Extraction d’information rythmique à partir d’enregistrements musicaux

Miguel A. Alonso Arevalo

TELECOM PARIS
école nationale
supérieure des
télécommunications

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

Présenté pour obtenir le grade de docteur de l'Ecole Nationale
Supérieure des Télécommunications

Spécialité : **Signal et Images**

Miguel A. ALONSO–AREVALO

EXTRACTION D'INFORMATION RYTHMIQUE À PARTIR D'ENREGISTREMENTS MUSICAUX

Soutenue le 13 novembre 2006 devant le jury composé de :

Mme. Myriam DESAINTE-CATHERINE	Rapporteur et Président
Mme. Régine ANDRÉ-OBRECHT	Rapporteur
M. Emmanuel SAINT-JAMES	Examinateurs
M. Anssi KLAPURI	
M. Gaël RICHARD	Directeurs de thèse
M. Bertrand DAVID	

A Eloïsa ,
ma belle femme,

A mes parents Ana Miriam et Miguel.

A mes sœurs Annie et Béatrice,
et à mon frère Paul.

Remerciements

Cette thèse est le fruit de plusieurs années de travail au sein du département de Traitement du Signal et des Images (TSI) à l'Ecole Nationale Supérieure des Télécommunications (ENST), à Paris.

J'ai eu la chance d'avoir deux directeurs de thèse, Gaël RICHARD et Bertrand DAVID, qui, par leur encadrement, m'ont apporté leur compétence scientifique et leur passion pour la recherche. Je leur suis redevable d'une somme inestimable de temps et d'énergie, ils ont toujours fait preuve d'une exceptionnelle disponibilité. J'apprécie vraiment beaucoup notre collaboration, qu'ils trouvent ici l'expression de toute ma reconnaissance.

J'adresse aussi tous mes remerciements aux rapporteurs de cette thèse Madame Régine ANDRÉ–OBRECHT de l'IRIT de Toulouse et Madame Myriam DESAINTE–CATHERINE du LaBRI de Bordeaux dont les bonnes suggestions m'ont permis d'améliorer la qualité de mon manuscrit. Au même titre, je remercie les examinateurs, Monsieur Anssi KLAPURI de l'Université de Tampere en Finlande et Monsieur Emmanuel SAINT–JAMES du LIP6.

Je remercie également Monsieur Yves GRENIER, qui m'a accueilli au sein du département TSI dans un environnement idéal pour mon travail de recherche.

J'ai énormément appris au contact de Roland BADEAU tout au long de ce travail de thèse grâce à sa discipline scientifique et à sa grande disposition.

J'adresse toute mon amitié à Mathieu GUILLAUME, mon compagnon de bureau pendant la thèse. Sa compagnie a toujours été un énorme plaisir, ma gratitude pour lui est très profonde.

Je voudrais également exprimer ma reconnaissance à Monsieur Claude MONTIGNY pour son aide pendant la rédaction de la partie en Français de cette thèse.

Travailler au sein du département TSI a été une expérience unique, toutes ces années écoulées au laboratoire n'auraient pas été si enrichissantes sans cette atmosphère chaleureuse et conviviale créée par les doctorants, les enseignants-chercheurs, l'équipe administrative et les stagiaires que j'ai rencontré tout au long de la thèse. J'adresse mes plus vifs remerciements à eux tous pour m'avoir accordé leur amitié.

Ce travail a été rendu en grande partie possible grâce au Peuple Mexicain qui m'a attribué une bourse à travers le Ministère Mexicain Pour la Science et la Technologie (CONACYT) et aussi grâce au Gouvernement Français qui m'a donné un support financier à travers le project Music Discover.

Je voudrais terminer ces remerciements par les personnes les plus importantes dans ma vie : mon amoureuse femme Eloïsa car elle m'a toujours encouragé et parce qu'elle est toujours là pour moi. Mes parents Ana Miriam et Miguel, mes sœurs Béatrice et Annie et mon frère Paul, ils m'ont toujours soutenu, encore plus quand je me suis éloigné du terreau familial pour m'aventurer en *terra incognita* pour continuer mes études.

Contents

List of Figures

List of Tables

Acronyms

ACF	Autocorrelation Function
AR	Autoregressive
ARM	Average Response Meter
COW	Correlation Optimized Warping
CSD	Complex Spectral Difference
dB	Decibels
DFT	Discrete Fourier Transform
EDS	Exponentially Damped Sinusoid
ERB	Equivalent Rectangular Bandwith
EVD	Eigenvalue Decomposition
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
H+N	Harmonic plus Noise
iFFT	Inverse Fast Fourier Transform
iid	Independent and identically-distributed
IIR	Infinite Impulse Response
IOI	Inter Onset Interval
ITU	International Telecommunications Union
MDCT	Modified Discrete Cosine Transform
MIREX	Music Information Retrieval Evaluation eXchange
ML	Maximum Likelihood
OLA	Overlap-Add
PSD	Power Spectral Density
SD	Spectral Difference
SEF	Spectral Energy Flux
STFT	Short Time Fourier Transform
SS	Spectral Sum
SP	Spectral Product
SVM	Support Vector Machines
TFR	Time–Frequency Representation
TWM	Two-way Mismatch
WGN	White Gaussian Noise

Synthèse des travaux exposés dans le manuscrit

La musique est un phénomène omniprésent de nos vies quotidiennes. En fait, la plupart des gens ont une capacité naturelle pour l'apprécier, qu'ils aient ou non suivi une formation musicale. Etudier comment les humains appréhendent la musique est un sujet de recherche fascinant et le voile se lève à peine sur de nombreuses questions afférentes. Parmi les travaux du domaine une large catégorie rassemble des méthodes qui visent à imiter, à simuler des processus cognitifs humains, ou encore à réaliser des taches similaires, à l'aide d'un ordinateur. Un des Graal du traitement numérique de l'information appliqué à l'analyse musicale consiste à réaliser une transcription automatique précise d'un enregistrement musical. Ce projet implique de résoudre un ensemble de tâches phares en recherche automatique d'informations musicales (MIR[1]) : détection de hauteur et de tonalité, détection des attaques, analyse du déroulement temporel, estimation du nombre de sources, identification des articulations et des expressions musicales, ... Klapuri & Davy (2006) présentent la transcription musicale comme un problème d'*ingénierie inverse* où il faut inférer le "code source" à partir duquel le signal de musique a été généré.

La transcription musicale automatique est ainsi un point d'intersection de plusieurs disciplines scientifiques : informatique, acoustique, musicologie, psychoacoustique, traitement du signal. Son champ d'application est vaste. Citons par exemple :

- Recherche et indexation automatique d'extraits musicaux. Le but ici est de localiser des extraits dans une large collection de pièces selon des critères de similarité ou selon des requêtes de l'utilisateur (genre musical, tempo, interprète,...). Un autre versant de cette recherche concerne l'étiquetage automatique de ces pieces.
- Codage structuré de signaux musicaux. Cette application présente beaucoup de points communs avec la précédente, mais dans ce cas le but est de développer des codeurs audio capables de gérer simultanément des requêtes de recherche d'extraits et la compression de ceux-ci.
- Edition musicale assistée par ordinateur. Il s'agit, par exemple, d'opérations telles que le copier–coller de signaux audio, les effets spéciaux commandés par la musique, la synchronisation rythmique, etc.
- Les applications où l'homme et la machine interagissent. Dans cette catégorie nous trouvons le suivi de partition musicale ou encore l'accompagnement automatique du musicien par l'ordinateur.

[1] Le terme anglais pour cette opération est connu sous le nom de *Music Information Retrieval* (MIR).

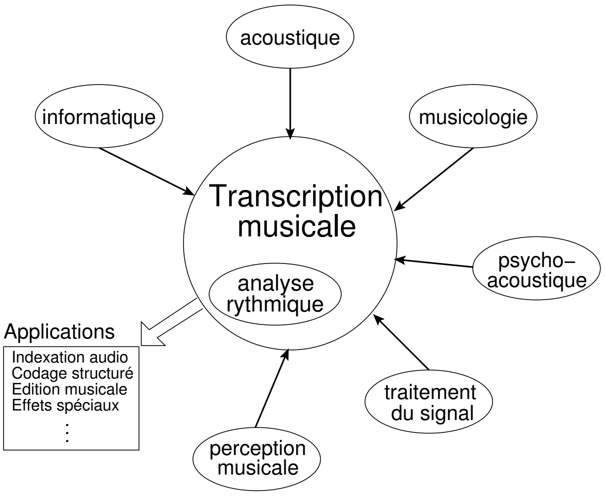


Figure 1: La transcription musicale vue comme point d'intersection de plusieurs disciplines scientifiques. L'analyse du rythme constitue l'un des sous-ensembles.

Le but principal de ce travail de thèse est de présenter une méthode pour analyser automatiquement le rythme des signaux musicaux à l'aide d'un ordinateur, ce qui constitue une des sous-tâches nécessaires de la transcription. L'analyse du rythme s'appuie sur les mêmes disciplines scientifiques et partage largement les directions applicatives mentionnées ci-dessus. La Figure 1 illustre ce contexte.

Le rythme musical

Pour Carterette & Kendall (1999), la musique est formée de trois éléments essentiels : la mélodie, l'harmonie et le rythme. Le rythme et l'harmonie sont vus en tant que parties complémentaires, un même morceau de musique pouvant être au besoin analysé selon un seul des deux aspects, rythmique ou harmonique.

Le rythme peut s'avérer comme un concept paradoxal : d'un côté tout un chacun peut le ressentir, mais d'un autre côté sa définition précise se heurte à des difficultés parfois importantes. Certains chercheurs vont jusqu'à nier l'existence d'une définition consensuelle. En fait, il est possible d'en trouver plusieurs selon le sujet d'intérêt. Après une revue de la littérature liée à ce sujet, nous adoptons comme définition du rythme musical la compilation recueillie par Parncutt & Drake (2001). Selon eux, la perception du rythme implique une organisation perceptive et cognitive des événements temporels, de manière à situer chaque événement sonore par rapport à ceux qui se sont déjà produits (mémoire) et ceux à venir (attente). Maintenant, nous allons brièvement décrire les processus cognitifs prenant place aux différents niveaux temporels.

⋆ **Organisation au niveau de la surface.** Le signal audio est segmenté en événements séparés correspondant aux points d'attaque des événements musicaux tels que les débuts de notes où les changements d'accords. Dans ce contexte, nous appelons "l'attaque perceptive" l'instant où l'événement est perçu comme se produisant.

L'intervalle de temps entre l'attaque d'un événement et l'attaque de l'événement suivant s'appelle l'intervalle inter-*onset* (IOI).

- ⋆ **Groupement et métrique.** Les événements d'un rythme peuvent être hiérarchiquement organisés de deux manières différentes connues sous le nom de *groupement* et *métrique*. D'un point de vue perceptif, le rythme est caractérisé par (et est parfois même défini comme) une combinaison de ces deux formes d'organisation. Celles-ci peuvent être analysées séparément mais elles sont étroitement liées dans une analyse du rythme. Le *groupement* traduit l'existence d'assemblage de notes formant des motifs musicaux courts qui se combinent pour former des phrases musicales, qui à leur tour s'agrègent pour former des passages, des mouvements s'étendant éventuellement jusqu'à former des morceaux entiers.

  La *métrique* traduit une forme d'organisation perceptive basée sur la régularité temporelle (battement ou pulsation fondamentale) et sur les éventuels schémas d'accentuation récurrents qu'elle contient. La sensation de pulsation peut être évoquée à n'importe quel niveau d'une séquence formée d'événements sonores et peut être exprimée concrètement par le battement (claquement des doigts ou mouvement synchrone, marquage du temps au pied, battue,...). Le processus perceptif d'extraction de la régularité s'apparente alors à une synchronisation de notre horloge interne avec la pièce de musique. Par exemple, lors d'un intermède de silence, l'auditeur anticipe une continuité de la sensation de pulsation. De manière générale son attention est dirigée sur les instants où les événements sont en rapport étroit avec la pulsation attendue.

  Notre travail de recherche s'intéresse plus particulièrement à la manière d'estimer la métrique d'un morceau de musique.

- ⋆ **Prédominance.** La structure métrique des signaux musicaux se décompose classiquement à plusieurs niveaux hiérarchiques de pulsation, aussi appelées "couches rythmiques".

  Pour Lerdahl & Jackendoff (1983), ces différents niveaux de pulsation évoquent différentes régularités temporelles dans l'esprit de l'auditeur; et ce dernier tend à préférer un niveau de tempo modéré (période aux alentours de 500 ms) et à percevoir les autres niveaux de façon hiérarchique (et par conséquent tous les événements) par rapport à cette couche en particulièr. Ce niveau distinctif est connu comme le *tactus*.

- ⋆ **Accents.** Souvent le terme d'*accent* fait référence à la notion de sonie, car l'attention de l'auditeur est attirée par un événement audio plus fort ou plus faible que son contexte. L'accent est ainsi vu comme synonyme de prépondérance de l'événement. Plus généralement, d'après Jones (1987), toute propriété qui rend un événement sonore plus remarquable que les événements adjacents peut être considéré comme un accent. Par exemple, il peut s'agir des variations rapides et légères de la dynamique d'un signal (*i.e.*, trémolo) ou des petites modulations de la hauteur d'un son (*i.e.*, vibrato).

- ⋆ **Organisation rythmique et tempo.** Selon Handel (1993), la façon dont les humains perçoivent l'organisation d'un morceau de musique dépend du tempo auquel celui-ci est exécuté. Le tempo peut affecter le groupement et la métrique. Handel considère que le niveau métrique auquel le tactus est situé dépend du tempo, car les distributions des cadences de battement en musique ne sont pas liées au tempi annotés dans les partitions musicales. Par exemple, un auditeur pourrait taper des croches quand un morceau est joué lentement ou des noires si le même morceau est joué deux fois plus vite.

La structure métrique

Nous avons mentionné auparavant que le rythme est hiérarchiquement organisé de deux manières différentes : le groupement et la métrique (Lerdahl & Jackendoff, 1983). Nous avons aussi signalé que dans ce travail nous nous intéressons à ce dernier. Dans cette structure hiérarchique, la notion de métrique aide à organiser la musique en une suite d'impulsions, créant avec ceci une base de temps musical. Si le tempo est constant, la base de temps est isochrone, *i.e.*, l'intervalle de temps entre deux impulsions consécutives quelconques est constant.

D'après Lerdahl & Jackendoff (1983), dans la musique occidentale traditionnelle, la hiérarchie métrique est établie à partir de deux propriétés fondamentales :

1. chaque impulsion dans un niveau métrique donné coïncide avec une impulsion à tous les niveaux métriques inférieurs;

2. Le rapport entre deux niveaux métriques quelconques obéi une relation binaire ou ternaire; c'est à dire, les périodes des impulsions entre deux niveaux métriques consécutifs sont reliées par un facteur deux ou trois.

La Figure 2 présente un exemple sur les propriétés de la structure métrique et ses niveaux hiérarchiques. Cette figure montre un arrangement de plusieurs couches métriques formant une *grille métrique*. Les niveaux métriques sont empilés les uns sur les autres, avec les niveaux inférieurs situés dans la partie basse et les niveaux supérieurs dans la partie haute. Cette figure montre aussi la structure cognitive correspondant à une métrique $\binom{3}{4}$. Elle inclut des impulsions au niveau de noires (le tactus), des blanches pointées (la mesure), et aussi des impulsions au niveau de la croche. Nous considérons qu'après le tactus, le prochain niveau en importance dans la structure métrique et le *tatum*. Ce terme désigne le niveau métrique le plus bas (voir Figure 2) et dans la pratique il indique la pulsation qui coïncide le mieux avec toutes les attaques.

Panorama actuel de l'informatique dans l'analyse du rythme

Quand les gens écoutent certains types de musique, ils ressentent immédiatement que celle-ci possède un *beat* (battement), qu'ils peuvent claquer des doigts pour l'accompagner. Le plus important c'est que, d'une certaine manière (consciemment ou inconsciemment), ils perçoivent des motifs régulièrement espacés et qu'ils peuvent se synchroniser avec. Depuis quelques années, le but d'un groupe de chercheurs en informatique musicale a été de répéter ce processus en utilisant des machines et de leur *apprendre* comment la musique est organisée en beats. Les raisons pour soutenir cette idée ont déjà été soulignées au début de ce chapitre. De façon générale, nous pouvons définir l'analyse automatique du rythme comme une tentative à reproduire de manière artificielle le processus par lequel les humains *appréhendent* le rythme.

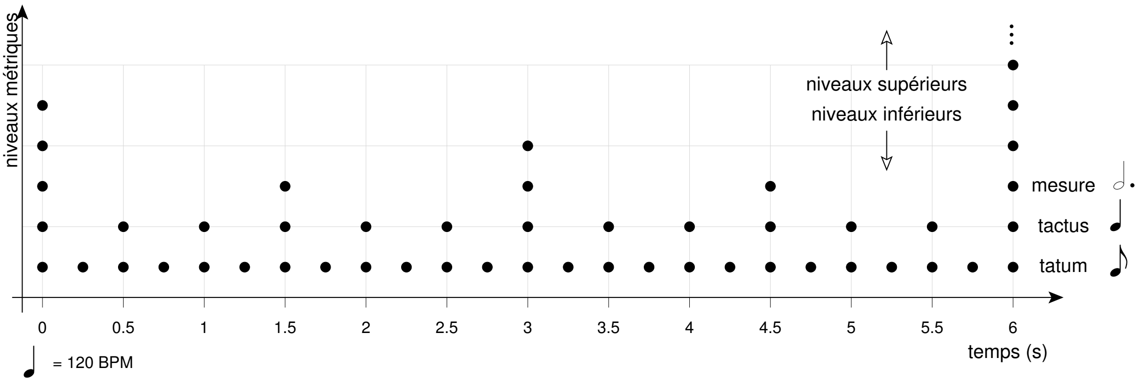


Figure 2: Exemple d'une grille métrique.

D'après la définition du rythme musical présentée dans §, pour qu'un algorithme conçu pour analyser le rythme soit complet, il doit accomplir les tâches suivantes : décomposer la musique en événements sonores isolés, estimer l'importance liée à chaque événement, séparer syntaxiquement la musique en motifs, trouver la couche métrique à laquelle appartient chaque événement, identifier et retrouver les patrons qui se répètent et être capable de s'adapter aux changements de tempo liés à l'interprétation musicale. Actuellement, un tel système n'existe pas, toutefois nous considérons qu'il est envisageable dans un avenir proche pour la musique avec un rythme prononcé et dans plusieurs années pour les cas plus exigeants. Tels que, les pièces de piano de Chopin ou de Mendelssohn dont les changements de tempo sont très nombreux à l'intérieur de courts intervalles du temps.

Les méthodes actuelles d'analyse de rythme par ordinateur peuvent être classées de plusieurs manières différentes. Cependant, la différence la plus importante entre toutes ces approches est la nature du signal musical qu'elles traitent. Les premières méthodes, plus connues comme *modèles symboliques*, utilisaient comme entrée une représentation symbolique du signal audio. Plus précisément, l'entrée de ces systèmes se compose d'un ensemble de données formé des *instructions* qui décrivent les événements musicaux à interpréter et parfois cette entrée comporte aussi des informations pour rendre les événements audibles. Sans doute, l'exemple le plus connu est celui du format MIDI. A présent, puisque la grande majorité des signaux musicaux sont disponibles sous forme numérique (ou plus récemment aussi en plusieurs formats compressés comme MP3 ou AAC), la plupart des méthodes récemment développées ont opté pour traiter directement des enregistrement musicaux. Ces méthodes sont connues comme *modèles acoustiques*. La méthode que nous développons dans ce travail de thèse appartient à cette deuxième catégorie.

Principe général de l'analyse automatique du rythme

Le principe général de l'estimation automatique du rythme est composé de quatre étapes.

- ⋆ D'abord, le degré d'accentuation musical en fonction du temps doit être mesuré, c'est-à-dire que les notes ou accords doivent être détectés. Dans le cas des méthodes symboliques cette tâche n'est pas requise puisque toute l'information sur les événements musicaux est déjà disponible. Pour les méthodes acoustiques cette tâche est nécessaire.

- ⋆ Deuxièmement, les périodes et les phases (emplacements) des pulsations métriques doivent être estimées. Plusieurs méthodes ont été proposées, par exemple : la fonction d'autocorrélation, la transformée de Fourier discrète, réseaux d'oscillateurs.
- ⋆ Troisièmement, le système doit être capable d'identifier les couches métriques. Ceci peut être fait à partir d'une connaissance musicale *a priori* de la distribution des pulsations ou en appliquant des techniques de reconnaissance de formes.
- ⋆ Enfin, les événements musicaux (y compris leur emplacement) liés à chacun des couches métriques doivent être sélectionnés.

Travaux précédents en analyse automatique du rythme

Modèles symboliques. A l'origine, l'analyse automatique du rythme trouve ses fondements dans les modèles cherchant à expliquer la façon dont l'auditeur humain arrive à interpréter la métrique particulière d'un morceau de musique (Lee, 1991). Les premiers modèles traitaient des signaux symboliques et ils étaient basés sur un *ensemble de règles* utilisées pour définir le degré d'accentuation d'un événement musical et, à partir des accents, estimer la métrique du signal musical (Steedman, 1977; Longuet-Higgins & Lee, 1982; Povel & Essens, 1985; Lee, 1991; Parncutt, 1994). Desain & Honing (1999) présente une comparaison détaillée d'une grande partie de ces modèles.

Un autre type d'approche sont les modèles à *hypothèses multiples*. Ils créent un certain nombre de conjectures indépendantes (appelées *agents*) sur les périodes et les emplacements des pulsations. Ensuite, un score dynamique est calculé de façon itérative pour chacune des hypothèses. À la fin de l'analyse, la conjecture ayant le score le plus élevé est considérée comme la périodicité du morceau (Allen & Dannenberg, 1990; Rosenthal, 1992; Dixon, 2001).

Une des manières les plus intuitives d'obtenir des informations sur la métrique d'un signal de musique se base sur des oscillateurs. Les méthodes appliquant cette approche utilisent un *réseau d'oscillateurs* (aussi appelé banc d'oscillateurs), où chacun d'entre eux est construit à partir d'un prototype, mais chaque oscillateur est réglé pour fonctionner à une périodicité différente. Puisque l'oscillateur prototype réagi seulement à une gamme de fréquences très spécifique (en général petite), si la période du signal d'excitation est proche de la fréquence caractéristique, l'oscillateur entre en résonance. Dans ce cas la période est indiquée par l'oscillateur ayant l'énergie de sortie la plus élevée, *i.e.*, celui qui résonne le plus (Large & Kolen, 1994; Toiviainen, 1998).

Un autre type d'approche pour analyser automatiquement la métrique est basé sur des *modèles probabilistes*. Ce genre de technique suppose que les accents ont une nature stochastique et qu'il existe un modèle aléatoire qui commande le processus rythmique dont les paramètres de contrôle doivent être estimés (Raphael, 2001; Cemgil et al., 2001).

Modèles acoustiques. Afin de rendre plus simple la description sur les modèles acoustiques, nous supposons qu'ils se composent de trois étapes. La première effectue la *conversion* du signal du format audio à un type du signal appelé "fonction de détection" qui porte l'information sur les événements musicaux. La sortie de ce module est formée d'une suite de pulsations qui indiquent les endroits où se produisent les accents. La deuxième étape est connue comme le bloc d'*induction de la pulsation* et elle s'occupe d'estimer la périodicité des accents musicaux. La dernière étape se charge de suivre les variations temporelles des accents musicaux. Plusieurs variantes ont été proposées pour chacune de ces étapes.

⋆ **Conversion.** La manière la plus directe d'estimer le profil d'accentuation musicale d'un enregistrement consiste à calculer son *enveloppe énergétique*. C'est à dire, la somme des carrés des échantillons sur des segments courts du signal audio. Cette méthode donne des bons résultats pour la musique percussive, mais elle s'avère peu efficace pour traiter des cas plus complexes (Dixon, 2001; Gouyon et al., 2002; Eck & Casagrande, 2005).

Banc de filtres. Parmi les techniques les plus utilisées pour calculer la fonction de détection nous trouvons le banc de filtres. Cette méthode décompose le signal d'entrée en canaux fréquentiels. Puis, dans chaque canal la dérivée de l'enveloppe énergétique est calculée, et après toutes les enveloppés énergétiques sont intégrées pour former le profil d'accentuation musicale. Les bancs de filtres les plus courants effectuent des décompositions logarithmiques ou autres types de décomposition liées à la perception humaine (Scheirer, 1998; Seppänen, 2001b; Uhle & Herre, 2003). Dans cette catégorie on trouve aussi des approches qui utilisent la Transformée de Fourier à Court Terme comme banc de filtres (Laroche, 2001, 2003; Klapuri et al., 2006; Jehan, 2004).

Attributs de bas niveau. Des travaux plus récents ont exploré l'utilisation d'attributs de bas niveau pour identifier les pulsations présentes dans les signaux musicaux. Le principe de fonctionement qui a motivé ces méthodes se base sur les travaux de classification développés dans le domaine de traitement de la parole (voir par exemple (Jensen & Andersen, 2003; Sethares et al., 2005; Gouyon, 2005)).

⋆ **Estimation de la périodicité.** Plusieurs types de méthodes ont été proposées pour estimer la périodicité des pulsations. Plusieurs d'entre elles se sont inspirées des modèles symboliques. Dans le contexte des modèles acoustiques on trouve aussi des techniques utilisant les *hypothèses multiples* (Goto & Muraoka, 1994, 1997b; Dixon, 2001), les *réseaux d'oscillateurs* (Scheirer, 1998; Klapuri et al., 2006; Jehan, 2004) ou des *modèles probabilistes* (Laroche, 2001; Hainsworth & Macleod, 2003a; Sethares et al., 2005). D'autres méthodes se basent sur les *histogrammes des intervalles inter-attaque*, où le principe est de calculer et de grouper les distances entre deux attaques. De cette façon, les intervalles qui se ressemblent sont réunis dans une même classe. A la fin de l'analyse, la classe contenant le plus d'éléments est considérée comme la période.

La *fonction d'autocorrélation* est une autre méthode très répandue dans le cadre de l'estmation de la périodicité (Foote & Uchihashi, 2001; Uhle et al., 2004; Peeters, 2005; Davies & Plumbley, 2005).

⋆ **Suivi des variations rythmiques.** Il existe deux méthodologies différentes pour suivre le déroulement temporel du rythme. Gouyon & Dixon (2005), appellent la première "suivi par induction répétée" et, comme l'indique son nom, cette méthodologie consiste à répéter itérativement le processus d'induction de la périodicité en obtenant à chaque itération une seule valeur (ou observation) pour la période des pulsations. Dans ce cas, l'évolution du rythme est obtenue en connectant directement les observations sur la périodicité (Sethares & Staley, 2001; Foote & Uchihashi, 2001; Dixon, 2001; Alonso et al., 2004). L'inconvénient principal de cette approche c'est le manque de continuité entre observations successives.

La deuxième méthodologie est également basée sur un processus d'induction itérative, mais au lieu d'estimer une seule valeur de la périodicité, un ensemble de périodes potentielles est calculé. Dans ce cas, pour trouver la courbe de variation de la période "optimale" à travers le temps, il faut calculer la meilleur façon de connecter les hypothèses successives à chaque itération. Si le problème du suivi est développé dans un cadre déterministe, il peut être résolu par programmation dynamique (Laroche, 2003; Alonso et al., 2005b). Si le problème est développé dans un contexte probabiliste, l'algorithme de Viterbi (Klapuri, 2004; Peeters, 2005; Klapuri et al., 2006) ou aussi des techniques de filtrage particulier peuvent être employés (Hainsworth & Macleod, 2003a; Sethares et al., 2005).

Evaluation

D'après Temperley (2004), le schéma d'évaluation pour les méthodes d'analyse automatique du rythme doit accomplir quatre conditions fondamentales:

1. définir de façon claire la manière de représenter l'information à rechercher,
2. disposer d'une base de données de taille considérable qui soit suffisamment représentative de la musique à traiter,
3. conduire une annotation (manuelle) appropriée de l'information à rechercher dans la base de données,
4. préciser la manière de comparer les résultats obtenus par les méthodes d'analyse automatique du rythme avec les annotations manuelles.

La première étape vers l'établissement d'une méthodologie commune d'évaluation et de comparaison pour les algorithmes d'induction de tempo a été prise en 2004 par le comité de direction de la conférence internationale ISMIR. L'annexe A présente des détails au sujet de cette évaluation internationale.

Bases de test

Une des premières tâches réalisées dans ce travail était le rassemblement d'une base de données d'extraits musicaux conforme aux recommandations décrites précédemment. Ce corpus a été extrait à partir d'enregistrements commerciaux. Pour chaque enregistrement un extrait caractéristique a été choisi, qui a été alors converti en signal monophonique et échantillonné à 16 kHz avec une résolution 16 bit. Ce corpus de données à été formé après fusion de deux autres bases.

- ⋆ **Base de données de l'ENST.** Cette base contient 961 morceaux musicaux avec une durée globale de 18759 secondes (5 heures 12 minutes et 39 secondes). Le plus petit

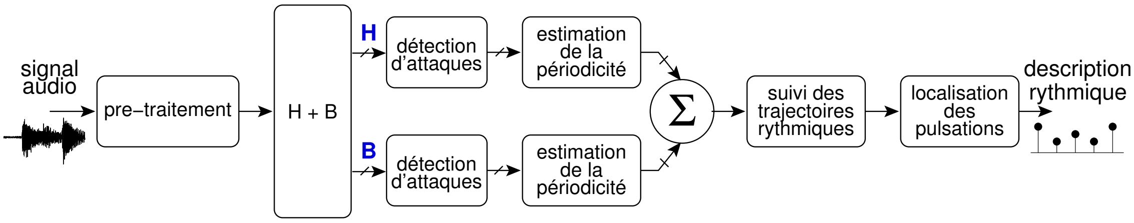


Figure 3: Système général pour l'analyse automatique du rythme.

morceau a une durée de 10 secondes, le plus long a une durée de 30 secondes et la durée moyenne est de 19.5 secondes. Cette base contient onze genres différents et une grande variété de tempi (voir figure page 51).

- ⋆ **Base de données de l'université de Tampère.** Cette base contient 474 morceaux musicaux avec une durée globale de 126595 secondes (35 heures 9 minutes et 55 secondes). Le morceau plus petit a une durée de 42 secondes, le morceau plus long a une durée de 829 secondes et la durée moyenne est de 267 secondes. Cette base contient sept genres différents et une grande variété de tempi (voir figure page 52).

Calcul de la fonction de détection

Dans le cadre de notre travail de recherche nous avons développé un système pour estimer le tactus et le tatum des signaux musicaux. Une illustration graphique de ce système est présentée dans la Figure 3.

Pré-traitement

Approche causale Afin d'obtenir une bonne estimation des paramètres fréquentiels dans chaque sous-bande, il est nécessaire d'utiliser des filtres suffisamment réjecteurs pour que la puissance du signal dans la bande atténuée ne dépasse jamais le niveau de bruit dans la bande passante. Or la densité spectrale de puissance des sons émis par de nombreux instruments de musique est une fonction décroissante de la fréquence. Ainsi, la sélection d'une bande en hautes fréquences nécessiterait d'utiliser un filtre plus réjecteur qu'en basses fréquences, donc un filtre plus long. Pour éviter cette distinction et pour pouvoir appliquer la même réjection en hautes fréquences qu'en basses fréquences, il est préférable d'égaliser approximativement la puissance du signal en entrée du banc de filtres. Une façon simple mais suffisante de procéder consiste à appliquer un filtre de pré-accentuation pour compenser la tendance décroissante de la Densité Spectrale de Puissance (DSP), par exemple de fonction de transfert

$$H(z) = 1 - 0.98z^{-1}. \tag{1}$$

Approche non-causale La méthode causale décrite ci-dessus est une solution partielle au problème puisque la tendance décroissante de la densité spectrale a été réduite, mais pas entièrement compensée (voir la Figure 3.2). Des meilleurs résultats peuvent être obtenus à l'aide d'un filtre blanchisseur.

Supposons dans un premier temps que le signal audio $x(n)$ soit un processus autoregressif ou AR (sans sinusoïdes), obtenu en filtrant un bruit blanc de variance σ^2 par un filtre de fonction de transfert $\frac{1}{H(z)}$, où tous les zéros de $H(z) = 1 + a_1 z^{-1} + ... + a_p z^{-p}$ sont à l'intérieur du cercle unité. Il est connu que les coefficients du filtre $H(z)$ et la variance σ^2 s'estiment par prédiction linéaire à partir d'un estimateur de la fonction d'autocovariance $r_x(t) = E[x(u)^* x(u+t)]$. Définissons un estimateur de $r_x(t)$

$$\hat{r}_x(n) = \frac{1}{N} \left([\tilde{g} \cdot \tilde{x}] \star [g \cdot x] \right)(n) \tag{2}$$

où $g(n)$ est une fenêtre d'analyse de taille finie N, en plus $\tilde{x}(n) = x(-n)^*$ et $\tilde{g}(n) = g(-n)^*$. La fonction $\hat{r}_x(n)$, définie comme un produit de convolution, a un support de longueur $2N - 1$. Il est possible de le calculer de manière rapide par le biais de l'algorithme FFT (Transformée de Fourier Rapide). Ainsi, $\hat{r}_x(n)$ s'obtient en calculant la transformée de Fourier inverse du périodogramme

$$\widehat{R}_x(e^{j2\pi f}) = \frac{1}{N} |X(e^{j2\pi f})|^2. \tag{3}$$

L'algorithme d'estimation du filtre blanchisseur se décompose en quatre étapes

1. multiplication du signal $x(n)$ par la fenêtre $g(n)$;
2. calcul du périodogramme défini dans l'Equation 3;
3. calcul de $\hat{r}_x(n)$, obtenue par transformée de Fourier inverse;
4. estimation du filtre $H(z)$ par prédiction linéaire à partir de $\hat{r}_x(n)$.

Supposons maintenant que le signal $x(n)$ soit perturbé par la présence de sinusoïdes qui viennent s'ajouter au processus AR. Le périodogramme $\widehat{R}_x(e^{j2\pi f})$ est alors perturbé par des pics centrés aux fréquences de ces sinusoïdes, qui se superposent à la DSP du processus AR. Il est possible de les éliminer en introduisant une étape de lissage du périodogramme à l'aide d'un filtre de rang utilisé entre les étapes 2 et 3. Dans la pratique le pré-traitement se fait par trames. Le signal est multiplié par une fenêtre de Hann de même longueur, et le périodogramme est calculé. Il est ensuite lissé en appliquant un filtre de rang de longueur q. Pour calculer la valeur du périodogramme lissé en chaque point, les q valeurs extraites sont triées par ordre croissant, puis celle d'ordre $\frac{q}{3}$ est sélectionnée. La fonction $\hat{r}_x(n)$ est en suite obtenue en calculant la transformée de Fourier inverse du périodogramme filtré. Finalement, le filtre blanchisseur $H(z)$ est calculé par prédiction linéaire à l'ordre $p = 6$.

Décomposition Harmonique plus Bruit

Une des nouveautés que nous proposons dans notre recherche se base sur l'idée de décomposer le signal audio en deux parties : une déterministe et l'autre stochastique. Puis, nous analysons chacune d'entre elles de façon séparé et finalement nous combinons les résultats. Plus spécifiquement au sujet de la décomposition, nous modelons le signal audio $x(n)$ comme une somme linéaire de deux éléments. La première partie (à laquelle nous nous sommes référés en haut comme déterministe) est constituée seulement des composantes sinusoïdales et nous l'appellerons *harmonique* et sera notée $s(n)$. La deuxième partie (à laquelle nous nous sommes référés en haut comme stochastique) est constituée de tous les éléments provenant du signal audio original qui ne peuvent pas être considérés comme sinusoïdes et nous l'appellerons *bruit* et sera notée $w(n)$. Donc, $w(n) = x(n) - s(n)$.

A notre connaissance, dans le contexte de l'analyse automatique du rythme, il n'existe pas d'approches précédentes traitant séparément la partie harmonique et la partie bruit. Pour cette raison, un des buts de notre recherche est d'explorer le potentiel de cette décomposition dans le contexte de l'analyse de la métrique des signaux musicaux. Dans notre travail nous utilisons deux méthodes de décomposition harmonique plus bruit (H+B).

Méthode basée sur le modèle de Sinusoïdes Modulées Exponentiellement Après le pré-traitement, le signal audio $x(n)$ est décomposé en P sous-bandes uniformes (sans recouvrement) en utilisant un banc de filtres en cosinus modulé avec un filtre prototype (à réponse impulsionelle finie) d'ordre 200 et 80 dB d'atténuation dans la bande de réjection. L'utilisation d'un filtre très sélectif est nécessaire puisque cette méthode est très sensible à des sinusoïdes situées dans la bande de réjection.

Le modèle de sinusoïdes modulées exponentiellement est basé sur une technique de décomposition en sous-espaces, il se base sur le principe de la décomposition du signal en deux parties.

- La partie harmonique, celle-ci est formée par la somme de M sinusoïdes qui peuvent subir un affaiblissement exponentiel. Afin de prendre en compte les bruits polyphoniques, les fréquences de ces sinusoïdes ne sont pas contraintes à être uniformément distribuées.

- La partie bruit est définie comme la différence entre le signal original et la partie harmonique.

Le principe d'analyse de cette technique est le suivant, des *instantanés* consécutifs ayant un longueur de L échantillons sont extraits du signal. Ensuite, l'espace L-dimensionnel engendré par ces vecteurs est décomposé en deux parties: le *sous-espace signal* et le *sous-espace* bruit. Le sous-espace signal sert à caractériser les M sinusoïdes et sa dimension est $2M$. Au contraire, le sous-espace bruit contient tout ce qui ne peut pas être considéré comme sinusoïdal et sa dimension est $L - 2M$. Dans la pratique, L doit être beaucoup plus grand que $2M$ pour augmenter la robustesse de l'algorithme.

La propriété la plus remarquable de cette méthode nous permet d'éviter l'estimation et la soustraction des sinusoïdes, puisque la partie bruit peut être directement obtenue en projetant le signal de chaque sous-bande sur le sous-espace bruit correspondant. Plus précisément, soit $\mathbf{U}^S(n)$ une base orthonormale de la p-ème sous-bande engendrant le sous-espace signal pour la fenêtre d'analyse $[n - L + 1, n]$. Pour plus de clarté nous allons omettre l'index de sous-bande p, sachant que ce procédé est répété pour chacune d'entre elles. Il est possible de calculer $\mathbf{U}^S(n)$ à l'aide d'une décomposition en valeurs propres de la matrice de données où de la matrice d'autocovariance ou aussi à l'aide des algorithmes de poursuite de sous-espaces (voir (Badeau, 2005)). Ensuite, un vecteur de la partie bruit de la forme

$$\mathbf{w}(n) = [w(n-L),\, w(n-L+1), \ldots, w(n)]^\top \qquad (4)$$

peut être obtenu en appliquant le projecteur de sous-espace bruit $\mathbf{I}_L - \mathbf{U}^S(n)\mathbf{U}^S(n)^H$, où $\mathbf{A}^H$ indique la matrice transposée hermitienne de $\mathbf{A}$, au vecteur des données

$$\mathbf{x}(n) = [x(n-L),\, x(n-L+1), \ldots, x(n)]^\top, \tag{5}$$

dans ce cas nous obtenons les parties harmonique et bruit par:

$$\begin{aligned} \mathbf{s}(n) &= \mathbf{U}^S(n)\mathbf{U}^S(n)^H\mathbf{x}(n) && (6)\\ \mathbf{w}(n) &= \mathbf{x}(n) - \mathbf{s}(n). && (7) \end{aligned}$$

Afin d'obtenir ces composantes pour tout le signal nous répétons ce processus combiné avec la méthode addition/recouvrement, de la façon suivante:

1. la trame d'analyse $[n-L+1,\, n]$ est récursivement décalée (un recouvrement de $3L/4$ échantillons donne des bons résultats),

2. le sous-espace signal $\mathbf{U}^S(n)$ est poursuivi à l'aide de l'algorithme *Sequential iteration EVD* présenté dans la Table 3.1 (voir aussi (Badeau et al., 2002; Badeau, 2005)),

3. le vecteur de la partie signal ($\mathbf{s}(n)$) et le vecteur de la partie bruit ($\mathbf{w}(n)$) sont calculés comme indiqué par les Equations 6 et 7,

4. à chaque itération les vecteurs des parties harmonique et bruit sont multipliés par une fenêtre de Hann et sommés aux signaux harmonique et bruit respectifs.

Pour chaque bloc, le coût de cette opération de décomposition par projection en sous-espaces est celui de la deuxième étape, qui est la plus complexe. Badeau et al. (2002) ont montré que sa complexité est $O(Ln(n + \log(L)))$.

Méthode basée sur la Transformée de Fourier à Court Terme La deuxième méthode de décomposition H+B que nous utilisons est basée sur le vocoder de phase (Portnoff, 1980). Cette technique permet d'effectuer des modifications aux amplitudes et aux phases des composantes sinusoïdales spécifiques directement dans le domaine fréquentiel. Puis, cette représentation modifiée du signal est resynthétisée dans le domaine temporel. Cette technique est également connue comme *filtrage FFT*, puisqu'à l'intérieur du vocoder de phase on trouve la Transformée de Fourier à Court Terme (TFCT).

De la même façon que la méthode précédente, cette technique effectue la décomposition du signal d'entrée par trames. Pour chaque fenêtre d'analyse (trame), les pics les plus importants dans le spectre d'amplitude sont considérés comme sinusoïdes et sélectionnés, le reste du spectre est éliminé. Alors, chaque trame de la partie harmonique est obtenue en synthétisant ce spectre modifié à l'aide de la Transformée de Fourier Inverse. Chaque trame de la partie bruit est calculée en soustrayant de la fenêtre d'analyse du signal original la trame correspondante de la partie harmonique. Ensuite, la fenêtre d'analyse est décalée et tout le processus est réitéré pour tout le signal d'entrée.

On suppose que le signal d'entrée à déjà été pré-traité. On calcule la TFCT du $x(n)$

$$\widetilde{X}(m,k) = \sum_{n=-0}^{N-1} g(n)x(Mm+n)e^{-\jmath\frac{2\pi}{N}kn} \tag{8}$$

où $m \in \mathbb{Z}$ est l'index du temps (trame), $g(n)$ est une fenêtre de longueur finie N qui détermine la partie de $x(n)$ à analyser à l'instant m, M est le décalage temporel de la fenêtre d'analyse et $k = 0, \ldots, K-1$ est l'index de fréquence (*bin*).

Afin d'estimer la partie harmonique, nous supposons que les maxima les plus importants dans le spectre d'amplitude représentent des sinusoïdes dans le signal d'entrée. Soit ν_ℓ, où $0 < \ell \leq L \ll K$, les fréquences (bins) correspondant à ces maxima. Alors, nous définissons la représentation fréquentielle de la partie harmonique comme $\widetilde{S}$, où

$$\widetilde{S}(m,k) = \begin{cases} \widetilde{X}(m,k) & \text{si} \quad k \in \nu_\ell \\ 0 & \text{sinon.} \end{cases} \tag{9}$$

C'est-à-dire, un nouveau signal est formé où seules sont gardées les fréquences correspondants aux pics les plus significatifs (*i.e.*, sinusoïdes) et le reste du spectre est mis à zéro. Á partir de ce signal modifié nous synthétisons (trame par trame) la partie harmonique

$$s(n) = \sum_{m=-\infty}^{\infty} f(n+Mm) \left(\frac{1}{K} \sum_{k=0}^{K-1} \widetilde{S}(m,k) e^{\jmath \frac{2\pi}{K} kn} \right). \tag{10}$$

La partie bruit $w(n)$ est obtenue en soustrayant la partie harmonique du signal d'entrée (trame par trame)

$$w(n) = x(n) - \sum_{m=-\infty}^{\infty} f(n+Mm) \left(\frac{1}{K} \sum_{k=0}^{K-1} \widetilde{S}(m,k) e^{\jmath \frac{2\pi}{K} kn} \right). \tag{11}$$

Afin de rendre compatible la sortie de cette méthode de décomposition H+B avec la précédente, nous filtrons les signaux $s(n)$ et $w(n)$ en utilisant les mêmes bancs de filtres présentés pour la méthode d'analyse en sous-espaces.

Estimation du profil musical d'accentuation

Dans le cadre de notre travail de recherche nous avons développé un système de détection d'attaques musicales spécifiquement adapté à l'analyse du rythme (Alonso et al., 2005c). Néanmoins, nous considérons qu'il peut être utile pour d'autres applications dans le domaine du traitement de la musique par ordinateur.

La technique d'estimation du profil musical d'accentuation que nous proposons est basée sur le principe du *Flux Energétique Spectral*. Cette méthode utilise un traitement par sous-bandes, le schéma d'analyse est présenté dans la Figure 4. L'approche générale est la suivante : d'abord la partie harmonique $s(n)$ ou bruit $w(n)$ de chaque sous-bande est décomposée en canaux fréquentiels à l'aide de la TFCT. Après, cette représentation temps–fréquence est réallouée pour améliorer sa lisibilité. Finalement, le flux spectral dans chaque sous-bande est calculé comme indiquée dans la Figure 4.

Réallocation dans le plans temps–fréquence L'utilisation de la méthode de réallocation améliore de manière significative l'estimation du contenu temps–fréquence d'un signal. Cette information est très importante pour une estimation adéquate des enveloppes des notes musicales, car c'est à partir de celles-ci que nous détectons les attaques ("*onsets*").

