. thesis, Elektrotechnik und Informatik der Technischen Universität Berlin.

Purwins, H., Blankertz, B., & Obermayer, K. (2000). A new method for tracking modulations in tonal music in audio data format. In *Proceedings of the International Joint Conference on Neural Networks*, (pp. 270–275). Como, Italia.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Rameau, J. (1722). *Traité de l'harmonie réduite à ses principes naturels*.

Reed, J., Ueda, Y., Siniscalchi, S., Uchiyama, Y., Sagayama, S., & Lee, C. (2009). Minimum classification error training to improve isolated chord recognition. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, (pp. 609–614). Kobe, Japan.

Rocher, T., Robine, M., Hanna, P., & Strandh, R. (2009). Dynamic chord analysis. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/RRHS.pdf`.

Ryynänen, M., & Klapuri, A. (2008a). Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, *32*(3), 72–86.

Ryynänen, M., & Klapuri, A. (2008b). Chord detection method for MIREX 2008. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2008/CD_ryynanen.pdf`.

Sailer, C., & Rosenbauer, K. (2006). A bottom-up approach to chord detection. In *Proceedings of the International Computer Music Conference (ICMC)*, (pp. 612–615). New Orleans, USA.

Sheh, A., & Ellis, D. (2003). Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 185–191). Baltimore, MD.

Shenoy, A., Mohapatra, R., & Wang, Y. (2004). Key determination of acoustic musical signals. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, (pp. 1771–1774). Taipei, Taiwan.

Shenoy, A., & Wang, Y. (2005). Key, chord, and rhythm tracking of popular music recordings. *Computer Music Journal*, *29*(3), 75–86.

Sumi, K., Itoyama, K., Yoshii, K., Komatani, K., Ogata, T., & Okuno, H. (2008). Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 39–44). Philadelphia, USA.

Temperley, D. (2001). *The cognition of basic musical structures*. MIT Press.

Tukey, J. (1953). The problem of multiple comparisons. Princeton University. Unpublished manuscript.

Varewyck, M., Pauwels, J., & Martens, J.-P. (2008). A novel chroma representation of polyphonic music based on multiple pitch tracking techniques. In *Proceedings of the ACM international conference on multimedia*, (pp. 667–670). Vancouver, Canada.

Weil, J., Sikora, T., Durrieu, J.-L., & Richard, G. (2009). Automatic generation of lead sheets from polyphonic music signals. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. Kobe, Japan.

Weller, A., Ellis, D., & Jebara, T. (2009). Structured prediction models for chord transcriptino of music audio. Abstract of the Music Information Retrieval Evaluation Exchange. `http://www.music-ir.org/mirex/abstracts/2009/WEJ.pdf`.

Yoshioka, T., Kitahara, T., Komatani, K., Ogata, T., & Okuno, H. (2004). Automatic chord transcription with concurrent recognition of chord symbols and boundaries. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 100–105). Barcelona, Spain.

Zenz, V., & Rauber, A. (2007). Automatic chord detection incorporating beat and key detection. In *Proceedings of the IEEE International Conference on Signal Processing and Communications (ICSPC)*, (pp. 1175–1178). Dubai, United Arab Emirates.

Zhang, X., & Gerhard, D. (2008). Chord recognition using instrument voicing constraints. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, (pp. 33–38). Philadelphia, USA.

Zhu, Y., Kankanhalli, M., & Gao, S. (2005). Music key detection for musical audio. In *Proceedings of the International Conference on Multimedia Modeling (MMM)*, (pp. 30–37). Melbourne, Australia.