De façon plus générale, le principe de la réallocation a souvent été utilisé pour rehausser plusieurs types de représentations temps–fréquence (Auger & Flandrin, 1995; Hlawatsch & Auger, 2005). Dans le cadre de l'estimation du profil musical d'accentuation, la réallocation a déjà été utilisée avec succès par Hainsworth & Wolfe (2001); Röbel (2003) and Peeters (2005).

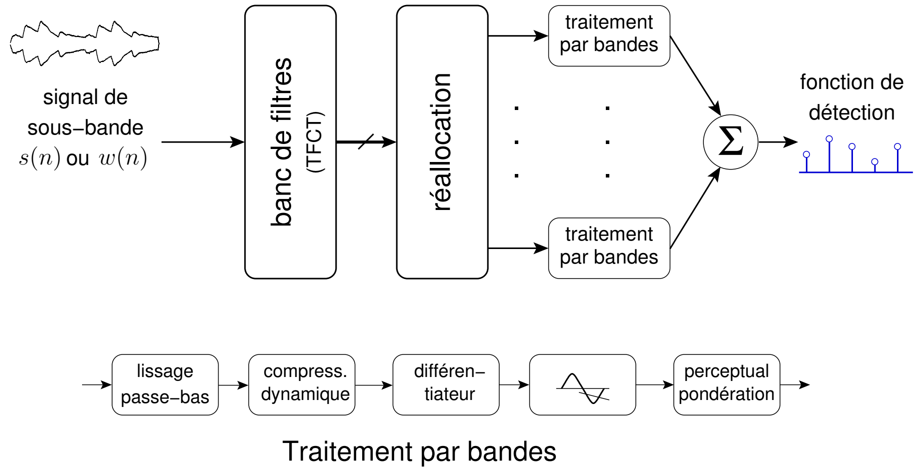


Figure 4: Schéma d'estimation du profil musical d'accentuation.

Puisque les fondements théoriques de l'opération de réallocation se basent sur la définition continue de la TFCT, nous commutons pendant un moment du domaine temporel discret au continu. Soit $s_c(t)$ (où $t \in \mathbb{R}$) le signal réel, la TFCT à temps continu est définie comme :

$$\widetilde{S}_c(\tau, f) = \int_{-\infty}^{\infty} s_c(t) g_c(t-\tau) e^{-\jmath 2\pi f t} dt. \tag{12}$$

Le handicap principal dans l'estimation de l'amplitude et de la fréquence des composantes sinusoïdales de $s_c(t)$ est directement lié à la longueur et à la largeur de bande de la fenêtre d'analyse $g_c(t)$. Il est possible d'écrire la TFCT en termes de son module et phase comme :

$$\widetilde{S}_c(\tau, f) = |\widetilde{S}_c(\tau, f)| \, e^{\jmath\varphi(\tau, f)}.$$

Les opérateurs de réallocation sont obtenus à partir des dérivées partielles de $\varphi(t, f)$ par rapport à chacune de ses variables, menant respectivement à la fréquence instantanée

$$F_i(\tau, f) = \frac{1}{2\pi} \frac{\partial \varphi(\tau, f)}{\partial \tau}, \tag{13}$$

et au retard du groupe

$$T_g(\tau, f) = -\frac{1}{2\pi} \frac{\partial \varphi(\tau, f)}{\partial f}. \tag{14}$$

Ces deux équations peuvent être interprétées de la manière suivante : si nous considérons l'énergie $|\widetilde{S}_c(\tau_0, f_0)|^2$ répandue autour de la position (τ_0, f_0) dans le plan temps-fréquence, son centre de gravité est le point situé à la fréquence $F_i(\tau_0, f_0)$ et à l'instant $\tau_0 + T_g(\tau_0, f_0)$. Par conséquent, chaque point d'énergie est dit d'être *réalloué* à une nouvelle position dans le plan temps-fréquence. Dans la pratique, cette opération a lieu dans le domaine de temporel discret.

Flux énergétique spectral La méthode que nous avons sélectionnée pour calculer l'index d'accentuation musicale est connue comme le *flux énergétique spectral*. Cette technique a été employée précédemment dans la littérature, par exemple par Laroche (2001, 2003) aussi dans le contexte de l'analyse du rythme. Les fondements de cette technique reposent sur la supposition générale que l'apparition d'un *onset* (ou de façon plus générale un événement musical) dans un signal audio provoque une variation rapide du contenu spectral du signal audio.

Pour détecter les variations spectrales, l'approche la plus naturelle consiste à calculer la dérivée de la représentation temps–fréquence par rapport au temps. Dans notre cas, il faut dériver la TFCT réallouée $\widetilde{\mathcal{S}}(m,k)$:

$$E(m,k) = V(m,k) \star h(m) = \sum_{l} h(m-l)\, V(l,k), \tag{15}$$

où $E(m,k)$ est connu comme le flux énergétique spectral et $h(m)$ est une approximation d'un différentiateur idéal :

$$H(e^{j2\pi f}) \simeq j2\pi f \tag{16}$$

et

$$V(m,k) = \mathcal{F}\{\widetilde{\mathcal{S}}|(m,k)|\} \tag{17}$$

est une transformation qui sert à calculer une enveloppe énergétique perceptuelle pour chaque canal fréquentiel k de la représentation temps–fréquence. La partie inférieure de la Figure 4 indique les étapes associées au calcul du flux spectral.

Dans le cas idéal, le profil d'accentuation musicale (aussi connu comme "fonction de détection" dans la communauté d'analyse du rythme) devrait être composé d'une série d'impulsions pondérées et localisées aux instants où se trouvent les *onsets*, c'est à dire, les attaques des événements musicaux.

La Figure 5 montre un exemple du calcul de la fonction de détection pour un signal de piano. Les attaques de piano ne sont pas particulièrement difficiles à détecter, mais notre méthode donne de bons résultats en distinguant même deux événements à peine séparés de quelques millisecondes aux alentours de 3.25 s. Les quatre représentations de la Figure 5 correspondent respectivement de haut en bas à :

(a) la forme d'onde du signal de piano, les attaques ont été réperées manuellement (lignes verticales rouges) à fin de montrer une réference de la sortie souhaitée;

(b) module de la TFCT réallouée, on peut distinguer la structure harmonique de ce son;

(c) le flux énergétique spectral $\widehat{E}(m,k)$, les points alignés en forme de traits verticaux indiquent les régions où le flux d'énergie est grand ; et

(d) fonction de détection $d(m)$, les attaques ont été à nouveau manuellement annotées et elles sont marqués par les lignes verticales pointillées.

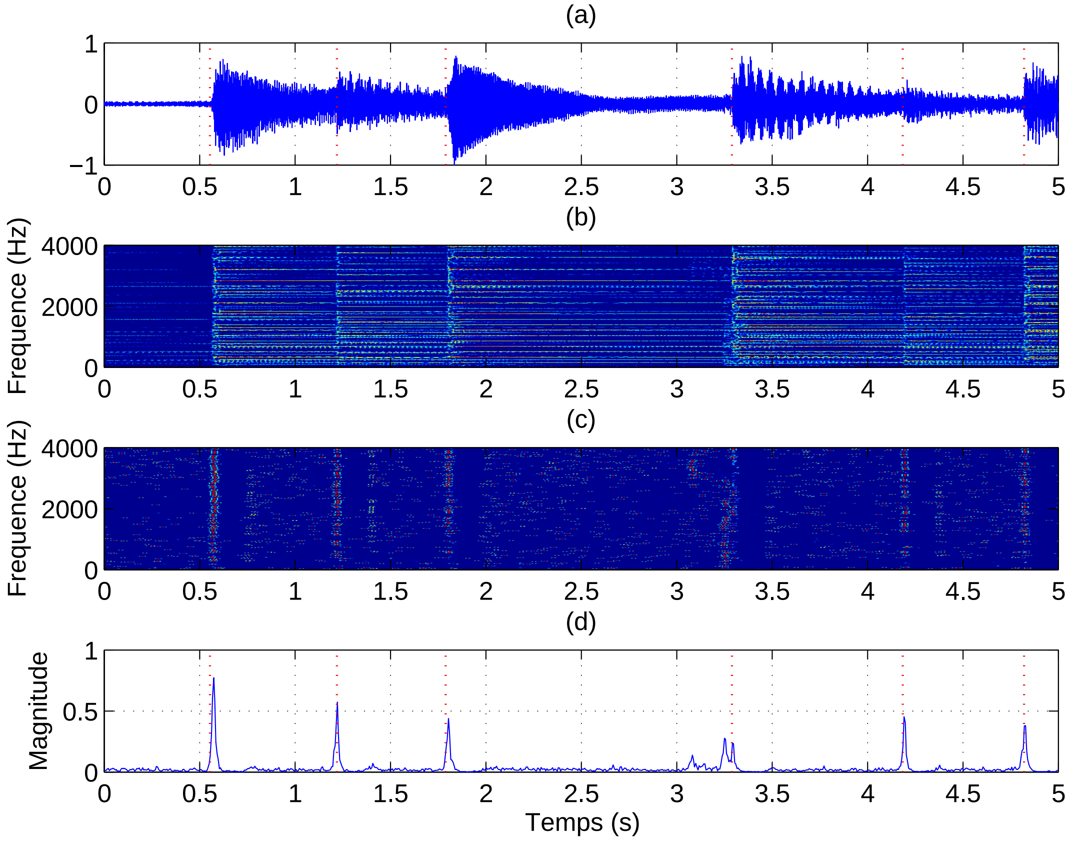


Figure 5: Flux spectral pour un signal de piano. (a) Forme d'onde du signal de piano; (b) module de la TFCT réallouée; (c) flux énergétique spectral $\widehat{E}(m,k)$, les points alignés en forme de traits verticaux indiquent les régions où le flux d'énergie est grand; (d) fonction de détection $d(m)$. Les attaques (onsets) ont été manuellement annotées et elles sont marquées par les lignes verticales pointillées.

Induction de la métrique du rythme

Une fois que le profil d'accentuation musicale a été calculé, l'étape suivante consiste à estimer la périodicité des événements musicaux, à suivre son déroulement à travers le temps et à trouver la localisation des battements qui constituent les couches métriques.

Estimation de la périodicité

D'après (Parncutt, 1994), le rythme musical peut être défini comme une séquence acoustique qui évoque chez l'auditeur une sensation de pulsation. Cette idée nous fait penser à une répétition régulière des événements sonores qui en général est perçue sans effort par les humains. Dans le cadre de l'analyse du rythme par ordinateur, il est nécessaire de reproduire cette *machinerie* perceptuelle à l'aide d'une méthode qui cherche des périodicités dans la fonction de détection. A ce propos, nous utilisons quatre algorithmes différents qui ont déjà été utilisés dans le cadre de l'estimation de la périodicité dans le contexte de la détection de hauteur : la fonction d'autocorrélation, banc de filtres en peigne, la somme spectrale et le produit spectral.

Fonction d'autocorrélation La fonction d'autocorrélation est probablement la méthode d'estimation la plus utilisée dans le cadre de l'analyse du rythme. En fait, c'est un outil mathématique utilisé très fréquemment dans le traitement des signaux pour analyser des fonctions ou des séries de valeurs. L'autocorrélation calcule le degré auquel le signal est semblable à une version décalée de lui-même

$$\hat{r}_d(k) = \frac{1}{N} \sum_{t=k}^{N-1} d(n)d(n-k), \quad 0 \leq k \leq N-1 \tag{18}$$

où $d(n)$ est la fonction de détection, N indique la taille de la fenêtre d'observation de la fonction de détection $\{d(0), \ldots, d(N-1)\}$ et k indique le décalage temporel.

Banc de filtres en peigne L'utilisation d'un banc de filtres en peigne pour l'analyse du rythme des signaux acoustiques a été proposé à l'origine par Scheirer (1998, 2000). La mise en œuvre du banc de filtres en peigne que nous utilisons dans ce travail est celle qui a été proposée par Klapuri (2004); Klapuri et al. (2006). La sortie du filtre en peigne avec un retard τ, avec $d(n)$ comme entrée, est donné par

$$y(n) = \alpha\, y(n-\tau) + (1-\alpha)d(n) \tag{19}$$

où le gain de rétroaction, qui est différent pour chaque filtre en peigne, est donné par $\alpha = 0.5^{\tau/T_0}$ avec $T_0 = 3\mathrm{F_s}$ (où $\mathrm{F_s}$ correspond à la fréquence d'échantillonnage de la fonction de détection). D'après Klapuri, la valeur de T_0 est assez courte pour réagir rapidement aux changements de tempo et assez large pour estimer avec précision la période des pulsations de l'ordre de 4 secondes.

Méthodes spectrales Nous proposons aussi deux techniques fréquentielles pour effectuer l'analyse de périodicité : la somme spectrale et le produit spectral. Ces deux méthodes ont été développées indépendamment par Noll (1970) et Wise et al. (1976) dans le contexte de l'analyse de la parole et elles sont basées sur un critère de maximum de vraisemblance.

Ces méthodes se fondent sur la supposition que la densité spectrale de la fonction de détection est constituée d'harmoniques très énergétiques situés aux multiples entiers de la fréquence fondamentale de $d(n)$. Pour trouver les périodicités, la densité spectrale de $d(n)$ ($|D(e^{\jmath 2\pi k f})|^2$) est comprimée d'un facteur k, puis la densité obtenue est sommée/multipliée à la densité spectrale originale, menant à une fréquence fondamentale très renforcée. En termes de la fréquence réduite, la somme spectrale est donnée par:

$$S(e^{\jmath 2\pi f}) = \sum_{k=1}^{K_{\text{max}}} |D(e^{\jmath 2\pi k f})|^2 \tag{20}$$

où K_{max} est la limite supérieure de compression. De façon très similaire, le produit spectral s'obtient en remplaçant la somme par un produit :

$$P(e^{\jmath 2\pi f}) = \prod_{k=1}^{K_{\text{max}}} |D(e^{\jmath 2\pi k f})|^2. \tag{21}$$

Fusion de données L'estimation de la périodicité est répétée pour chaque sous-bande des parties harmonique et bruit à l'aide des méthodes présentées ci-dessus. Une fois que cette opération a été accomplie, les informations sur la périodicité du signal audio provenant de chaque sous-bande des parties harmonique et bruit ($\mathbf{v}_{m,p}^{s,w}$) sont fusionnées dans un seul vecteur. Cette nouvelle opération se fait en deux étapes. D'abord, tous les vecteurs contenant des informations sur la périodicité du signal sont normalisés en divisant chacun par sa valeur la plus grande. Après, chacun des vecteurs est pondéré par un coefficient ($c_{m,p}^{s,w}$) qui varie dans l'intervalle $[0, 1[$ et qui mesure l'importance de l'information du vecteur de périodicité. La dernière partie de la fusion de données consiste à intégrer dans un seul vecteur tous les index de périodicité pondérés :

$$\boldsymbol{\gamma}_m = \frac{1}{2P}\sum_{p=0}^{P-1} c_{m,p}^{s}\mathbf{v}_{m,p}^{s} + \frac{1}{2P}\sum_{p=0}^{P-1} c_{m,p}^{w}\mathbf{v}_{m,p}^{w} \tag{22}$$

où $p \in [0, \ldots, P-1]$ indique le numéro de sous-bande, m la localisation temporelle et finalement les indices s et w indiquent respectivement les parties harmonique et bruit.

La méthode de fusion de données décrite ci-dessus nous a permis de calculer le profil de périodicité pour une seule fenêtre de la fonction de détection. Dans la pratique il faut répéter cette opération de façon itérative. A chaque itération, nous obtenons un vecteur colonne $\boldsymbol{\gamma}_m$ qui indique le profil de périodicité à l'instant m. Après avoir analysé tout le signal on obtient la matrice temps–périodicité

$$\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_{M-1}]. \tag{23}$$

$\boldsymbol{\Gamma}$ est une représentation bidimensionelle des pulsations présentes dans le signal audio. Les lignes indiquent le degré de périodicité à plusieurs fréquences tandis que les colonnes indiquent l'index temporel.

Suivi des pulsations

Une fois que le profil de périodicité du signal audio a été estimé l'étape suivante consiste à analyser les pulsations présentes dans les colonnes de $\mathbf{\Gamma}$ afin de trouver à chaque instant m les meilleures candidats aux couches métriques et aussi pour suivre leur déroulement temporel. La programmation dynamique (PD) est une technique qui a été intensivement employée pour résoudre ce genre de problèmes qui demandent une analyse séquentielle. En fait, le suivi des pulsations dans le contexte de l'analyse de rythme n'est pas une exception et Laroche (2003), Peeters (2005) et Collins (2005b) ont employé cette technique pour résoudre ce type de problème. Des informations plus détaillées au sujet de la mise en œuvre de l'algorithme de PD peuvent être trouvées dans Rabiner & Juang (1993).

Dans notre application, pour chaque instant m il existe K candidats potentiels appelés $\mathbf{\Gamma}_{(m,k)}$ où $k \in [0, \ldots, K-1]$. Pour trouver la trajectoire "optimale", la méthode de PD résout ce problème de combinatoire et d'optimisation en examinant toutes les combinaisons possibles de façon itérative et rationnelle. La trajectoire optimale est formée par l'enchaînement d'une suite de candidats ψ_m sélectionnés parmi les $\mathbf{\Gamma}_{(m,k)}$. La technique de PD définit itérativement un score $\mathcal{S}_{(m,k)}$ pour chaque trajectoire arrivant au candidat $\mathbf{\Gamma}_{(m,k)}$, ce score est une fonction de trois paramètres : le score de la trajectoire à l'instant antérieur $\mathcal{S}_{(m-1,\psi_{m-1})}$ où ψ_{m-1} représente le candidat par lequel la trajectoire vient de passer à l'instant $m-1$; l'intensité de la périodicité du candidat analysé $\mathbf{\Gamma}_{(m,k)}$ à l'instant m; et finalement une pénalité de transition $D_{(m-1,\psi_{m-1})}$, aussi appelée contrainte locale, qui pénalise le score pour une transition du candidat ψ_{m-1} à l'instant $m-1$ au candidat ψ_m à l'instant m selon la règle indiqué par la Figure 4.9.

Une autre nouveauté de notre algorithme d'analyse du rythme consiste à modifier le bloc de suivi de la pulsation à fin de traquer non seulement la trajectoire optimale, mais aussi d'autres trajectoires. Dans la pratique, à l'intérieur de l'algorithme de suivi, la *meilleure* trajectoire est aussi la plus énergétique. Afin de trouver une deuxième (et davantage de) trajectoires, nous exécutons l'algorithme en imposant la restriction suivante : aucune nouvelle trajectoire ne doit partager des segments de chemin ou être trop proche ($<$ 10 BPM) des autres trajectoires déjà trouvées par l'algorithme. Alors, il est possible de réitérer sur cette restriction pour trouver un certain nombre de trajectoires (qui dépend du morceau analysé) plus faibles en termes énergétiques. En général, toutes les trajectoires sont liées entre elles par un facteur rationnel. Deux exemples du fonctionnement du système complet de suivi de la périodicité sont présentés dans les Figures 4.10 et 4.11 (voir pages 108 et 109).

Sélection d'une trajectoire de périodicité comme tempo

A la sortie du module de suivi des périodicités nous avons un ensemble de trajectoires où chacune d'entre elles possède une énergie directement liée a sa prépondérance dans l'enregistrement musical. L'étape suivante consiste à estimer parmi ces trajectoires celle qui représente le mieux le tempo du signal audio. A ce propos, nous utilisons une distribution *a priori* qui modèle les préférences humaines par rapport au rythme. Cette distribution est utilisée sous la forme d'une courbe de pondération, c'est à dire que nous multiplions la prépondérance de chaque trajectoire de périodicité par une valeur qui dépend directement de la période de battement. Puis, la trajectoire avec la plus grande valeur est considérée comme le “vrai” tempo. La courbe de pondération que nous utilisons est celle proposée par (Moelants, 2002). Il faut noter que cette méthode de calcul du tempo a été précédemment utilisée dans la littérature, plusieurs autres types de courbes de pondération ont été aussi proposés (voir §4.2.3).

Performance du système proposé

Dans les sections précédentes nous avons introduit les étapes qui forment notre système d'analyse du rythme représenté dans la Figure 3. Le but de cette partie est de présenter une évaluation quantitative de la performance de ce système.

L'évaluation quantitative des systèmes d'analyse de la métrique des signaux musicaux est actuellement un problème partiellement résolu. Des méthodologies adéquates à ce sujet ont été indépendamment proposées par Goto & Muraoka (1997a) et Temperley (2004), toutefois elles se fondent sur un processus laborieux ou extrêmement long pour obtenir la base de données des annotations manuelles. En raison de telles limitations d'ordre pratique, la plupart des évaluations quantitatives décrites dans la littérature se limitent seulement à l'estimation du tempo (en BPM) d'un morceaux de musique. Dans notre évaluation, nous adoptons aussi cette approche. Idéalement, la procédure d'évaluation devrait être plus approfondie et inclure dans le processus plusieurs couches métriques ainsi que l'estimation de leur phase (localisation temporelle) à l'intérieur du signal musical.

Pendant l' évaluation du système, nous considérons l'analyse d'un morceau de musique comme “correcte” si le tempo trouvé par le système ne varie plus de 5% par rapport à la valeur du tempo obtenu pendant l'annotation manuelle. Nous considérons aussi comme correctes toutes les estimations dans un rapport de moitié, double, tiers ou triple de la valeur du tempo annoté.

Il est intéressant de savoir si la combinaison des résultats des quatre algorithmes de périodicité que nous utilisons (SS, SP, AC, CF) peut obtenir un score plus élevé que celui obtenu par chaque méthode de façon individuelle. Pour cette raison nous avons créé une cinquième technique appelée *Fusion de méthodes* (MF) qui intègre en une seule valeur les résultats des autres méthodes à l'aide d'un algorithme de décision.

La Figure 6 présente les résultats (par méthode de périodicité) sur l'efficacité du système à estimer le tempo des morceaux musicaux dans les bases de test de l'ENST et de l'université de Tampère. Ces résultats ont été obtenus en utilisant le pre-traitement non-causal et un banc de filtres uniforme à 8 bandes (voir §3.3.1.3). Les Figures 6-(a)-(b) indiquent respectivement la performance pour chacune des méthodes de décomposition H+B (celle basée sur le modèle EDS et celle basée su la TFCT). La méthode EDS donne des résultats légèrement supérieurs, mais toutes les deux présentent des performances comparables. De la même façon, parmi les méthodes d'estimation de la périodicité la somme spectrale obtient la meilleur performance. Toutefois, les autres méthodes montrent aussi des résultats proches à l'exception de la technique de banc de filtres en peigne. La Figure 6 compare aussi notre système avec la méthode d'analyse du rythme proposé

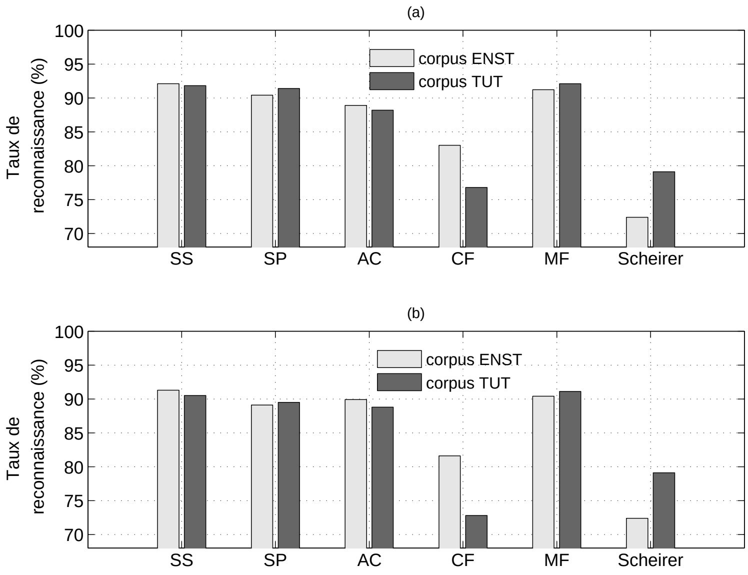


Figure 6: Performance par méthode de périodicité utilisant la technique de décomposition H+B (a) basée sur le modèle EDS et (b) basée sur la TFCT. La signification des sigles est la suivante, SS : somme spectrale, SP : produit spectral, AC : fonction d'autocorrelation, CF : banc de filtres en peigne, et MF : *fusion de méthodes*.

par Scheirer (1998), on peut voir que notre approche obtient une meilleure performance.

Après l'analyse des résultats présentés dans la Figure 6, on peut se poser comme question : *quelle est l'influence de la décomposition H+B sur la performance du système?* Pour trouver la réponse, nous avons mesuré la performance de notre algorithme pour trois configurations de la méthode H+B : décomposition basée sur le modèle EDS, décomposition basée sur la TFCT et dans le dernier cas, pas de décomposition. Ces tests on été effectués en utilisant la méthode de la somme spectrale pour estimer la périodicité. La Figure 7-(a) montre les résultats obtenus sur la totalité des morceaux dans les bases de test, les barres d'erreur indiquent l'intervalle de confiance à 95% (l'explication de la méthode de calcule de cet intervalle est présentée dans la page 131). L'examen de ces résultats nous indique que la faible amélioration obtenue après avoir effectué la décomposition H+B n'est pas statistiquement significative.

Après un examen plus détaillé des résultats, nous avons découvert que la décomposition H+B n'a pas la même influence sur tous les genres musicaux présents dans les bases de test. Plus précisément, nous avons trouvé que l'impact de cette approche est pratiquement négligeable pour la musique percussive, mais que la contribution de la décomposition pour d'autres genres plus difficiles est bien plus importante. La Figure 7-(b) montre l'influence de la décomposition H+B dans le cas spécifique de la musique classique. Pour ce genre musical cette influence est plus notable et l'amélioration obtenue est très nette, mais elle ne garantit pas que la décomposition H+B soit statistiquement

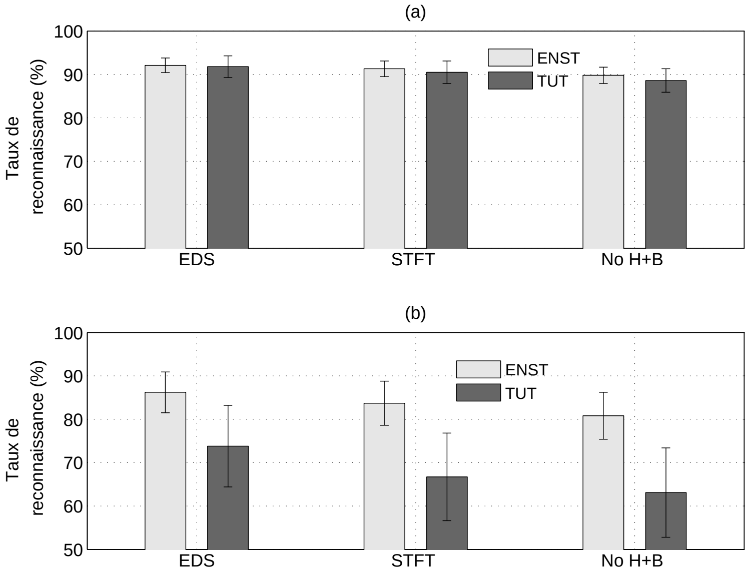


Figure 7: Influence de la décomposition H+B. Les barres montrent uniquement le résultat pour la méthode de périodicité SS. Résultats pour (a) la totalité des morceaux dans base et (b) pour les morceaux de musique classique.

significative[2]. Toutefois nous considérons ces résultats très satisfaisants, car ce genre est particulièrement difficile à traiter.

Dans les paragraphes précédents nous avons présenté notre évaluation du système d'analyse du rythme où il a montré une bonne performance. En outre, nous ne nous sommes pas limités aux mesures internes et nous avons soumis notre algorithme à une évaluation externe dans le cadre de la compétition MIREX (voir Annexe A), où nous avons obtenu en 2005 la première place dans la catégorie d'estimation du tempo parmi plus d'une dizaine d'algorithmes.

Conclusions et perspectives

Cette thèse a été consacrée au développement des mécanismes pour essayer de donner aux ordinateurs la capacité de comprendre certains éléments fondamentaux du rythme musical. Plus précisément, il s'agit de construire un système qui prend comme entrées des enregistrements musicaux et dont la sortie doit être comparable à la réponse générée par un auditeur humain (habitué à la musique occidentale) lorsqu'il lui est demandé de *battre* avec le signal musical.

La première partie du manuscrit est consacrée à la présentation des concepts fondamentaux, à savoir : une définition appropriée du rythme musical et une notion de la structure métrique couplée à l'idée de présenter le rythme sous forme de couches métriques superposées.

Puis, une introduction au champ de l'analyse du rythme par ordinateur a été présentée.

[2] Il faut aussi prendre en compte que le nombre d'échantillons dans la base de musique classique est très inférieur aux nombre de morceaux dans toute la base, ce qui provoque une augmentation considérable de la taille des barres d'erreur.

Une contribution importante de cette thèse a été la présentation d'un panorama complet sur l'état actuel de l'analyse automatique du rythme. Nous avons classifié les approches existantes selon la nature du signal d'entrée dans deux catégories générales : les *modèles symboliques* et les *modèles acoustiques*. Nous avons présenté l'état de l'art sur ces deux approches, bien que nous avons mis plus l'accent sur les modèles acoustiques car ils concernent directement l'objectif de notre travail.

Nous nous sommes exhaustivement intéressés à la question de mesurer le degré d'accentuation musicale en fonction du temps. A notre avis, c'est de loin le problème le plus complexe à résoudre afin de développer un système d'analyse du rythme performant. Nous avons proposé une nouvelle méthode pour faire face aux cas plus difficiles, *i.e.*, ceux formés par des sons contenant des attaques faibles.

Cette méthode est basée sur l'idée de séparer dans un signal audio, la partie harmonique et la partie bruit[3] (H+B).

Le but de cette séparation est de souligner les accents musicaux en les séparant des éléments qui les entourent et qui peuvent rendre leur détection plus compliquée. Deux méthodes différentes ont été employées pour effectuer cette décomposition : d'une part une technique d'analyse en sous-espaces qui exploite le modèle des sinusoïdes amorties exponentiellement (appelée EDS en anglais); d'autre part une approche plus traditionnelle qui utilise la transformée de Fourier.

Une autre contribution importante de notre travail de recherche a été l'amélioration de la technique connue sous le nom de Flux Energétique Spectral (SEF en anglais), qui mesure le degré de changement de la densité spectrale de puissance en fonction du temps. Nous avons découvert qu'une bonne estimation de l'enveloppe temporelle à l'intérieur des canaux fréquentiels est fondamentale pour les techniques de détection d'attaques basées sur des critères énergétiques. Pour obtenir cette estimation, nous avons proposé un filtre lissant (*i.e.*, passe-bas) décrit par Meddis (1988) et qui imite la réponse du nerf auditif aux stimulus sonores soudains. Nous avons discuté aussi sur l'importance de l'utilisation d'un bon filtre différentiateur et nous avons adopté celui proposé par Dvornikov (2003). La sortie idéale du module d'estimation du profil d'accentuation musicale consiste en un signal formé par des impulsions localisées aux emplacements des attaques.

Ensuite, nous avons abordé le problème de l'estimation de certains paramètres rythmiques des enregistrements musicaux à partir de leur profil d'accentuation musicale. Pour trouver les périodicités les plus saillantes nous avons utilisé quatre méthodes : la fonction d'autocorrélation, un banc de résonateurs formé de filtres en peigne, la somme spectrale et le produit spectral. Pour traiter la sortie du module d'induction de la périodicité nous utilisons une technique de suivi de la pulsation basée sur l'algorithme de programmation dynamique. Bien que des approches semblables aient déjà été proposées, notre méthode est innovatrice car nous sommes parmi les premiers à proposer une version modifiée capable de détecter et poursuivre simultanément plusieurs périodes.

[3] Par *bruit* nous nous référons à tous les éléments du signal audio original qui ne peuvent pas être modélisés comme des composantes sinusoïdales.

Après avoir évalué notre système, nous considérons sa performance globale comme satisfaisante. Concernant la décomposition harmonique plus bruit (H+B), dans le cas général elle n'a pas fourni l'amélioration attendue. Plus précisément, pour les genres musicaux riches en sons percussifs cette technique ne semble pas apporter une grande contribution. Cependant, nous considérons que cette décomposition H+B a montré un gain important pour les cas plus difficiles (avec peu ou sans sons percussifs), par exemple une amélioration moyenne de 6.4% pour la musique classique en utilisant le modèle EDS. Le schéma que nous avons proposé peut être vu comme un système modulaire qui peut être adapté en fonction des besoins, par exemple, en fonction de la musique à traiter ou en termes de conditions/limitations informatiques.

Il existe plusieurs manières potentielles d'étendre le système que nous avons proposé dans ce travail. A ce sujet, nous avons quelques idées.

- Nécessité de poursuivre la recherche pour mesurer le potentiel de la décomposition H+B. En fait, les deux méthodes que nous utilisons actuellement n'arrivent pas à séparer complètement les sinusoïdes de la partie bruit. Il est possible que cette séparation imparfaite réduise les avantages de cette méthode.

- A notre avis, le noeud du problème se trouve dans l'étape d'estimation de l'accentuation musicale. En fait, cette étape est sans doute l'élément le plus important de tout le système. L'amélioration de ce module devrait prendre en compte la non-stationnarité de sons en utilisant les informations de plusieurs canaux fréquentiels.

- Bien que nous soyons parmi les premiers à proposer un mécanisme de fusion des profils de périodicité dans un seul vecteur, l'approche que nous utilisons est un peu grossière. L'utilisation des méthodes d'apprentissage pour affiner la fusion de données est possible, ces techniques peuvent être employées pour établir si une sous-bande donnée apporte des informations utiles ou si elle doit être rejetée.

- L'intégration des connaissances musicales de haut niveau doit améliorer les capacités du système, car la méthode actuelle a très peu exploité l'information fournie par l'algorithme de poursuite des périodicités. L'inclusion de nouvelles étapes tenant compte des informations de haut niveau et de la périodicité permettrait, par exemple, de trouver la subdivision appropriée de beat/mesure, et d'estimer conjointement plusieurs couches métriques.

Chapter 1

Introduction

Music is a ubiquitous phenomenon that we experience in our daily lives. In fact, most humans have an intrinsic facility to enjoy music regardless of their musical background. Human comprehension of music is an exciting and only partially unveiled field of study. Like most areas related to perception, rationalizing and imitating the process by which humans understand music is a highly complex endeavor. The Holy Grail of computer music understanding is the ability to conduct an accurate transcription. As a matter of fact, this operation embraces in a single project all of the most important tasks in music understanding, namely: pitch and key estimation, onset detection, timing, number of sources, dynamics, articulation, recognition of phrases and so forth. According to Klapuri & Davy (2006), music transcription can be seen as discovering the *recipe*, or reverse-engineering the *source code* of a musical signal. Music transcription is also the convergence point of various disciplines required in this process such as computer science, acoustics, musicology, psychoacoustics, signal processing and music perception. In addition, there exists a large number of applications, among which:

- Music Information Retrieval (MIR). This field is concerned with the problem of locating pieces of music by their content, finding the best matches in a collection of music to a particular query. That is, a *search engine* for music signals.

- Structured coding of music signals. This subject is related to MIR, it refers to the development of audio codecs that specifically support content based retrieval while also providing a compact data representation (*i.e.*, audio compression).

- Music processing, for example, alignment of multiple instruments or musical pieces; cut-and-paste operations in audio editing; digital audio effects, such as beat-driven visualization, rhythm synchronized cross-fading and many other events that can be driven by music.

- Human-computer interaction such as automatic musical accompaniment, score following, meta data generation, music for computer games.

The purpose of this thesis is to discuss a subtask of the all-encompassing music transcription problem. More specifically, in this work we address the subject of computer-based rhythm analysis. In a similar way to music transcription, rhythm analysis is influenced by the same disciplines and also shares a large part of the applications mentioned above, as illustrated in Figure 1.1.

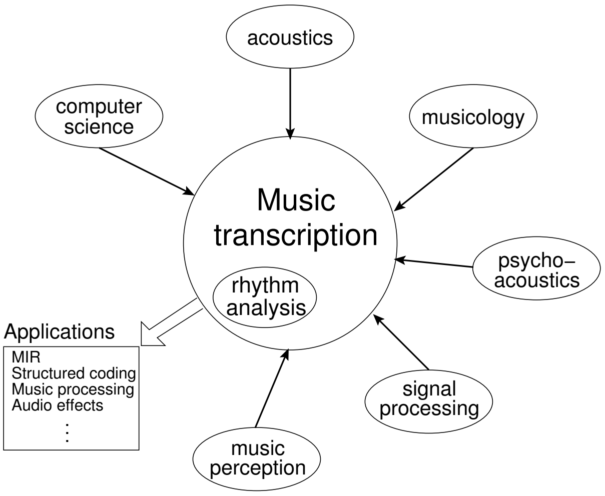


Figure 1.1: Music transcription seen as a convergence point of disciplines. Rhythm analysis can be seen as a subtask of this problem.

For Ellis & Rosenthal (1995), auditory perception can be viewed as a sequence of representations from ″low″ to ″high″ where low-level representations correspond to describing an acoustic stimulus reaching the cochlea, and high-level representations are those to which we have cognitive access, such as recognizing specific songs or musical instruments. For them, between these two levels there is a network of possible descriptions which they designate with the term *mid-level representations*. They also point that the general requirements of a mid-level hearing representation are that it may be computed efficiently from the input, and that it can readily answer the questions asked of it by the higher levels of processing.

Lerdahl & Jackendoff (1983) consider that musical rhythm can be seen as formed of two components: grouping and meter. In this thesis we only cover the latter and the main emphasis is laid on the analysis and processing of mid-level representations of polyphonic[1] music signals and not on a higher-level development which would be more focused on forming musical units/phrases. To be more precise, we propose a novel method to conduct musical meter recognition at two metrical levels: the tactus and tatum. Where the earlier, also known as *beat*, refers to the most salient level of musical meter and the latter to the lowest metrical level. In this work we adopt a bottom-up approach and we process acoustic audio signals without incorporating any high-level musical knowledge.

1.1 Musical rhythm

For Carterette & Kendall (1999) music is composed of three basic parts: melody, harmony and rhythm and all musical pieces are perceived based on these elements. Rhythm and harmony are seen as being complementary to each other in the sense that the same piece of music can be analyzed solely from a rhythmic or a harmonic aspect, if required.

Musical rhythm is a very contrasting term: from one side it is straightforward to feel it, and from the other side it is very difficult to define. Many authors have stressed that

[1] By polyphonic we refer to various sound sources playing simultaneously.

a consensual definition of the meaning of rhythm does not exist. In fact, we can find many of them, depending on the subject of interest. After reviewing the literature, for the present work we adopt as musical rhythm definition the compilation gathered by Parncutt & Drake (2001). In the following lines we outline its main parts.

According to Parncutt & Drake, the perception of rhythm involves the perceptual and cognitive organization of events in time, by which each sound event is situated in relation to those that have already occurred (memory) and those yet to come (expectancy). Next, we describe the different cognitive processes that occur over short and long time-spans.

- ⋆ **Surface organization.** The acoustical signal is first perceptually segmented into separate events corresponding to the attack points of musical elements such as tones and chords. In this context, we call ″perceptual onset″ the moment at which an event is perceived to occur. The time interval between the onset of one event and the onset of its successor is called the ″inter-onset interval″ (IOI). In addition, the physical duration of an event (i.e. the time interval between its onset and offset) may be shorter than its IOI (*e.g.*, in staccato) or longer (*e.g.*, in legato). Literature shows us that rhythmic organization is generally more influenced by IOI than by physical duration[2].

- ⋆ **Grouping and meter.** The events of a rhythm are hierarchically organized in two distinct ways, known as grouping and meter (Lerdahl & Jackendoff, 1983). From a perceptual standpoint, rhythm is characterized by (and may even be defined as) a combination of these two forms of organization. These two categories also complement each other, they can be observed separately but a complete rhythm analysis requires both. It is possible to define the grouping as follows: at the musical surface, groups correspond to short motifs. Motifs combine to form phrases, which in turn group into longer phrases, extended passages, movements and eventually whole pieces.

  On the other hand, meter is a form of perceptual organization based on temporal regularity (underlying beat or pulse). A sensation of pulse may be evoked by temporal regularity at any level within a sound sequence, or whenever relatively salient events (or motivic patterns) are perceived as roughly equally spaced in time. The musical behavior that perhaps most clearly reflects the perception of pulse is the famous term of *foot-tapping* to music. In fact, cognitively, the process of regularity extraction may be regarded as one of synchronizing our internal time-keeper or clock to music. For example, if a sequence abruptly stops, the listener expects the pulse to continue; thus attention is enhanced at the temporal locations of expected events.

  At this point we can emphasize that during the present work our research is carried out *only* on the ″meter″ estimation problem and that we do not address the ″grouping″ aspect of rhythm. More exactly, automatic rhythm analysis aims at providing

[2] In fact, to our knowledge none of the existing computer-based approaches to conduct rhythm analysis aims at estimating the offset of musical events. The reason is mainly practical, since for a large part of these events the offset takes place gradually and time instant where it occurs cannot be defined with adequate precision. However, obtaining offset information would considerably improve the rhythmic analysis. For example a series of events formed by concatenating a quarter-note followed by a quarter-rest would be correctly detected. On the contrary, by using only onset information it would be detected as a sequence of half-notes.

insights into the temporal structure of a music track by analyzing its acoustical content in terms of repetitions. However, as mentioned above these repetitions do not only exist at the ″note level″, but also take place at higher levels by clustering notes to form groups (*i.e.*, such as melodic phrases), and also clustering groups to form structures at a larger time scale. It is then possible to represent a music track not only as a series of notes, but as a set of sequences that take place at different scales. For example, in popular Western music it is very common to consider audio tracks as formed of different parts: first short melodic motifs which are then clustered to form other structural parts at larger scales called introduction, verse, chorus, bridge and others. In this work, we do not address this segmentation problem, we only work on the metrical part of rhythm analysis.

⋆ **Salience.** The perceptual salience of a pulse sensation depends on its tempo. In the vast majority of cases, musical pulses are confined to a tempo range of roughly 30 to 300 beats per minute (BPM), or an inter-beat interval ranging from 200 milliseconds to 2 seconds. For Fraisse (1982), the most salient pulses usually have tempi in the vicinity of the ″preferred tempo″, which was considered generally located around 600 ms (100 BPM). However, a number of recent studies (Moelants, 2002; McKinney & Moelants, 2004) suggest that a period slightly below 500 ms (120 BPM) is probably more realistic.

A metrical structure consists of hierarchical levels of pulsation or rhythmic layers. The multiple pulses that make up a conventional musical meter are mutually *consonant* in the sense that every event at every level (except the fastest) corresponds to an event at the next-faster level[3]. Due to their relevance in our work, we will come back with a more detailed explanation of the hierarchical levels later.

For Lerdahl & Jackendoff (1983), whenever temporal regularity is perceived at different levels, listeners tend to focus on (or attend to) a single level of moderate tempo (period near 500 ms) and perceive other levels (and hence all events) relative to this *especial* one. In the case of the consonant levels that make up a meter, this particularly important layer in the metrical structure is called the *tactus*.

⋆ **Accents.** In common usage the meaning of ″accent″ is used indistinctly with loudness, implying that attention is attracted to an audio event simply by playing it more loudly (or sometimes more softly) than its adjacent events. In this case accent is seen as a synonymous of event salience. But according to Jones (1987), anything that makes an event sound more important than adjacent events, or which attracts the attention of a listener to an event, may be considered as an accent. For example, rapid and slight alterations or changes in the dynamic level (*i.e.*, tremolo) or in the pitch of a sound (*i.e.*, vibrato) for expressive purposes add interest and substance to a sustained sound.

For Lerdahl & Jackendoff (1983), the grouping and metrical structures perceived in a piece of music depend ultimately on the timing and on the *phenomenal accent* of the events at the surface. And according to Steedman (1977), the most important contributor to phenomenal accent is typically the IOI between the event and its successor, that is, the longer the IOI following an event, the stronger the accent. Apart from IOI, phenomenal accents are generated by relative loudness (dynamic

[3] Simultaneous pulses can also be dissonant, although we do not address this variant in our work.

accents); by articulation (*e.g.*, by switching from legato to staccato); by timbral variation (manipulating the temporal or spectral envelope of events, for example changing of instrument); or by adjusting intonation.

⋆ **Rhythmic organization and tempo.** According to Handel (1993), the perceived organization of a piece of music depends on the tempo at which it is performed. Tempo may affect both grouping and meter. The metrical level at which the tactus is located depends on the tempo because the distributions of tapping rates to music are practically unconstrained by the tempo annotated in the musical score. For example, a listener might tap eighth-notes when a piece is played slowly and quarter-notes when the same piece is played twice as fast, thus keeping the tapping rate in the same absolute range. Clarke (1982) shows that in the case of grouping, the number of elements in a group increases as tempo increases, keeping their absolute length about constant.

As shown above, explaining and understanding musical rhythm is a complex matter. In fact, it is also very difficult to develop a full implementation on a computer, as will be discussed in detail in the following chapters.

1.2 Metrical structure

We mentioned in the previous section that rhythm is hierarchically organized in two distinct ways known as grouping and meter (Lerdahl & Jackendoff, 1983), and that in this work we focus on the latter. Along with the hierarchical (or metrical) structure, the concept of music meter has another important contribution that consists in organizing the music into pulses creating with this a musical time-base. If the tempo is constant, the musical time-base (inside any given metrical level) is said to be isochronous, *i.e.*, the time-interval between any two consecutive pulses is constant.

According to Lerdahl & Jackendoff (1983), in traditional Western music, the metrical hierarchy is built from two basic properties:

1. Every pulse on a given metrical level *coincides* with a pulse on all the lower[4] metrical levels.

2. Metrical levels obey a binary/ternary division, *i.e.*, the periods of pulses between any two consecutive metrical levels are related either by a factor of two or three[5].

For the sake of clarity, let us illustrate the afore mentioned properties and the hierarchical structure of metrical levels by an example. Figure 1.2 shows an arrangement of several metrical levels forming a *metrical grid*. Pulses at different levels and their respective locations are denoted by black dots. As seen in the figure, metrical levels are piled-up on top of each other with the low-ones situated in the bottom part and the high-ones on the top. Figure 1.2 also shows the cognitive structure corresponding to a $\binom{3}{4}$ meter, it includes pulses of quarter-notes (the tactus) and dotted half-notes (the measure), and usually also includes faster pulses (*e.g.*, eighth-notes) and slower pulses like groups of two measures and so on. This example also shows the coexistence of both binary/ternary

[4] In the context of metrical levels, the term "lower" means *faster* and the term "higher" means *slower*.

[5] In practice this assumption is not valid for all kinds of music. For example, it is possible to find jazz music with a $\binom{5}{4}$ meter which does not obey the second property.

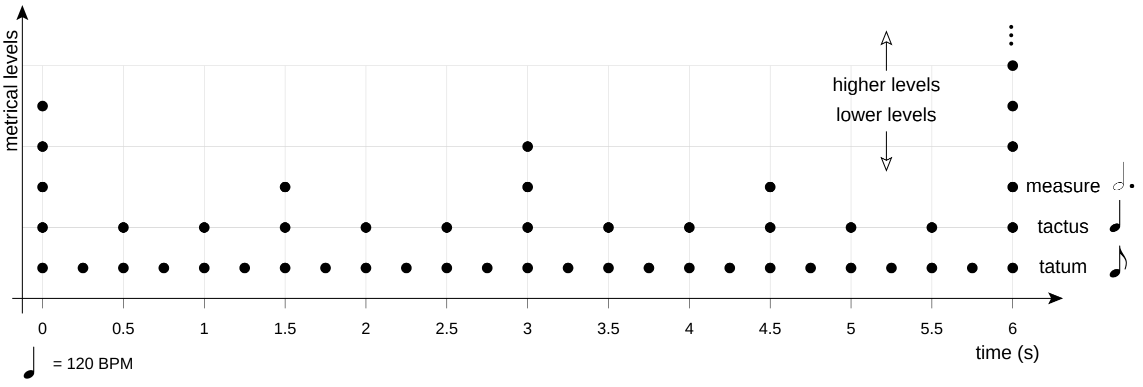


Figure 1.2: Example of a metrical grid.

levels, since there is a ternary relation between the tactus and the measure levels, while all other consecutive levels are linked by a binary relation.

Tatum. From our point of view, after the tactus, the next metrical level in importance is the so-called *tatum*. This term was coined by Bilmes (1993a,b) and derives from ″temporal atom″, although it was actually named after the famous jazz pianist ″Art Tatum″. The tatum is the lowest metrical level (see the bottom part in Figure 1.2) and in practice it refers to a rhythmic *quantum*, the pulsation that most highly coincides with all note onsets (Gouyon et al., 2002). A significant aspect is that pulsation at all rhythmic levels bear an integer-multiplicity with respect to the tatum[6], making of it a perfect short-time musical unit for segmentation and analysis purposes. This segmentation property has been outlined several times in the literature (Bilmes, 1993a,b; Seppänen, 2001a,b; Gouyon et al., 2002; Jehan, 2005a).

1.3 Goals and dissertation outline

The main goals of this thesis work are the following.

- ⋆ To describe the scope and objectives of computer-based rhythm analysis.
- ⋆ To explore the panorama of current and previous computational approaches to rhythm description.
- ⋆ To propose an approach to perform meter analysis on audio recordings at two metrical levels: the tatum and the tactus.
    - ◇ Introduce a novel front-end method to estimate a profile of the musical stress present in an audio stream as a function of time. This proposal is based on the decomposition of the audio signal in two parts: harmonic and noise. Then, an efficient way of computing the derivative of the energy envelope for the harmonic and noise parts is also presented.

[6] In practice this is not a rule, although it occurs in most cases.

  - ◇ Propose a number of methods to find the underlying periodicities present in the musical stress profile and a technique to keep track of their evolution through time.

- ⋆ To carry out a quantitative evaluation of the efficiency (mostly in terms of accuracy, but also of computational complexity) of our rhythm-description proposal.

- ⋆ To perform a qualitative assessment of the achievements and limitations of our proposal.

This report is organized as follows. In Chapter 2 we present the fundamentals and aims of computer-based rhythm analysis, which are essential to the understanding of the subsequent parts of this thesis. Next, we provide a survey of existing systems and we also describe their main characteristics, we also present a table to ease the comparison of these methods. In the end part of this chapter we address the issues of evaluating computer-based rhythm estimation systems and we describe the test database used in our work.

In Chapter 3 we introduce a framework to carry out metric analysis of music signals. This model is divided into two parts. The first one is presented in this chapter and describes the components which handle the conversion of an acoustic signal (*i.e.*, commercial audio recordings) to a symbol-like representation indicating the likelihood of finding a musical accent (*i.e.*, note onsets or chord-changes) as a function of time.

The remaining components of the metrical analysis framework are explained in Chapter 4. In this part we present a number of methods to perform periodicity induction on the symbolic representation of musical accents. We also present a method to keep track of accents progress through time as well as a technique to obtain their respective time-location. A method to estimate the tatum is also presented.

In Chapter 5 we use an heuristic approach to tune a number of key parameters in our rhythm analysis framework. Then we evaluate the performance of our system at inducing the tempo values of the test database instances. A number of different system variants is tested and their efficacy is evaluated. The ability to locate beat positions and to estimate the tatum are also assessed.

Then, Chapter 6 summarizes the main achievements and contributions of our research work. In this part we also address system shortcomings and we also propose potential paths for future research.

The last part of the document presents complementary information. Appendix A describes the recently adopted methodology for a systematic assessment and comparison of tempo extraction algorithms. In fact, during the second edition of this annual evaluation one of our algorithms obtained the first place. Appendices B and C briefly introduce two important components of our system, namely the differentiator and weighting filters. Appendix D provides numerical values for system efficiency evaluations and Appendix E presents the essence of two onset detection algorithms. Finally, Appendix F lists the articles published during this research work.

Chapter 2

A survey on computational rhythm description

In this chapter, we introduce the key concepts required to adequately understand how computer-based rhythm analysis systems work. More precisely, let us mention a few of the questions that we will answer herein: what do we mean by automatic rhythm analysis? What are its main goals? How does it work? What has been undertaken in this area before? How are such systems evaluated? In this chapter, besides providing answers to these questions, we also underline the principles which we believe are important when implementing a rhythm analysis system.

2.1 Computer rhythm analysis

When people listen to certain kinds of music, they feel immediately that it has a *beat*, that they can foot-tap or clap their hands to it. The most important is that, in some (conscious or unconscious) way, they perceive the regularly spaced motifs and they can synchronize with this sequence. Since many years, the ambitious goal of many researchers has been to repeat this process using computers and to *teach* them how music is organized into beats. There are various desirable reasons to support this idea: a computer can provide numerous alternatives as an improvising partner, facilitate cut and paste operations in audio editing, beat-drive special effects, transcribe live performance music and it opens the possibility to explore many other fascinating opportunities.

In a broad way, we can define computational rhythm analysis as an attempt to artificially replicate the process by which humans' comprehension of rhythm apparently takes place, that is, by *reproducing*[1] the part of the auditory information-processing functions that have an effect in rhythm understanding. For most humans foot-tapping along with a musical performance is generally very straightforward, that it might make us think that writing a computer algorithm to repeat this operation will be as easy.

While computers can do a rather good job when estimating the rhythm of a large part of modern pop music, it is indeed very common to see them fail when the turn comes to processing classical music. This problem has several possible causes: the number of

[1] The physiology and neurology of the human auditory system related to the perception of rhythm is only partly understood. These aspects of rhythm are not covered in the present work, but can be found in Moore (1995, 1997).

instruments playing simultaneously, the lack of clear onsets, erroneous cognitive (or cultural) presumptions. As Rosenthal (1992) mentions, we forget that the mental machinery that detects rhythm has been evolving for a long time, and has apparently become quite sophisticated that emulating it on a machine is a highly demanding task.

From our point of view, according to the definition of musical rhythm of §1.1, a complete computer algorithm designed to analyze rhythm should undertake the following tasks: parse the music into separate events[2], estimate the salience associated with each event, syntactically separate the music into motifs, identify the underlying pulsation levels, detect repeating patterns and adapt to sudden rate and timing changes representing musical expression. At the present time such a system does not exist, however we conceive its feasibility not so far in the future for music with a straightforward rhythm and still years ahead for more challenging cases, for instance romantic piano performances from Chopin or Mendelssohn whose timing varies very rapidly in short time segments.

As briefly mentioned, a complete computer-based implementation of musical rhythm encompasses many tasks. Current state-of-the-art systems are more modest and most of them aim at finding the meter or *rhythmic score* (explained below) of the music signal under analysis, generally without taking into account music gestures or expression[3]. The process of estimating the musical meter can be seen as a subtask of the afore mentioned definition of a complete system.

In order to ground the meaning of rhythmic score, Figure 2.1 shows a simple but illustrating graphical instance, which the reader might find familiar since it resembles to that presented in the previous chapter. The inside of the dotted box contains what would be the desirable result of rhythm analysis system: the underlying metrical levels and periodicities are identified, as well as the intervening musical events and their respective time locations. However, in practice it is almost never that straightforward.

Gouyon & Dixon (2005) point out that a serious drawback of automatic rhythm analysis resides on the impossibility to explicitly define the rhythm, coupled to the fact that computer-based implementations must be formulated using precise definitions. They also use clever arguments to put into evidence the ambiguity of rhythm and they formulate questions, for example: how many metrical levels are relevant? Is there one most important level? Is there only one uniquely *correct* tactus? Which metrical levels define the time signature? Are the answers to these questions common to all listeners? (Gouyon & Dixon, 2005).

Models categorization The computer-based rhythm analysis proposals found in the literature can be categorized in several different ways. However, the most straightforward and fundamental distinction between them is, almost undoubtedly, the nature of the input signal. The earliest methods, known as *symbolic models*, used as input a symbolic audio representation, *i.e.*, a structure based on tokens describing music events, their relationship and sometimes also providing information to render them audible, for example the ubiquitous MIDI format.

Nowadays, given that the vast majority of musical signals are available in raw audio (or since a few years now also in compressed format) and also influenced by the ever

[2] In its simplest form, an audio event is a musical tone. Nevertheless, the same term is used to refer to more elaborated situations: notes, an instrument playing a chord, various instruments playing (in synchrony) musical notes, chords or any other form of musical accentuation.

[3] Laroche (2001) and Gouyon et al. (2003) are two exceptions. They determine the *swing* ratio of the signal under analysis.

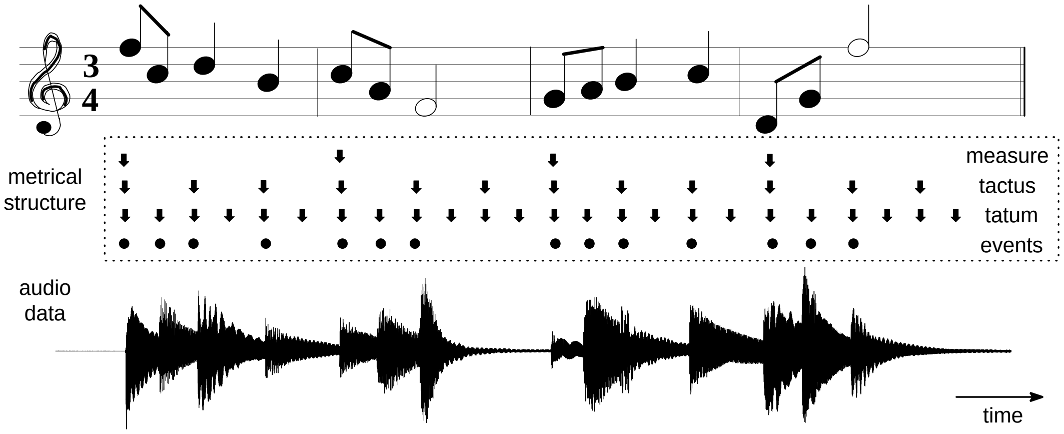


Figure 2.1: A complete computer-based rhythm analysis system should provide the full metric structure of the music signal under analysis, that is, the inside of the dotted box.

increasing computational power, most of the later developed methods leaned towards direct processing of the audio recording waveform (or on a number of its time parameters if working on the compressed domain) and are usually known as *acoustic audio models* or *signal processing models*.

The method that will be described throughout this report belongs to the second category. For this reason, more emphasis will be given below to this kind of approach. Nevertheless, it is necessary to point out that symbolic models play an important role in the consolidation of their counterparts by developing and settling many techniques later used in the signal processing models.

2.2 General principle of computational rhythm description

After having introduced in §2.1 the main goals and definitions of computer-based rhythm analysis, in this section we proceed to explain the general working principle from our point of view.

The four stages approach that we present here is heavily based on the scheme proposed by Klapuri (2004) and Klapuri et al. (2006). This scheme, graphically illustrated in Figure 2.2, is explained below.

- ⋆ First, the degree of musical accentuation as a function of time has to be measured, that is, phenomenal accents must be detected. In the case of symbolic audio this task is granted. For an audio recording input, the assignment is far from trivial and is closely related to the problem of onset detection. Some systems measure the likelihood of finding a phenomenal accent in a continuous manner, others extract discrete events.

- ⋆ Secondly, the periods and phases (locations) of the underlying metrical pulses have to be estimated. Numerous methods haven been proposed, for example: autocorrelation function, DFT, comb-filter oscillators.

- ⋆ Thirdly, the system has to identify the metrical levels such as the tactus, tatum or

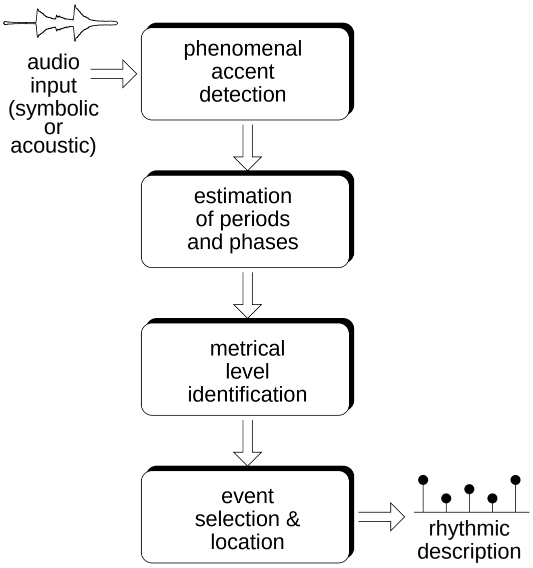

audio input (symbolic or acoustic)
phenomenal accent detection
estimation of periods and phases
metrical level identification
event selection & location
rhythmic description

Figure 2.2: Overview of the general principle used in computational rhythm description.

measure. This can be done by using *a priori* knowledge of pulse distributions or by applying pattern matching techniques.

- ⋆ Finally, the music events (including their respective time locations) related to each one of the previously found metrical levels must be selected.

Our proposition to address some of these problems will be described in detail later in chapter 3. In theory, once the afore mentioned steps have been accomplished, the result obtained must be comparable to that presented in Figure 2.1, certainly taking into consideration the inherent ambiguity of rhythm as pointed out by Gouyon & Dixon (see §2.1 page 34).

2.2.1 Symbolic and acoustic models seen as complementary approaches

At first sight, rhythm analysis algorithms processing symbolic and acoustic audio can be seen as quite similar. In effect, there are few theoretical differences between both approaches and their large discrepancies are more of practical order. The main distinctions between them are found mainly in the first two stages of scheme presented in Figure 2.2.

The foremost contrast is that symbolic models do not require to undertake (and therefore are not designed to face) the difficulty of the onset-detection step preceding beat analysis (first block in Figure 2.2). In other words, the periodicity and phase estimation stage of symbolic models operates solely on *clean* data instead of dealing with information corrupted by false detections, undetected onsets and loose attack positions. All these difficulties are inherent when processing acoustic music signals. In fact, only few of the algorithms processing symbolic audio have been successfully ported to signal processing models, for instance the methods proposed by Dixon (2001) and Raphael (2001). The remaining parts of the rhythm analysis procedure are rather similar for both kind of approaches. In the case of acoustic models some approaches extract discrete points indicating the location of potential onsets, while others process directly an onset likelihood signal that behaves like a *continuous* function.

It is important to remark that in many aspects, the symbolic *vs.* acoustic comparison is not adequate since they have different goals. A number of the symbolic models find their origins and are purely motivated by scientific research in fields like rhythm perception, music cognition research and music psychology. On the contrary, the most part of acoustic models usually obey an engineering motivation which is more related to developing an end-user device for executing tasks such as music transcription, score following, music driven special effects, all of these tasks using commercial audio recordings as input.

In addition, symbolic models (MIDI based for instance) have traditionally been a step "ahead" concerning rhythm analysis since they already have at their reach information whose availability is currently under development for signal processing models (*e.g.*multiple fundamental frequencies, chords changes, number of sources). A state-of-the-art symbolic model, for example that proposed by Meudic (2004), is capable of performing assignments such as: analyzing rhythm and motifs from polyphonic music with variable tempo, explore progressively a musical piece from its lower levels (starting at the note layer), identifying motif variations, comparing performances and extract the musical structure. Most of these tasks are still in an initial stage when dealing with audio recordings.

2.3 Literature survey: current automatic rhythm analysis

As shown below, automatic rhythm analysis has been a very active research field in recent years. Contrary to most literature surveys about rhythm analysis, we decided in the present work to separately examine symbolic and acoustic models. It should be noted that categorizations of rhythm analysis models can be, to a certain degree, subjective. Thus, boundaries in our classification are somewhat fuzzy. For instance, the rule-based approach can be seen as overlapping with the multiple-agent and histogramming models category.

2.3.1 Symbolic models

A large part of the research carried out in the metrical analysis field has been done on models that operate on a symbolic input. The main approaches can be classified as follows.

2.3.1.1 Rule-based models.

Pioneering work to develop a rhythm parsing system shared the rationale of being driven by a reduced set of heuristic *if–then* procedures to find the meter of the music signal and are thus known as *rule-based* models. They initially assume that the beat is equal to the time interval between the first two onsets, and then work their way through the incoming signal, shifting, doubling and stretching the beat. Each model postulates a state variable (the current beat hypothesis) and a small set of rules in which the test consists of a predicate on the rhythmic pattern and the current beat hypothesis, and the action modifies this beat hypothesis. A detailed explanation of this kind of approach can be found in Desain & Honing (1999).

Steedman (1977) describes a model of perception using note durations to infer accents and melodic repetition to infer the rhythmic structure of Bach's "Well Tempered Clavier" set of melodies.

Longuet-Higgins & Lee (1982) developed a model of rhythm perception that operates on a list of onset times taken from a monophonic melody and computes the beat and other higher metrical levels. This method successfully builds binary structures, but does not work for ternary meters. An extension of this work is presented in Longuet-Higgins & Lee (1984). It provides a formal description of syncopation[4] and describes a preferred rhythm interpretation as one which avoids syncopation.

Povel & Essens (1985) propose a model of perception of temporal patterns. They suppose that the listener tries to induce an internal clock which best coincides with the accents in the stimulus (short repeated tone bursts patterns) and allows it to be expressed in the simplest possible terms.

Lee (1991) summarizes and compares in a theoretical and experimental framework all of the above mentioned methods. He states that musical rhythm obeys a canonical accent pattern of strong and weak beats and that the listener can induce the meter by matching the accent patterns in music to a canonical pattern of possible rhythmic interpretations, that is, the model tries to recognize the metrical structure at several levels.

Parncutt (1994) adds two important novelties to previous models: the concept of phenomenal accents and the preference for a moderate tempo, although in his proposal he only uses durational accents. He presents a model with a direct relationship between inter-onset intervals durational accents, moderate tempo and the perceived beat. This model also tries to estimate perceived meter and expressive timing information.

Temperley & Sleator (1999) propose a hybrid meter–harmony recognition system heavily based on Lerdahl & Jackendoff (1983) GTTM's work. They align beats with event onsets and use a *length* rule, that is, longer notes are aligned with stronger beats. They use dynamic programming algorithm to search the best solution in the space of possible mappings of events to a pulse. In addition, Temperley (2004) compares his model to other methods (including some non-symbolic) and proposes an evaluation and comparison framework for metrical systems.

2.3.1.2 Multiple-agent models

In this kind of approach, a number of distinct and independent conjectures (*agents*) about pulse-periods and phases are made. Then a dynamic score is iteratively computed through time for each of these agents. Since conjectures can be pruned or split at any time, the number of agents is variable. In addition, their score increases whenever an onset coincides with a pulsation belonging to the conjecture. At the end of the analysis, the agent with the highest score wins and is considered to represent the correct pulse-period. A drawback of this approach is that it requires an initialization stage to provide an adequate number of conjectures.

[4] In music, syncopation consists in a temporary displacement of the regular metrical accent caused typically by stressing the weak beat. Syncopation is used only occasionally in most musical styles, but it is fundamental in others like jazz.

Allen & Dannenberg (1990) propose a beat-tracking system that uses beam-search[5] to find the most salient hypothesis of beat-period and placement. They use a rule that penalizes short events and event absence.

Rosenthal (1992) proposes a complete meter analysis system for polyphonic music that attempts to model the human rhythm perception. This model produces multiple-agents with a hierarchical structure and computes a score corresponding to the likelihood that a human listener would choose that interpretation of the rhythm. This model is capable of adapting to changes in tempo and meter.

Dixon's (2001) beat-tracking system although originally developed for symbolic data, is one of the few models that process also acoustic audio. In his approach, agents behavior resembles to that of Allen & Dannenberg (1990), except that he allows events to fall close to the expected beat. The beat-period agents are initialized as in the method proposed by Rosenthal (1992).

2.3.1.3 Oscillator models

Perhaps one of the most intuitive ways of obtaining metrical information from music signals is by using phase-locking oscillators. In most cases, this kind of approach consists of a bank containing several oscillators, each tuned to a different periodicity and built from a single prototype. Since a basic oscillator in general only responds to a specific frequency range (usually small), if the period of the excitation is close to the characteristic frequency, the oscillator output will resonate. The better the match between the excitation and the oscillator is, the higher the resonance energy will be. Thus, pulse-period is obtained by searching the oscillator having the highest output energy. In addition, pulse-location can be obtained by examining the phase of the most resonating oscillator.

Large & Kolen (1994) describe a non-linear oscillator prototype that uses a gradient-descent method to continuously adapt the oscillators characteristic frequency and phase. The major drawback is that basic tempo and initial phase must be supplied to the algorithm, since it only tracks pulse-period variations. The aim of this system is to model the listener expectation to a regular pulse in the music. Toiviainen (1998) proposes an extension by including short and long-term adaptation mechanisms, where the former is intended to deal with local timing deviations and the latter to follow tempo changes.

2.3.1.4 Probabilistic models

Other kind of metrical analysis systems are based on probabilistic models. They suppose that phenomenal accents have a stochastic nature and that there exists an underlying random model that rules the rhythmic process whose control parameters must be estimated.

Raphael's (2001) proposal is based on Bayesian networks and hidden Markov models. His approach handles symbolic and acoustic audio data. The tempo and pulse location are modeled as hidden variables and the results are obtained by using a maximum *a posteriori* estimation.

Cemgil has developed two different statistical modeling approaches. The first one (Cemgil et al., 2001) is related to the theory of linear dynamical systems and presents a

[5] Beam search is a heuristic search algorithm which only expands the n most promising nodes at each depth, where n is a constant number known as the *beam width*. Initially, the n nodes are chosen according to some prefixed rules The successors of these n states are all calculated. If the Goal Node is reached, the algorithm halts. Otherwise, the best n states of these successors are taken and the steps repeated.

beat-tracking method based on a Kalman filter who searches the smoothest path through a local periodicity representation (computed from an onset location vector) that he calls *tempogram*. Cemgil & Kappen (2003) present a probabilistic generative model for timing deviations in expressive music performance. As in Raphael (2001), Cemgil formulates the tempo tracking as a filtering and maximum *a posteriori* state estimation problem that he solves using particle filtering and Markov Chain Monte Carlo methods.

2.3.2 Signal processing models

During the last years there has been a considerable increase in research devoted to developing rhythm analysis models for acoustic signals. In this part we aim at providing a general overview of the approaches commonly found in the literature. As in the symbolic models case, these methods are categorized according to the broad principles they use. Although somewhat simplistic, for the sake of clarity we suppose that the signal processing models are composed of three stages: a front-end which estimates the *degree of musical accentuation*, the periodicity estimation or *pulse induction block* and finally the *pulse-tracking* block. In the following part we describe a number of the existing proposals for each of these tasks.

2.3.2.1 Estimating the degree of musical accentuation

The first stage of any meter analysis system processing music recordings consists in breaking-down the audio data into a temporal series of features which convey the predominant rhythmic information. Gouyon et al. (2006) refer to this task as the "feature list creation" block, we call this process the "acoustic-to-symbolic conversion". Various techniques have been proposed, they can be classified as follows.

– **Signal energy**. This is a straightforward way of obtaining a musical stress profile at a low computational cost, it simply consists in computing the energy envelope of the signal as the sample-wise sum of the squares over successive and overlapping segments of the audio signal. It displays a good performance for percussive music, but in general it cannot cope with more challenging cases without additional processing. It has been used by Dixon (2001), Gouyon et al. (2002) and more recently by Eck & Casagrande (2005).

– **Filter bank methods**

  ⋆ **STFT based methods**. One of the most common techniques to conduct the acoustic to symbolic conversion is based on the use of the Short-Time Fourier Transform (STFT) as a filter bank. That is, the STFT is merely used as an efficient way to decompose the input signal in frequency channels. Then, these frequency bands are processed to highlight the phenomenal accents. The processing usually consists of computing the derivative of the energy envelope and integrating all the channels to form a single musical stress profile (Laroche, 2001, 2003; Alonso et al., 2004; Peeters, 2005).

  There are also proposals which only integrate certain frequency regions and therefore use various musical stress profiles simultaneously (Klapuri, 2003; Klapuri et al., 2006; Alonso et al., 2005b). Others proposals also employ this principle, but using a set of profiles for each critical band in the spectrum according to a perceptual scale such as Bark or ERB[6] (Sethares & Staley, 2001; Jehan, 2004; Uhle et al., 2004).

  Other methods, focusing more specifically on modern popular music, consider that phenomenal accents are located at time instants where the upper part of the spectrum is more energetic, thus they only compute a high-frequency-content profile Dannenberg (2005).

[6] ERB stands for Equivalent Rectangular Bandwidth.

A rather different approach obtains the musical stress profile by measuring the acoustic self-similarity of the audio signal as a function of time-lag (Foote, 2000). In this case the STFT is used to obtain a sequence of feature vectors. This information is then embedded into a 2-dimensional representation by finding a similarity measure computed over all possible combinations of the feature vectors (Foote & Uchihashi, 2001).

⋆ **Other filter banks**. The working principle is exactly the same to that used by the STFT methods, *i.e.*, to decompose the audio signal in frequency bands and then compute the derivative of the energy envelope, thus obtaining a musical stress profile for each subband signal. In general, for this kind of approach the subbands are logarithmically distributed in frequency (Scheirer, 1998; Paulus & Klapuri, 2002; Seppänen, 2001b; Uhle & Herre, 2003). The use of a uniform frequency decomposition has also been proposed by Alonso et al. (2003a).

Tzanetakis & Cook (2002) have proposed an equivalent approach, but using the discrete wavelet transform as a constant–Q (center frequency/bandwidth) filter bank with octave spacing between the centers of the filters.

Hainsworth & Macleod (2003a) have developed a hybrid accent detector. It uses a filter bank for detecting transient events with strong energy changes associated. Just like the afore mention models, it is obtained by computing the energy envelope in three bands containing low, middle-high and high frequencies respectively. Another part of the system aims at detecting harmonic transitions not related to large energy changes. This is done by computing the STFT and then measuring the cosine distance between two consecutive frames.

Wang & Vilermo (2001) propose an algorithm for music with a very regular rhythmic structure (pop, techno). This model has a distinctive characteristic that makes it slightly unusual from the rest of signal processing approaches because it processes encoded audio bit-streams in MP3 format directly in the compressed domain. This beat detector employs directly the window-type[7] data and the decoded MDCT coefficients to detect onsets.

– **Low-level descriptors**. Undoubtedly, the concept of phenomenal accents or onsets as temporal events in an audio stream has a great relevance in rhythm analysis. If we were asked to find a common property that encompasses the totality of the front-ends previously discussed, perhaps the best answer will be that (practically) all of them search for onset clues using methods solely based on one kind of *descriptor*: energy variations in one or several frequency bands. This heuristic approach has proved to be highly successful since phenomenal accents produce variations in the loudness, pitch or timbre, which consequently yield to energy variations in the audio signal.

In the quest for improving rhythm analysis, recent research has explored additional alternatives by searching low-level acoustic descriptors that are adequate for identifying musical beats from a computational perspective. In fact, extracting descriptors from audio recordings to characterize aspects of the audio content is by no

[7] The MP3 format employs four different window types: long, long-to-short, short and short-to-long. This window size parameter is introduced to better cope with transient signals.

means a new area of research. Much effort has been spent on descriptors extraction in areas like speech processing and more recently on audio signal analysis. It is out of the scope of this document to give an overview of the audio descriptors currently employed, but a well-grounded introduction of their use for characterizing music can be found in (Widmer et al., 2005).

To our knowledge, the research carried out by Seppänen (2001a) is the first endeavor to explore the use of diverse low-level audio descriptors for analyzing the rhythmic content of music signals. This model does not use this information to segment the audio stream or to find onsets, but as a beat recognition device meant to model phenomenal accentuation capable of differentiating strong from weak accents in the music.

Research on the exploration and assessment of low-level audio descriptors for metrical analysis has been carried out by various groups (Jensen & Andersen, 2003; Sethares et al., 2005; Gouyon, 2005). The perspective of these methods is different from the approach taken by Seppänen (2001a), who views the classification process as an actual method for finding beats. These methods aim at determining the low-level descriptors (of audio signals) that best convey the rhythmic information in music. More precisely, to select among several low-level features computed at a regular sampling rate, those whose temporal behavior would best indicate the presence and localization of beats.

One of the most exhaustive studies in this domain has been carried out by Gouyon (2005), he evaluates the effectiveness of 274 singleton descriptors using a machine learning methodology. After evaluating the quality of these descriptors, he proposes a subset of 59 elements yielding good accuracy figures in the context of rhythm analysis (see Gouyon, 2005, page 102).

Two final observations can be made concerning these methodologies of musical stress estimation. From one part, many different frequency decompositions have been used, yielding comparable results. However, there is a tendency towards using frequency decompositions related to human perception, *e.g.*, using subbands distributed in a logarithmic, ERB or Bark-scale fashion.

The other important aspect is that some of these approaches carry out an explicit phenomenal accent detection, producing as output a *discrete* musical stress profile. More exactly, a precise list containing the time location and musical salience for each of the *potential* onsets (Dixon, 2001; Seppänen, 2001b; Gouyon et al., 2002). On the other side, the counter part of the discrete onset detection approaches are the so-called *Detection Function* models. These approaches does not aim at precisely extracting onset positions, but rather at obtaining a smooth or *continuous* musical stress profile usually known as the ″detection function″. This signal indicates the possibility of finding an onset as a function of time. In addition, this profile is usually built from the subbands time waveform envelope (Scheirer, 1998; Laroche, 2001; Klapuri, 2003).

2.3.2.2 Pulse induction block

After obtaining the musical stress profile, the next stage consists in estimating its periodicity. Various different methods have been proposed.

- **Multiple-agent models**. This approach is similar to that used in the symbolic audio models, a number of distinct and independent hypothesis (called *agents*) about pulse-periods and phases are made. Then a dynamic score is iteratively computed through time for each of these agents. The beat is obtained from the agent having the highest score. This kind of approach uses as input a discrete musical stress profile. The research carried out by Goto (2001) is one of the best known multiple-agent algorithms which process audio recordings. Dixon (2001) proposes another method also based on the same principle.

- **Histogramming models**. Rhythm analysis models based on the computation of an inter-onset interval (IOI) histogram have been proposed. The principle is to cluster similar IOIs into a histogram-alike "class" representation, where each IOI belongs to one class. The beat-period is obtained by selecting the class with the highest number of elements (Seppänen, 2001b; Gouyon et al., 2002; Jensen & Andersen, 2003).

- **Correlative models**. The autocorrelation (AC) is an ubiquitous method for finding periodicities in data which has been used in many fields. Hence, several rhythm analysis approaches are based on this method (Foote & Uchihashi, 2001; Uhle et al., 2004; Tzanetakis & Cook, 2002; Davies & Plumbley, 2005).

  Peeters (2005) uses an interesting method for estimating the periodicity jointly in frequency and time domains by combining the magnitude of the Discrete Fourier Transform (DFT) with a frequency-mapped autocorrelation function (ACF), *i.e.*, the absolute value of the DFT is weighted by the frequency-mapped autocorrelation[8]. The goal of this operation is to reduce the inherent tempo-octave ambiguity by allowing peaks present in both domains to be reinforced and by undermining peaks being present in only one domain.

  Paulus & Klapuri (2002) propose a beat analysis method built around an autocorrelation-like function called YIN, which includes a number of modifications to prevent errors. This technique was originally developed by Cheveigné & Kawahara (2002) as a fundamental frequency estimation algorithm for speech signals.

  Eck & Casagrande (2005) present a system that analyzes the meter of audio signals based on a characteristic usage of the AC. This approach computes the distribution of the AC energy in the phase space, yielding to the so-called *autocorrelation phase matrix*[9]. According to its authors, this method significantly improves the performance of the standard autocorrelation by taking advantage of how energy is stored and distributed at different lags in the autocorrelation matrix.

  Laroche (2003) has developed a system to determine the beat period and position at a given time. The solution is obtained by computing the cross-correlation between a

[8] In order to keep an acceptable resolution during the mapping operation, the ACF is resampled at a considerably higher rate.

[9] According to Eck & Casagrande (2005), the autocorrelation phase matrix is a data structure designed to overcome some limitations of the traditional AC. It stores phase and period information in the same data structure. In addition, it also adds statistical measures of spread (like variance or entropy) to the information stored for each lag.

set of "expected" musical stress profiles (*i.e.*, pulse-train like signals) and the actual stress profile obtained from the audio signal.

A common characteristic of the correlative methods cited above is that all of them use as input a continuous musical stress profile.

– **Oscillator models**. Another popular method for finding the most likely pulsation periodicity consists in using an oscillator network, or also known as an oscillator bank. Multiple copies of a basic oscillator are used to account for different period hypothesis. Then, the set of oscillators is excited by a continuous musical stress signal and the filter which corresponds best to the frequency of the data receives the highest excitation. Pulse location can be calculated by examining the phase of the oscillator with the highest energy. This particular method is also suited for a continuous tracking of the pulsation Scheirer (1998); Klapuri (2003); Jehan (2004); Klapuri et al. (2006).

– **Probabilistic models**. The strategy used in the probabilistic approaches bears a high resemblance with the motivation used by multiple agents methods, but in the earlier case the reasoning employed is purely stochastic. Probabilistic models suppose that phenomenal accents have a stochastic nature and that there exists an underlying random model that rules the rhythmic process whose control parameters must be estimated. A number of techniques have been proposed to estimate these parameters. For example, the maximum likelihood of set of variables which best fit the audio data (Laroche, 2001) or particle filtering algorithms (Hainsworth & Macleod, 2003b; Sethares et al., 2005).

– **Other models**. This kind of models cannot be characterized with a common property, since we consider that they only apply a signal processing technique to the problem of estimating the pulsation period.

Sethares & Staley (2001) have developed a particular pulse period induction method using the so-called *periodicity transforms* (Sethares & Staley, 1999). In this case, a continuous musical stress profile is decomposed into a sum of periodic sequences by projecting it onto a set of periodic subspaces. The algorithm finds its own set of non-orthogonal basis elements based on the data, rather than assuming a fixed predetermined basis as in the Fourier-like transforms. This approach is able to discern the pulsation at several layers of the rhythmic structure, even measures and musical phrases. A drawback of this method consists in its sensitivity to the sampling rate of the stress profile, since it is primarily designed to search for integer periodicities and it is not always possible to assure that the period will be an integer number of samples.

Alonso et al. (2003b, 2004, 2005b) have proposed two frequency domain techniques to carry out periodicity analysis: the spectral sum and the spectral product. These methods were originally developed for estimating the pitch period of voiced speech sounds. They will be explained in detail later in §4.1.1.

In the method developed by Dannenberg (2005), pulse induction is based on a predefined pattern matching which models the alternating strong and weak beats in the stress profile with a fixed period. To perform beat induction, the pattern is stretched in small increments inside a beat period range and time-shifted. For a given pulsation period and time-shift amount, the "goodness of fit" is computed.

The tempo and shift values are further refined using a gradient descent algorithm to find the best local fit to the stress profile function.

2.3.2.3 Pulse tracking

The pulse tracking module is the last stage of the simplified rhythm analysis model that we use to provide an overview of the field. There are two different methodologies used in the pulse tracking stage.

Gouyon & Dixon (2005) call the first of them "tracking as repeated induction". As indicated by the name, this methodology consists in iteratively repeating the pulse induction. More exactly, the periodicity is induced in a short analysis window of the musical stress profile (usually of a few seconds), then the window is time-shifted to include new data (generally by a fraction of the window length) and then the induction process is repeated again. In this case, observations to the tracking process are no longer stress profile segments, but the period value (and in some implementations also the pulse phase location). Thus, the pulse evolution is obtained by directly linking the periodicity and phase observations at each iteration (Sethares & Staley, 2001; Foote & Uchihashi, 2001; Dixon, 2001; Alonso et al., 2004). A drawback of the tracking approach described above is the potential lack of continuity between successive observations.

In the second, pulse tracking methodology is also based on a repeated induction process, but this time each pulse induction iteration produces a set of potential periods (and in some implementations also a set of potential pulse phases). Therefore, finding the "optimal" pulse period curve and pulse locations reduces to computing the best path that connects all the successive hypothesis. If the pulse tracking problem is developed in a deterministic framework, the solution can be found using techniques such as dynamic programming (Laroche, 2003; Alonso et al., 2005b). If the problem is developed in a probabilistic framework, the Viterbi algorithm (Klapuri, 2004; Peeters, 2005; Klapuri et al., 2006) or particle filtering techniques (Hainsworth & Macleod, 2003a; Sethares et al., 2005) can be used.

2.3.3 Comparison Table

In order to provide a quick panorama of the current metrical analysis field for acoustic signals, we have put together a number of systems. Table 2.1 lists these techniques in chronological order as well as a number of important attributes and characteristics that they have. This compilation is by no way an exhaustive catalog of the currently available methods, but a record of those algorithms that we have found in the literature during the last years.

Table 2.1: Comparison and characteristics of various metrical analysis systems.

	Method	Metrical levels	Approach	Musical stress profile	Public evaluation	Source code available	Evaluation material
1	Goto & Muraoka (1994) Goto (2001)	measure, half-note and quarter-note	multiple agents	discrete			85 pop-music pieces
2	Scheirer (1998, 2000)	tactus	network of oscillator filters	continuous	✓	✓	60 pieces with ”strong beat”
3	Laroche (2001)	tactus, swing	probabilistic (GMM)	discrete			not available
4	Dixon (2001)	tactus	multiple agents, rule-based	discrete	✓	✓	10 pieces with ”strong beat”
5	Mayor (2001)	tactus	multiple agents, heuristic	discrete			not available
6	Seppänen (2001b)	tatum, tactus	IOI histogram	discrete		✓	50 pieces various genres
7	Wang & Vilermo (2001)	tactus	IOI histogram	discrete			6 pop-music pieces
8	Foote & Uchihashi (2001)	tactus	correlative	continuous			8 pieces various genres
9	Sethares & Staley (2001)	meter[10]	periodicity transform	continuous	✓	partially	a few pieces
10	Seppänen (2001a)	meter	probabilistic (GMM, LDA)	low-level descriptors			330 pieces various genres
11	Gouyon et al. (2002)	tatum	IOI histogram, TWM	discrete			57 short drum sequences
12	Tzanetakis & Cook (2002)	tactus	ACF	continuous	✓	✓	not available
13	Paulus & Klapuri (2002)	meter	correlative	continuous			365 pieces various genres
14	Dixon et al. (2003)	meter and style	ACF	continuous	✓		170 pieces of dance music

Continues on the next page...

[10] tatum, tactus and measure (bar).

Table 2.1: Comparison and characteristics of various metrical analysis systems (second part).

	Method	Metrical levels	Approach	detection function	Public evaluation	Source code available	Evaluation material
15	Laroche (2003)	tactus	correlative	continuous			a few pieces of various genres
16	Jensen & Andersen (2003)	tactus	IOI histogram	discrete		✓	2164 pieces of popular music
17	Hainsworth & Macleod (2003a)	tactus	probabilistic (particle filters)	discrete			175 pieces various genres
18	Klapuri (2004); Klapuri et al. (2006)	meter	oscillator filters	continuous	✓		474 pieces various genres
19	Uhle et al. (2004)	meter	ACF, TWM	continuous	✓		445 pieces various genres
20	Davies & Plumbley (2004)	tactus	ACF, oscillators	continuous	✓		222 pieces various genres
21	Jehan (2004, 2005b)	tactus	oscillator filters	continuous		✓	not available
22	Chua & Lu (2004, 2005)	perceptual tactus	ACF	continuous			50 pieces various genres
23	Peeters (2005)	tactus & meter/beat subdivision	ACF, DFT	continuous	✓		1038 pieces various genres
24	Collins (2005b)	tactus	correlative	continuous		✓	1 drum kit sequence
25	Eck & Casagrande (2005)	tactus	correlative	continuous	✓		1163 pieces various genres
26	Sethares et al. (2005)	tactus	probabilistic (particle filters)	low-level descriptors			9 pieces various genres
27	Gouyon (2005)	tactus and style	DFT, ACF, oscillator filters	low-level descriptors	✓		3223 various genres
28	Dannenberg (2005)	tactus	pattern-matching, gradient-descent	continuous			16 pop-music pieces
29	Alonso et al. (2003b, 2004, 2005b)	tactus	spectral sum, spectral product, ACF	continuous	✓	partially	961 pieces various genres

2.4 Evaluation

As presented above, many researchers in the computer music community have devoted a large effort to developing algorithms capable of analyzing automatically the rhythm of audio recordings. Whatever the goals and assumptions of a given analysis method, an important and rather obvious question to ask is: "how good is it?". There exists no simple answer and in fact, this question has been *per se* subject of a number of publications or a constitutive element in others, for instance (Temperley, 2004; Cemgil et al., 2001; Goto & Muraoka, 1997a).

Temperley (2004) states that a successful evaluation system for metrical models must fulfill four basic requirements:

1. an agreement upon the way of representing the information to be retrieved,
2. a suitably large and representative corpus of data,
3. a correct analysis (*i.e.*, annotation) of the corpus representing the information to be retrieved, this process is better known as computing the *ground-truth* of the test database or also as database annotation; and
4. an agreement upon the way of comparing a model's analyses of the corpus to the correct analyses and scoring the model on its success at matching the correct analyses.

Until recently there was no consensus on these requirements and most of the work to evaluate the algorithms described in §2.3.2 have been isolated efforts. As a result, in many cases such methods cannot be objectively compared for a number of reasons. For example, Temperley (2004) proposes a comprehensive framework for testing and comparing metrical models who addresses some inefficiencies of the earlier frameworks proposed by Goto & Muraoka (1997a) and Cemgil et al. (2001), but it has the major disadvantage of being limited to symbolic audio input.

The proposal by Goto & Muraoka (1997a) targets audio recordings, but requires that the input signal to be supplemented with markers, added by hand[11], indicating the exact location of every beat. Unfortunately, these stipulations are totally impractical from the point of view of manual annotation if we consider using a test database containing several hundreds (not to say thousands) of musical pieces in order to obtain performance figures with a high degree of confidence. In reality, due to the arduousness and the intrinsic time-consumption of the task, almost all researchers have to content with a much simpler way of annotating the test data.

The ubiquitous fashion of annotating the test corpus only considers one single rhythmic level (the tactus) and consists in listening to each musical excerpt while tapping along the corresponding rhythm (see for example Alonso et al. (2003b)). Simultaneously, the tapping signal is recorded with the help of a microphone, then the average of the inter-beat intervals is calculated and used as the *ground-truth* tempo value. Some researchers (McKinney & Moelants, 2004; Klapuri et al., 2006) also set up a recording installation where the annotator's tapping signal is in perfect synchrony with the musical signal, providing also a *ground-truth* for the beat locations.

[11] To label the exact beat-positions, Goto & Muraoka (1997a) developed an editor program that enables the annotator to mark beat positions in the digitized signals while listening to the corresponding audio and visually inspecting its spectrogram and waveform.

In addition to the ground-truth requirement (the third from the list presented above), it is well known that human manners of tapping along with the music contain a strong subjective and cognitive component[12]. It is, thus, reasonable to deduce that the result produced by a given analysis method must be defined as "correct" if it is in accordance with the rhythm that would be inferred by a (suitable) human listener. A closely related issue is that the rhythm found by human listeners might not be unique, that is, there might be individual differences among listeners. This topic has been studied and documented in the literature by various researchers: McKinney & Moelants (2004), Todd (1999), van Noorden & Moelants (1999) and others. According to McKinney & Moelants (2004), the notated tempo is that which can be obtained from the music score. Then, it is straightforward to attach this value to a musical excerpt. For those instances that do not have an "official" tempo annotation available, it is also possible to annotate the "perceived" tempo. This is not a straightforward task and needs to be done carefully. If someone asks a group of listeners (including musicians and non-musicians) to annotate the tempo of musical excerpts, they can provide different answers (they tap at different metrical levels) if they are unfamiliar with the piece. For some excerpts the perceived pulse or tempo is less ambiguous and everyone taps at the same metrical level, but for other excerpts the tempo can be quite ambiguous and a complete split across listeners can be obtained (McKinney & Moelants, 2004).

Also of considerable importance in the evaluation procedure is the fact that in general, researchers private test databases (consisting of at most a few hundreds of excerpts usually with a duration of less than a minute) are extracted from commercial audio recordings, which are copyrighted material. This is a legal barrier that prevents researchers from sharing their test corpora, which also complicates quantitative comparisons between systems.

The first step towards establishing a methodology of evaluation and comparison for tempo induction algorithms, along with several other *music information retrieval* (MIR) tasks, was taken in 2004 by the steering committee of the International Conference on Music Information Retrieval (ISMIR). See Appendix A for details about this international evaluation.

2.5 Test corpus description

The aim of this section is to give a thorough description of the two test databases which will provide us the ground-truth information. These corpora will be extensively used to examine the performance of the algorithms presented in subsequent chapters.

2.5.1 Database for tempo analysis

One of the first tasks accomplished in this work was the collection of a database of musical excerpts. This corpus was extracted from commercial audio CDs. From each recording a characteristic excerpt was selected, which was then converted to a monophonic signal sampled at a rate of 16 kHz with 16 bit resolution. The corpus contains 961 excerpts with a global length of 18759 seconds (5 hours 12 minutes and 39 seconds) of music in

[12] For example, Drake et al. (2000) suggest that musicians have the greatest range of available metrical layers and that they prefer to tap at slower levels than nonmusicians.

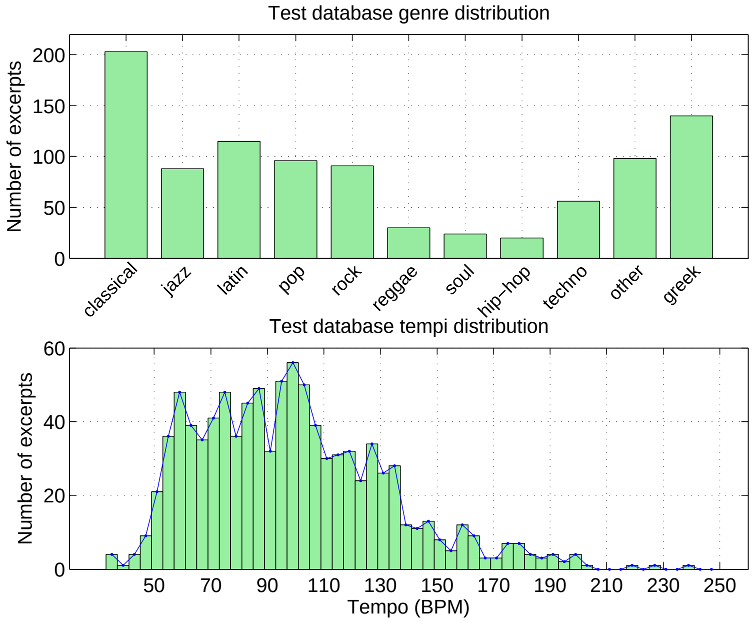


Figure 2.3: Test database information. On top the genre distribution and in the bottom the tempi distribution.

total. The minimum excerpt length is 10 seconds, the maximum length is 30 seconds, and the mean length is 19.52 seconds.

This database was formed after merging a database gathered at ENST (52% of the corpus) and a free database (remaining 48% of the corpus) provided by the Music Technology Group (MTG) at Pompeu Fabra University[13] (Barcelona, Spain). The audio recordings were selected to cover many kinds of instruments, dynamic ranges and a large tempi region: from 36 to 240 BPM. The music included is: with and without percussions, with and without vocals, and with live and studio recordings. In addition, the corpus represents eleven different musical genres: classical, jazz, latin, pop, rock, reggae, soul, hip-hop (& rap), techno, other (film sound tracks, folk) and traditional greek music. The genre categories and selection were made according to those of *Amazon.com*. Figure 2.3 summarizes the musical genres and tempi distribution of the test corpus.

The reader might wonder: why the signals are not sampled at 44.1 kHz? We consider that for a large part of cases, the audio signal bandwidth required for an adequate phenomenal accent detection lies in the low and middle frequency ranges, *i.e.*, below 8 kHz. Undoubtedly, the use of upper frequencies during the accent estimation is desirable, however their inclusion in the analysis entails a higher computational complexity. In this trade-off involving estimation accuracy *vs.* algorithm complexity, we opted for the latter and we use signal sampled at 16 kHz.

All of the corpus instances have a *clear* and stable rhythm. For the fraction of the test corpus collected at the ENST, each excerpt was meticulously manually annotated by three skilled musicians. Separately, each annotator tapped along with the music while the tapping signal was being recorded. The *ground-truth* tempo was computed in a two

[13] This database was also used during the first “tempo extraction contest” (Gouyon et al., 2006). It can be obtained at: `http://www.iua.upf.es/mtg/ismir2004/contest/tempoContest/node3.html`.

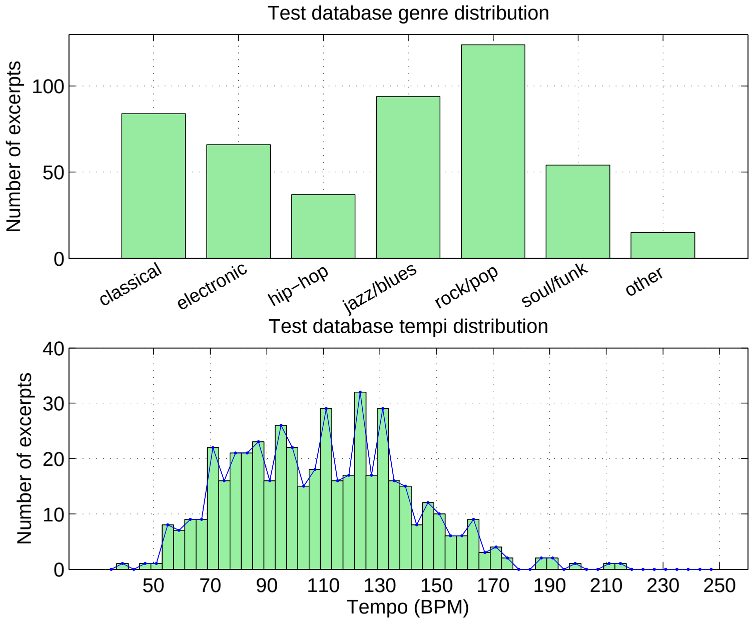


Figure 2.4: TUT test corpus information. On top the genre distribution and in the bottom the tempi distribution.

step process. First, the median of the inter-beat intervals was calculated. Then, concording annotations from different annotators were directly averaged, while annotations disagreeing by a *permissible* integer multiple (as explained in §4.2.3) were normalized in order to match with the majority before being averaged.

The song excerpts database provided by the MTG was annotated in a similar way. A professional musician placed beat marks on the music instances excerpts and the groundtruth was computed as the median of the inter-beat intervals (Gouyon et al., 2006).

2.5.2 TUT database

The second test corpus that we use was recently obtained by means of a database exchange with the Audio Research Group of the Tampere University of Technology (Tampere, Finland). This database was created using a similar procedure as the one described in §2.5.1. A more detailed description of this database is provided in (Klapuri, 2004; Klapuri et al., 2006). Klapuri et al. (2006) points that this corpus was gathered for the purpose of musical signal classification in general and the balance between genres is according to an informal estimate of what people listen to.

This database contains 474 instances. Opposite to the test corpus mentioned above, this one includes entire music songs and not excerpts. The total length is 126595 seconds (35 hours 9 minutes and 55 seconds). The minimum instance length is 42 seconds, the largest instance length is 829 seconds, and the mean instance length is 267 seconds. The tempi range is from 41 to 216 BPM. Figure 2.4 shows genre statistics (taken from (Klapuri et al., 2006)) tempi distribution of the TUT test database. Instances are formatted as monophonic signals sampled at a rate of 44.1 kHz and 16 bit resolution.

The corpus was provided with annotations at the tactus and tatum levels. Each instance was manually annotated for approximately one-minute, where the excerpt was selected to represent each piece. Beat annotations were made by a musician who tapped along with the pieces (Klapuri et al., 2006). The time position of the manual annotations is also available, thus providing a *ground-truth* for the beat location.

2.6 Conclusions

In this chapter we have introduced the main goal of computer rhythm analysis as an attempt to artificially replicate the process by which humans understand musical rhythm. We have pointed out that the existing proposals to carry out automatic rhythm analysis can be classified in several ways, but the most important characteristic concerns the nature of the input signal. Under this distinction, we categorize such models as *symbolic* or as *acoustic*, the latter are also known as *signal processing* models. We have also stressed that during the present work we develop a framework which belongs to the second category.

We have introduced the general principle of computational rhythm description as composed of four stages:

- the degree of accentuation as a function of time has to be measured;
- the periods and phases of the underlying metrical levels have to be estimated;
- the system has to identify the metrical levels,
- the music events corresponding to each metrical level must be selected and located in time.

We also conducted a literature survey about symbolic and acoustic models with a significant emphasis to the latter. Table 2.1 provides a panorama of various recent proposals on acoustic methods and it also highlights their most important characteristics.

In addition, we addressed the problem of evaluating rhythm analysis models. We also highlighted the importance of the so-called *ground-truth* during this process, as an assessment tool to verify their proper operation.

Finally, we described the composition of the ground-truth corpus that will be used to evaluate the performance of the framework that will be presented in the following chapters.

Chapter 3

Estimating the degree of musical accentuation

In the previous chapter we presented a general perspective of the automatic metrical analysis field and we briefly outlined the main tasks involved. We also pointed out that every system processing music recordings has a front-end that estimates the degree of musical accentuation, *i.e.*, a system whose input consists solely on the digitized waveform of an audio signal and its respective output is a profile formed of *impulses* representing the rhythmic activity. This output signal not only contains information about the location of the note onsets but it also provides clues about their salience. In this chapter we provide a detailed description of an original approach we have developed to accomplish this conversion task. According to the scheme presented in Figure 2.2, this chapter only addresses the top block of the analysis.

Although intimately related with the next chapter, the method to estimate the degree of musical accentuation presented here is independent of a full rhythm analysis system (see Figure 3.1). In fact, it can be employed in any other application requiring a likelihood profile of onset presence or information about the salience of the acoustic events *embedded* in the audio signal.

3.1 Introduction

We have stated beforehand that the goal of this work is the design of mechanisms contributing to the estimation and recognition of the rhythmic structure in musical signals at two metric levels: the tatum and the tactus. For that purpose we found our research on the somewhat abstract scheme laid down in Figure 2.2. To ground it we have developed a block-wise framework, which is graphically illustrated in Figure 3.1. This diagram outlines the main stages involved in our approach. For the sake of clarity the whole system has been divided into two parts. Only the stages truly processing audio signals and required during the estimation of musical accents (*i.e.*, pre-processing, harmonic-plus-noise decomposition and musical stress estimation) are described in this chapter. The literature shows that there exist many ways of accomplishing these assignments, hence comparisons to state-of-the-art systems will be made and whenever possible a set of potential solutions will be explored.

With the intention of building a system capable of dealing with a large music variety, our approach avoids as far as possible the use of any high-level information about the

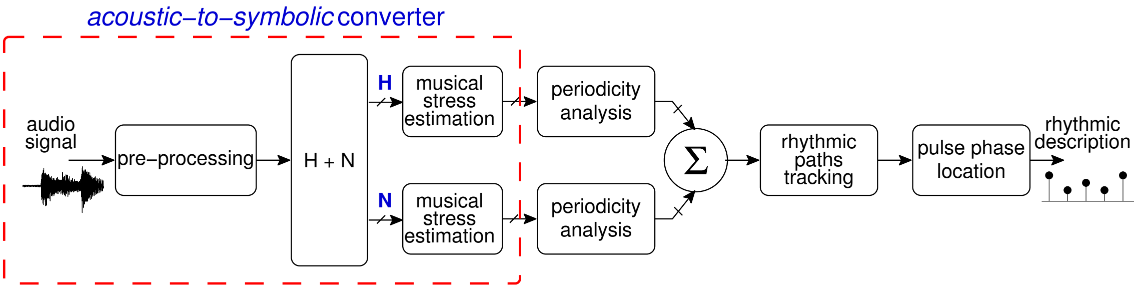


Figure 3.1: Flow diagram of the rhythm analysis framework

input audio. Such as, for example, data concerning the number or kinds of intruments playing in the recording being analysed.

The reader might notice that under certain configurations, the system shown in Figure 3.1 duplicates components. This redundancy was preferred over efficiency in order to privilege a full modularity and independence between blocks and thus ease the exploration of more possibilities[1]. In addition, we provide specific details about the causality of every block and in some cases also about the computational requirements.

3.2 Pre-processing

The first stage of the system consists in a preliminary processing of the audio data to prepare it for the harmonic-plus-noise (H+N) decomposition. There exists a ubiquitous assumption that the average power spectral density (PSD) of most audio signals behaves as a decreasing function of frequency. For example, Figure 3.2-a shows the average spectral density computed using a 20 s audio recording. The decreasing trend in magnitude as frequency increases is very noticeable. In order to ensure an appropriate harmonic-plus-noise separation and a correct estimation of the music signal parameters, it is important to assure that the noise-level power at low-frequencies does not exceed the signal level at high-frequencies. It is then necessary to compensate this power slope in frequency by *whitening* the music signal before continuing the analysis. To perform this task we propose two different techniques as explained below.

3.2.1 Causal approach

The first technique refers to the case where causality constrains are imposed to the rhythm analysis system. Under those requirements, a straightforward and computationally simple method yet producing acceptable results consists in filtering the input signal using an *equalization filter* which tries to level the signal power in the passband. In practice a simple preaccentuation filter can be used

$$H(z) = 1 - a_1 z^{-1}$$

where a_1 is smaller than 1 but very close to it. Figure 3.2 presents an example for $a_1 = 0.97$. The top part shows (blue trace) the average spectrum of the input signal and the fre-

[1] As the mathematician Donald E. Knuth once stated: “Premature optimization is the root of all evil (or at least most of it)”.

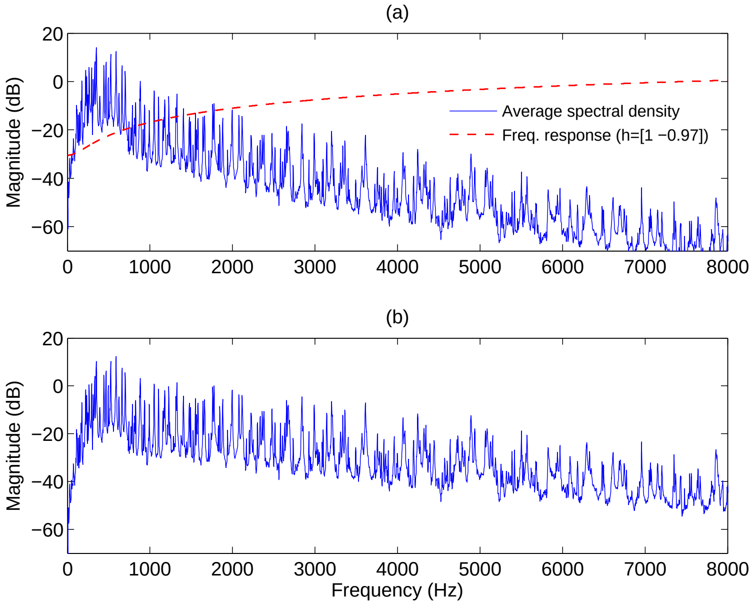


Figure 3.2: Causal pre-processing example. (a) average spectrum of the input signal (blue trace) and (b) detrended spectrum by filtering the audio signal with $H(z)$.

quency response of $H(z)$ (red trace). The bottom part shows the detrended average spectrum. We observe that an increase of about 20 dB was obtained in the upper part of the spectrum.

3.2.2 Non-causal approach

The causal method described above is a partial solution to the problem since the decreasing trend was reduced, but not entirely compensated. Much better results can be obtained by lifting the causality constraint and by using an approach originally proposed by Badeau (2005, page 153). This procedure is based on the assumption that the audio signal can be modeled as an autoregressive (AR) process plus a sum of sinusoid components.

3.2.2.1 Whitening of an AR process

Let us suppose that the audio signal $x(n)$ is stationary and thus it can be modeled as an AR process obtained after filtering a white noise signal (with variance σ^2) using a filter with transfer function $\frac{1}{H(z)}$, where all the zeros of

$$H(z) = 1 + a_1 z^{-1} + \ldots + a_p z^{-p}$$

are located inside the unit circle, thus $x(n)$ is a centered stationary process. Linear prediction is a widely used technique to compute the filter coefficients of $H(z)$ and the variance σ^2 using an estimator of the autocorrelation function $r_x = \mathbb{E}\{x(u)^* x(n+u)\}$ (Scharf, 1991; Hayes, 1996). Let $g(n)$ be an analysis window of finite support N, and let us define an estimator of the autocorrelation function $\hat{r}_x(n) = \frac{1}{N}\left([\tilde{g} \cdot \tilde{x}] \star [g \cdot x]\right)(n)$, where

$\tilde{x}(n) = x(-n)^*$ and $\tilde{g}(n) = g(-n)^*$ are the respective time-inverted and conjugated versions[2]. The estimator $\hat{r}_x(n)$, of support $2N-1$, is defined as a convolution and thus can be efficiently computed using the Fast Fourier Transform (FFT). In other words, let N' be the next higher power of 2 bigger than $2N-1$. Then, the $2N-1$ non-zero samples of $N\hat{r}_x(n)$ are extracted from the *circular* convolution between x' and $\tilde{x}'$, where x' of length N' is the zero padded version of the (non-zero) windowed signal $g(n)x(n)$ of length N and $\tilde{x}'(n) = x'(-n)^*$. The circular convolution is obtained by computing the inverse FFT of the square of the absolute value of the FFT of x', $|X(e^{i2\pi f})|$ where $f \in \frac{1}{N'}\mathbb{Z}$. Thus, $\hat{r}_x(n)$ is obtained by computing the inverse Fourier transform of the periodogram

$$\widehat{R}_x(e^{j2\pi f}) = \frac{1}{N}|X(e^{j2\pi f})|^2. \tag{3.1}$$

The estimation of the whitening filter can be summarized in five stages:

1. windowing of $x(n)$ using the analysis window $g(n)$,
2. forward Fourier transform (with the respective zero padding),
3. calculation of the periodogram defined in Eq. (3.1),
4. calculation of $\hat{r}_x(n)$ using the inverse Fourier transform,
5. estimation of $H(z)$ by linear prediction using $\hat{r}_x(n)$ as observation (Scharf, 1991).

This method has two advantages:

◇ the estimator of $\frac{1}{H(z)}$ obtained by linear prediction is a stable[3] filter (Scharf, 1991),

◇ there are no restrictions in the selection of the analysis window $g(n)$.

3.2.2.2 Whitening of a signal carrying sinusoids

Now, let us suppose that the audio signal $x(n)$ is formed of an AR process plus a sum of sinusoids. The periodogram of $\widehat{R}_x(e^{j2\pi f})$ is altered by the presence of peaks, centered around the frequencies of sinusoids, superimposed over the PSD of the AR process. It is possible to eliminate those peaks by performing a smoothing operation of the periodogram. A straightforward way to implement the periodogram smoothing is to use an order filter[4] (Bovik et al., 1983; Tagare & Figueiredo, 1985). In the frequency domain, the shape of the peaks corresponds to the Fourier transform of the analysis window $\frac{1}{N}(\tilde{g} \star g)(n)$. Thus, it is important to select $g(n)$ having in mind the trade-off between the main lobe width and the side lobe height.

In practice, we apply this whitening technique to long audio sequences. Since calculating the periodogram for a signal of several tens of seconds might be computationally expensive, to estimate the PSD we use the method proposed by Welch (1967) that

[2] In fact, $\hat{r}_x(n)$ can be seen as the standard biased autocorrelation function with expected value $\mathbb{E}\{r_x(n)\} = \frac{1}{N}[\tilde{g}(n) \star g(n)] \cdot r_x(n)$, where $g(n)$ is a rectangular window and $\frac{1}{N}[\tilde{g}(n) \star g(n)]$ is a Bartlett (or triangular) window.

[3] The estimator is also causal, although this property is not really necessary for our purposes since the estimation of the periodogram requires full access to the whole signal beforehand. The stability property is guaranteed by the biased estimator of $\hat{r}_x(n)$ (Scharf, 1991).

[4] Also called "order statistic filter". Bovik et al. (1983) present a description of order filters as a generalization of the median filter concept.

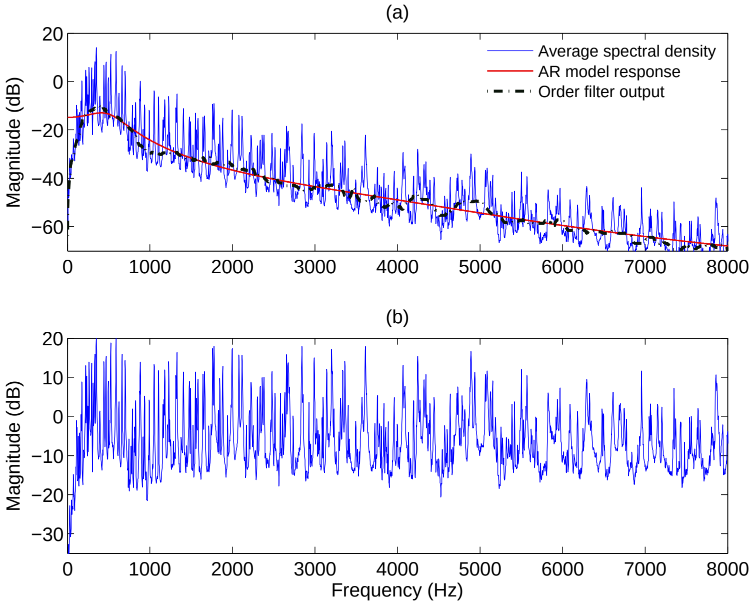


Figure 3.3: Non-causal pre-processing example. (a) Average spectral density in blue trace, order filter output (or smoothed periodogram) in black trace and AR model response in red trace. (b) Whitened spectral density.

consists in segmenting the signal in a number of (overlapping) small excerpts, computing their respective periodograms and averaging them. This technique not only reduces the number of operations and storage requirements but also computes an average PSD estimation with smaller variance (Welch, 1967; Stoica & Moses, 1997). We carry out the analysis over windows of size $N = 16000$ samples (1 second for Fs=16 kHz) with an overlapping of 4000 samples, $g(n)$ is a Hann window. The periodograms are calculated using $N' = 2^{14} = 16\,384$ points. It is then smoothed using an order filter of length $q = \frac{N'}{128} = 128$ points covering approximately 125 Hz, which is significantly larger that the average sinusoidal peak-width. The point-wise values of the smoothed periodogram are obtained by arranging the q samples inside the order filter in increasing order and then taking the sample located at $\frac{q}{3}$ (≈ 43 samples) of the order filter length. This value was found using a heuristic approach. The autocorrelation estimator $\hat{r}_x(n)$ is then obtained by computing the inverse Fourier transform of the smoothed periodogram. Finally, the coefficients of the whitening filter $H(z)$ are obtained by linear prediction using a filter order $p = 6$.

Figure 3.3 presents an example for of this approach using the same audio signal of Figure 3.2. The top part shows (blue trace) the input signal average spectral density, superimposed the smoothed periodogram (order filter output in black trace) and the AR model response (with red trace). The bottom part shows the detrended average spectral density. We can note a significant increase of about 55 dB obtained in the upper part of the spectrum and a much flatter frequency profile.

3.3 Harmonic-plus-Noise decomposition

One of the novelties of our research relies on the idea of decomposing the audio signal into two parts, one deterministic and the other stochastic; then we analyze each of them separately and the results are finally fused. More specifically about the decomposition, we model the audio signal $x(n)$ as the linear sum of two elements. The first part, to which we referred above as deterministic, is constituted solely of sinusoidal components it will be hereinafter called *harmonic* and denoted as $s(n)$[5]. The second part, to which we referred above as stochastic, is formed of all the elements in the audio signal that cannot be modeled as sinusoidal and it will be subsequently called *noise* and denoted by $w(n)$, therefore, $w(n) = x(n) - s(n)$.

To our knowledge, in the context of beat analysis there exists no previous approach processing separately the harmonic and noise components[6]. For this reason, one of the goals of our research is to explore the potential of this decomposition in the scope of metrical analysis of music signals.

Our main motivation to decompose the music signal is the idea of emphasizing phenomenal accents by separating them from the surrounding disturbing events, we explain this idea using an example. When processing a piano signal (percussive or plucked string sounds in general) the sinusoidal components could hamper the detection of the non-stationary mechanical noise of the attack, in this case the sound of the hammer hitting the cords. Conversely, when processing a violin signal (bowed strings or wind instrument sounds in general) the non-stationary friction noise of the bow rubbing the cords hampers the detection of the sinusoidal components.

In this section we explain two different harmonic plus noise (H+N) decomposition procedures used throughout the present work. In the context of the rhythm analysis framework mentioned in §3.1, both of these techniques refer to the second block of the scheme presented in Figure 3.1. In addition, both procedures share the characteristic of being causal. The first approach is based on the Exponentially Damped Sinusoidal (EDS) model. The second one has not been previously published in detail, but the theoretical principle used to perform the decomposition is more classical and is based on the Fourier transform (FT). Henceforth, to differentiate these H+N decomposition techniques they will be referred to as the "EDS" and "FT" respectively.

3.3.1 Exponentially Damped Sinusoidal model

The H+N model described in this part is based on a subspace analysis technique (sometimes referred to as high resolution methods) and it is founded on the Exponentially Damped Sinusoidal (EDS) model (Badeau et al., 2002). If the reader is interested about the potential of this technique, a comprehensive mathematical formulation on subspace analysis models and their application to digital music processing is provided in (Badeau, 2005). The use of this method has given place to two articles in the ambit of rhythm analysis (Alonso et al., 2003a, 2005b).

For the sake of clarity, the EDS model is described in three stages. First, the subspace analysis problem is posed and solved for a small data excerpt (one frame). Then, it is

[5] No hypothesis is made on the relationship between the different frequencies present in the audio signal.

[6] Paulus & Klapuri (2002) have developed a system to measure the similarity of rhythmic patterns which uses a harmonic-plus-noise decomposition to extract audio features. Contrary to our approach, theirs does not use this information to carry out metrical analysis.

generalized to the whole signal by introducing the concept of *subspace tracking*. Finally, it is applied to the audio signal context.

3.3.1.1 Subspace filtering

The first of the separation techniques used in this work belongs to the so-called *subspace filtering* methods (De Moor, 1993; Ephraim & Van Trees, 1995). This kind of approach has the notable advantage that the estimation of the signal parameters is not required to perform the H+N decomposition. Although subspace filtering methods do not succeed to completely separate the noise and the theoretical harmonic part, they have been successfully applied in speech processing as a denoising mechanism (Hermus & Wambacq, 2004; Wang et al., 2004).

Let $x(n),\ n \in \mathbb{Z}$, be the real signal[7] under analysis. By definition, we suppose that it can be written as the sum:

$$x(n) = s(n) + w(n), \tag{3.2}$$

where

$$s(n) = \sum_{i=1}^{2M} \alpha_i z_i^n \tag{3.3}$$

is referred to as the deterministic part of x, and where $w(n)$ is a real valued wide-sense stationary white gaussian noise[8] (WGN) with zero-mean and variance σ_w^2, and is denoted as the noise part of x.

In Eq. (3.3), the α_i are the complex amplitudes bearing magnitude and phase information and the z_i are the complex poles $z_i = e^{d_i + j2\pi f_i}$ where $f_i \in [-\frac{1}{2}, \frac{1}{2}[$ are the frequencies with $f_i \neq f_k$ for all $i \neq k$ and $d_i \in \mathbb{R}$ are the damping factors. It should be noticed that since s is a real sequence, the z_i's and α_i's can be grouped in M pairs of conjugate values.

Let us define L-dimensional data vector

$$\mathbf{s}(n) = [s(n-L+1), \ldots, s(n)]^T$$

where usually $2M \ll L$. Subspace analysis techniques rely on the property that $s(n)$ belongs to the $2M$-dimensional subspace spanned by the basis $\mathbf{V}$, given by the Vandermonde matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{L-1} & z_2^{L-1} & \cdots & z_{2M}^{L-1} \end{bmatrix}.$$

This subspace is the so-called *signal subspace*. As a consequence, $\mathbf{V} \perp \mathrm{span}(\mathbf{W}_\perp)$ where $\mathbf{W}$ denotes a $L \times 2M$ matrix spanning the signal subspace and $\mathbf{W}_\perp$ an $N \times (N - 2M)$ matrix spanning its orthogonal complement, referred to as the *noise subspace* . The H+N decomposition is performed by projecting the signal x respectively onto the signal subspace and the noise subspace.

[7] We take for granted that $x(n)$ has already been preprocessed as explained in §3.2 and it is a zero-mean signal.

[8] As a reminder: a zero-mean white gaussian noise $w(n)$ with variance σ_w^2 is a sequence of independent and identically distributed random variables with probability density function $p(w) = \frac{1}{\pi\sigma^2} e^{-\frac{w^2}{\sigma^2}}$ having a constant PSD over the passband.

Let the symmetric $L \times L$ real Hankel matrix $\mathbf{H}_s$ be the data matrix:

$$\mathbf{H}_s = \begin{bmatrix} s(0) & s(1) & \cdots & s(L-1) \\ s(1) & s(2) & \cdots & s(L) \\ \vdots & \vdots & \ddots & \vdots \\ s(L-1) & s(L) & \cdots & s(N-1) \end{bmatrix}, \tag{3.4}$$

where $N = 2L - 1$, with $2M \leq L$. Since each column of $\mathbf{H}_s$ belongs to the same $2M$-dimensional subspace, the matrix is of rank $2M$ and thus is rank deficient. Its eigenvalue decomposition (EVD) yields

$$\mathbf{H}_s = \mathbf{U}\mathbf{\Lambda}_s\mathbf{U}^H \tag{3.5}$$

where $\mathbf{U}$ is an orthonormal matrix, $\mathbf{\Lambda}_s$ is the $L \times L$ diagonal matrix of the eigenvalues, of which $L - 2M$ are zero-valued. $\mathbf{U}^H$ denotes the Hermitian transpose of $\mathbf{U}$. The $2M$-dimensional space spanned by the columns of $\mathbf{U}$ corresponding to the non-zero entries of $\mathbf{\Lambda}_s$ is the signal subspace.

Because of the surrounding additive white noise $\mathbf{H}_x$ is full rank and the signal subspace, $\mathbf{U}_S$, is formed by the $2M$-dominant eigenvectors of $\mathbf{H}_x$, *i.e.*, the column of $\mathbf{U}$ associated to the $2M$ eigenvalues having the highest magnitudes.

Using as observation the noisy sequence $x(n)$, the underlying harmonic part can be obtained by projecting $x(n)$ onto its signal subspace as follows:

$$\mathbf{s} = \mathbf{U}_S\mathbf{U}_S^H\mathbf{x} \tag{3.6}$$

where $\mathbf{x}(n) = [x(n - L + 1), \ldots, x(n)]^T$ is the input data vector. As mentioned above, a remarkable property of subspace filtering methods is that for calculating the noise part of the signal, the estimation and subtraction of the sinusoids is not required explicitely. Thus, the noise part is obtained by projecting $x(n)$ onto the noise subspace as follows:

$$\mathbf{w} = \mathbf{x} - \mathbf{s} = (\mathbf{I} - \mathbf{U}_S\mathbf{U}_S^H)\mathbf{x}. \tag{3.7}$$

In reality, the H+N model of Eq. (3.2) does not fully hold for a number of practical order difficulties (*e.g.*, $s(n)$ is not formed of pure sinusoids, neither $w(n)$ is a WGN). For this reason the harmonic and noise parts obtained in Eqs. (3.6) and (3.7) respectively, are only approximations. In fact $\mathbf{s}(n)$ can be seen as a least squares estimation, defined as the best $2M$-rank approximation to the data vector $\mathbf{x}(n)$. There exists the possibility to use other subspace filtering methods such as *singular values adaptation* and *minimal variance*, but in practice the procedure described above is the most straightforward to implement yielding a good compromise between signal distorsion and noise level (Badeau, 2005, page 158).

3.3.1.2 Subspace tracking

Since the harmonic plus noise decomposition of $x(n)$ involves the calculation of one EVD of the data matrix $\mathbf{H}_x$ at every time step, decomposing the whole signal would require a too highly demanding computational burden. In order to reduce this cost, there exist adaptive methods that avoid the computation of the EVD, a survey of such methods can be found in (Badeau, 2005). For the present work, we use an iterative algorithm called *sequential iteration* (Badeau et al., 2002), shown in Table 3.1. Assuming that it converges faster than the variations of the signal subspace, the algorithm operation involves two

Initialization: $\mathbf{U}_S = \begin{bmatrix} \mathbf{I}_{2M} \\ \mathbf{0}_{(N-2M)\times 2M} \end{bmatrix}$

For each time step m iterate:
1- $\mathbf{A}(n) = \mathbf{H}_x(n)\mathbf{U}_S(n-1)$ fast matrix product
2- $\mathbf{A}(n) = \mathbf{U}_S(n)\mathbf{R}(n)$ skinny QR factorization

Table 3.1: Sequential Iteration EVD algorithm.

auxiliary matrices at every time step $\mathbf{A}(n)$ and $\mathbf{R}(n)$, in addition of a skinny QR factorization. The harmonic and noise parts of the whole signal $x(n)$ can be computed by means of an overlap-add method:

1. the analysis window is recursively time-shifted. In practice, we choose an overlap of $3L/4$,
2. the signal subspace $\mathbf{U}_S$ is tracked by means of the previously mentioned sequential iteration algorithm presented in Table 3.1.
3. the harmonic, s, and noise, $\mathbf{w}$, vectors are computed according to Eqs. (3.6) and (3.7),
4. finally, consecutive harmonic and noise vectors are multiplied by a Hann window and respectively added to the harmonic and noise parts of the signal.

The overall computational complexity of this decomposition method for each analysis block is that of step 2, which is in fact the most computationally demanding task of the whole metrical analysis system. Its complexity is $\mathcal{O}(L\,M(M + \log(L))$.

3.3.1.3 Implementation

As mentioned before, the theory of the subspace analysis model that we use relies on the principle that $w(n)$ is a WGN. In §3.2 we described two methods for whitening the smoothed average periodogram of $x(n)$, but it is still not sufficient, especially for the causal approach. In practice, additional signal conditioning must be conducted to satisfy the WGN property as much as possible. One simple way to flatten even more the smoothed PSD is by splitting the audio signal in frequency bands, in this way the WGN assumption becomes true inside each subband and the EDS model can be applied individually to all bands. The frequency decomposition also assures that the average power discrepancies in the subband boundaries are small. Moreover, when using a bandwise processing approach there are less sinusoids per subband (compared to the full band signal) which allows to significantly reduce the overall computational complexity, this problem is revisited below.

In the metrical analysis community there exists an implicit consensus about decomposing the music signal in frequency bands. Some experiments carried out by Scheirer (1998, page 591) show the importance of this operation, although he claims that there exists no optimal frequency decomposition since many subband layouts lead to comparable satisfactory results. On the other side, during his research Gouyon (2005, page 147) argues "the superiority of the ERB[9] frequency subband decomposition over others (pro-

[9] ERB (Equivalent Rectangular Bandwidth) is a critical band scale proposed by Moore (1995).

posed in the literature) as the basis for the computation of effective energy feature sets". For this reason, we have decided to investigate ourselves the impact of the frequency decomposition by comparing two different ways of breaking-down the signal's spectral content. The first one uses a uniform frequency decomposition and the second one is based on a logarithmic decomposition.

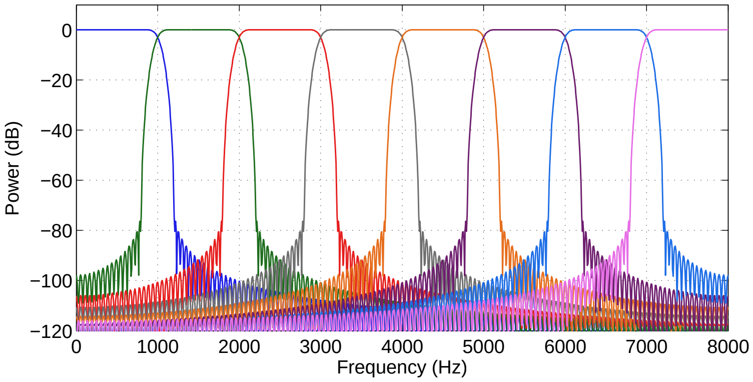


Figure 3.4: Uniform filter bank.

Uniform filter bank For the uniform frequency decomposition we use a maximally decimated cosine modulated filter bank (Vaidyanathan, 1992, page 353) with $P = 8$ bands and where the prototype filter is implemented as a 180$^{\text{th}}$ order FIR filter with at least 80 dB of rejection in the stop band. Using such a highly selective filter is important because subspace filtering methods are sensitive to spurious sinusoids outside the passband. Figure 3.4 shows the uniform filter bank layout and Figure 3.5 shows an example of the respective output corresponding to 1 s of a piano chord. We can see that the noise floor is fairly stable with small oscillations around -30 dB. As mentioned above, an advantage of using a uniform frequency decomposition is that further processing in the subbands is the same for all channels.

Until now, we have introduced the EDS model assuming that we know the number of sinusoidal components present in $x(n)$, as shown by Eqs. (3.2) and 3.3. Unfortunately in practice this assumption is far from true and in fact estimating the exact value of M is indeed a difficult task which has attracted much attention in various research fields. Badeau et al. (2006) review various solutions proposed in the literature and presents a new and promising method. However, these procedures remain highly expensive in computational terms.

To circumvent this problem, we adopted a more pragmatic solution for estimating M. Let us call r the true number of sinusoidal components in the harmonic part of $x(n)$. Badeau (2005, page 55, theorem 4.2.2) formally shows that underestimating the model order, that is choosing $M < r$, perturbs the position of the z_i and in general does not guarantee that the true pole values are found. On the contrary, if the model order is overestimated $M > r$, it can be shown (Badeau, 2005, page 54, proposition 4.2.1) that the r correct pole values are contained among the M poles found. Even if the overestimation has the unwanted side effect of producing a higher computational cost, it remains perfectly tractable in the practical case. However, this choice leads to a dilemma: *what is the smallest M that we can choose in order to overestimate the number of sinusoids and at the same*

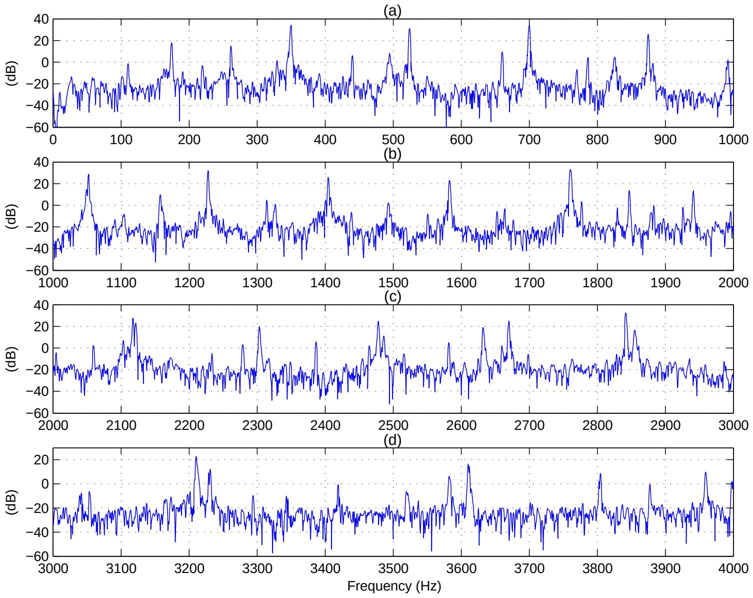


Figure 3.5: Uniform filter bank output corresponding to the first four bands of a piano chord. From top to bottom, as frequency increases, we can see that the noise floor is fairly stable with small oscillations around -30 dB.

time keep the computational burden as low as possible? There is not a unique answer, since it totally depends on the nature of the audio input.

Our goal is to remain as general as possible concerning the audio source. We opted for a heuristic solution to the problem of evaluating the number of sinusoids: we estimated the average number of components present in another data corpus. This operation was carried out on the ”RWC music database: music genre” gathered by Goto et al. (2003). We consider that this set is well balanced, it contains 100 instances from ten different musical genres. The operation was conducted as follows: every signal in the set is pre-processed, then its respective PSD was computed every 30 ms, thresholded using an order filter[10], next all maxima were located and those having a bandwidth larger than 33 Hz (and not having other larger peaks in the vicinity) were selected[11] as sinusoids. Figure 3.6 shows an example, in blue trace the PSD corresponding to one frame of a violin signal, in red trace the threshold and in black circles the maxima selected as sinusoids. After repeating this procedure for whole database and counting the number of sinusoids, we obtained the results presented in Table 3.2. This table presents the number of components rounded to the closest integer, the second row indicates the average maximum number of sinusoids and the third row the average mean number of sinusoids.

The value of M in every subband was fixed to that shown in the second row of table 3.2. We opted for the average maximum since this value provides a higher degree of confidence that the noise part will not contain sinusoidal components. One of the advantages of using a uniform frequency decomposition is that further processing in the subbands is

[10] This order filter is very similar to that described before, but it was tuned to keep the spectral peaks and to trim off the rest.

[11] To compute the PSD we use a Hann window whose side-lobes have a bandwidth of 33 Hz, thus we consider that a true sinusoid should have a larger bandwidth.

Range (Hz)	0 to 1000	1000 to 2000	2000 to 3000	3000 to 4000	4000 to 5000	5000 to 6000	6000 to 7000	7000 to 8000
max. sinusoids	**14**	**13**	**11**	**11**	**8**	**9**	**9**	**8**
mean sinusoids	6	6	6	6	6	6	5	5

Table 3.2: Average number of sinusoids per band. These values have been rounded to the closest integer.

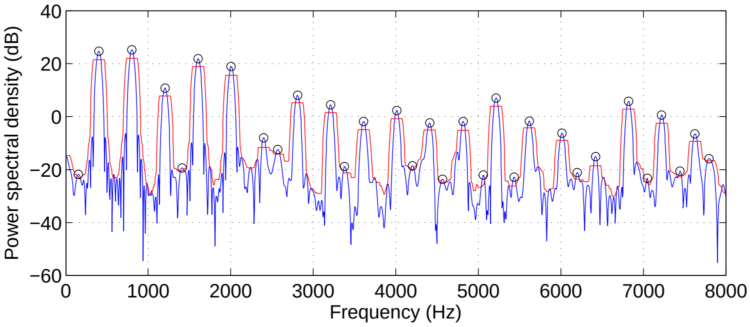


Figure 3.6: In blue trace the PSD of a violin signal, in red trace the dynamic threshold and in black circles the maxima selected as sinusoids.

the same for all channels.

The output of the H+N decomposition stage consists of two signals per subband: $s_p(n)$ carrying the harmonic and $w_p(n)$ the noise part of $x_p(n)$ respectively, where $p \in [1 \quad P]$ indicates the band number. Figure 3.7 shows an illustrative example: (a) presents the spectrogram for $x_1(n)$ corresponding to a piano signal, only the first subband (0 Hz to 1 kHz) is considered, where it is possible to see the appearance of 12 onsets. (b) shows the harmonic part $s_1(n)$, there is no much visual difference between this image and the previous one. On the contrary, (c) depicts the noise part $w_1(n)$ where all sinusoids have disappeared, but the twelve onset attacks can be seen as vertical stripes located at the same place where notes begin in the two earlier images.

Logarithmic filter bank This filter is built in two steps. First, a $P = 16$-band uniform filter bank is designed using the same cosine modulated technique (Vaidyanathan, 1992, page 353). In this case the prototype filter is a 270th order FIR filter with at least 80 dB of rejection in the stopband[12]. Then, the filter outputs are merged as explained in Table 3.3 to produce a filter bank with $P' = 5$ subbands, as illustrated in Figure 3.8-(a).

Figure 3.8-(b–e) shows an example of the filter bank output corresponding to the same piano signal used in the example of Figure 3.5. As in the uniform filter bank case, we can see that the noise floor is fairly stable with small oscillations around -30 dB.

A logarithmic filter bank has the disadvantage that the analysis window length of the EDS model depends on the subband under analysis. The values of L, derived from the

[12] Compared to the 8-band uniform filter bank, since the passband of this filter is narrowed, the filter order must increase to keep constant the rejection ratio.

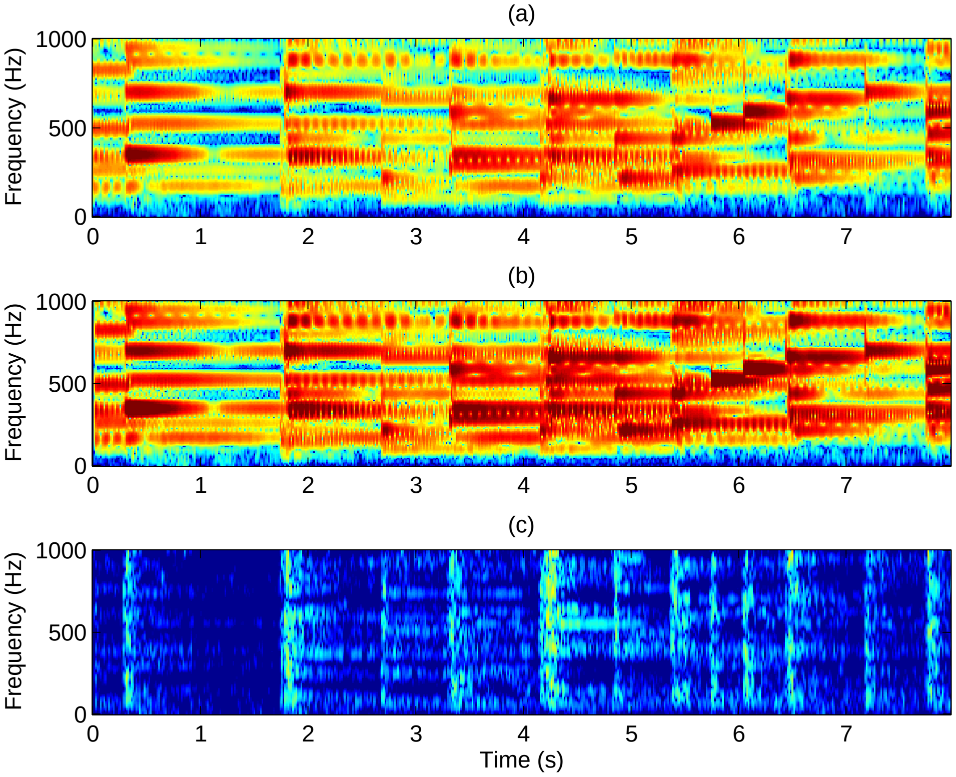


Figure 3.7: EDS model output, the input corresponds to a piano signal. Only the first frequency band (0 Hz to 1 kHz) is presented. (a) spectrogram of the original signal, (b) spectrogram of the harmonic part and (c) spectrogram of the noise part.

Subband number	1	2	3	4	5
Merged bands	1	2	3–4	5–8	9–16
Range (Hz)	0–500	500–1000	1000–2000	2000–4000	4000–8000
decimation factor	16	16	8	4	2

Table 3.3: Logarithmic filter bank structure.

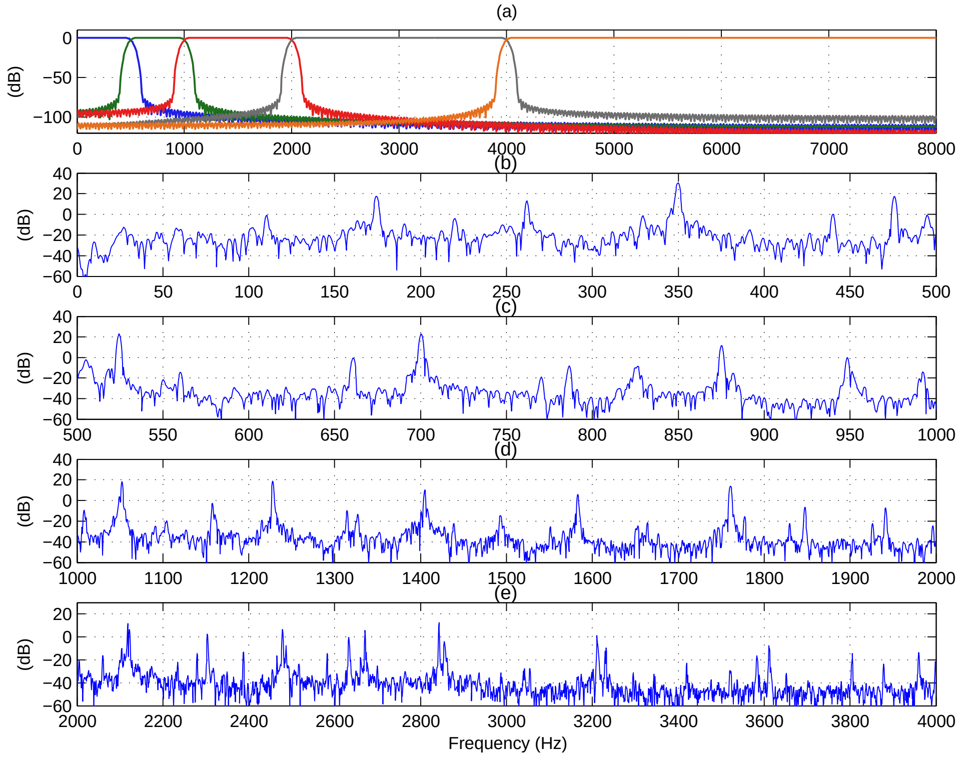


Figure 3.8: Logarithmic filter bank.

Subband number	1	2	3	4	5
Range (Hz)	0–500	500–1000	1000–2000	2000–4000	4000–8000
L (samples)	32	32	64	128	256
M (sinusoids)	7	7	13	22	34

Table 3.4: EDS parameters for the logarithmic filter bank.

uniform filter bank case, are presented in Table 3.4. In addition, this table also indicates the number of sinusoids (M) extracted in each subband, the values were obtained from those of Table 3.2.

In a similar way to the uniform filter bank, the output of the decomposition stage consists of two signals per subband: $s_p(n)$ carrying the harmonic and $w_p(n)$ the noise part of $x_p(n)$ respectively, where $p \in [1 \;\; P']$ indicates the band number. Figure 3.9 shows an example with the same piano signal that we used before, but a different subband: (a) presents the spectrogram for $x_4(n)$ (the same piano signal), only the fourth subband (2 kHz to 4 kHz) is considered. Once again, it is possible to see the appearance of 12 onsets. (b) shows the harmonic part $s_4(n)$, with a high visual resemblance to the original signal image shown above. (c) depicts the noise part $w_4(n)$, we can appreciate how the note attacks have been emphasized by isolating them from the sinusoidal part.

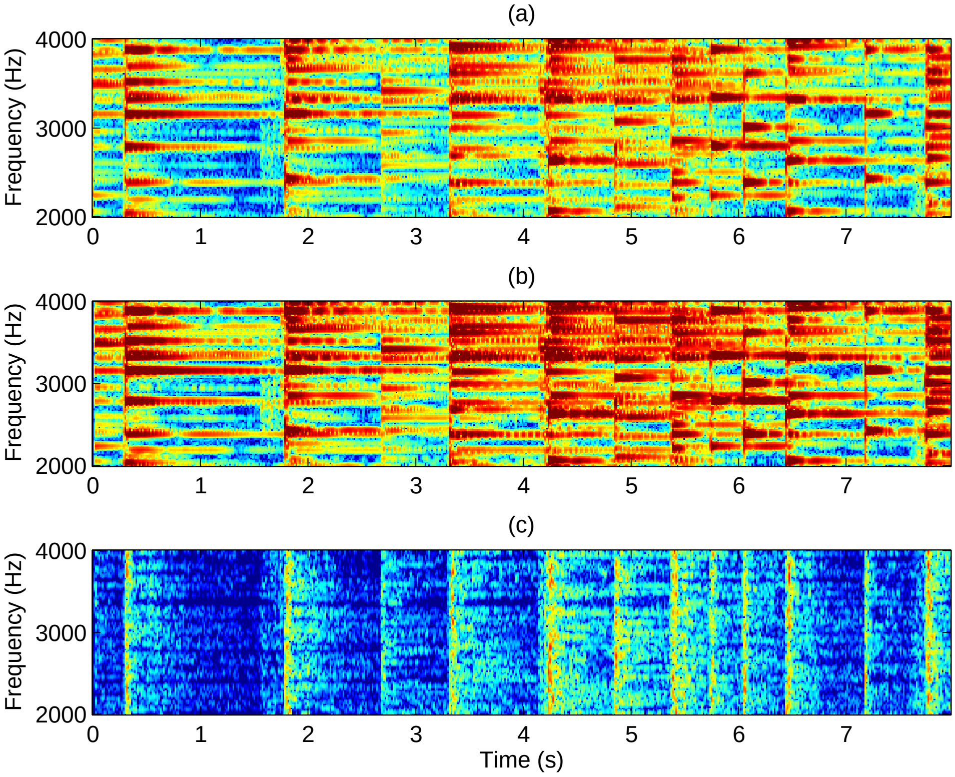


Figure 3.9: EDS model logarithmic filter bank output, the input corresponds to the same piano signal used in the uniform bank case. Only the fourth frequency band (2 kHz to 4 kHz) is presented. (a) spectrogram of the original signal, (b) spectrogram of the harmonic part and (c) spectrogram of the noise part. In this particular case, the decomposition presented in this figure using the logarithmic bank seems to work better than that using the uniform filter bank (in Figure 3.7). However, no concluding remarks can be made based solely on this example. The filter bank comparison will be addressed in the forthcoming chapters.

3.3.2 Decomposition based on the Fourier transform

The second H+N model used in this work is based on the phase vocoder principle which allows modifications to the amplitudes or phases of specific sinusoidal components of the audio signal in the frequency domain. Then, this modified representation is resynthesized into the time domain. This technique is also known as *FFT filtering*, since at the heart of the phase vocoder lies the Short Time Fourier Transform (STFT).

The theoretical principle of this decomposition model bears some resemblance to those developed by McAuley & Quatieri (1986) or Serra (1989); Serra & Smith (1990), except that we do not carry out the trajectory matching in the frequency domain. In addition, this technique has a lower computational burden compared to the subspace analysis approach.

This method analyzes the input signal in a frame-wise fashion as follows. For every analysis window (frame), the salient peaks in the magnitude spectrum are considered as sinusoids and selected, and the rest of the spectrum is discarded. A frame of the

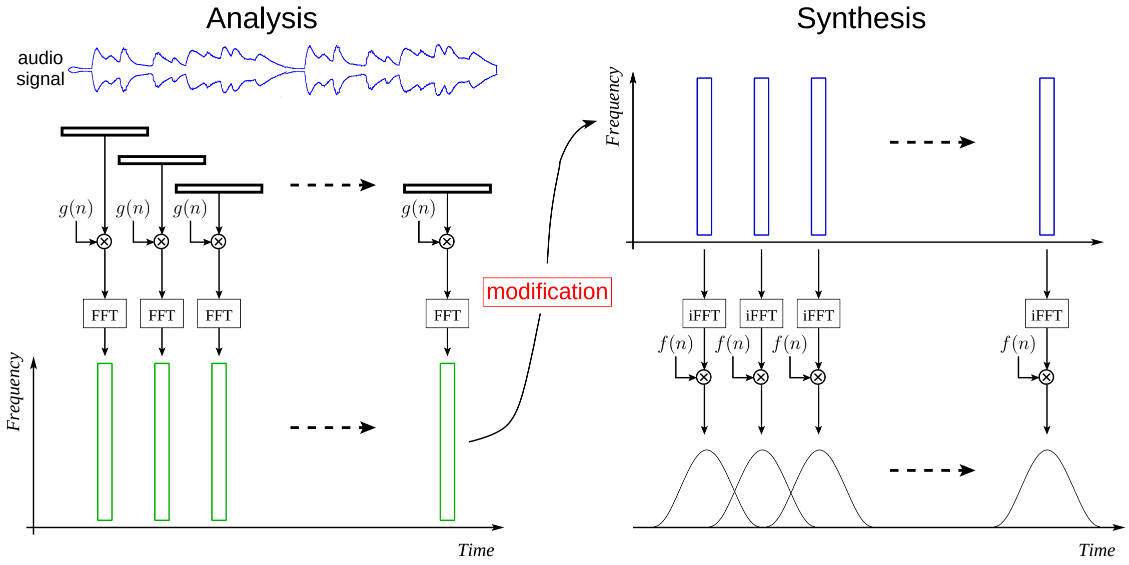


Figure 3.10: Scheme of analysis/synthesis based on the STFT.

harmonic part, given by the modified spectrum, is synthesized and subtracted from the input-signal window. This subtraction produces a frame of the noise part. The analysis window is shifted in time and the whole process is iterated. The details of this mechanism are described below.

3.3.2.1 Short-time Fourier transform analysis/synthesis

The short-time Fourier transform (STFT) consists in computing the DFT over a set of regularly-spaced windowed signal segments (called signal frames) which are obtained by weighting the input signal with a window of length N and then shifting the window M samples. For each frame the FFT is computed, providing frequency representation of the signal between two consecutive time instants. To synthesize a signal from its STFT, the inverse FFT (iFFT) of every frame is computed. Then the reconstructed signal is obtained using a technique called *OverLap-Add* (OLA) where every frame is weighted by a synthesis window, which is then added to the overlapping portion of the previous-frame. Perhaps the most important characteristic of the signal description provided by the STFT is its capability to allow signal modifications between the forward and the inverse Fourier transform. We exploit this attribute to separate the signal into harmonic and noise parts. A descriptive image of the STFT analysis/synthesis principle can be seen in Figure 3.10.

Once again, we suppose that the audio signal has been pre-processed beforehand as described in §3.2. We start by computing the analysis part of the STFT of $x(n)$ defined as follows[13]

$$\widetilde{X}(m,k) = \sum_{n=-0}^{N-1} g(n)x(Mm+n)e^{-\jmath\frac{2\pi}{N}kn} \tag{3.8}$$

where $m \in \mathbb{Z}$ is the time (frame) index, $g(n)$ is a real window of finite length N which determines the portion of $x(n)$ that is under analysis at a particular time instant m, M is

[13] There exist two different conventions for describing the STFT, the so-called *band-pass* representation and *low-pass* representation (Portnoff, 1980). In this work we use the band-pass representation.

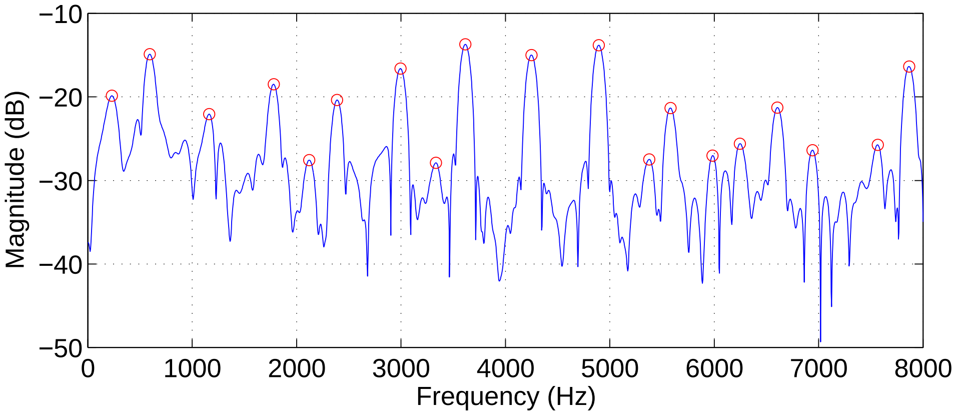


Figure 3.11: In blue trace the magnitude spectrum of piano signal frame. In red circles the peaks considered as sinusoids.

defined as the hop-size or time-shift for the window and $k = 0, \ldots, K-1$ is the frequency (bin) index.

As we can see, the STFT $\widetilde{X}(m,k)$ is a function of two variables, but for the moment we prefer to consider m as constant, that is, to see $\widetilde{X}$ as the DFT of the finite sequence $g(n)x(Mm+n)$. Our goal is to detect the sinusoidal part of this windowed signal segment, and for this purpose we rely on the assumption that the maxima in the magnitude spectrum represent sinusoids in the input signal. Now let us call ν_ℓ, where $0 < \ell \leq L \ll K$, the set of frequencies (bins) corresponding to those maxima. In Figure 3.11 we present an example of this hypothesis. This figure shows in blue trace the magnitude spectrum of a *pitched* frame of a piano signal, in addition the peaks located at the ν_i frequencies and considered as sinusoids are marked by red circles. In practice, a robust peak-detection algorithm was implemented to assure that only those maxima having a bandwidth larger than 33 Hz are selected.

Then, we define the frequency representation of the harmonic part as $\widetilde{S}$, where

$$\widetilde{S}(m,k) = \begin{cases} \widetilde{X}(m,k) & \text{if} \quad k \in \nu_\ell \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

That is, a new signal is formed where only the frequencies corresponding to the selected maxima (*i.e.*, sinusoids) are kept and the rest of the components are set to zero. From this modified signal we synthesize frame-by-frame the harmonic part $s(n)$ as

$$s(n) = \sum_{m=-\infty}^{\infty} f(n - Mm) \left(\frac{1}{K} \sum_{k=0}^{K-1} \widetilde{S}(m,k) e^{\jmath \frac{2\pi}{K} kn} \right). \tag{3.10}$$

From this expression we see that the reconstructed signal is calculated by adding overlapping frames obtained by inverse Fourier transform and weighted by a synthesis window $f(n)$. Then, the noise part $w(n)$ is obtained directly by subtracting the harmonic part from the input signal in a frame-by-frame basis

$$w(n) = x(n) - \sum_{m=-\infty}^{\infty} f(n + Mm) \left(\frac{1}{K} \sum_{k=0}^{K-1} \widetilde{S}(m,k) e^{\jmath \frac{2\pi}{K} kn} \right). \tag{3.11}$$

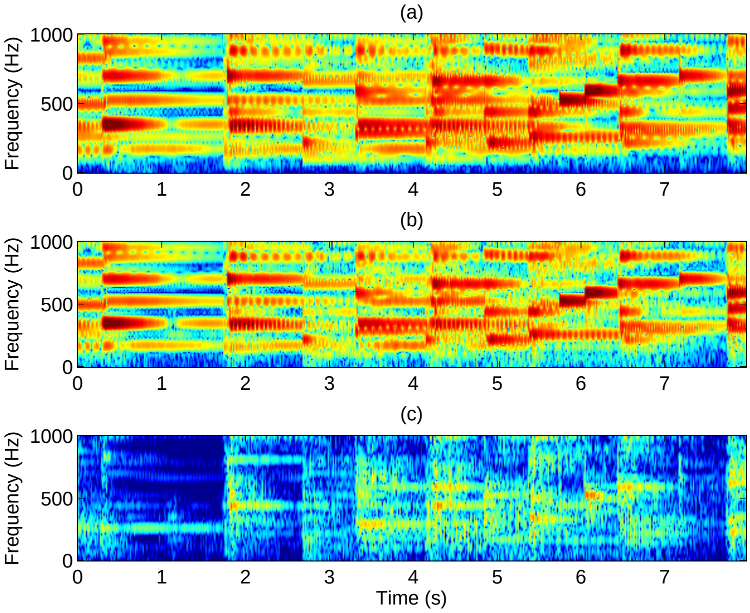


Figure 3.12: Uniform filter bank output of the Fourier based H+N model. First frequency band (0 Hz to 1 kHz) corresponding to a piano signal. Spectrograms of the (a) original signal, (b) harmonic part, and (c) noise part.

We have already mentioned the importance of decomposing the audio signal in frequency bands in the context of metrical analysis. We also saw how naturally this frequency decomposition fits with the requirements of the subspace analysis method to locally whiten the noise floor. In order to render the output of this method compatible with the model explained above, we filter the signals $s(n)$ and $w(n)$ using the same two filter banks described above. In this way both H+N models have a similar output: a set of sixteen signals $s_p(n)$ and $w_p(n)$, with $p \in [1 \;\; 8]$, when using the uniform filter bank; and a set of ten signals $s_p(n)$ and $w_p(n)$, with $p \in [1 \;\; 5]$ when using the logarithmic filter bank.

In Figures 3.12 and 3.13 we present two examples of the H+N decomposition based on the STFT. The first one is the counterpart of Figure 3.7 and corresponds to the first frequency band (0 Hz to 1 kHz) of the uniform filter bank output. The bottom graph shows the spectrogram of the noise part. Figure 3.13 presents the fourth subband (2000 Hz to 4000 Hz) of the logarithmic filter-bank. Although less explicit at first sight, compared to the EDS model graphs, the presence of the twelve note attacks is still noticeable in the noise part spectrograms of the Fourier based H+N model.

3.3.3 Comparing the H+N decomposition algorithms

In this part we compare the performance of the H+N decomposition algorithms presented in the previous sections. This comparison is based on two different criteria: the separation quality and the computational requirements.

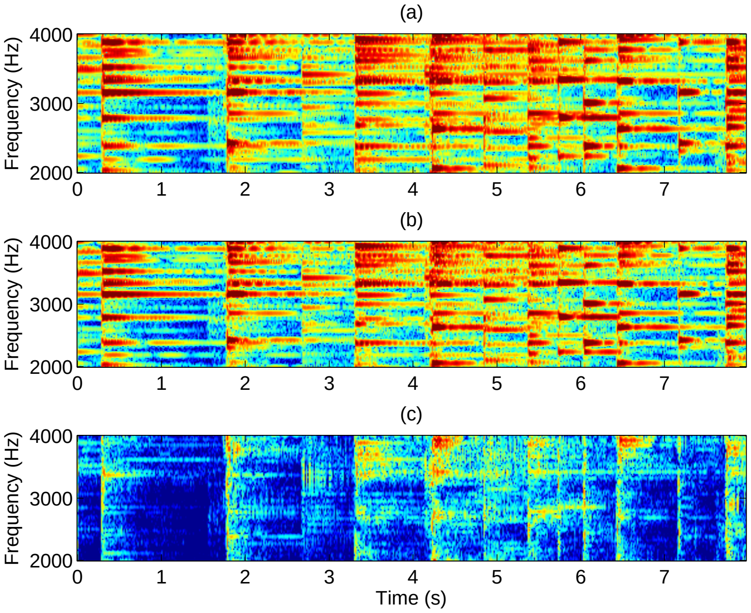


Figure 3.13: Log filter bank output of the Fourier based H+N model. Third frequency band (2 kHz to 4 kHz) corresponding to a piano signal. Spectrograms of the (a) original signal, (b) harmonic part, and (c) noise part.

Decomposition performance To evaluate the separation quality of the decomposition algorithms we have designed a scenario which accentuates the situations that the methods have to deal with. We have set up a synthetic signal $x_s(n)$ of unitary power composed of three different elements (modulated chirps) which have a time varying frequency response, additionally, this signal is immersed in complex white Gaussian noise (WGN) with a signal-to-noise ratio of 40 dB. Figure 3.14-(a) ilustrates the frequency behavior of the three components, one of them has a piece-wise constant frequency, another one is a piece-wise linear modulated chirp, and the last one is a cosine-modulated chirp. Towards the end of $x_s(n)$ all the components have a constant frequency.

Figure 3.14-(b) shows the magnitude of the estimation error $e_1(n) = |x_s(n) - \hat{x}_1(n)|$, where $\hat{x}_1(n)$ is the harmonic part of $x_s(n)$ obtained by the EDS method. As argued before, for this test the model order was slightly overestimated and we set $M = 4$, *i.e.*, the number of components to extract is eight. The large error at the left end of the figure is related to an algorithm boundary effect. From the figure it can be seen that this method handles well smooth frequency variations, on the contrary it, has more difficulty dealing with abrupt transitions as shown by the error overshots. The average error magnitude (without considering the boundaries) is -24.4 dB, and the average error magnitude in the last part (where the components have constant frequencies) is -29.1 dB.

In a similar way, Figure 3.14-(c) shows the magnitude of the estimation error $e_2(n) = |x_s(n) - \hat{x}_2(n)|$, where $\hat{x}_2(n)$ is the harmonic part of $x_s(n)$ obtained by the FT method. Although this method shows a better response to the abrupt frequency transitions, it also exhibits a higher average error magnitude of -19.9 dB (without considering the boundaries). The average error magnitude in the region where the chirp components have a constant frequencies reduces to -25.3 dB. It is interesting to see that this method exhibits

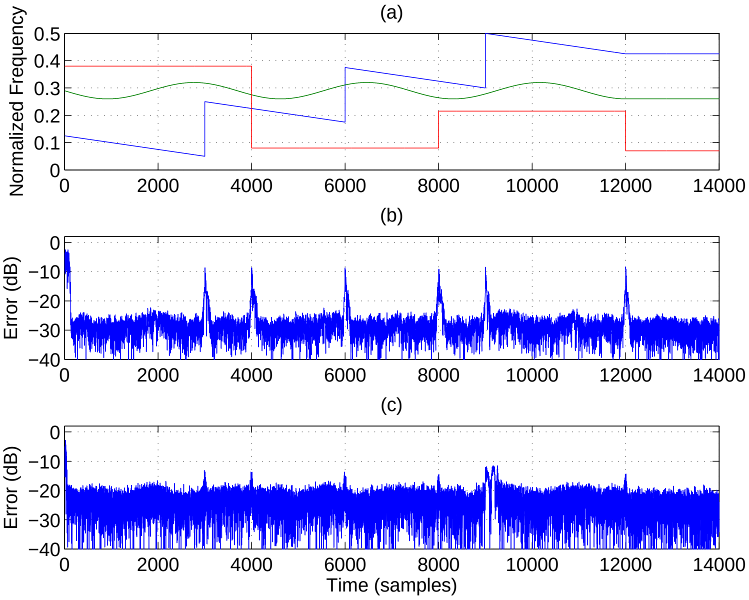


Figure 3.14: Assesing the separation quality of the H+N algorithms. (a) Frequency behavior of the synthetic signal $x_s(n)$ used to test the decomposition methods. (b) Error magnitude ($e_1(n)$) between the noiseless synthetic signal and the harmonic part obtained by the EDS method. (c) Error magnitude ($e_2(n)$) between the noiseless synthetic signal and the harmonic part obtained by the method based on the STFT.

an error peak around $n = 9000$, where one of the components of $x_s(n)$ takes frequency values very close to the Nyquist frequency.

However, the afore mentioned test does not guarantee that a proper separation was made, *i.e.*, that the noise part is free of sinusoidal components. To inspect if the noise parts contain sinusoids we have computed their spectrograms, Figures 3.15-(a)–(b) show the outcomes for the EDS and FT methods respectively. From the top Figure we can see that the noise part obtained by the EDS model is practically free of sinusoidal components. The abrupt frequency transitions can be seen as vertical stripes. Towards the end, some low-energy sinusoidal components can be distinguished at low frequencies. The noise part obtained with the FT model (3.15-(b)) shows traces of sinusoidal components, especially in the parts where the chirps vary and to a lesser extent towards the end of the analysis. In addition, the background noise is more energetic.

Based on the results obtained by both decomposition approaches, we can suggest that in our implementantions the EDS method outperforms, in terms of separation quality, the FT method.

Computational complexity In any real world application, upon the necessity of choosing among two or more alternatives, the computational burden of a given algorithm plays a fundamental role. In this part we present a brief analysis examining the computation time of the H+N methods described above. Since both algorithms were implementend under Matlab using a number of built-in functions[14], a meticulous evaluation of the com-

[14] For instance `fft`, `ifft`, `qr`.

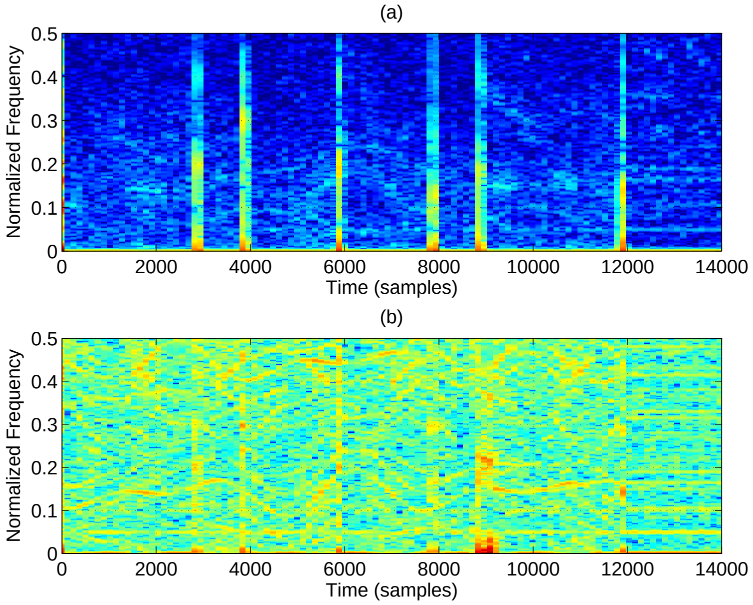


Figure 3.15: (a) Spectrogram of the noise part for the EDS method $e_1(n)$, and (b) spectrogram of the noise part for the FT method $e_2(n)$.

Method	Bandwise computation time (s)								Total
EDS uniform FB	2.32	2.31	2.26	2.27	1.99	1.84	1.79	1.80	16.58
EDS log FB		0.73	0.73		2.26	9.81		42.75	56.28
STFT & OLA	9.24								9.24

Table 3.5: H+N models and their computation time when processing a 10 s length audio signal.

putational burden appears to be rather complex. The approach we adopted to estimate the computational burden is not the most, but it is very straightforward and helps to provide a tangible opinion about the computational requirements of the algorithm. We measured just the time it takes to both algorithms to calculate the decomposition when processing a 10 s music signal sampled at 16 kHz on a Pentium IV computer running at 2.4 GHz with 1 GB of RAM. The time it takes to decompose the signal in subbands (uniform or logarithmic) was not taken into account. Table 3.5 presents the effective computation time for both algorithms and for both filter bank (FB) variants.

The EDS model performing a logarithmic subband decomposition is by far the most time consuming task among the three listed above. Especially when processing the last subband (4 kHz to 8 kHz), but this is not a surprise since the EDS model complexity is $O(L\,M(M + \log(L))$, where M is the model order.

3.4 Calculation of the musical stress profile

Perhaps the most critical part in any rhythm analysis system processing acoustic signals is that of extracting acoustic events from a music stream. In fact, this block is in charge of *converting* the *raw* audio signal into a *symbolic* representation indicating the exact location of the beginning of a musical event (a single note in simple cases, but more generally a whole pack of them).

In Chapter 1, we pointed out the importance of phenomenal accents as discrete sound events playing a fundamental role in metrical analysis (Lerdahl & Jackendoff, 1983). Humans hear them in a hierarchical structure, that is, a phenomenal accent is related to a motif, several motifs are clustered into a pattern and a musical piece is formed of several patterns that may be different or not. In the present work, we attempt to be acute (in a computational sense) to the physical events in an audio signal related to the moments of musical stress, such as magnitude changes, harmonic changes and pitch leaps. That is, acoustic effects that can be heard and are musically relevant for the listener. The attribute of being sensitive to these events does not necessarily imply the need of a specific algorithm for detecting harmonic or pitch changes, but solely a method which reacts to variations in these characteristics.

In the computer music community, phenomenal accents are better known as onsets. Therefore, calculating the profile of the musical stress present in a music signal as a function of time is intimately related to the task of detecting onsets. One of the goals of this stress profile is to simplify the decomposition of the input signal into musically relevant segments. This operation plays a significant role not only for rhythm analysis, but also for a large number of computer music applications. For instance, automatic transcription, score following, music retrieval, audio editing and special effects.

According to the framework illustrated in Figure 3.1, this section stands for the third block from the left. We start by providing a brief description about the nature of musical notes followed by a general overview of the onset detection methods available in the literature. Then we present our approach. Although based on earlier work, it considerably improves preceding methods based on the same principle.

3.4.1 The nature of a musical note

Before describing our proposal, it is important to understand the structure of musical notes. Natural sounds, including those coming from acoustic instruments, do not just instantly switch from *on* to *off* or vice versa. They are almost never static but change their "character" through time, they always have a fade-in and fade-out period. To take an example, a drum hit begins very sharply as the drumstick hits the skin and also fades away quite fast. The sound volume of a note on the piano will also rise rather quickly, but will dampen much more slowly. The sound of some (especially bowed string) instruments like the violin can last for a long time, while the sound of a drum inevitably fades away after each stroke, this behavior is the so-called *envelope* of the sound.

Figure 3.16 illustrates this simplistic model of the nature of a musical note. In practice, this is just a coarse description of the true envelope curve since the real ones from acoustic sounds are considerably more complex than this. However, from this abstraction we can still identify the main parts of an envelope[15]:

◇ *attack*: is characterized by an increase in the envelope's magnitude and marks the perceptual beginning of the sound,

[15] In the literature, this kind of representation is often called the "ADSR envelope model". The acronym is obtained from the name of the different parts: Attack, Decay, Sustain and Release. Although apparently reductive, this model has been widely applied in sound synthesis.

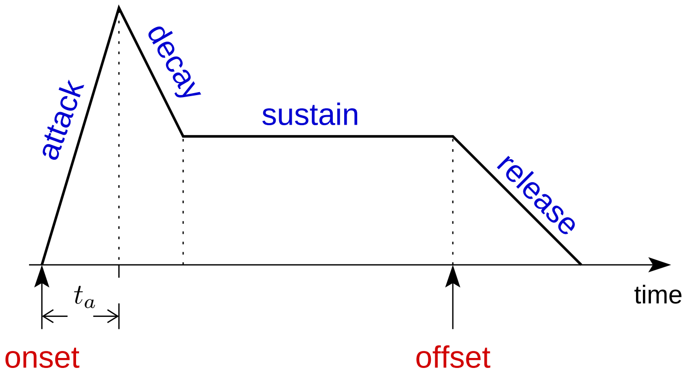


Figure 3.16: Structure of the envelope of a musical note.

- ◇ *decay*: is the first fading of the sound,
- ◇ *sustain*: this envelope part only exists in instruments where the excitation is continuous (*e.g.*, bowed string) and refers to the level at which the sound is held as long as there is a stimulus,
- ◇ *release*: indicates the fade-out of the sound.

Ideally we should always detect the leftmost point of the envelope indicating the start of the sound and marked as "onset" in Figure 3.16, but in the all-inclusive case it is very difficult to define its precise location. Moreover, in general the moment at which we perceive the onset often coincides with the "true onset", although this is not a rule but a parameter related to the sound source (Gordon, 1987). In practice, it is very difficult to detect the "true onsets" and in most cases detection algorithms only aim at locating the onset within the attack duration window t_a, shown in Figure 3.16. The attack duration parameter heavily depends upon the nature of the musical instrument and to a lesser extent to the stress applied to a given note. Gordon (1987) presents a quantitative study on the perceptual attack time of musical tones for a number of instruments.

A closely related concept, but more difficult to define, is that of audio *transient*. For our purposes, a transient can be informally described as locally distinctive short-duration behaviour in a sound. It contains a high degree of non-periodic frequency components and regularly covers the whole signal bandwidth or a at least a significant part of it.

3.4.2 Overview of current onset detection methods

In a realistic situation dealing with polyphonic signals, robust onset detection becomes a very challenging task due to the large variety of instruments that can be employed, whether they play simultaneously or not and the different kinds of attacks and dynamic ranges that they can produce. In recent years a considerable effort has been invested to solve this problem, and a large step into a systematic *showcase* and evaluation of new proposals has been taken by the MIREX[16] committee. Bello (2003); Bello et al. (2005) and Collins (2005a) provide an in-depth survey and evaluate a number of commonly used

[16] Further information on MIREX or concerning the onset detection contest is available at `http://www.music-ir.org/mirexwiki/index.php/Audio_Onset_Detection`.

methods. In general, onset detection approaches can be divided into two broad categories according to the working principle:

- ⋆ *deterministic techniques*, they use time–frequency or time–scale features of the audio signal,
- ⋆ *statistical techniques*, lie on the assumption that the onset appearance can be described by a probabilistic model.

Several onset detection systems have been proposed since the dawn of computer music research in the early 80's. The first endeavors to detect note onsets in music signals used to process the amplitude envelope of the waveform as a whole. This approach has proved to be very vulnerable since note onsets can be easily masked in the bulk signal by continuous tones of higher amplitude. For instance, effects such as heavy dynamic compression (widely used in modern popular music) tend to reduce the attack sharpness. In the following part we briefly describe a few, but representative onset detection algorithms according to their working principle.

Deterministic methods Most of these techniques use a time–frequency representation (TFR) to compute onsets. Amongst the most common approaches are filter banks (Klapuri, 1999), the STFT magnitude (Masri, 1996; Hainsworth & Macleod, 2003b; Collins, 2005a), phase (Bello & Sandler, 2003), or both (Bello et al., 2004). Compound systems using both filter-banks and the STFT magnitude have also been proposed (Duxbury et al., 2002; Hainsworth & Macleod, 2003a). Other onset detection methods improve their performance by using an enhanced time-frequency representation, *i.e.*, a reassigned spectrogram which considerably improves the more conventional STFT (Hainsworth & Wolfe, 2001; Röbel, 2003, 2005; Peeters, 2005). Practically all of these systems find the onsets by computing the difference between successive frames. The use of time–scale representations has also been explored (Daudet, 2001) displaying interesting properties for transient detection.

Statistical methods As mentioned above, statistical methods for onset detection are based on the assumption that the music signal can be described by some probability model, *i.e.*, they use the available observations to *guess* about the potential moments when abrupt changes have place in the audio signal. Obviously, the success of this approach entirely depends on the *goodness of fit* between the model's premise and the "real" behaviour of the observations. Perhaps the best known approach is based on the sequential probability ratio test who has been successfully used in speech segmentation applications (André-Obrecht, 1988; Di Francesco, 1990). This technique has also been proposed for audio transient detection (Jehan, 1997; Thornburg & Gouyon, 2000) but very few application examples have been presented. Comparable models, but using Independent Component Analysis (ICA) instead of a Bayesian framework have been proposed (Abdallah & Plumbley, 2003; Bello et al., 2005). Good results are obtained for percussive sounds. Unfortunately, in the context of onset detection and musical recordings segmentation, the lack of publications presenting tangible and successful results suggest (at least so far) that statistical methods have not found much acceptance. From our point of view the reason is rather practical, since in music the simple and tractable model assumptions merely do not hold and any attempts to improve them produces too complex formulations[17] that discourage their use.

More recent research (Davy & Godsill, 2002; Desobry et al., 2005) has opened new alternatives by using Machine Learning techniques and constructing a novelty function using Support Vector Machines (SVM). In this case the SVM measure the dissimilarity between two consecutive feature vectors corresponding to a discretized Cohen's class TFR (Hlawatsch & Auger, 2005, chapter 6). A rather similar approach based on SVM who uses as input features the frame-wise fundamental frequency, amplitude, and the relative strengths of the first three harmonics has also been tested (Kapanci & Pfeffer, 2004).

Onset detectors based on other machine learning methods have also been proposed. In fact, a technique based on artificial neural networks and using features derived from the STFT obtained the first place in the MIREX'05 annual contest in the "onset detection" category (Lacoste & Eck, 2006). Another technique based on a similar principle has been proposed but it has not been exhaustively evaluated yet, but tested on a few speech and music signals (Smith & Fraser, 2004). In spite of their appeal, ML methods present some drawbacks compared to more conventional STFT based methods, namely: a high computational cost, and the need of a large and well-annotated corpus for training.

3.4.3 Our approach to estimate the musical stress profile

Within the scope of our research we have developed an onset detection system to specifically fit our rhythm analysis requirements (Alonso et al., 2005c). Nevertheless, we believe it can be used for various other computer music applications too. This algorithm has proven to be fairly effective when processing a wide range of music signals (Alonso et al., 2005a). Our approach falls in the category of deterministic methods and its principles are founded on previous work carried out by Klapuri (1999), Duxbury et al. (2002) and Laroche (2003). Figure 3.17 displays the flow diagram of our proposal.

This method uses a band-wise processing rationale, as motivated by many approaches encountered in the literature. The general approach is as follows: first the harmonic $s(n)$ or noise $w(n)$ subband signal[18] is decomposed into frequency channels, for computational convenience we use the STFT merely as a filter bank. Then, this Time–Frequency Representation (TFR) is reassigned in order to produce a sharper description. Next, each frequency band is processed as depicted in the lower part of Figure 3.17 to find the time-location and intensity of its onset components. Finally, contributions from all frequencies are summed producing the system output, the so-called *detection function*. This is a signal that bears peaks with magnitude and location related to the onsets' perceptual intensity and position.

3.4.3.1 Reassignment: a method to sharpen time–frequency representations

The use of the reassignment method significantly improves the estimation of the time and frequency content of a given signal, which is fundamental for an effective computation

[17] For example, consider modeling the number of possible sources (musical instruments), the number of different ways to generate musical notes (onsets), the combinations that can be made. The number of potential combinations is just too large.

[18] for convenience we drop the subband index p since the processing principle is identical for all bands regardless of the filter bank and H+N decomposition under consideration

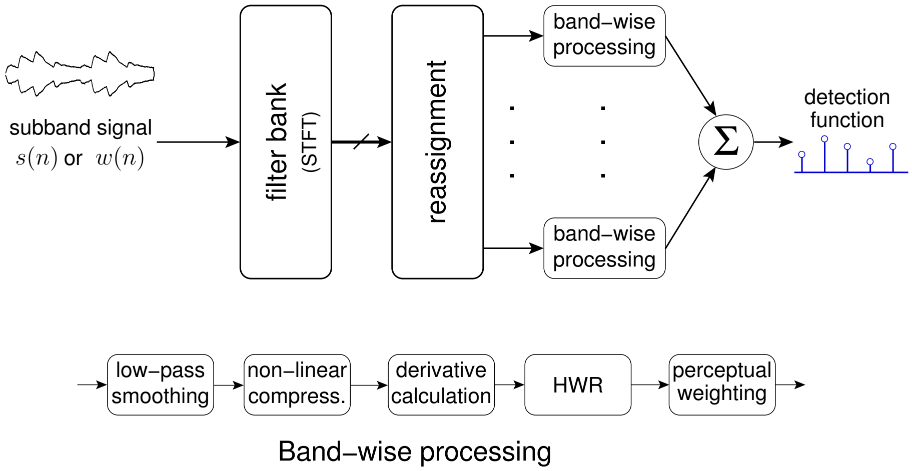


Figure 3.17: Spectral Energy Flux (SEF) flow diagram.

of the musical note envelopes required in our onset detection approach. Although originally called the *modified moving window method* and developed in the context of the STFT (Kodera et al., 1978), the reassignment finds its origins in the *Principle of Stationary Phase* (Papoulis, 1962, chapter 7). In fact, this method has been used to sharpen various time–scale and TFR's in order to make them more *intelligible* as developed in Auger & Flandrin (1995) and Hlawatsch & Auger (2005). As pointed before, Hainsworth & Wolfe (2001), Röbel (2003, 2005) and Peeters (2005) have used this method to enhance their onset detection algorithms. Below we provide a brief description outlining the main concepts and advantages of the reassignment procedure.

Let us remember the definition of the conventional STFT stated in Eq. (3.8) as:

$$\widetilde{S}(m,k) = \sum_{n=-\infty}^{\infty} g(n+Mm)s(n)e^{-\jmath\frac{2\pi}{N}kn} \tag{3.12}$$

where $m \in \mathbb{Z}$ is the time (frame) index, $g(n)$ is a real window of finite length N which determines the portion of the signal[19] $s(n)$ that is under analysis at a particular time instant m, M is defined as the hop-size or time-shift for the window and $k = 0, \ldots, K-1$ is the frequency (bin) index. In this definition, the data is sampled at a rate equal to the analysis window hop size (M), so information in the resulting TFR is stored in a regular temporal grid corresponding to the (geometrical) centers of the short-time analysis windows. We can make the sampling of this frame-based representation as dense as desired by choosing an appropiate hop-size (under the limitation $M \geq 1$). Nevertheless the temporal *smearing* due to long analysis window needed to achieve a high-resolution in frequency estimations (especially at low-frequencies) cannot be relieved by this denser sampling.

Although it is widely accepted that the STFT phase component bears important temporal information, it is typically discarded and only the magnitude part is considered in the representation. On the contrary, the so-called method of reassignment computes sharpened time and frequency estimates for each spectral component in $\widetilde{S}(m,k)$ from

[19] For the sake of clarity, we drop the subband index and we only refer to $s(n)$ knowing that the very same procedure is applied to all subbands in both signal components.

the partial derivatives of the short-time phase spectrum. Instead of locating the time–frequency components at the geometrical center of the analysis window (m, k), as in the conventional STFT, the components are reassigned to the center of gravity of their complex spectral energy distribution.

Continuous time STFT Since the theoretical foundation of the reassignment operation relies on the continuous definition of the STFT, we switch for a moment from the discrete-time domain to continuous time. Let $s_c(t)$, with $t \in \mathbb{R}$, be the real signal under analysis, the associated STFT is formulated as:

$$\widetilde{S}_c(\tau, f) = \int_{-\infty}^{\infty} s_c(t) g_c(t-\tau) e^{-\jmath 2\pi f t} dt. \tag{3.13}$$

The main handicap in the estimation of the magnitude and frequency of the sinusoidal components of $s_c(t)$ is directly related to the length and bandwidth of $g_c(t)$ (perhaps better known as the time–frequency Heisenberg box). As mentioned before, the reassignment improves the estimation of the TF content by using the phase information. Let us write the STFT in terms of its magnitude and phase as:

$$\widetilde{S}_c(\tau, f) = |\widetilde{S}_c(\tau, f)|\, e^{\jmath\varphi(\tau, f)}.$$

The reassignment operators are derived from the partial derivatives of $\varphi(t, f)$ with respect to each of its variables, leading respectively to the instantaneous frequency

$$F_i(\tau, f) = \frac{1}{2\pi}\frac{\partial\varphi(\tau, f)}{\partial\tau}, \tag{3.14}$$

and to the group delay

$$T_g(\tau, f) = -\frac{1}{2\pi}\frac{\partial\varphi(\tau, f)}{\partial f}. \tag{3.15}$$

Eqs. (3.14) and (3.15) can be interpreted as follows: if we consider the energy $|\widetilde{S}_c(\tau_0, f_0)|^2$ smeared around a given position (τ_0, f_0) in the time–frequency plane, its center of gravity is the point with normalized frequency $F_i(\tau_0, f_0)$ and time location $\tau_0 + T_g(\tau_0, f_0)$. Therefore, each energy *spot* is said to be reassigned to a centroid, *i.e.*, the time-frequency signal content is *re-mapped* on the plane.

Discrete-time implementation Let us rewrite Eq. (3.12) in its polar form as:

$$\widetilde{S}(m, k) = |\widetilde{S}(m, k)|\, e^{\varphi(m, k)} \tag{3.16}$$

Since the derivative of functions given in discrete points are not defined, we must find a way to approximate the partial derivatives of Eqs. (3.14) and (3.15) by means of numerical processing. A clever way to circumvent this limitation is by using a FIR differentiator filter. In this work we decided to use a differentiator based on the formulae for central differentiation developed by Dvornikov (2003), the reason will be justified later.

$\varphi(m, k)$ is then extracted and unwrapped for each channel k, then its derivative is computed to obtain the instantaneous frequency, that is:

$$F_i(m, k) = \varphi(m, k) \star h(m) \qquad \text{for a fixed value of } k, \tag{3.17}$$

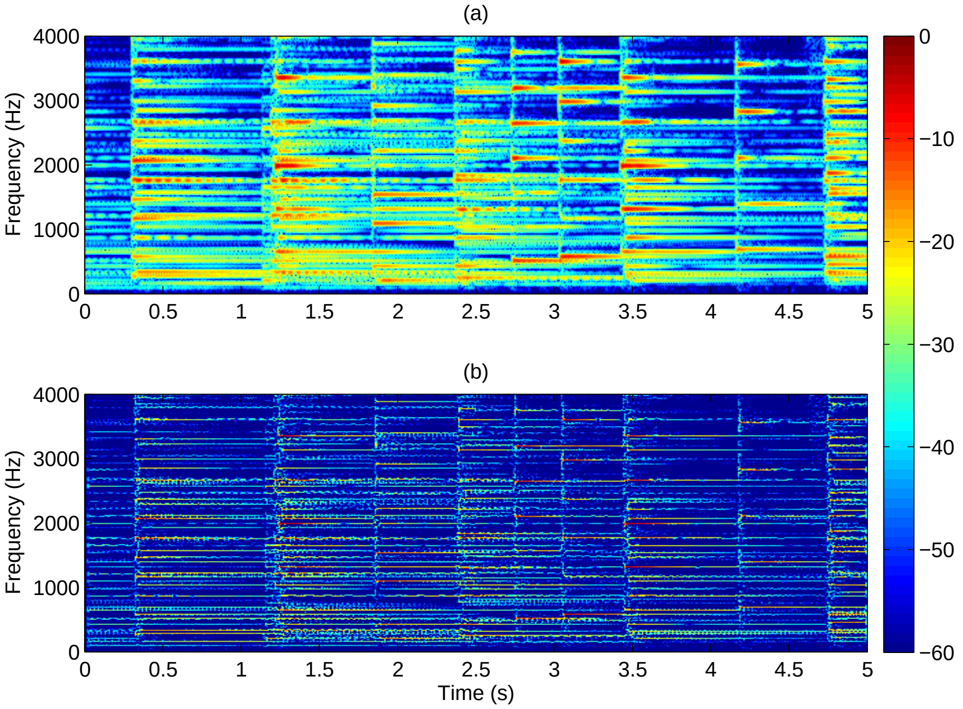


Figure 3.18: STFT reassignment: piano example. (a) Conventional STFT and (b) reassigned STFT.

where h is the differentiator filter. The same procedure is applied along the frequency axis, yielding the group delay:

$$T_g(m,k) = \varphi(m,k) \star h(k) \qquad \text{for a fixed value of } m. \tag{3.18}$$

As described above, the spectral content in $\widetilde{S}(m,k)$ is then remapped according to F_i and T_i. Hereinafter, we denote as $\widetilde{\mathcal{S}}(m,k)$ the reassigned STFT.

Figures 3.18 and 3.19 show two reassignment examples. The first one presents a piano signal of 5 s, on top the conventional STFT and in the bottom the reassigned version. In both representations, the dynamics have been normalized and limited to the range $[-60\text{ dB} \quad 0\text{ dB}]$. We can notice that a significant amount of the spectral *debris* present in the conventional TFR has been removed in its reassigned counterpart. Additionally, the spectral leakage has been removed and now the spectral components have a precise localization. The second example presents a 5 s violin signal and the same observations can be made.

3.4.3.2 Spectral Energy Flux

The method that we use to compute the musical stress profile is the so-called *Spectral Energy Flux*. It has been used before in the literature, for instance by Laroche (2001, 2003) also in the context of metrical analysis. This technique resides on the general assumption that the appearance of an onset (event) in an audio stream leads to a variation in the signal's frequency content. For example, in the case of a violin producing pitched notes, the

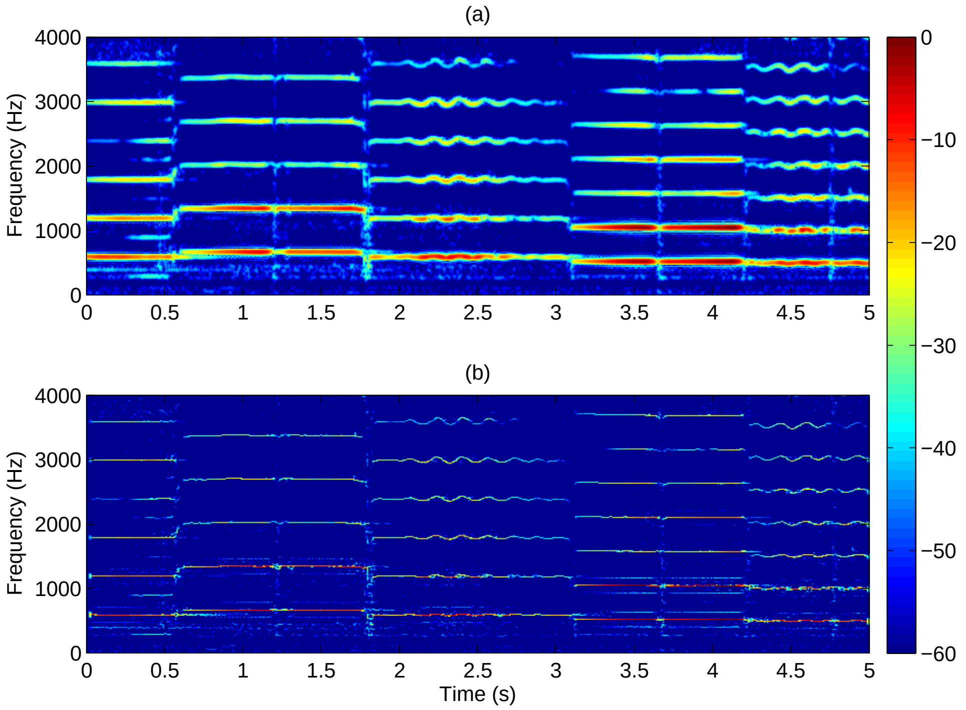


Figure 3.19: STFT reassignment: violin example. (a) Conventional STFT and (b) reassigned STFT.

resulting signal will have a strong fundamental frequency as well as the related harmonic components at integer multiples of the fundamental attenuating as frequency increases. In the case of a percussive instrument, the resulting signal will tend to have sharp energy boosts.

To detect the above mentioned variations in the frequency content of the audio signal, the most natural approach is to compute the derivative of the TFR with respect to time. In our case that means computing the derivative of the reassigned STFT $\widetilde{\mathcal{S}}(m,k)$:

$$E(m,k) = V(m,k) \star h(m) = \sum_{l} h(m-l)\, V(l,k) \tag{3.19}$$

and where $E(m,k)$ is known as the Spectral Energy Flux (SEF), $h(m)$ is an approximation to an ideal differentiator where

$$H(e^{j2\pi f}) \simeq j2\pi f \tag{3.20}$$

and

$$V(m,k) = \mathcal{F}\{\widetilde{\mathcal{S}}|(m,k)|\} \tag{3.21}$$

is a transformation that accentuates some of the psychoacoustically relevant properties of $\widetilde{\mathcal{S}}(m,k)$. The precise details about this operation are given below.

Digital differentiator In solving many physical problems by means of numerical methods, it is a challenge to seek derivatives of functions given in discrete points. Due to its considerable relevance in many disciplines, this subject has attracted much attention and

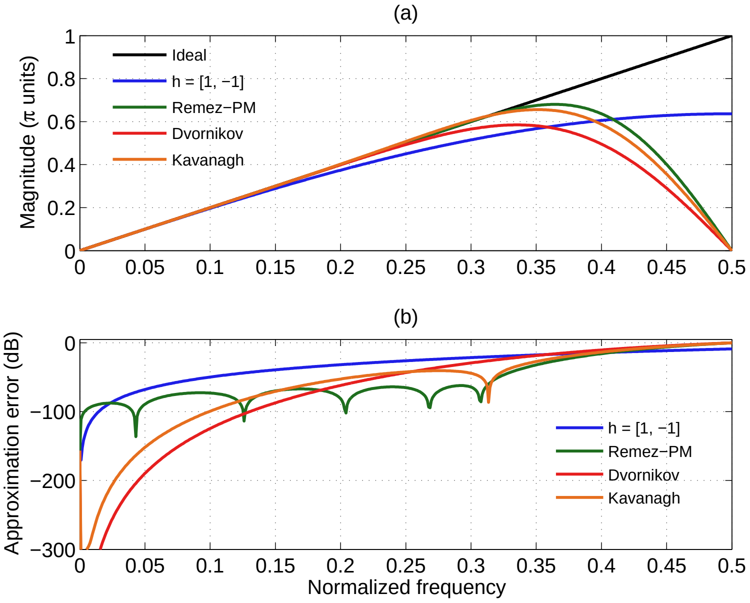


Figure 3.20: Digital FIR differentiators comparison. (a) Differentiator approximation to an ideal filter and (b) accuracy of the approximation.

in consequence a large number of *digital differentiators* have been proposed in the literature. A comprehensive search of the available methods is beyond the scope of our work, but in the context of our research we have evaluated four different FIR digital differentiators. The first one is the classical first-order difference where $h = [1, \, -1]$. This differentiator has been employed in several metrical analysis systems, for instance Klapuri (1999, 2003), Laroche (2001, 2003), Peeters (2005) and Alonso et al. (2003a). The second method is based on the Remez-Parks-McClellan (RPM) optimisation procedure which leads to the best approximation to Eq. (3.20) in the minimax sense (Proakis & Manolakis, 1996, page 652). In (Alonso et al., 2004) we have used this method to carry out metrical analysis. The third differentiator proposal is based on the formulae of numerical differentiation proposed by Dvornikov (2003). The fourth method has been developed by Kavanagh (2001) and was especially conceived for quantized signals differentiation. The underlying principle of these last two differentiation methods is the calculation of interpolating polynomials passing through discrete points.

Figure 3.20 compares the performance of all these digital differentiators in approximating Eq. (3.20). Based on heuristic tests, we concluded that a differentiator filter of order[20] $2L = 10$ was the best compromise of accuracy *vs.* memory and computational requirements. Our differentiation requirements do not demand a computationally expensive and less performant full-band rectifier. In fact, the sampling frequency (or frame rate) of the STFT is well above the frequency range of the rhythmic phenomena (< 25 Hz) we wish to detect. For this reason we focus our attention in the low-frequency region $0 \leq f \leq \Delta_f$, where $\Delta_f = 0.15$.

With the exception of the first-order difference, strictly speaking, the other digital

[20] Due to their design principle, the Kavanagh and Dvornivok filters are obliged to be of even order, *i.e.*, to have an odd filter-length equal to $2L + 1$.

Method	First order difference	Remez-Parks-McClellan (Proakis & Manolakis, 1996)	Dvornikov (2003)	Kavanagh (2001)
MSE	0.4913e-6	0.0097e-6	< 0.0001e-6	< 0.0001e-6

Table 3.6: Digital differentiators mean-square error. It was computed in the normalized frequency interval $0 \leq f \leq \Delta_f$.

differentiators are not causal. Nevertheless, if real-time requirements are imposed by the application, the use of these differentiators can be considered since the *future* data demand goes as far as L STFT frames ahead (usually below 30 ms).

Under the settings mentioned above, the differentiator proposed by Dvornikov (2003) gives the best results. It has the smallest mean-square error (MSE) (see Table 3.6)

$$\text{MSE} = \frac{1}{\Delta_f} \int_0^{\Delta_f} [H_i(f) - H(f)]^2 df \tag{3.22}$$

in the frequency range $0 \leq f \leq \Delta_f$, where $H_i(f)$ is the frequency response of an ideal differentiator and $H(f)$ is frequency response of any of the filter differentiators under analysis.

In addition, Dvornikov (2003) differentiator also displays the best performance in a set of heuristic tests we carried out. The first-order difference was ranked last among the four differentiators we evaluated. This is not a surprise since the time span of the other filters is considerably larger. In other words, additional information coming from several analysis frames is taken into account, while the plain spectral difference only uses the information from two consecutive analysis frames.

As justified above, we opted for using a digital differentiator $h(m)$ of order $2L$ (thus of length $2L + 1$) based on the formulae for central differentiation (Dvornikov, 2003), see Appendix B. The analytical expression of the first L coefficients of the antisymmetric FIR digital differentiator is given by

$$\gamma(l) = \frac{1}{l\alpha(l)}$$

where

$$\alpha(l) = \prod_{\substack{k = 1 \\ k \neq l}}^{L} \left(1 - \frac{l^2}{k^2}\right) \tag{3.23}$$

and $l = 1, \ldots, L$. The coefficients of $h(m)$ are then given by

$$h = [-\gamma(L), \ldots, 0, \ldots, \gamma(L)]. \tag{3.24}$$

Figure 3.21 shows the waveform of a tenth order ($L = 5$) digital differentiator.

Perceptual transformation In our onset detection proposal, the transformation $V(m, k)$ calculates a perceptually plausible power envelope for frequency channel k and is formed of two steps. First, psychoacoustic research on computational models of mechanical to neural transduction (Meddis, 1988) shows that the auditory nerve adaptation response

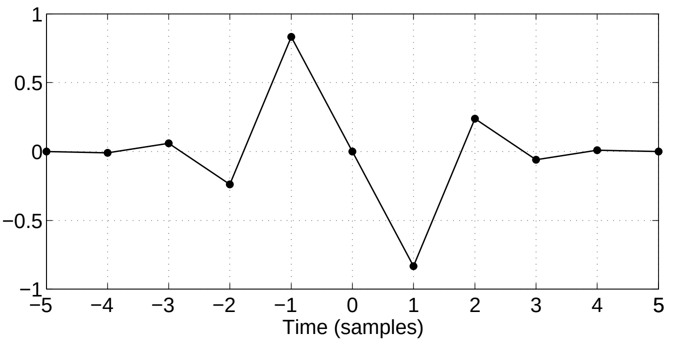


Figure 3.21: Digital differentiator waveform obtained by the method proposed by Dvornikov (2003) with $L = 5$.

following a sudden stimulus change can be characterized as the sum of two exponential decay functions:

$$\phi(m) = \alpha e^{-m/T_1} + \beta e^{-m/T_2} \quad \text{for} \quad m \geq 0 \tag{3.25}$$

formed by a rapid decline component with time constant (T_1) in the order of 15 ms and a slower short-term decline with a time constant (T_2) in the region of 70 ms. This adaptation function performs energy integration, emphasizing the most recent stimulus but masking rapid modulations. From a signal processing standpoint, this can be viewed as two smoothing low-pass filters whose impulse response has a discontinuity that preserves edge sharpness and avoids dulling signal attacks. In practice, the smoothing window is implemented as a 2nd-order IIR filter with z-transform

$$\Phi(z) = \frac{\alpha + \beta - (\alpha z_2 + \beta z_1)\, z^{-1}}{1 - (z_1 + z_2) z^{-1} + z_1 z_2\, z^{-2}}. \tag{3.26}$$

where $\alpha = 1$, $\beta = 5$, $z_1 = e^{-1/T_1}$, $z_2 = e^{-1/T_2}$ $T_1 = 15$ ms, and $T_2 = 75$ ms. Several authors have pointed the importance of this low-pass or smoothing step prior to computing the derivative. For instance, some approaches use a half-Hann window (descending-part), this was originally proposed by Todd (1994) and then used by Scheirer (1998), Klapuri (1999), Alonso et al. (2003a) and Jehan (2004). On the other side, Paulus & Klapuri (2002) and Klapuri et al. (2006) use a 6th order low-pass Butterworth filter and Peeters (2005) opted for a 5th order low-pass elliptic filter, in these last three examples the filters were designed to have a cut-off frequency $f = 10$ Hz.

Figure 3.22 presents the frequency response of the four smoothing filters mentioned above. We suppose that the inter-frame distance in the STFT is 5.6 ms, this is equivalent to stating that the TFR is sampled at 180 Hz. Undoubtedly, the smoothing filters proposed by Paulus-Klapuri and Peeters are more effective for removing the high-frequency signal content.

Now, let us take a closer look at the duty that those low-pass filters are exposed to. Figure 3.23-(a) shows a quite typical situation found in onset detection, it displays a (normalized) *pitched* channel of the TFR corresponding to a piano signal. We can see that the rapid variations in the power-envelope are very pronounced yielding true note attacks not easily detectable, even for the human eye (true onsets are marked by a red vertical dashed-line).

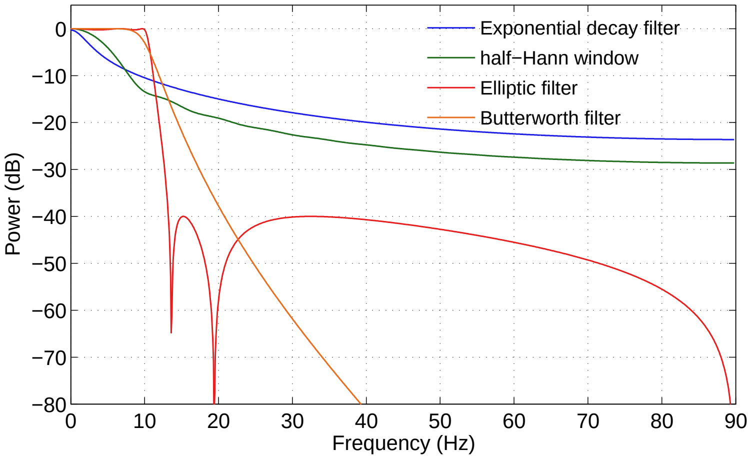


Figure 3.22: Smoothing filters and their frequency reponse.

Figure 3.23-(b)–(e) show how the afore mentioned *smoothers* react under this stimulus. Figure 3.23-(b) presents our approach using the two exponential decay functions Alonso et al. (2005c). Figure 3.23-(c) presents that using a half-Hann window (of length 200 ms) proposed by Todd (1994) and others. Figure 3.23-(d) shows the output of the elliptic filter proposed by Peeters (2005) and finally Figure 3.23-(e) the output of the Butterworth filter proposed by Klapuri et al. (2006).

Although at first sight all of the output envelopes look rather similar, there exist some significant distinctions. For example (d) and (e) have completely cleared the high frequency variations, but they display low-frequency oscillations of significant power after every attack which might produce important peaks after taking the derivative and eventually yielding to false note onsets. Plots (b) and (c) are very similar with the latter slightly smoother. Both still exhibit small amplitude high-frequency oscillations after every note onset, but their derivative is small compared to that of the signal attacks. In terms of computational resources, (b) has the lowest requirements since it is a 2nd-order all-pole IIR filter[21]

Concerning the power envelope smoothing, we consider that the time-domain aspect of the *smoother* plays a more important role than its frequency behaviour.

The second part of the envelope extraction consists in a logarithmic compression, this operation was originally proposed by Klapuri (1999). Similarly to the smoother filter, this procedure also has a perceptual relevance since the logarithmic difference function gives the amount of change in a signal's intensity in relation to its level, that is

$$\frac{d}{dt}\log I(t) = \frac{\Delta I(t)}{I(t)}. \tag{3.27}$$

The perceived increase in signal level is in relation to its level, in other words, this means that the same amount of increase is more prominent in a quiet signal. According to Moore (1995), the smallest detectable change in intensity is approximately proportional to the intensity of the signal, *i.e.*, the Weber fraction ($\frac{\Delta I}{I}$) is a constant (Klapuri, 1999).

[21] In the figure, (c) is a FIR filter with 36 coefficients (200 ms length), (d) is an IIR filter with 5-poles and 5-zeros and (e) is also an IIR filter with 6-poles and 6-zeros.

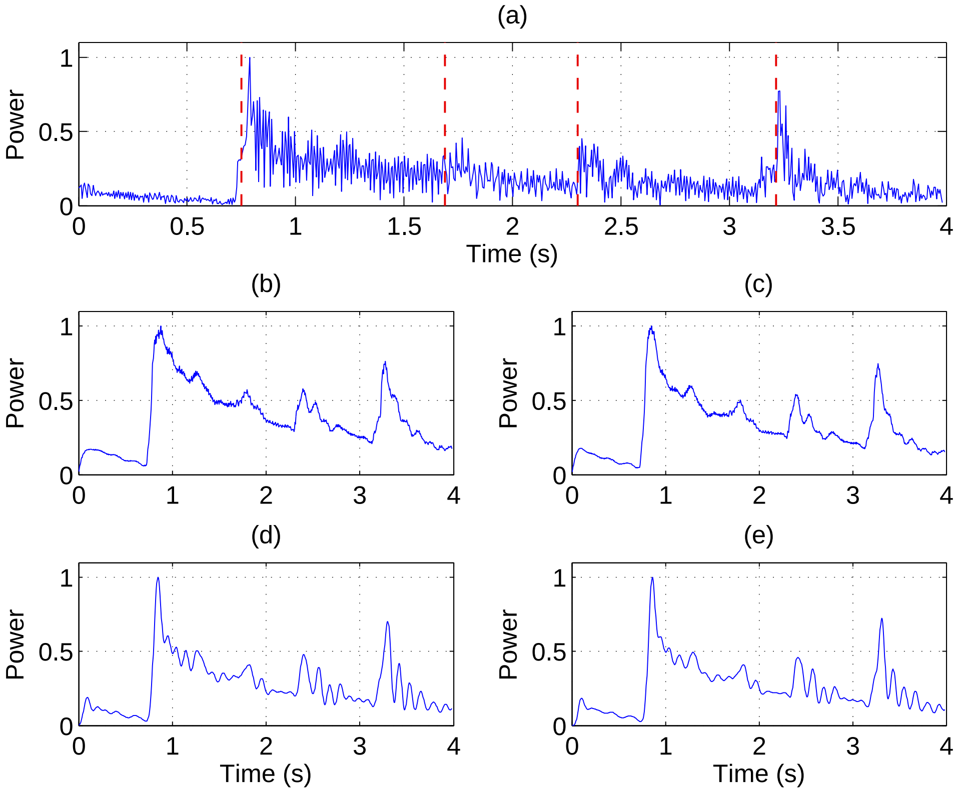


Figure 3.23: Smoothing effects corresponding to different low-pass filters. (a) noisy envelope of a *pitched* STFT channel belonging to a piano signal. Envelope obtained after smoothing with (b) two exponential decay functions, (c) a half-Hann window, (d) an elliptic low-pass filter, and (e) a Butterworth low-pass filter.

Implementation In practice, the algorithm implementation is straightforward and is carried out step by step exactly as presented in the Figure 3.17. The TFR in Eq. (3.12) is computed using an N point Fast Fourier Transform (FFT) and its reassigned version is obtained using Eqs. (3.17) and (3.18). Then the square of the absolute value of every frequency channel, $|\tilde{S}(m,k)|$ is convolved with the low-pass filter $\phi(m)$. The smoothing operation is followed by a logarithmic compression. The resulting compressed and smoothed envelope $V(m,k)$ is given by

$$V(m,k) = 20\log_{10}\left(\sum_i |\tilde{S}(i,k)|\phi(m-i)\right). \tag{3.28}$$

Then the SEF is computed as indicated by Eq. (3.19). Every frequency channel k is time-convolved with the differentiator filter $h(m)$. At those time instants where the frequency content of $s(t)$ changes and new frequency components appear, $E(l,k)$ exhibits positive peaks whose amplitude is proportional to the energy and rate of change of the new components. In a similar way, when frequency components disappear from $s(t)$, the SEF exhibits negative peaks, marking the *offset* of a musical event. Since we are only interested in onsets, we apply a half-wave rectification (HWR) to $E(m,k)$, *i.e.*, we only

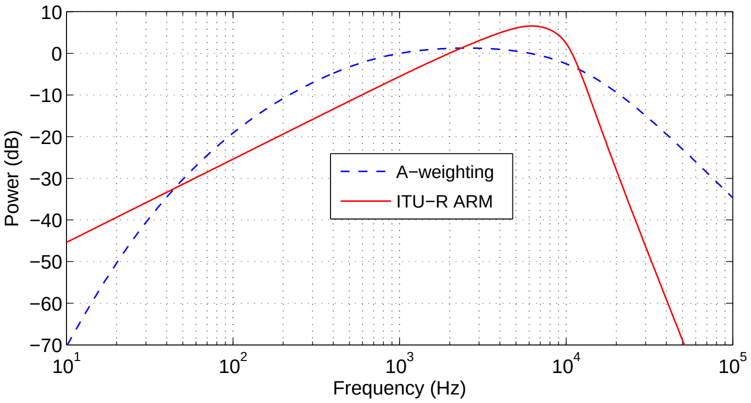


Figure 3.24: Weighting functions comparison. In blue-dashed trace the A-weighting curve which is said to reflect the equal-loudness contours. In red trace, the more recent "ITU-R ARM" weighting filter. According to Dolby et al. (1979), the latter has better agreement with subjective assessments.

keep positive values

$$\widehat{E}(m,k) = \begin{cases} E(m,k) & \text{if } E(m,k) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Before integrating the contributions from all frequency channels, we must keep in mind that during the pre-processing stage the power level of frequency componentes was altered. In order to counterbalance this effect and to obtain a detection function which takes into account some psychoacoustic principles, we decided to weight $E(l,k)$ by a perceptual curve. A possible candidate is the familiar A-weighting curve (IEC268-1, 1968) which is said to reflect the *equal-loudness contours*. However, Dolby et al. (1979) are critic with respect to these curves, they argue that they only relate to the subjective loudness of pure tones and not to broad-band audio signals. For this reason, we opted for a more recent and perceptually plausible weighting curve usually known as the "ITU-R ARM" (average response meter) derived from the standard ITU-R468 (1986). According to its developers (Dolby et al., 1979) this weighting curve has better agreement with subjective assessments and is widely employed in professional and commercial audio-level meter equipment. Appendix C presents the mathematical expression to compute the respective weighting curve $\mathcal{W}(k)$. In addition, Figure 3.24 presents the frequency-shape of the ITU-R ARM weighting function along with that of A-weighting.

To obtain the so-called onset detection function $d(m)$, contributions from all channels are weighted and integrated across frequency

$$d(m) = \sum_k \widehat{E}(m,k)\mathcal{W}(k). \tag{3.29}$$

Finally, $d(m)$ should ideally display sharp peaks at transients and note onsets, those instants where the positive energy flux is large. In addition, the peaks amplitude bears a relation with the loudness of the acoustic events.

Figure 3.25 shows an example using a piano signal. Although it is a rather simple case, our onset detection method exhibits a good performance even distinguishing two very close events at about 3.25 s. The four plots of Figure 3.25 respectively correspond from top to bottom to:

(a) the waveform under analysis where the hand-annotated onsets are marked by dotted vertical lines;

(b) the respective modulus of the reassigned-STFT, the signal's harmonic structure is visible;

(c) the SEF $\widehat{E}(m,k)$, the dotted points aligned indicate the regions where the positive energy flux is large; and

(d) presents the corresponding detection function $d(m)$, the note onset instants and a perceptually-related intensity are indicated by the location and the height of the peaks respectively, true onsets are also marked by red vertical lines.

For onset detection developers, bowed string instruments are the *bogeyman*, since they can produce notes with soft and long attacks which can easily pass undetected by the stress estimation block. Moreover, the frequency partials that these instruments produce can also be non-stationary (*e.g.*, when playing vibrato) and such behavior might cause the appearance of false onsets in the detection function. Now we present a more challenging example using a violin signal. The four plots of Figure 3.26 respectively correspond from top to bottom to:

(a) the waveform under analysis where the hand-annotated onsets are marked by dotted vertical lines;

(b) the modulus of the reassigned-STFT, we can see that the signal's frequency structure exhibits strong vibratos at about 0.75 s and 3.75 s;

(c) the SEF $\widehat{E}(m,k)$, since the attacks are not very sharp the image is more difficult to interpret; and

(d) the corresponding detection function $d(m)$, in this signal only the peaks of six (out of seven) note attacks were detected by the algorithm. Although in this case $d(m)$ is less suitable for a direct *peak-picking* like onset detection, it is still fairly useful for periodicity induction in its present state.

Ideally, the musical stress profile should be as close as possible to a series of weighted impulses positioned at onset locations. The detection function corresponding to the piano example (Figure 3.25) has a good resemblance to this theoretical appearance. In this case, onset positions can be distinguished in the time-waveform as sudden changes in the amplitude going from low-energy to higher-energy values. In the violin example (Figure 3.26), conducting onset detection by visual inspection of the time-waveform is less evident since onsets are located in low-energy regions. However, the detection function correctly resolved most of the onsets as prominent impulses. In this second case the problem is that the algorithm also produced *false onsets*, *i.e.*, prominent peaks that do not belong to an onset.

The computation of the detection function is the final stage, this signal provides the degree of musical accentuation as a function of time. musical. The output of this front-end blocks is formed of two groups of signals: the band-wise detection functions of the harmonic part called $d_p^s(m)$, and the band-wise detection functions of the noise part refered as $d_p^w(m)$. Where $p \in [1, \ldots, 8]$ for the uniform filter bank and $p \in [1, \ldots, 5]$ for the logarithmic filter bank.

3.5 Conclusions

Throughout this chapter we have described a novel method which analyzes audio recordings in order to compute the so-called *musical stress profile*. This profile can be seen as a signal bearing "symbols" pulsations which indicate the likeliness[22] of finding a musical accent (or note onset). The novelty of our method resides on the idea of separating the audio signal in order to emphasize the phenomenal accents that they contain. This separation consists in decomposing the audio input into a harmonic (or deterministic) part and a "noise" part which contains all the elements from the original signal that cannot be modeled as sinusoidal components.

We have proposed two different methods to conduct this decomposition. The first one uses a subspace analysis technique (sometimes referred to as high resolution methods) based on the Exponentially Damped Sinusoidal (EDS) model. The second one is based on a more traditional Fourier-based method. Then, we introduced a technique to calculate the musical stress profile of the harmonic and noise components. In fact, this procedure is a significant improvement of a previously existing method called the Spectral Energy Flux (SEF) or also Spectral Difference (SD). This method is founded on the idea of measuring the rate of change of the power-spectrum as a function of time. These enhancements consists of computing a reassigned Short Time Fourier Transform (STFT). Next, a perceptually motivated method to calculate the smoothed power envelope of each STFT channel is presented. Then, the derivative of frequency components is computed using an efficient differentiator filter. Finally, contributions from individual components are added to create the stress profile. This signal displays sharp maxima at transients and note onsets, i.e., those instants where the energy flux is large.

In order to reduce the computational complexity of the stress estimation block, it is possible to disable some of its components at the expense of reducing its efficiency. For example, when processing strong beat music, a fairly accurate detection function can be obtained at a lower cost by disabling the H+N decomposition block. Another way to reduce even more the computational burden consists in using a traditional STFT instead of a reassigned version.

Given the distinctive nature of musical instruments, an important question to ask is: *is it better to estimate the stress profile using various algorithms adapted to a number of potential sound sources or to develop a single algorithm to handle all of them?* In this work we opted for the latter, *i.e.*, one algorithm using the same set of parameters for processing all kinds of music signals. While the earlier idea of developing source-specific algorithms might seem theoretically advantageous, a very important and highly sensitive problem is how to fuse the information from each of these methods when processing a musical instance containing multiple and different sound sources. More precisely, how to prevent false onsets produced by any of the algorithms from damaging the others? Otherwise, the applicability of these algorithms would be considerably restricted to solo performances.

[22] By likeliness we mean "the possibility of occurring" and not the probabilistic sense of the word.

Figure 3.25: The four plots correspond from top to bottom to: (a) the waveform under analysis; (b) the respective modulus of the reassigned-STFT, the signal's harmonic structure is visible; (c) the SEF $\widehat{E}(m,k)$, the dotted points aligned indicate the regions where the positive energy flux is large; and (d) presents the corresponding detection function $d(m)$. Manually annotated onsets are marked by red vertical lines.

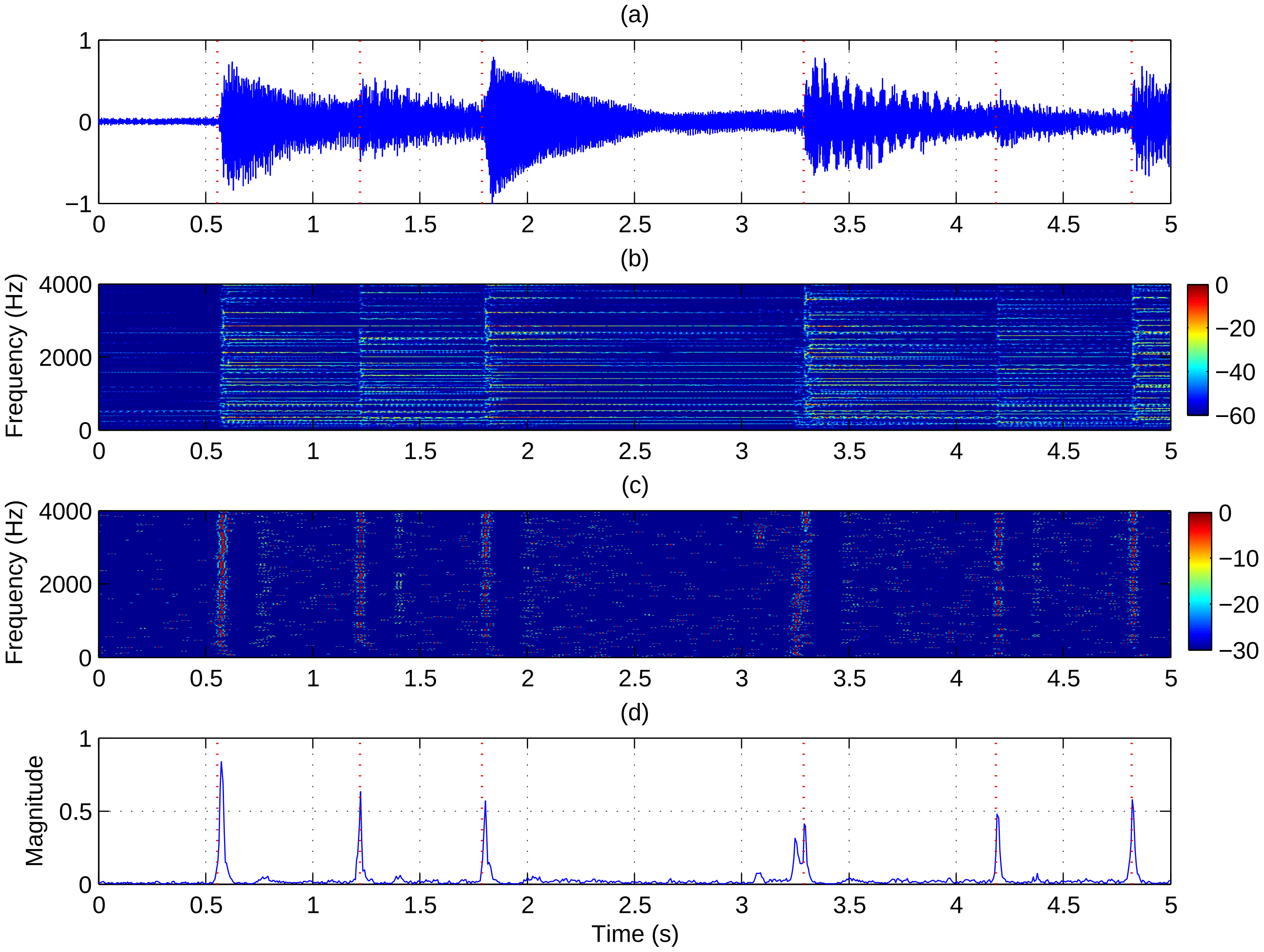
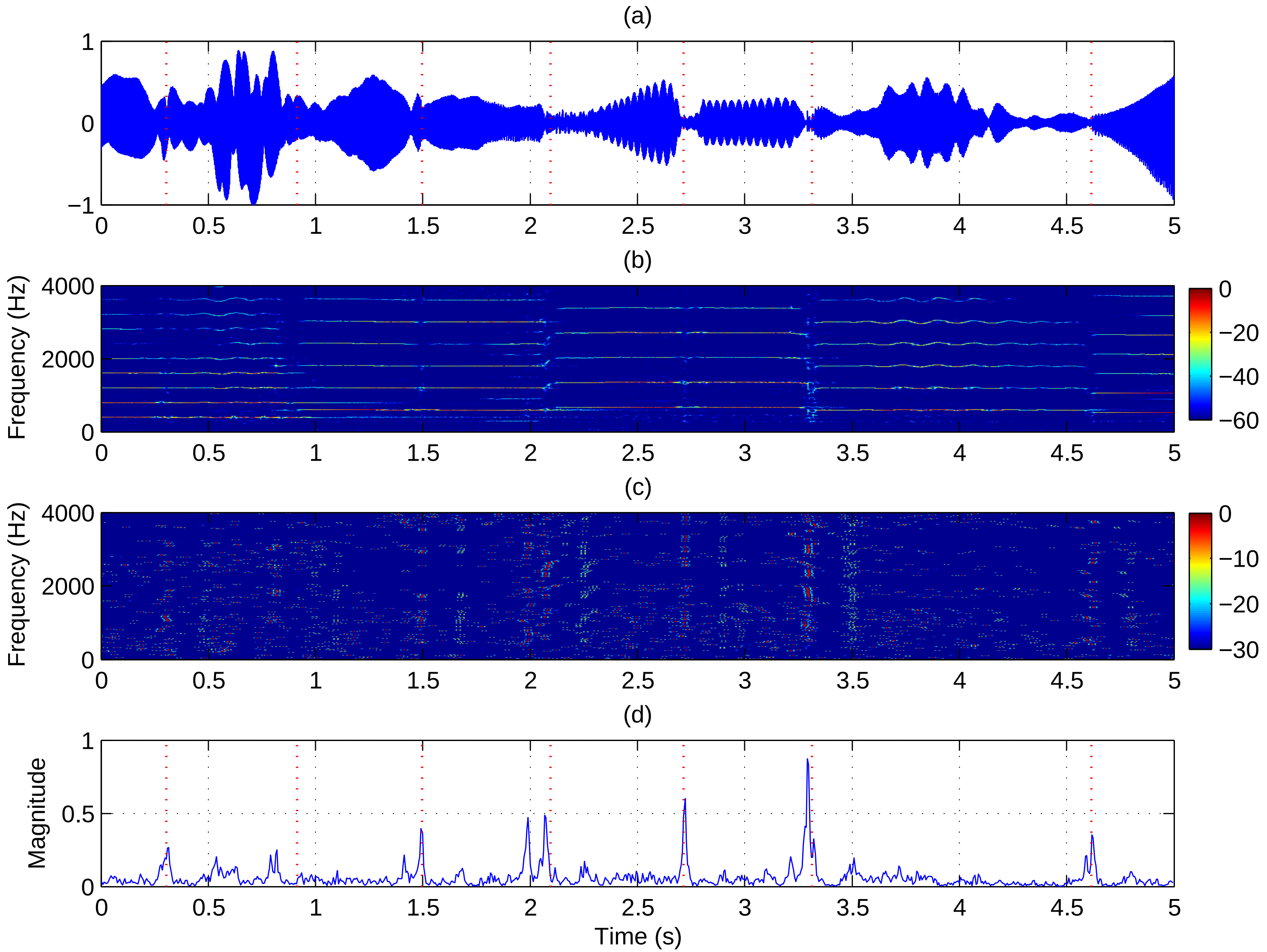


Figure 3.26: The four plots correspond from top to bottom to: (a) the waveform under analysis; (b) the modulus of the reassigned-STFT, we can see that the signal's frequency structure exhibits strong vibratos at about 0.75 s and 3.75 s; (c) the SEF $\widehat{E}(m,k)$, since the attacks are not very sharp the image is more difficult to interpret; and (d) the corresponding detection function $d(m)$. Manually annotated onsets are marked by red vertical lines.



Chapter 4

Inducing rhythm metrics

In this chapter, we exploit a symbolic representation input to induce the periodicity of music accents (*e.g.*, note onsets, chord changes), to track their evolution through-time and to estimate the individual locations of metrical pulses at the tactus level. According to the framework depicted in Figure 4.1, in this chapter of the system description we cover the following blocks: periodicity analysis and data fusion, tracking of the rhythmic paths and pulse phase location. In addition, we also present a method to estimate the tatum.

As seen from above, this chapter is intimately related to the musical stress profile computed in the previous part. We must point out that other methods for computing the detection function can be used, as will be shown during the evaluation chapter. In other words, there exists no strict dependence between the elements introduced in the previous chapter and our approach to induce rhythm metrics presented hereafter.

4.1 Periodicity analysis

In §1.1 musical rhythm was defined as an acoustic sequence evoking a sensation of pulse (Parncutt, 1994). This statement makes us think of a regular recurrence of sound events, which is in general effortlessly detected by humans. In computer-based metrical analysis of music, this machinery has to be *reproduced* by an algorithm which searches for periodic behaviors in the detection function. For this purpose, a large number of researchers have called upon methods developed in the more mature field of fundamental frequency (or f_0) estimation, usually referred to as "pitch detection" methods. In §2.3.3 can be found an exhaustive list containing those procedures currently used in the context of rhythm analysis.

In this section we describe four methods to estimate the periodicities inherent in the detection function as well as their integration into the context of our analysis framework (see Figure 4.1).

4.1.1 Predominant f_0 estimation

We must grasp and fully understand that our experiments are very often an imperfect representation of an idealized world. In our case, we model the detection function as a perfectly periodic signal although we know that in practice it is not true. Nevertheless, we exploit the fact that in cases where the tempo remains relatively stable the detection

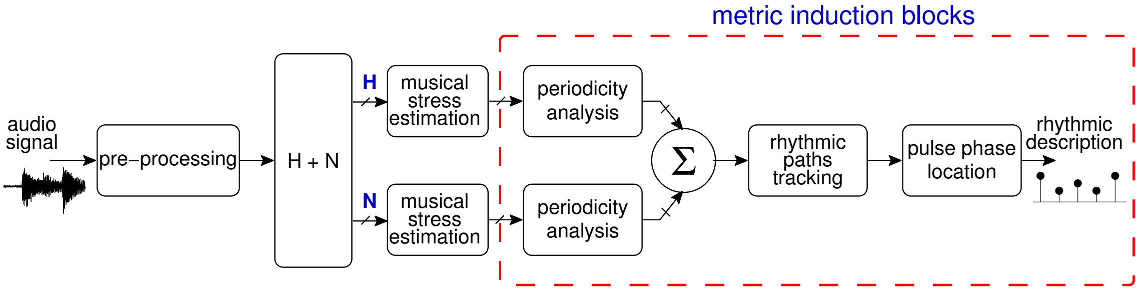


Figure 4.1: Flow diagram of the rhythm analysis framework. The dotted box gathers the building blocks discussed in this chapter.

function exhibits a quasi-periodic nature and modelling errors do not compromise the admissibility of the results.

During our research we use four periodicity estimation algorithms. Two of them can be classified as time-domain methods and the two others as frequency-domain methods. Detailed explanations are provided below.

4.1.1.1 Temporal methods

The temporal methods that we use to conduct periodicity analysis are the ubiquitous autocorrelation function (ACF) and a bank of comb-filter resonators. Both techniques have already been proposed a number of times in the rhythm analysis literature. For instance the ACF by Foote & Uchihashi (2001) and Dixon et al. (2003) and the comb-filter bank by (Scheirer, 1998) and Klapuri (2004).

Autocorrelation function As illustrated in Table 2.1, the biased estimator of the autocorrelation function (ACF) is probably the most used periodicity induction method in the context of metrical analysis of music signals. In fact, it is a mathematical tool used very frequently in signal processing for analyzing functions or series of values. The ACF computes the degree to which the signal is similar to a time-shifted version of itself. The deterministic sample ACF, also called sample autocovariance sequence, is usually defined as the cross-correlation of a signal with itself

$$\hat{r}_d(k) = \frac{1}{N}\sum_{n=k}^{N-1} d(n)d(n-k), \quad 0 \leq k \leq N-1 \tag{4.1}$$

where $d(n)$ is the real-valued detection function under analysis, N is the length of the analysis window and k indicates the time lag. The sample correlations for negative lags are constructed using the property $\hat{r}(-k) = \hat{r}(k)$ for $k = 0, \ldots, N-1$. The expression in Eq. (4.1) is called the standard biased ACF estimate. This operator is usually preferred than the non-biased version since it is likely to be a more accurate estimator of the $r(k)$ for relatively large values of k (compared to N) and it is guaranteed to be positive semidefinite.

As an example, Figure 4.2-(a) displays the detection function corresponding to 5 s of the song *Le bruit du frigo*[1], the annotated tempo for this excerpt is 185 BPM (*i.e.*, a

[1] Composed and performed by the group *Mano Negra*, album "King of Bongo" (french pop-rock).

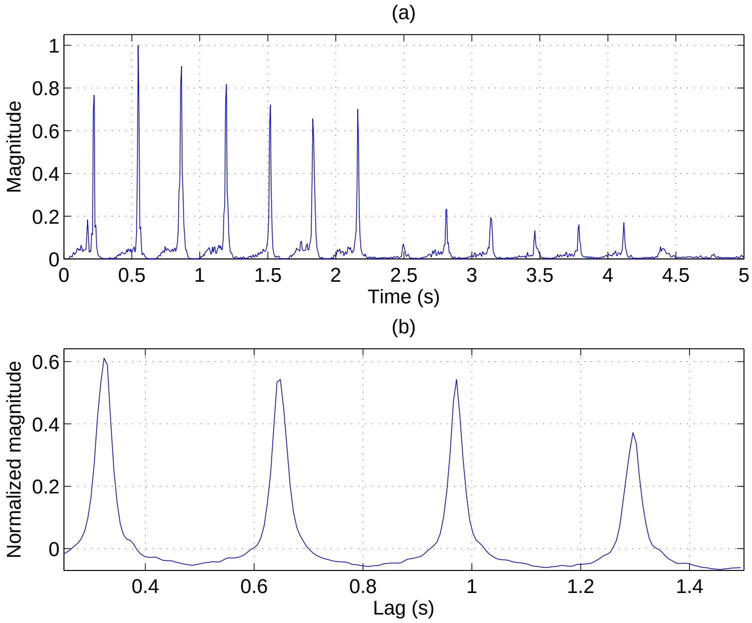


Figure 4.2: (a) An example detection function and (b) the corresponding periodicity profile computed via the ACF.

fundamental period of approximately 0.32 s). Figure 4.2-(b) presents the corresponding ACF computed in the range from 0.25 s (240 BPM) to 1.5 s (40 BPM). The ACF exhibits prominent peaks located at the first four integer multiples of the fundamental period (approximately 0.32 s, 0.65 s, 0.97 s and 1.29 s).

Bank of comb-filter resonators In the context of rhythm analysis, the use of a bank of comb-filter resonators with a constant half-time was originally proposed by Scheirer (1998). The comb-filters that we use here were developed by Klapuri (2004); Klapuri et al. (2006). They have an exponentially-decaying impulse response where the *half-time* term refers to the delay during which the response decays to a half of its initial value. The output of a comb-filter with delay τ when processing a detection function $d(n)$ is given by

$$y(n) = \alpha\, y(n-\tau) + (1-\alpha)d(n) \tag{4.2}$$

where the feed-back gain (distinct for each comb-filter) is given by $\alpha = 0.5^{\tau/T_0}$ and is calculated based on a selected half-time T_0 in samples. We use the value indicated by Klapuri (2004), which corresponds to a half-time equivalent to 3 s, *i.e.*, $T_0 = 3\mathrm{F_s}$, where $\mathrm{F_s}$ corresponds to the sampling frequency of the detection function. This value is short enough to react to tempo changes but long enough to reliably estimate pulse-periods of up to 4 s in length.

The prototype comb-filter of Eq. (4.2) has the following frequency response:

$$|H(e^{\jmath 2\pi f})|^2 = \frac{(1-\alpha)^2}{1+\alpha^2-2\alpha\cos(2\pi f\tau)}. \tag{4.3}$$

This filter has the maxima (resonates)

$$|H(e^{\jmath 2\pi f_{\text{max}}})|^2 = 1 \quad \text{for } f_{\text{max}} = \frac{l}{\tau}$$

where $l = 0, \ldots, \lfloor \tau/2 \rfloor$. In addition, it has the minima

$$|H(e^{\jmath 2\pi f_{\text{min}}})|^2 = \left(\frac{1-\alpha}{1+\alpha}\right)^2 \quad \text{for } f_{\text{min}} = \frac{2l+1}{2\tau}$$

where $l = 0, \ldots, \lfloor \frac{\tau-1}{2} \rfloor$. To obtain the overall power φ_α of a comb-filter with feed-back gain α, we integrate over the squared-impulse response, which yields

$$\varphi_\alpha = \frac{1-\alpha}{1+\alpha}. \tag{4.4}$$

In order to obtain a periodicity profile, we use a bank of such resonators where the delay gets values ranging from τ_{min} corresponding to .25 s (240 BPM) up to τ_{max} corresponding to 1.5 s (40 BPM). The instantaneous output energy for the prototype resonator of Eq. (4.2) at time n is given by

$$E_y(n) = \frac{1}{\tau} \sum_{i=n-\tau+1}^{n} y(i)^2. \tag{4.5}$$

This operation is equivalent to convolving the square of the resonator output with a rectangular window, *i.e.*, $E_y(n) = \frac{1}{\tau}(y^2 \star g)(n)$, where $g(n) = 1$, $t \in [0, \ldots, \tau - 1]$ and zero everywhere else.

Instead of using Eq. (4.5) as the output of the comb-filter bank, Klapuri (2004) improves the performance by proposing a normalization step

$$\hat{y}(n) = \frac{1}{1-\varphi_\alpha}\left(\frac{E_y(t)}{\hat{d}(n)} - \varphi_\alpha\right) \tag{4.6}$$

where $\hat{d}(n)$ is the energy of the detection function computed by applying a leaky-integrator, *i.e.*, a resonator which has $\tau = 1$ and the same half-time response:

$$\hat{d}(n) = \alpha\hat{d}(n-1) + (1-\alpha)d(n)^2.$$

This normalization compensates for the differences in the overall power responses, given by Eq. (4.4), for different values of α. This guarantees a unity response at the peak frequencies, while removing a τ-dependent trend. Figure 4.3 presents an example using the detection function depicted in Figure 4.2-(a). On top $E_y(n)$, given by Eq. (4.5), containing a τ-dependent trend. In the bottom, the normalized energy $\hat{y}(n)$.

Klapuri (2004) evaluated a number of periodicity induction methods, most of them performing equally well in terms of accuracy. The computational complexity of the comb-filters is $\mathcal{O}(1)$ per input sample per resonator.

4.1.1.2 Spectral methods

We propose two spectral techniques to carry out periodicity analysis: the spectral sum and the spectral product. These methods were independently developed by Noll (1970)

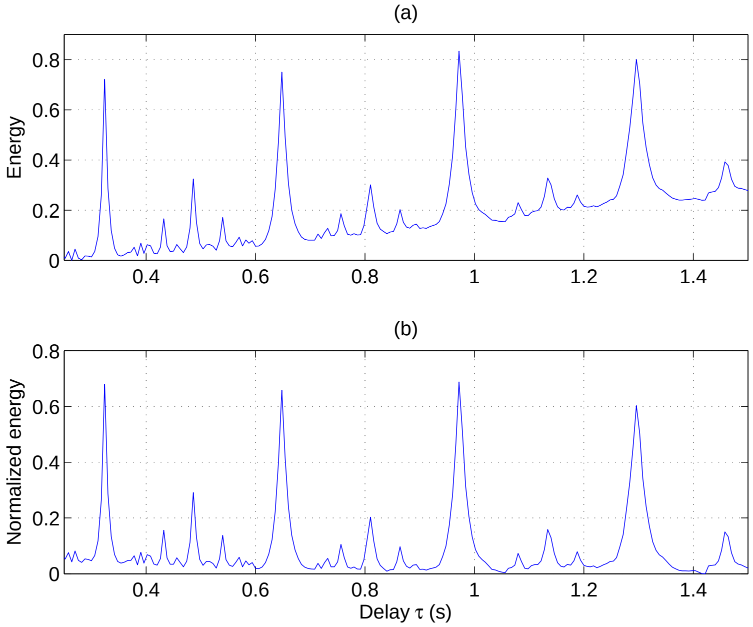


Figure 4.3: Periodicity profile obtained using a comb-filter bank. In (a) energy $E_y(n)$ containing a τ-dependent trend and (b) normalized energy $\hat{y}(n)$.

and Wise et al. (1976), for estimating the pitch period of voiced speech sounds and are based on a maximum likelihood formulation. Although more expensive in computational terms when compared to the comb-filter approach, the spectral sum and product are two highly reliable methods. In addition, to our knowledge they have not been applied elsewhere in the context of rhythm analysis.

Spectral sum We are interested in analyzing a periodic discrete time signal with period T, *i.e.*, $a(n+T) = a(n)$ and $n \in \mathbb{Z}$. In practice, we do not have direct access to $a(n)$. For that reason, we suppose that the observed signal (in our case the detection function), is a deterministic periodic signal surrounded by additive white Gaussian noise

$$d(n) = a(n) + w(n) \tag{4.7}$$

where the noise power is σ_w^2. Let us consider only a realization inside an analysis window $n \in [0, \ldots, N-1]$. In addition, we also suppose that N is an integer multiple of T, *i.e.*, $\exists K \in \mathbb{N}^* \,|\, N = KT$. This assumption does not affect the applicability of the algorithm, but reduces the mathematical development of the solution[2]. If we write Eq. (4.7) as $w(n) = d(n) - a(n)$, knowing that it is a sequence of i.i.d. random variables, then the likelihood function can be written as

$$p(d|T, a, \sigma_w^2) = \frac{1}{(2\pi\sigma_w^2)^{N/2}} e^{-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(d(n)-a(n))^2}.$$

[2] If the reader is interested, Wise et al. (1976, page 419) show how to obtain the same result without making this assumption.

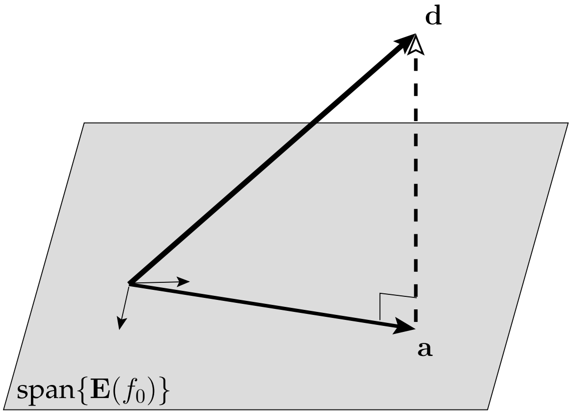


Figure 4.4: Orthogonal projection of $\mathbf{d}$ over the space of complex signals with period T given by span$\{\mathbf{E}(f_0)\}$.

And the log-likelihood as

$$L(T, a, \sigma_w^2) = -\frac{N}{2}\ln(2\pi\sigma_w^2) - \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(d(n) - a(n))^2. \tag{4.8}$$

For a fixed value of T and σ^2, maximizing L with respect to $a(n)$ is equivalent to minimizing the square of the distance between $d(n)$ and $a(n)$. That is, to find the signal $a(n)$ of period T having the shortest Euclidean distance to $d(n)$. This means that the vector $\mathbf{a} = [a(0), \ldots, a(N-1)]^\top$ is the orthogonal projection of $\mathbf{d} = [d(0), \ldots, d(N-1)]^\top$ over the space of complex signals with period T, as depicted in Figure 4.4. Now, for all $k \in [0, \ldots, T-1]^\top$, let

$$\mathbf{e}_k(f_0) = [1, e^{\jmath 2\pi k f_0}, \ldots, e^{\jmath 2\pi k f_0 (N-1)}]^\top$$

where $f_0 = \frac{1}{T}$, and

$$\mathbf{E}(f_0) = [\mathbf{e}_0(f_0)\ \ldots\ \mathbf{e}_{T-1}(f_0)]_{N\times T}.$$

The N-dimensional vector space of complex signals with period T is span$\{\mathbf{E}(f_0)\}$, and the orthogonal projection over this space is given by

$$\mathbf{a} = \frac{1}{N}\mathbf{E}(f_0)\mathbf{E}(f_0)^H\mathbf{d}. \tag{4.9}$$

Then, putting Eq. (4.9) into (4.8)

$$\begin{aligned} L(T, \mathbf{a}, \sigma_w^2) &= -\frac{N}{2}\ln(2\pi\sigma_w^2) - \frac{1}{2\sigma_w^2}\left(\|\mathbf{d} - \frac{1}{N}\mathbf{E}(f_0)\mathbf{E}(f_0)^H\mathbf{d}\|^2\right) \\ &= -\frac{N}{2}\ln(2\pi\sigma_w^2) - \frac{1}{2\sigma_w^2}\left(\|\mathbf{d}\|^2 - \frac{1}{N}\|\mathbf{E}(f_0)^H\mathbf{d}\|^2\right). \end{aligned} \tag{4.10}$$

For a fixed f_0, L is maximum when

$$\hat{\sigma}_w^2 = \frac{1}{N}\left(\|\mathbf{d}\|^2 - \frac{1}{N}\|\mathbf{E}(f_0)^H\mathbf{d}\|^2\right) \tag{4.11}$$

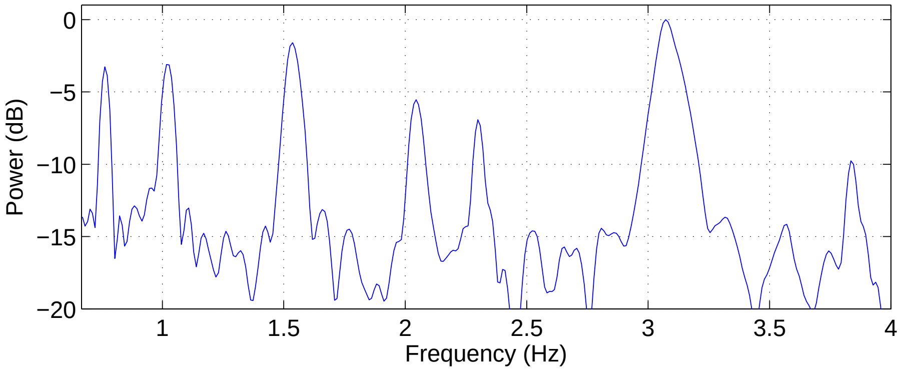


Figure 4.5: Periodicity profile computed via the spectral-sum method.

then

$$L(T) = -\frac{N}{2}\ln(2\pi e\hat{\sigma}_w^2).$$

This means that maximizing L with respect to f_0 is equivalent to minimizing $\hat{\sigma}_w^2$. For that purpose, we must maximize the second term on the right side of Eq. (4.11),

$$\|\mathbf{E}(f_0)^H\mathbf{d}\|^2 = \sum_{k=0}^{T-1}|\mathbf{e}_k(f_0)^H\mathbf{d}|^2 = \sum_{k=0}^{T-1}|D(e^{\jmath 2\pi k f_0})|^2 \tag{4.12}$$

where $|D(e^{\jmath 2\pi f})|^2$ is the PSD of $d(n)$. The last term of Eq. (4.12) represents the sum of the PSD values corresponding to integer multiples of f_0. For this reason this method is usually known as the "spectral sum". In this function, the predominant periodicities are visible as salient peaks. Since in practice we must avoid aliasing, the spectral sum is computed as follows

$$S(e^{\jmath 2\pi f}) = \sum_{k=1}^{K_{\max}}|D(e^{\jmath 2\pi k f})|^2 \tag{4.13}$$

where $K_{\max}$ is an upper limit who ensures that half the sampling frequency is not exceeded, *i.e.*, $fK_{\max} < \frac{1}{2}$.

Figure 4.5 shows a periodicity profile computed using the spectral sum method, it corresponds to the detection function of Figure 4.2-(a). The most salient periodicity is noticeable as the largest peak at approximately 3.1 Hz.

Spectral product This method is quite similar to the above mentioned spectral sum, the only difference consists in substituting the sum in Eq. (4.13) by a product, that is

$$P(e^{\jmath 2\pi f}) = \prod_{k=1}^{K_{\max}}|D(e^{\jmath 2\pi k f})|^2. \tag{4.14}$$

The spectral product is also a robust technique, Figure 4.6 shows the periodicity profile computed using this technique when applied to the detection function of Figure 4.2-(a). This method has the largest dynamic range among the four procedures presented.

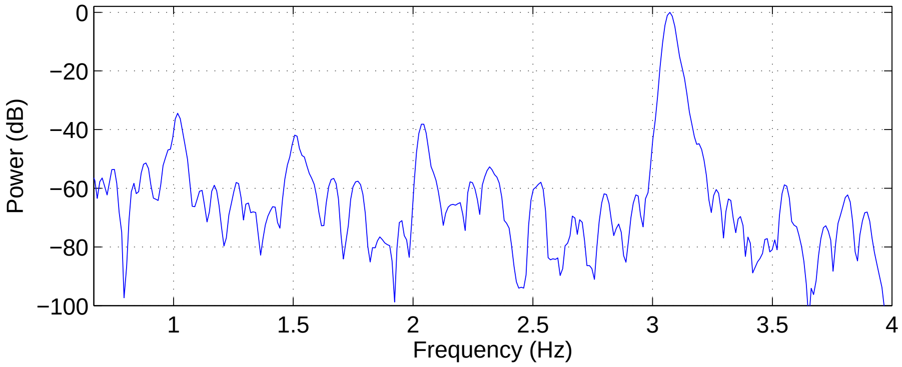


Figure 4.6: Periodicity profile computed via the spectral-product method.

4.1.2 Implementation

In the context of our analysis framework, the output of the sound-to-symbol conversion is a set of signals (see §3.4) corresponding to the band-wise detection functions of the harmonic ($d_p^s(n)$) and noise ($d_p^w(n)$) parts. The next step in the analysis consists in estimating the periodicities embedded in those signals using the methods presented in §4.1.1. For this purpose, the signals are processed as indicated in Figure 4.1.

For a given periodicity induction method, the procedure is repeated $2p$ times to account for the harmonic and noise detection functions in all subbands. Where $p \in [1, \dots, 8]$ if the uniform filter bank is used or $p \in [1, \dots, 5]$ if it is the logarithmic filter bank. After the periodicity induction in every channel has been calculated, profiles from all subbands are merged into a single periodicity vector. A detailed explanation is provided below.

Block-wise induction From the four periodicity induction methods described above, the ACF, the spectral sum (SS) and the spectral product (SP) operate in a block-wise fashion. More exactly, these methods use as input a series of contiguous and overlapping segments taken from the detection function. Figure 4.7 illustrates this process, the detection function[3] $d(n)$ is decomposed into a series of data vectors $\mathbf{u}_m$ where $m \in [0, \dots, M-1]$. Each vector has a length of ℓ samples and between two consecutive blocks there exists and overlapping of ρ samples. Then, periodicity induction for every block $\mathbf{u}_m$ is computed producing a vector signal

$$\mathbf{v}_m = \mathcal{T}\{\mathbf{u}_m\}$$

where $\mathcal{T}\{\cdot\}$ stands for any of the three methods mentioned above (ACF, SS, and SP). For the large majority of cases, rhythmic phenomena take place at relatively low-frequencies (<12 Hz, *i.e.*, periods beyond 0.08 s). For this reason, periodicity induction vectors $\mathbf{v}_m$ are trimmed, so that they only contain information in the frequency (or period) range of interest.

For every frequency channel in the harmonic or noise parts ($d_p^{s,w}$), we can see the output of the periodicity induction as a time–period (for the ACF) or time–frequency (for the SS and SP) matrix formed by the concatenation of the periodicity vectors:

$$[\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{M-1}]_{K \times M} \tag{4.15}$$

[3] Since this process is exactly the same for all $d_p^{s,w}(n)$, for the sake of simplicity we drop the subscript and superscript indexes.

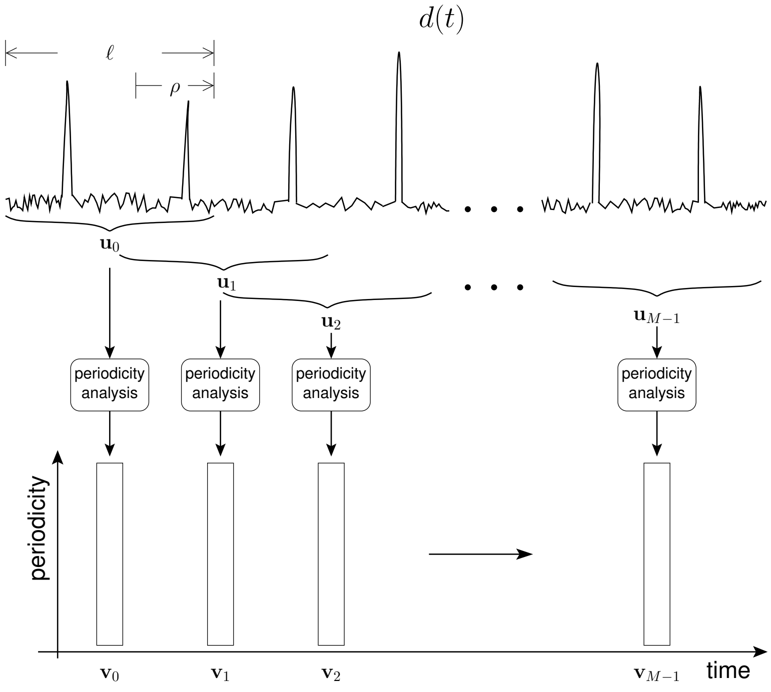


Figure 4.7: Block-wise decomposition and periodicity processing of $d(n)$.

where K indicates the number of frequencies or periods considered during the periodicity analysis.

Continuous induction Contrary to the rest of the methods, the bank of comb-filter resonators does not operate in a block-wise, but rather in a continuous fashion. In fact, this method provides better instantaneous *read-outs* about periodicity behavior than the others. There is a resonator filter $h_k(n)$ tuned for every periodicity we want to analyze and its respective output has the same sampling frequency as the input signal $d(n)$. Since the total number of resonators is in the order of a few hundred[4], the data size at the filter bank output seriously increases when compared to the input size. In order to reduce the data amount and to render this periodicity estimation method compatible with the afore mentioned block-wise procedures, the filter bank output is decimated by a factor $\frac{\ell}{\ell-\rho}$. This process is illustrated in Figure 4.8. Then, periodicity induction vectors are obtained by decimation as follows

$$\mathbf{v}_m = [\hat{y}_0(\hat{\ell}m), \hat{y}_1(\hat{\ell}m), \dots, \hat{y}_{K-1}(\hat{\ell}m)]^T \quad \text{for} \quad \hat{\ell} = \ell - \rho.$$

where the comb-filter output transformation $\hat{y}_k(n) = \mathcal{T}\{(d \star h_k)(n)\}$ is given by Eq. (4.6) and $k \in [0, \dots, K-1]$.

[4] To be more precise: in a "typical" situation a subband signal $d(n)$ is sampled at ≈200 Hz. Then, to cover the periodicity range from 0.1 s (600 BPM) to 1.5 s (40 BPM) at least 250 resonator filters are required for each subband. Based on this estimation, we can suggest that this technique has large storage memory requirements.

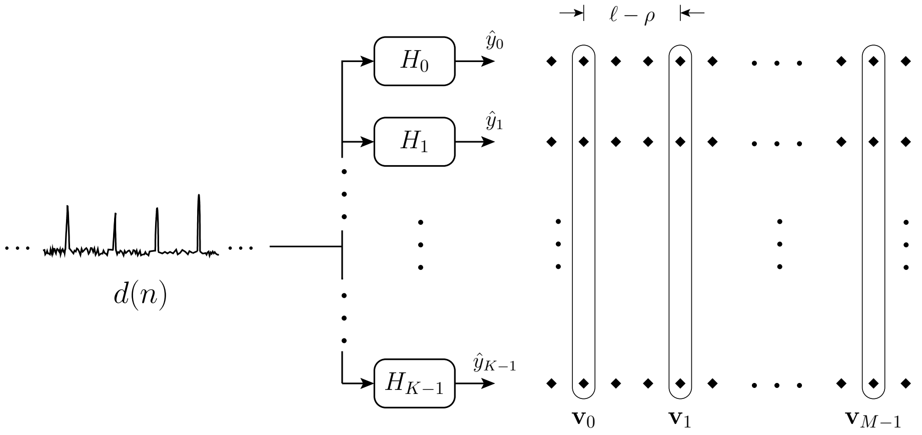


Figure 4.8: Continuous periodicity induction of $d(n)$.

Periodicity data fusion In practice, the periodicity estimation methods described before operate in a sequential (and causal) fashion. It means that at every time instant m, a total of $2P$ periodicity vectors $\mathbf{v}_p^{s,w}$ are computed, where $P \in [5, 8]$ depends on the filter bank in use and $p \in [0, \ldots, P-1]$. They represent the pulse salience of the $d_p^{s,w}$ different detection functions. The next part consists in merging all periodicity analysis profiles into a single vector providing comprehensive pulse salience information.

This operation is carried in a two step process. First, every periodicity vector coming from the harmonic and noise parts is normalized by its largest value and weighted by a *peakness* coefficient $c_{m,p}^{s,w}$ calculated over the corresponding $\mathbf{v}_{m,p}^{s,w}$.

It is important to notice that after normalizing and during the data fusion, all vectors have the same *weight* regardless of their effective contribution to the general periodicity profile. During the merge, if a subband signal (from the noise or harmonic part) does not contain any periodicity information (*i.e.*, it has a rather "flat" profile) it should not have the same weight of a subband conveying a larger amount of useful information. Therefore, the objective of this coefficient is to penalize flat profiles ($\mathbf{v_m}$ bearing little information) by a low weighting coefficient. On the contrary, a peaky profile leads to a high-valued coefficient. We obtain this peakness coefficient as $c = 1 - \phi$ where

$$\phi = \frac{\left(\prod_{k=1}^{K} v_m(k)\right)^{\frac{1}{K}}}{\frac{1}{K}\sum_{k=1}^{K} v_m(k)}.$$

Since the ratio of the geometric mean to the arithmetic mean is a flatness measure bounded to the region $0 < \phi \leq 1$, when $c \approx 1$ means that $\mathbf{v}_m$ has a peaked-shape. On the contrary, if $c \approx 0$, means it has a flat-shape.

The second step consists in adding information from all subbands coming from both harmonic and noise parts:

$$\gamma_m = \frac{1}{2P}\sum_{p=0}^{P-1} c_{m,p}^{s}\mathbf{v}_{m,p}^{s} + \frac{1}{2P}\sum_{p=0}^{P-1} c_{m,p}^{w}\mathbf{v}_{m,p}^{w} \tag{4.16}$$

where the superscript s and w on the right side indicate the harmonic and noise part respectively. Since this frame-wise process is repeated M times, then all the resulting γ_m

are arranged as the column vectors to form a periodicity matrix of size $K \times M$ as follows

$$\mathbf{\Gamma} = [\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_{M-1}]. \tag{4.17}$$

$\mathbf{\Gamma}$ can be seen as a *time–periodicity* representation of the pulsations present in the audio stream, since columns provide periodicity information while rows indicate the time position. Further analysis will be carried on this signal representation.

A natural question to ask is: why dealing with $\mathbf{\Gamma}$ and not selecting directly the most salient pulse periods by performing a frame-wise peak-picking operation on the periodicity profile? By adopting this approach it would be possible to throw away the rest of the periodicity information. At first sight it might seem more advantageous from the point of view of algorithm complexity. For example, by visual inspection of the periodicity profiles in §4.1.1 it is possible to identify the peak corresponding to the most salient period or frequency. In fact, we adopted this approach in (Alonso et al., 2004), by taking frame-wise decisions using only a single scalar for representing the periodicity profile at a given instant. While there is no noticeable difference when processing *strong beat* music, we found this approach more vulnerable when dealing with challenging instances. To render a meter analysis system more robust, we consider it is much preferable to avoid local decisions about pulse periods and instead gather periodicity salience information from a longer time-span prior to determining any period values. Additionally, with more periodicity data available it is possible to search for interconnections between metrical pulses at different levels.

4.2 Pulse-period tracking

At this point of the analysis, we have a series of metrical level candidates whose salience over time is registered in the columns of $\mathbf{\Gamma}$. The next stage consists in parsing through the successive columns to find at each time instant m the best candidates and thus track their evolution. During the present work we use the concept of Dynamic programming to carry out this task. In fact, this technique has been extensively used to handle many kinds of situations where sequential decisions are required. Actually, pulse-periods tracking in the context of rhythm analysis is not an exception, Laroche (2003), Peeters (2005) and Collins (2005b) have used this technique for solving the same task. Other techniques have also been proposed, for instance Hidden Markov models (Klapuri, 2004; Klapuri et al., 2006) and particle filters (Hainsworth & Macleod, 2003a; Hainsworth, 2003; Sethares et al., 2005).

4.2.1 Dynamic programming

As briefly mentioned, Dynamic Programming (DP) is a mathematical concept used for the analysis of sequential-decision problems. The basic principle has been applied throughout the history of mathematics, but DP was popularized as we know it today by the mathematician Richard Bellman in the early 1950's. *The Principle of Optimality*, as Bellman called the fundamental idea behind DP, has been applied since then to analyze hundreds of optimization problems in areas such as engineering and economics, among others. According to Silverman & Morgan (1990), the name of DP was probably affixed because, at that time, decision-making, mathematical methods which could be solved on early digital computers often had the word "programming" in their name, such as linear programming. Decision making for dynamical systems, or feedback decision making, probably implied ”dynamic programming”.

The principle of optimality Traditionally, DP has been used to find ”optimal” solutions to series of related events structured in a logical order. We consider an ”optimal” solution as one that minimizes or maximizes a performance or cost function. Optimization over time can often be regarded as ”optimization in stages”. In general, during optimization we must trade off our desire to obtain the lowest possible cost at a given present stage against the implication this would have for costs at future stages. The best action minimizes the sum of the cost incurred at the current stage and the least total cost that can be incurred from all subsequent stages, consequent on this decision. This is known as the Principle of Optimality.

Definition 4.1 (Principle of optimality). *An optimal sequence of decisions has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Bellman & Dreyfus, 1962).*

If the reader is interested, general purpose details about implementing the DP algorithm can be found in Rabiner & Juang (1993) and Cormen et al. (2001), in this part we provide the insights about its operational principle and then we adapt it to our context.

Our periodicity tracking problem amounts to selecting a sequence of track states $(\theta_m, \theta_{m-1}, ..., \theta_0)$ that maximizes a scoring function $\mathcal{S}$ reflecting the local pulse salience for different periodicity values. In the general case, an exhaustive brute-force search over all possible candidates has a too large computational cost and in some cases it is even intractable. Our DP approach formulates the problem as a first-order Markov process, *i.e.*, the scoring function is separable into the sum of functions

$$\mathcal{S}(\theta_m, \theta_{m-1}, \ldots, \theta_0) = \mathcal{S}_m(\theta_m, \theta_{m-1}) + \mathcal{S}_{m-1}(\theta_{m-1}, \theta_{m-2}) + \cdots + \mathcal{S}_1(\theta_1, \theta_0) \tag{4.18}$$

where the set θ_m represents the pulse-salience state vectors at each observation interval m. Because each function $\mathcal{S}_m(\theta_m, \theta_{m-1})$ depends only on a subset of the pulse-salience state variables, the optimization can be carried out in stages. In this case, the result of each progressive stage depends only on the results of the previous stage (the Markov process principle), a single reduced-dimension function (which we will call later *local constraint*), and the current pulse-salience observations. This procedure forms the basis for DP, which can be summarized by the nested expression

$$\tilde{\mathcal{S}}(\tilde{\theta}_m, \tilde{\theta}_{m-1}, \ldots, \tilde{\theta}_0) = \max_{\theta_m}\left[\max_{\theta_{m-1}}\left[\mathcal{S}_m(\theta_m, \theta_{m-1}) + \max_{\theta_{m-2}}\left[\mathcal{S}_{m-1}(\theta_{m-1}, \theta_{m-2}) + \cdots + \max_{\theta_0}\left[\mathcal{S}_1(\theta_1, \theta_0)\right]\right]\right]\right] \tag{4.19}$$

where $\tilde{\mathcal{S}}$ denotes the maximum achievable value of the track-scoring function, and the set of $\tilde{\theta}_t$ are the state values (*i.e.*, pulse-saliences) producing this maximum.

The DP maximization procedure is carried out by forming a sequence of intermediate one-dimensional functions $h_{t-1}(\theta_t)$ that represent the maximum partial sum (or score) of $\mathcal{S}_l(\theta_l, \theta_{l-1})$ for $l = 1, 2, \ldots, t$. Beginning at the innermost nesting level, h_0 is created by

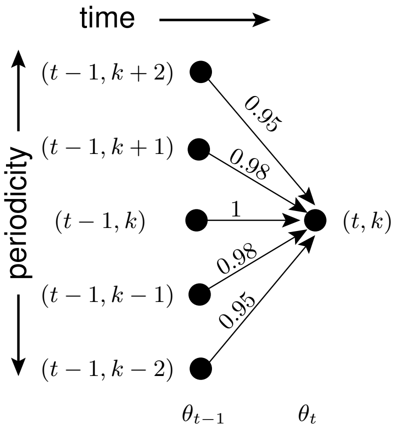


Figure 4.9: Dynamic Programming local constraint for path tracking.

locating the maximum value of θ_0 that maximizes $\mathcal{S}_1$ for each possible discrete value of θ_1:

$$h_0(\theta_1) = \max_{\theta_0}[\mathcal{S}_1(\theta_1, \theta_0)]. \tag{4.20}$$

Both the maximizing value of θ_0 and the corresponding maximum value $h_0(\theta_1)$ are stored for each θ_1. In the second and every subsequent t-th DP processing stage, the value of θ_{t-1} that maximizes the partial sum, along with the achieved maximum value, is chosen and stored for each possible value of θ_t:

$$h_{t-1}(\theta_t) = \max_{\theta_{t-1}}[h_{t-2}(\theta_{t-1}) + \mathcal{S}_t(\theta_t, \theta_{t-1})]. \tag{4.21}$$

At some stage m where a decision is required, the end-state θ_m providing the optimal solution is found by maximizing h_{m-1} over all candidate θ_m:

$$\tilde{\mathcal{S}}(\tilde{\theta}_m, \tilde{\theta}_{m-1}, \ldots, \tilde{\theta}_0) = \max_{\theta_m}[h_{m-1}(\theta_m)]. \tag{4.22}$$

In our implementation, using $\mathbf{\Gamma}$ as the DP input data (see Eq. (4.17)), the maximization procedure is performed M times, with only $\widehat{K}$ candidate state transitions evaluated (as explained below) for each of the K possible states (potential pulse-saliences) in θ_t. The computational load for DP is of $K\widehat{K}M$ operations, where $\widehat{K} \ll K$.

4.2.2 Finding and tracking the best paths

During the implementation, we restrain the range of variation of the metrical pulses. This restriction is based on the fact that in conventional music the periods of the metrical levels generally vary slowly in time, and *sharp* speed or rhythm transitions are less frequent. For this reason, instead of evaluating (with a higher computational cost) $K \times K$ potential transitions when going from a state in θ_{t-1} to a state in θ_t, we only allow a total of $K \times \widehat{K}$ potential transitions. Moreover, from those $\widehat{K}$ allowed transitions for each possible state in θ_{t-1}, we introduce a penalization that very lightly discourages period variations and favors horizontal paths (*i.e.*, metrical levels with constant periods).

In practice we deal with the columns of $\mathbf{\Gamma}_{K\times M}$, given by Eq. (4.17), as the observations θ_t. Motivated by heuristical reasons, we have implemented $\mathbf{\Gamma}$ such that a displacement of

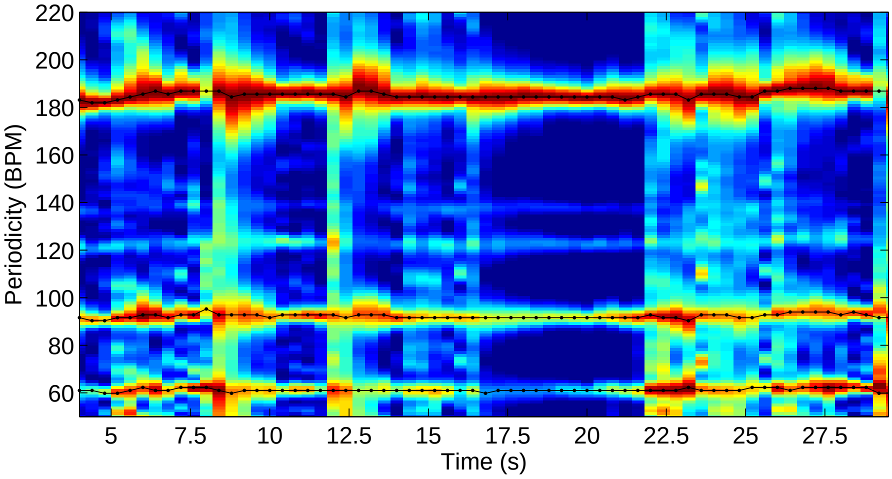


Figure 4.10: Tracking of the three most salient pulsation paths for the song *Le bruit du frigo* (by the french pop-rock group "Mano Negra"). The relationship between periodicity paths is $1 : \frac{3}{2} : 3$.

one position in the periodicity (vertical) axis corresponds to about 1 BPM, the exact value depending on the method used to estimate the periodicity. Based on this considerations, in practice we control the transitions going from the state θ_{t-1} to the state θ_t using the constraint illustrated in Figure 4.9. That is, we fix $\widehat{K} = 5$ and we only allow any sudden *accelerando* or *ritardando* of slightly above 2 BPM between two successive time instants[5].

In addition, the DP stage has been designed to track not only the best path, but also others. Inside the tracking algorithm, the *best* path is seen as the most *energetic* trajectory from $t = 0$ to $t = M$. To search for a second and other paths we impose the following restriction: any path sharing segments with or being too close ($<$ 10 BPM) to another more energetic paths is pruned. It is possible to iterate upon this restriction and this operation will outline other *secondary* high-energy paths, which are (practically always) related to the most energetic path by a rational factor.

Next we present two examples of how the DP algorithm detects and smoothly tracks the periodicity paths while avoiding abrupt transitions. This examples were calculated using the SS method, with an analysis window of size $\ell = 4$ s and $\rho = \frac{7}{8}\ell$, that corresponds to a time step of 0.5 s between consecutive observations. In addition, a displacement of one position in the vertical axis corresponds to an increase/decrease of 1.22 BPM in the periodicity index.

First we present a pop-music example, the music signal has a $\binom{3}{4}$ time-signature. The name of the song under analysis is *Le bruit du frigo* performed by the french group "Mano Negra", it is the same signal that we used for the detection function example back in Figure 4.2-(a). The annotated tempo for this excerpt is 185 BPM, additionally for this particular example the tactus and the tatum are represented by the same metrical level. In Figure 4.10 we can see the respective time–periodicity matrix, where strong pulse salience

[5] In practice, there exists no universal step size. This value entirely depends on the nature of the signal under analysis. If it has a fairly stable tempo, large windows ($\ell > 6$ s and $\rho < \frac{1}{2}\ell$) can be used, *i.e.*, a time step of 3 seconds. If the tempo is likely to change, smaller values should be employed (*e.g.*, $\ell \approx 4$ s and $\rho < \frac{7}{8}\ell$), corresponding to a time step around 0.5 s.

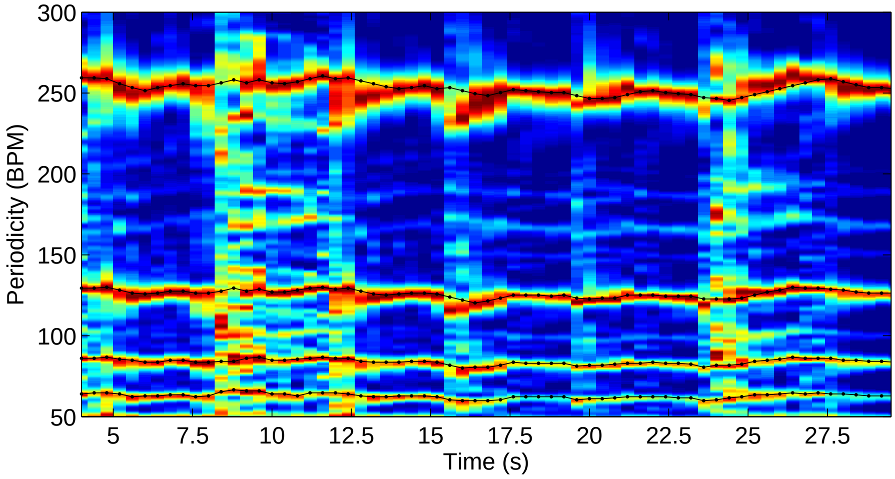


Figure 4.11: Tracking of the three most salient pulsation paths for Mozart's *Rondo Alla Turca*. The relationship between levels is $1 : \frac{4}{3} : 2 : 4$

locations are indicated by red tones and dark-blue zones indicate the periods for which there exists no activity. In the upper part of the figure, pictured in black-trace we can see the most energetic path detected by the DP algorithm. During most part of the analysis it perfectly coincides with the annotated tempo and only exhibits small fluctuations. In a similar way, the DP algorithm identifies the second most salient periodicity at about 62 BPM and the third most salient at 92 BPM. From the bottom path to the top, the relation between the periodicity levels is $1 : \frac{3}{2} : 3$.

The second example is a more challenging case, the music signal has a $\binom{2}{4}$ time-signature and its name is *Sonate Opus KV–331 Rondo Alla Turca* from W. A. Mozart. As in the previous case, we analyze an excerpt of 30 s length, but opposite to the earlier example this one has a tempo that varies through time, as can be seen from the time–periodicity matrix of Figure 4.11. The average of the tempo annotation for this excerpt is 125 BPM. By visual inspection we could expect that the most salient periodicity is the thick red-line, in the upper part, oscillating around 254 BPM. Nevertheless the DP algorithm considers it is the thinner and less energetic path with average periodicity of 126 BPM. The reason is that the tracking algorithm *hooks-up* better with this trajectory since the other one displays large variations (for example at 8 s, 15 s, 25 s) that it cannot follow. And in fact, this oscillating path is second most salient periodicity. The third one is located at 84 BPM and the fourth most salient at 63 BPM. From the bottom path to the top, the relation between the periodicity levels is $1 : \frac{4}{3} : 2 : 4$.

4.2.3 Selecting a periodicity path as the tempo

At this point of the analysis, we have a set containing the most energetic periodicity paths and their respective salience. The next step consists in estimating which is the "best" tempo candidate from this list. For this purpose, we use an approximate *a priori* distribution of beat period hypotheses as a weighting curve to find the most likely beat period , *i.e.*, we multiply the salience of each element in the periodicity path by a value which directly depends on the beat period, then the path having the largest value is considered as the actual tempo. This method has been previously used in the literature and a number of weighting curves have been proposed as shown below.

According to Moelants (2002), the borders of the "existence region" of tempo lie between 40 and 300 BPM (periods between 200 and 1500 ms). However, Moelants points that these numbers are approximate and should be interpreted as labels for a small 'tempo zone'. Determining the borders exactly is difficult, mainly because the transition is not abrupt, but gradual: tempi near the edges of the existence region are still not so easy to perceive or to perform. This implies that somewhere in the middle there must be a point or a zone where tempo perception is optimal, the so-called *preferred tempo*. Based on his experiments, Moelants suggests that the preferred tempo is close to 120 BPM. Then, using as premise the existence of this optimal region for periodicity perception, he develops a resonance model for pulse perception which is given by the expression:

$$W_R(f) = \frac{1}{\sqrt{(f_0^2 - f^2)^2 + \beta f^2}} - \frac{1}{\sqrt{f_0^4 + f^4}}, \tag{4.23}$$

where W_R is the effective resonance amplitude, f is the beat frequency (in Hertz) f_0 is the resonant frequency (or preferred tempo, also in Hertz) and β is a constant which controls the damping of the function: with higher damping the distribution gets broader, the peak becomes less prominent and slightly moves to a slower tempo than the actual resonance frequency. Figure 4.12 shows an example where W_R has been plotted for $\beta = 1$ and $\beta = 4$ for $f_0 = 125$ BPM.

Parncutt (1994) models prior distribution of beat period hypotheses as a two-parameter log-normal distribution given by

$$W_L(\tau) = \frac{1}{\tau\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}\left(\ln\left(\frac{\tau}{\mu}\right)\right)^2} \tag{4.24}$$

where τ is the beat period, σ and μ are the shape and scale parameters respectively. Figure 4.12 presents the beat frequency aspect[6] of this weighting curve for $\mu = 0.55$ s (109 BPM) and $\sigma = 0.28$ s. The value of these parameters were estimated by Klapuri et al. (2006) by fitting the log-normal distribution to the TUT hand-labeled test database (see §2.5.2).

Davies & Plumbley (2005, 2006) use as weighting function the Rayleigh distribution function

$$W_G(\tau) = \frac{\tau}{\mu^2} e^{-\frac{\tau^2}{2\mu^2}} \tag{4.25}$$

where the μ parameter sets the strongest point of the weighting and τ indicates the beat period. Figure 4.12 shows the beat frequency aspect of W_G for $\mu = 0.48$ s (125 BPM).

As seen from Figure 4.12, all these weighting functions have highly resembling shapes. Nevertheless, we opted for the resonance model developed by Moelants (2002) with $\beta = 0.4$, since it has proved to have a close fit to the perception of tempo in a wide variety of cases (McKinney & Moelants, 2004).

4.3 Beat location

Throughout the previous sections of this chapter we have solely focused on estimating the periodicities of the acoustic events that constitute the audio signal. In practice it is

[6] Notice that it is not the Fourier transform of $W_L(\tau)$, but a plot in terms of beat frequency (in BPM) instead of the beat period.

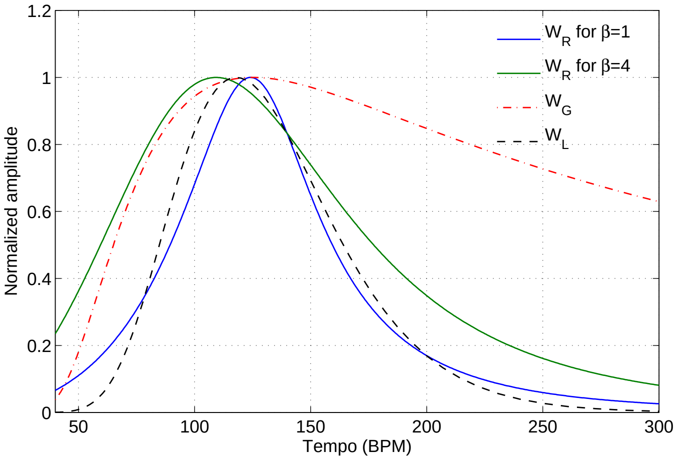


Figure 4.12: Weighting curves representing the prior distribution of beat period hypotheses.

fundamental to know not only the rate at which these events occur, but also to identify their precise location (see for example Figure 2.1). This operation is usually known as *beat phase location* or simply *beat location*. During our development, we suppose that a specific pulse period (*e.g.*, the tempo) has been selected from the list of periodicities obtained in §4.2.3 and we use this information to locate the instants where the related acoustic events occur or *should* occur. Expressed in pragmatic terms: given an audio signal and a selected periodicity level, our intention is to play the inverse role of a metronome by placing a *beep* at every time instant where a musical onset is detected *if* there exists one nearby, or simply mark the position where it would have had place in case there is no acoustic event *detected* in neighboring zone. Furthermore, positions are always calculated as the elapsed time from the beginning of the analysis until the instant where an event has *or* should have place.

We have developed two different methods to perform beat location. The first one is causal, has a minimal computational complexity and is based on the idea of *predicting* beats. The second one is non-causal, has a higher computational cost than its above mentioned counter part and is based on the idea of time-alignment of data sequences.

Although it is possible to carry out the beat location for every detection function in the set (remember that we have a total of $2P$ subband signals) and then fusion the results, based on our heuristical experience we consider that it is not necessary. On the contrary, band-wise beat location would increase the already non-neglectable computational load of our system. Thus, we obtain a new signal by summing the contributions of the detection functions from all subbands in the harmonic and noise parts, that is:

$$d(n) = \sum_{p=0}^{P-1} (d_p^s(n) + d_p^w(n)). \tag{4.26}$$

Both of the methods described in this section use $d(n)$ as input signal. In addition, we call $\mathcal{T}$ the period at which the beat location is performed.

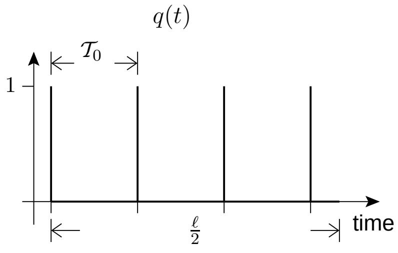


Figure 4.13: Beat period comb $q(n)$.

4.3.1 Causal beat-location

This proposal is based on a cross-correlation between a comb-like signal and the detection function. A rather similar approach to find beat locations was independently developed by Davies & Plumbley (2004). In addition, this model also bears resemblance to the beat-tracking system developed by Laroche (2003). For the development, the signal analysis is carried out in a block-wise fashion.

This method mainly targets signals with strong and stable beat, thus it uses fairly large analysis windows ($\ell \geq 5$ s and an overlapping factor $\rho < \frac{1}{2}\ell$), since $\mathcal{T}$ is not likely to change abruptly between successive frames.

First, we create an artificial pulse-train $q(n)$ of length $\frac{1}{2}\ell$ and tempo $\tau_0 = \frac{60}{\mathcal{T}_0}$, where $\mathcal{T}_0$ is the beat period for the first analysis window. Figure 4.13 depicts the aspect of $q(n)$. Next, we compute the cross-correlation between this comb-signal and the first analysis window $\hat{d}(n)$, that is:

$$r_{\hat{d}q}(n) = \sum_{\tau} \hat{d}(\tau)q(n+\tau). \tag{4.27}$$

Let n_0 be the time-index where this cross-correlation is maximal, from now we will consider as the initial beat location. Based on the premise that the tempo of the signal under analysis is fairly constant, for the second and successive beats in the j^{th} analysis window a beat period $\mathcal{T}_j$ is added to the previous beat location, *i.e.*, $n_i = \lfloor n_{i-1} + \mathcal{T}_j \rfloor$, for $i = 1, 2, \ldots$, $j = 1, 2, \ldots$ and a corresponding peak in $q(n)$ is searched within the *expectancy* region $n_i \pm \Delta$. We consider that $\Delta \approx 0.15\mathcal{T}_j$ is a reasonable value to tolerate non-intentional deviations. If no peak is found, the beat is placed in its expected position n_i. When the last beat in the current analysis windows occurs, its location is stored in order to assure the continuity with respect to the first beat of the new analysis window. If the tempo of the new analysis window ($\mathcal{T}_{j+1}$) differs by more than 10% with respect to the previous value, *i.e.*, $\frac{|\mathcal{T}_{j+1} - \mathcal{T}_j|}{\mathcal{T}_j} \geq 0.1$, a new "initial" beat is estimated using the same process described above. The subsequent beat locations are searched using the new beat period and no reference is made to the previous value.

Figure 4.14 presents an example of this method, the excerpt under analysis was taken from *Le bruit du frigo*. From top to bottom, the first plot shows the cross-correlation between $\hat{d}(n)$ (blue trace) and $q(n)$ (red) trace). In the second one, the initial beat phase location (marked with a black line) is used to detect the regularly spaced beats (marked as red diamonds), the last location (at about 4.8 s) is stored. The bottom plot uses last beat location in the previous analysis window as the initial phase (marked with a black line) who is used to detect the upcoming beats (marked with a black line). It can be seen that one beat was *missed* at about 7 s, because the peak was located off the search region.

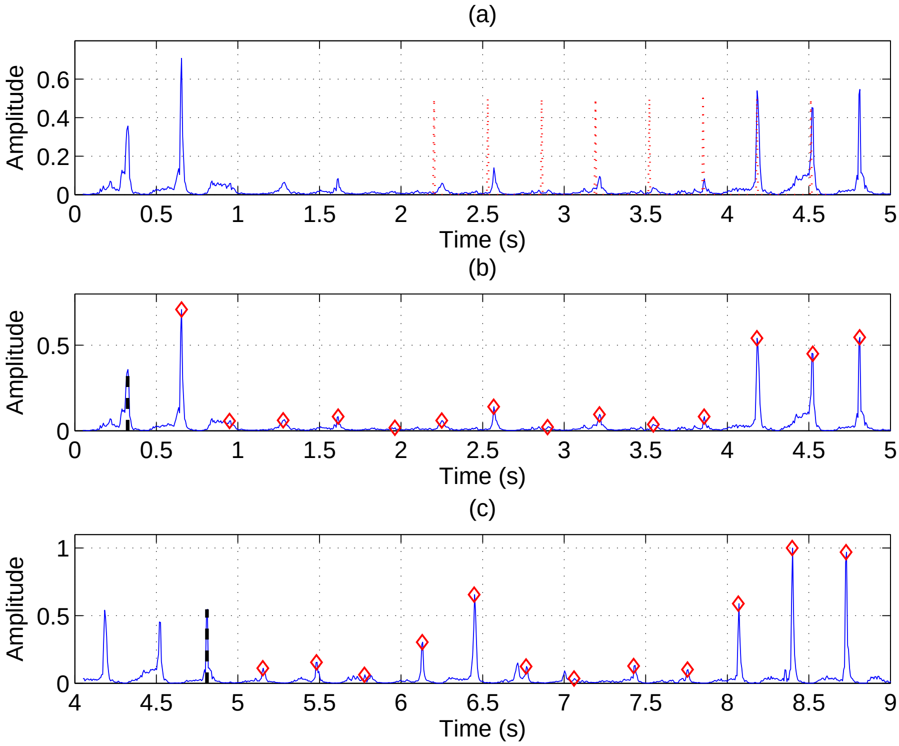


Figure 4.14: Causal beat location example using the signal *Le bruit du frigo*. (a) Cross-correlation between $d(n)$ (blue trace) and $q(n)$ (red trace). (b) Initial beat phase location (black trace) and detected beats (red trace). (c) Second analysis frame: initial beat phase (black trace) and detected beats (red).

Nevertheless the algorithm hooked up at the next beat position.

This method has the advantage of having a very low computational complexity yet it displays a rather good performance if the required conditions are met. On the contrary, it lacks of robustness if they are not properly satisfied. For this reason we decided to implement another method more robust to tempo variations.

4.3.2 Non-causal beat-location

This method is based on the idea of aligning two data series. For this purpose we *import* a technique called Correlation Optimize Warping[7] (COW) which was originally proposed by Nielsen et al. (1998) in the context of chromatography[8] as an analysis tool to compare the profiles of chemical substances. In fact, the visual resemblance between our detection functions and chromatographic profiles is remarkable, see Figure 4.15 for an example of the alignment of two chromatographic profiles. This technique can be seen as a high-performance block-wise Dynamic Time Warping (DTW). In fact, Tomasi et al. (2004) conduct a functional comparison of both methods and they argue that COW is much more efficient in computational terms since it has a smaller search space.

[7] The source code of this algorithm is available in the author's website.

[8] Chromatography is a common form of mass spectrometry which is often used in physical and analytical chemistry for the identification of substances, through the spectrum emitted or absorbed.

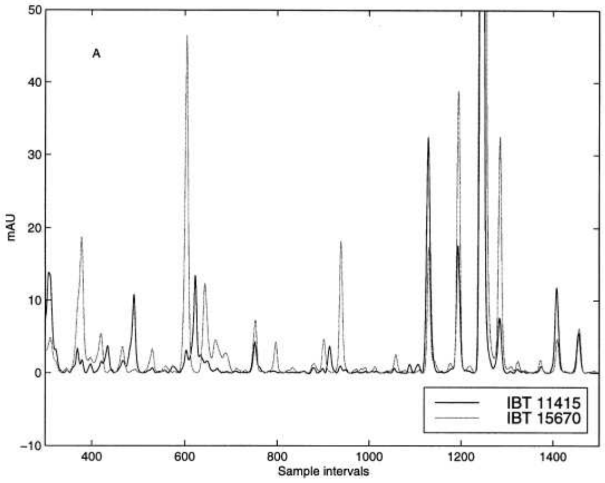


Figure 4.15: An example using Correlation Optimized Warping (COW) to align two chromatographic profiles (reproduced from (Nielsen et al., 1998)).

In our context, the alignment involves $d(n)$ as one of the sequences and the other one is the pulse train $q(n)$, where $\mathcal{T}$ is the distance between two consecutive pulses and is considered to be constant. Both signals have the same length and our goal is to independently move every "rigid" pulse in $q(n)$ and to match it with a beat pulse in $d(n)$.

Initialization Since the COW algorithm has been designed to match shapes, we must create a pulse train $\hat{q}(n)$ whose pulses look like the beat pulses in $d(n)$. The best way to assure the likeness is by extracting the pulse shape from $d(n)$, hence let $g(n)$ be the average pulse-shape as depicted in Figure 4.16, extracted from 20 s of *Le bruit du frigo*. We consider that $g(n)$ has a width of approximately 11 samples ($\approx$ 60 ms), since any additional samples are close to zero and it is important to keep the pulse-width as short as possible. Then we form $\hat{q}(n)$ as

$$\hat{q}(n) = \sum_{k=0}^{K_{\max}} g(n) \star \delta(n - k\mathcal{T}), \tag{4.28}$$

where $K_{\max}$ is merely an upper limit to ensure that the length of $d(n)$ is not exceeded. The time warping algorithm is based on an optimization procedure. If we manage to bring nearer the pulses in $\hat{q}(n)$ to the pulses in the detection function, we will provide a good initialization clue to the warping algorithm[9]. Let $q(n)$ be the time-shifted version of $\hat{q}(n)$ whose pulses best match those in $d(n)$, and where both signals have the same length. Now we are ready to start warping individually each pulse in $q(n)$ to match one (*if* it exists) in $d(n)$.

Correlation Optimized warping The COW method (Nielsen et al., 1998) aims at aligning two data sequences by piecewise linear stretching and compression (better known

[9] It can be done by computing cross-correlation between $d(n)$ and $\hat{q}(n)$. In fact it is only necessary to calculate this correlation in one period, *i.e.*, $0 \leq \tau < \mathcal{T}$.

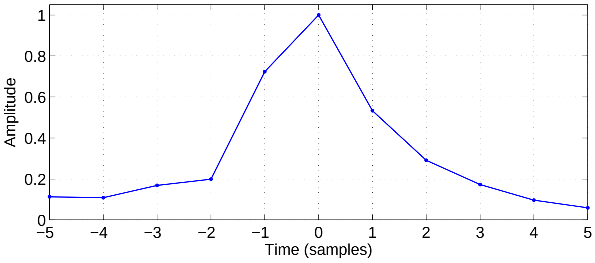


Figure 4.16: Average pulse-shape in $d(n)$.

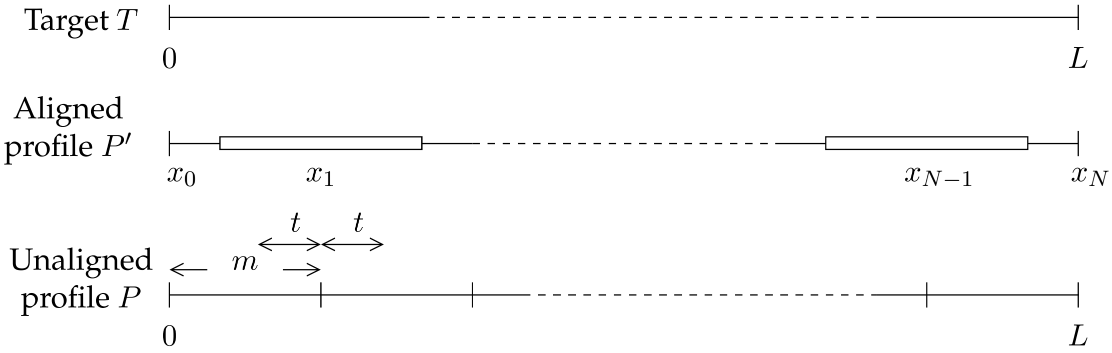


Figure 4.17: Schematic presentation of the warping problem.

as warping) of the time axis of one of the signals. Aligning by piecewise linear warping involves dividing the signals into a number of blocks that are each warped linearly. Because the number of sections and the number of ways each section can be warped is finite, the number of possible solutions is also finite. The algorithm sets up the question of finding the optimal warpings as a combinatorial optimization problem, which is later solved by DP.

For the sake of clarity, the same notation from (Nielsen et al., 1998) will be used in this part. For the rest of the explanation $d(n)$ will be known as the *target signal* T, $q(n)$ as the profile P to align with it, yielding the aligned profile P' . We suppose that P and T have $L+1$ data points and L sample intervals. These signals can be divided into sections of length m as shown in Figure 4.17. The number of sections is $N = \frac{L}{m}$. Each section may be warped to a smaller or greater length. The end-points of the sections are referred to as the nodes and the position of the starting point of section i after warping is denoted x_i. Node 0 is the starting point of the first section ($x_0 = 0$) and node N is the end point of the entire profile after warping ($x_N = L$). The warpings examined consist of the integer values from 0 to t, where t will be referred to as "the slack" (see Figure 4.17).

Determining the optimal alignment is now a question of finding the optimal combination of warpings of the N sections where no section may be warped more than t sample intervals. The warping of section i is called u_i. The quality of the alignment is determined separately for each section i by calculating the correlation coefficient[10] ρ between section i after warping and the corresponding segment of the target. Nielsen et al. (1998) justifies

[10] The correlation coefficient $\rho(x, y)$ indicates the strength of the linear relationship between two data sets

the use of the correlation coefficient arguing that it provides a good measurement of the covariations in data sets, thus making of it a good choice for calculations of similarity. The alignment quality function (ρ in this case) is termed the benefit function f, using the short notation $f(I) = \rho(I_P, I_T)$, where I denotes an interval between two node positions. By defining the optimal combination of warpings as the one that gives the largest value of the summed correlation coefficients, this problem now directly solvable by DP.

The optimal combination of warpings defines the optimal set of node positions after warping $\boldsymbol{x}^*$. The problem is then described by the constraint intervals $x_0 = 0 < x_1 < \cdots < x_{N-1} < x_N = L$ and $u_i \in [-t; t]$, $i = 0, \ldots, N-1$, then for all $x_{i+1} = x_i + m + u_i$

$$\begin{aligned} \boldsymbol{x}^* &= \arg\max_x \left(\sum_{i=0}^{N-1} f([x_i; x_{i+1}]) \right) \\ &= \arg\max_x \left(\sum_{i=0}^{N-1} \rho(P'[x_i; x_{i+1}], T[x_i; x_{i+1}]) \right). \end{aligned} \tag{4.30}$$

This problem is then solved using a DP algorithm to examine all possible combinations of the variables in a rational fashion: for each section i the optimal warping u_i is calculated for possible node x_i.

Once the pulse alignment between both signals has finished, we can detect the beat locations by just searching the closest peak (within a small range) in T to every pulse in P'. In case there exists no acoustic event at given position, the beat location will still be marked due to the pulse spacing constraints. One way to reduce the computational complexity and to speed-up calculations is by conducting a block-wise alignment. This operation reduces not only reduces the search space in the DP algorithm, but also allows larger tempo deviations.

Below, we present an alignment example. To compute it, the section size was set to $m = 15$ and the slack to $t = 3$. The intention of these values is to favor similarities between blocks who contain whole pulses.

Figure 4.18 shows an example with the same signal used in the causal location case. The top plot shows the initialization step. We can see that some pulses are practically aligned (towards the end), while others are in between. In the bottom shows that all pulses in both signals are perfectly aligned.

4.4 Tatum estimation

In Chapter 1 we highlighted that the tatum can be a perfect short-time musical unit for segmentation and analysis purposes. As a reminder, the term tatum refers to the lowest metrical level, *i.e.*, a regular pulse train that a listener intuitively infers from the timing of the musical events. For Bilmes (1993b) and Gouyon et al. (2002), it is roughly equivalent to the time division that most highly coincides with note onsets: a sort of trade-off between how well a regular grid explains the onsets, and how well the onsets fit to that

x and y of n points (Weisstein, 1999). It is calculated as:

$$\rho(x, y) = \frac{(\sum xy - n\bar{x}\bar{y})^2}{(\sum x^2 - n\bar{x}^2)(\sum y^2 - n\bar{y}^2)}. \tag{4.29}$$

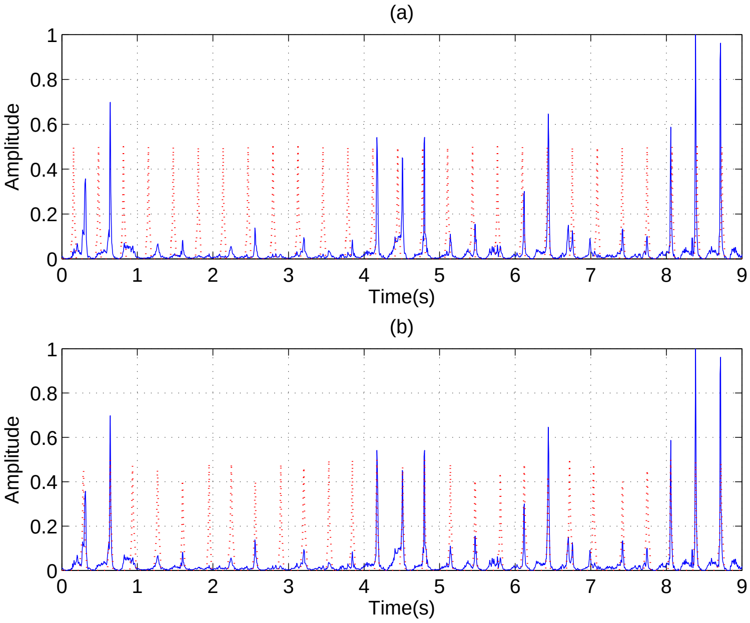


Figure 4.18: Non-causal beat location by aligning two data sequences. The signal under analysis is *Le bruit du frigo*. (a) Non-aligned signals P (red dotted trace) and target signal T (blue line trace). (b) Output of the COW, aligned profile P' (red dotted trace) and T (blue line trace).

grid. In the last part of our research we have focused on developing a tatum estimation system. Although the technique described below is still preliminary work, we have decided to include it in this report since it has produced encouraging results.

A number of tatum estimation techniques have been proposed in the literature. One of the first algorithms was proposed by Seppänen (2001b), it uses a time-varying IOI histogram with an exponentially decaying window for past data, enabling the tracking of accelerandos and ritardandos. Gouyon et al. (2002) proposed later a comparable system also based on an adaptive IOI histogram, in both cases the tatum period is found by calculating the greatest common divisor (GCD) integer that best estimates the histogram harmonic structure. To find the tatum, Gouyon et al. (2002) and Uhle et al. (2004) introduced a two-way mismatch (TWM) error procedure which was originally proposed for the estimation of the fundamental frequency in audio signals (Maher & Beauchamp, 1994). The principle of this method is as follows: two error functions are computed, one that illustrates how well the grid elements of period candidates explain the peaks of the measured histogram, another one illustrates how well the peaks explain the grid elements. The TWM error function is a linear combination of these two functions. The proposal of Klapuri et al. (2006) is based on a comb-filter resonator and a probabilistic back-end[11]. The method developed by Jehan (2005a) is built around a moving autocorrelation computed on the detection function. To refine that estimation and detect the phase, the moving autocorrelation is aligned against a set of templates (or patterns).

In this section, we introduce a non-causal technique to calculate the tatum period. We suppose that the tempo of the instance under analysis is stable during the tatum

[11] This method jointly computes three metrical levels: the measure, the tactus and the tatum.

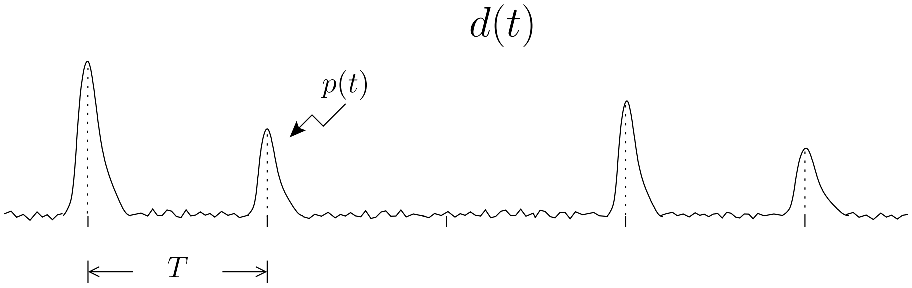


Figure 4.19: Detection function $d(t)$ seen as digital message.

estimation procedure. Our proposal considers the problem of estimating the tatum as peculiar digital communications situation. In fact, the detection function can be modeled as a baseband Pulse-Amplitude Modulated (PAM) signal. In addition, let us suppose that the detection function is a continuous time signal $d(t)$ where $t \in \mathbb{R}$. This idea is illustrated in Figure 4.19 and can be explained by the expression

$$d(t) = \sum_n a_n \, p(t - nT), \tag{4.31}$$

where $\{a_n\}$ is an arbitrary and stationary data sequence, $p(t)$ is the elementary pulse-shape and T is the tatum period. Contrary to the traditional case in digital communications, our goal is not to detect the data sequence but to determine the rate $(\frac{1}{T})$ at which the information is transmitted. We must point that this model represents a reductive panorama of the actual situation in metrical analysis: the elementary pulse-shape ($p(t)$) representing the onsets is unknown and may vary from attack to attack; in addition, this scheme does not take into account the presence of additive and impulsive (false onsets) noise. However, we consider it is adequate enough to estimate the tatum period.

The spectral occupancy and composition of many digital signaling techniques have attracted much attention in the literature. These spectral properties can be derived from the knowledge of the Power Spectral Density (PSD) of the line code. In Win (1998); Proakis (2000) and Simon et al. (1995), it has been shown that the PSD, $S_d(f)$, of Eq. (4.31) is formed of continuous as well as discrete components

$$S_d(f) = S_d^c(f) + S_d^d(f), \tag{4.32}$$

where the continuous part is

$$S_d^c(f) = \frac{1}{T}|P(f)|^2(R_a(0) - \mu_a^2) + \frac{2}{T}|P(f)|^2 \sum_{n \neq 0} [R_a(n) - \mu_a^2] \cos(2\pi n T f), \tag{4.33}$$

where $|P(f)|^2$ is the PSD of the pulse-shape $p(t)$, μ_a and $R_a(n)$ are the mean and autocorrelation function of the sequence $\{a_n\}$; and the discrete part is given by

$$S_d^d(f) = \frac{1}{T^2} \sum_n \left| P\left(\frac{n}{T}\right) \right|^2 \mu_a^2 \, \delta\left(f - \frac{1}{T}\right). \tag{4.34}$$

It is difficult to manipulate the pulse-shape $p(t)$, since it entirely depends on the nature of the audio signal and it is susceptible to change from attack to attack for a given signal. For this reason we opted for an easier task: control the $\{a_n\}$.

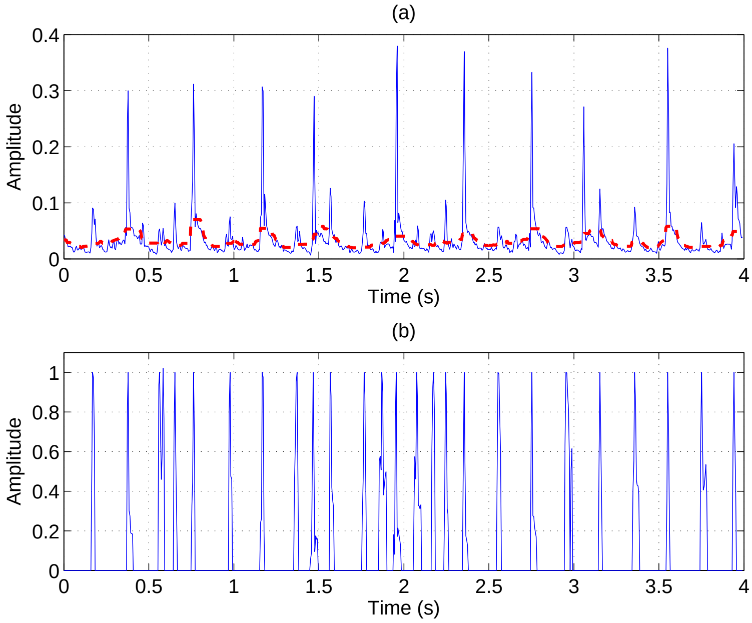


Figure 4.20: Detection function transformation. (a) Input signal (blue trace) $d(n)$ and (in dashed red-trace) the adaptive threshold $\theta(n)$. (b) the uniform-amplitude pulse train $\widehat{d}(n)$.

In fact, one of the reasons which makes the tatum rather difficult to detect is that short notes (bearing the tatum) are less energetic than notes having a larger time-span. For that reason, to ease the tatum estimation we have decided to reinforce the low-energy short notes and to level all of the acoustic events at a single amplitude level.

Now, let us consider again the detection function as a discrete signal $d(n)$ where $n \in \mathbb{Z}$. The first step is to distinguish the true onset peaks in $d(n)$. For that purpose, we assume that the noise and the unwanted peaks are considerably smaller in amplitude compared to true note attack peaks. A peak-picking algorithm that selects peaks above a dynamic threshold calculated with the help of a median filter is a simple and efficient solution to this problem. This solution (very similar to that used in §3.2 to whiten the audio signal) has already been proposed in the literature to solve this problem (Bello, 2003). The median filter is a nonlinear technique that computes the pointwise median inside a sliding window of length $2L+1$, formed by a subset of $d(n)$. The median threshold curve $\theta(n)$ is given by the expression

$$\theta(n) = \mathcal{C} \cdot \text{median}(g_m) \tag{4.35}$$

where $g_m = \{d_{n-L}, \ldots, d_n, \ldots, d_{n+L}\}$ and $\mathcal{C}$ is a predefined scaling factor to artificially rise the threshold curve slightly above the steady state level of the detection function. To ensure accurate detection, the length of the median filter must be longer than the average width of the peaks of the detection function. In practice, we set the median filter length to 150 ms. An example of this operation can be seen in Figure 4.20-(a), in blue trace is the detection function $d(n)$ and in the lower part of the figure in red trace the dynamic threshold.

A peak-processing stage selects as potential onset peaks those above the adaptive

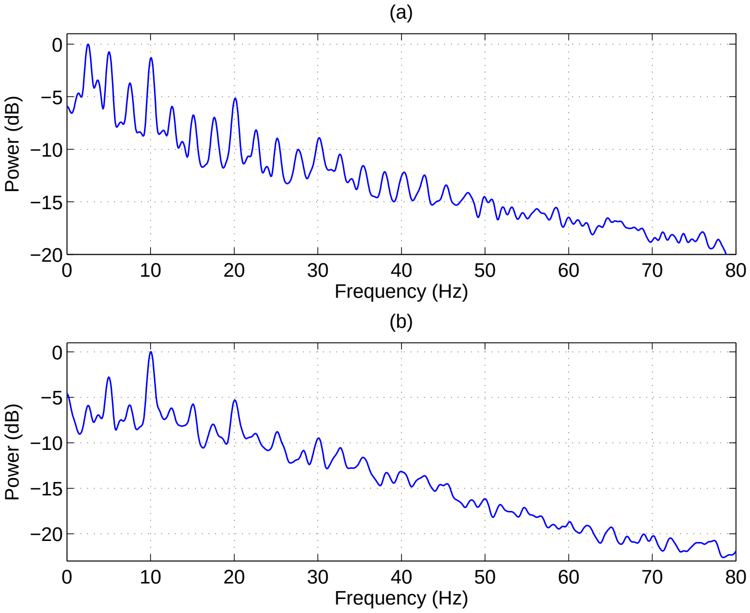


Figure 4.21: (a) Power Spectral Density of the detection function $d(n)$. (b) Power Spectral Density of the uniform-amplitude pulse-train $\widehat{d}(n)$

threshold and discards those having a too small width (<20 ms), considering them as artifacts. That is, we compute the signal $\tilde{d}(n)$, whose negative part is discarded:

$$\tilde{d}(n) = \begin{cases} d(n) - \theta(n) & \text{if } d(n) - \theta(n) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.36}$$

Then, each peak shape in $\tilde{d}$ is divided by its maximum value to obtain the uniform-amplitude pulse train $\widehat{d}(n)$. An example is illustrated in Figure 4.20-(b).

The last stage of the algorithm requires the calculation of the PSD of $\widehat{d}(n)$. To improve the PSD estimation, we use the method proposed by Welch (1967), which consists in decomposing the pulse train signal in several short-windows, computing their periodograms and in calculating their time average. The frequency corresponding to the tatum rate should coincide with the largest peak of the power spectrum. The accuracy of this method heavily depends on the quality of $d(n)$. In the presence of a defective detection function (containing dull peaks and/or many false onsets), this technique is not very useful.

Figure 4.21 presents and example using a pop/rock music signal. Figure 4.21-(a) shows the PSD of $d(n)$, we can see that the largest maximum is located at the tempo frequency at about 2.5 Hz (≈150 BPM). Figure 4.21-(b) presents the PSD of $\widehat{d}(n)$, this time we can see that the tatum frequency at about 10 Hz (≈600 BPM) is indicated by the largest peak of the power spectrum.

4.5 Conclusion

During this chapter, we addressed the problem of estimating certain rhythmic parameters from the musical stress profile. In fact, we know that this signal is formed of recurrent pulses conveying the rhythm information of the musical piece under analysis. To induce rhythmic metrics, the first step consists in estimating the underlying periodicities in the detection function $d(n)$. In this part, we have proposed four different procedures. Two of them are time-domain methods that have been previously used in the literature: the Autocorrelation Function (ACF) and a bank of comb-filter resonators (CF). The two others are spectral methods and are known as the spectral sum (SS) and the spectral product (SP). These techniques have not been used in the context of rhythm analysis before, however they were developed and have been applied as pitch estimation procedures. We have also showed how these approaches can be used to carry out block-wise periodicity induction (ACF, SS and SP) or continuous periodicity induction (CF).

Most of the systems proposed in the literature carry out the periodicity data fusion by directly summing information from all subbands. This operation has an inherent risk since it is not possible to guarantee that all subbands are informative and in some cases they might negatively affect the result. Two possible situations must be considered:

- ⋆ subbands carrying little periodicity information should have less influence during the data fusion;
- ⋆ a highly energetic subband containing many false onsets must not bias the periodicity description.

In order to reduce these potential problems, before integrating the periodicity vectors we have introduced a bandwise normalization and weighting stage.

We have also proposed a method to keep track of the most salient pulse periods present in the detection function. This approach is based on a multi-path dynamic programming algorithm which provides temporal stability as well as a robust multi-periodicity tracking, even in the presence of arrhythmic or quiet musical passages. The output of the tracking stage only provides information about the periodicities of the potential metrical levels. Next, we showed how the tempo can be estimated by weighting the saliences of the periodicities calculated by the tracking stage. The weighting function that we used is based on *a priori* knowledge of the tempo preferred by humans. Then we introduced two methods to conduct beat location. The first one is causal and is based on the idea of "predicting" beat locations. The second method is a novel technique in the context of rhythm analysis and is based on the alignment of an artificial pulse train. This approach is formulated as an optimization problem which is solved by dynamic programming. Finally, we introduced a tatum estimation algorithm based on the idea of pruning (thresholding) and modifying the amplitude of the peaks present in the detection function.

If the musical stress profile bears adequate rhythmic information, estimating and, to a lesser extent[12], tracking the most salient periodicities is only a partial solution to the metrical analysis problem stated in §2.2. The open question remains *selecting* and *connecting* certain of these salient pulsations into valid metrical levels. However, this process would require to build into the system a minimal musical knowledge. This additional information would help to estimate the time signature and the measure (or bar) period. Although these problems are not addressed in the present work, mainly because of the lack of the appropriate evaluation material, it is perfectly feasible to extend the current system capabilities to include them.

[12] We do not address the problem of tracking large timing deviations. As mentioned in §2.5, we suppose that the musical instance under analysis has a stable tempo.

Chapter 5

System performance: results and discussion

During the previous chapters we have introduced the constituent blocks of a modular system which carries out metrical analysis for audio recordings. The aim of this chapter is to present a quantitative evaluation of the algorithm performance, giving results for a number of different configurations that can be derived from the general system illustrated in Figures 3.1 and 4.1. In addition, we discuss the insights obtained from these results, highlighting the limitations of our approach as well as its advantages.

An exhaustive evaluation testing all the potential combinations of the internal parameters and blocks of the system proves to be computationally expensive and perhaps redundant. For this reason, only the efficacy of those configurations that we consider as most relevant and illustrative will be explored. First, §5.1 presents the effect of varying the (rhythm) analysis window-length and its overlapping factor, followed by §5.2 which provides a more detailed analysis by examining the results according to the musical genres present in the test database. Then, §5.4 discusses the effect of carrying out the harmonic plus noise decomposition on the input signal. In §5.5, we measure the impact of the frequency decomposition in the results by testing a number of different alternatives. In addition, §5.6 compares the performance of our proposal to estimate the musical stress profile to other methods found in the literature. §5.7 evaluates the computational complexity for a number of different variations of the general system. Then, §5.8 examines the accuracy of the beat-phase estimation methods and finally §5.9 evaluates the effectiveness of the tatum estimation procedure.

The quantitative evaluation of metrical analysis systems is an open issue. Appropriate methodologies have been independently proposed by Goto & Muraoka (1997a) and Temperley (2004), however they rely on an arduous or extremely time-consuming process to obtain the ground-truth database. Due to such limitations, the first part of our quantitative evaluation is confined to the task of estimating only the scalar value of the tactus (in BPM) of a given excerpt, instead of an exhaustive evaluation at several metrical levels involving beat-rates and phase locations.

As mentioned in §2.4, a first step towards benchmarking metrical analysis systems has been proposed by Gouyon et al. (2006). In our analysis, we adopt a similar approach to evaluate the output of our system by using two different metrics:

- ⋆ *Accuracy 1*: the tactus estimation must lie within a 5% precision window of the ground-truth tactus,

⋆ *Accuracy 2*: the tactus estimation must lie within a 5% precision window of the ground-truth tactus or half, double, three times or one-third of the ground-truth tactus.

The reason for using the second metric is motivated by the fact that the ground-truth used during the evaluation does not necessarily represent the metrical level that most human listeners would choose (Gouyon & Dixon, 2005), even if a perceptual weighting curve to find the most likely beat period is used (see §4.2.3). This is a widespread assumption among *beat-trackers* and it is largely employed during the evaluation of metric systems.

Throughout this chapter, whenever possible the results will be presented in separated fashion according to the test database. The reader must remember that we have access to two different test corpus as explained in §2.5. The first one is the ENST database which contains 961 instances and the second one is the TUT database containing 474 instances, both of them covering various musical genres.

Standard configuration As mentioned above, a full test for each of the parameters in our framework would require a too demanding computational cost and we do not consider it essential to explore the system capabilities and efficacy. For this reason, during the first two sections of this chapter the block corresponding to the preprocessing stage is configured to use the non-causal method (§3.2), the H+N decomposition technique employed is the EDS model (introduced in §3.3.1) using the uniform filter bank. The main motivation to select this particular configuration is that it displays satisfactory results compared to other settings, as will be shown in the forthcoming sections.

It is interesting to know if the combination of the four periodicity algorithms that we use (SS, SP, AC and CF) would reach a score higher than individual entries. For this reason we created a fifth entrant called *Method-Fusion* (MF) that combines results from the four other methods using a majority rule. This procedure takes as input the two most energetic tempi from each periodicity method and selects as tempo the candidate (entry) with most occurrences. For example, if the most energetic tempo from each periodicity method is different, but three of these methods have a common second most energetic tempo, then this candidate will be selected as tempo for the method-fusion (MF) procedure. If there exists no agreement between candidates and methods, preference was given to the first entry of the spectral sum (SS).

5.1 On the effect of the window length and overlapping

In the first part of this section, we study the effect of varying the parameter ℓ. This variable indicates the length of the analysis window (see Figure 4.7) for the block-wise periodicity methods: the Spectral Sum (SS), the Spectral Product (SP) and the Autocorrelation Function (ACF). When computing the periodicity in a continuous fashion, *i.e.*, using the bank of Comb-Filter (CF) resonators, the parameter ℓ controls the decimation factor[1] (see Figure 4.8).

To measure the impact of the window length ℓ, the overlapping factor was fixed to $\rho = 0.5\ell$. Then, several values of ℓ were tested as shown in Figure 5.1 for the ENST database and in Figure 5.2 for the TUT database. These values were obtained using the

[1] In fact, the decimation factor is simultaneously controlled by ℓ and ρ, see page 103 for details.

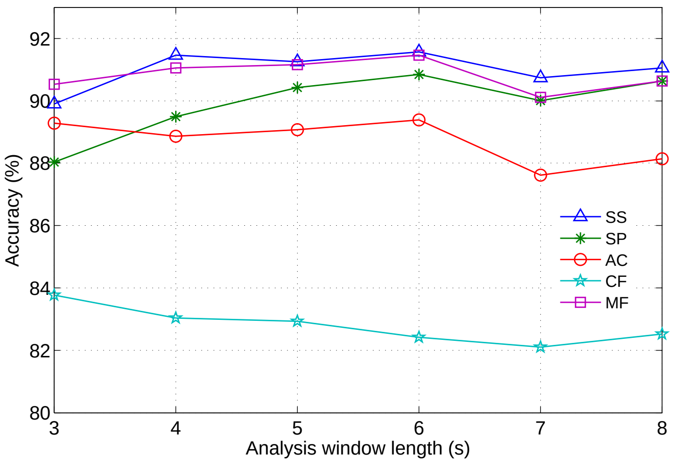


Figure 5.1: On the influence of window length for the ENST test database. This figures were obtained using the *Accuracy 2* evaluation metric.

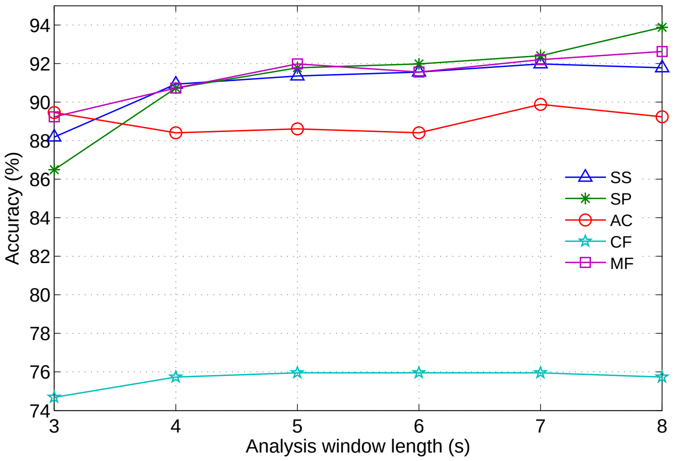


Figure 5.2: On the influence of window length for the TUT test database. This figures were obtained using the *Accuracy 2* evaluation metric.

Accuracy 2 criterion and the numerical figures can be found in Tables D.1 and D.2 in Appendix D.

For the ENST database, the spectral methods display a performance gain as ℓ increases, this improvement is more important for the SP approach. In the case of the time domain methods (AC and CF), increasing ℓ was not as productive. For small window values the performance of the AC remains rather constant and for the CF case accuracy gradually decays as the window length increases. Except for the leftmost point ($\ell = 3$ s), the MF was not able to outperform the SS. The small decrease in performance that exhibit all the methods for large window values ($\ell \geq 7$) is caused by the reduction in the number of (detection function) frames used to track the periodicity paths in the dynamic

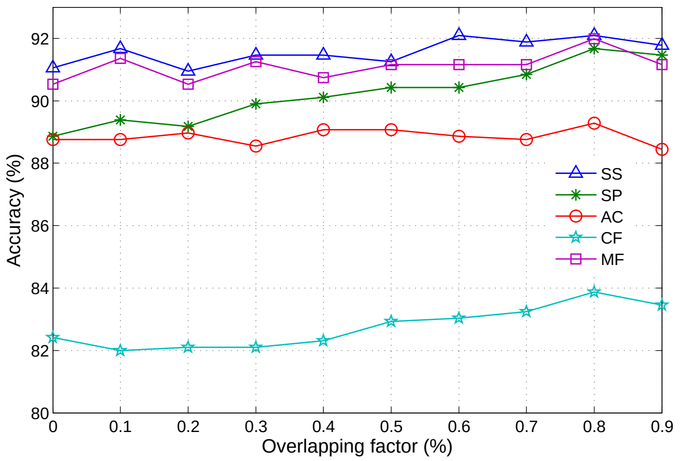


Figure 5.3: On the influence of the window overlap for the ENST test database.

programming (DP) stage. For example if $\ell = 8$ the matrix $\mathbf{\Gamma}$ will only contain 4 columns[2], which seems to be a rather small horizon to let periodicity paths settle, specially if the detection function contains an important number of spurious peaks.

System performance using the TUT database shows comparable behavior to that displayed on the ENST data. The spectral methods and the MF display a performance gain as ℓ increases, specially the SP approach. For the time domain methods (AC and CF), increasing ℓ does not seem to have a large impact on the accuracy. It is noteworthy that the CF method collapsed on the TUT data, it exhibited a performance drop of about 7% compared to its performance on the ENST data, this aspect will be examined in more detail during the next section.

There exists a trade-off between window length and adaptability to rhythmic fluctuations. From Figures 5.1 and 5.2 it can be seen that accuracy for the SS, SP and MF methods is close to its maximum when $\ell = 5$ s. Now, we focus on the influence of the overlapping (ρ) parameter on the overall performance for a fixed window length ($\ell = 5$ s).

A number of overlapping values as a function of the window length were tested: $\rho = k\ell$ with $k \in [0, 0.1, \ldots, 0.9]$. Results are shown in Figures 5.3 and 5.4 for the ENST and the TUT databases respectively. These values were obtained using the *Accuracy 2* criterion and the numerical figures can be found in Tables D.3 and D.4 in Appendix D.

Concerning the ENST database, the figure shows that introducing redundancy in the time–periodicity matrix $\mathbf{\Gamma}$ (by increasing the overlapping) produces a small but consistent gain in performance for the SP and CF, and to a lesser extent on the SS. The same impact is barely noticeable for the MF strategy, and we can say that ρ does not play any important role for the AC. The performance improvement can be explained by the fact that the DP stage has a larger data horizon and adapts better to metrical levels paths. Ironically, too large overlapping values ($\rho > 0.8$) also seem counter productive. After inspecting this small degradation in performance, we found that for those instances (mostly classical music) lacking of clear onset attacks during passages of several seconds, the small advance step in the analysis fills the time–periodicity matrix with ambiguous information which in certain cases lead the DP algorithm to lose track of the rhythm.

[2] If $\ell = 8$, the average number of columns per instance (for the ENST database) is four.

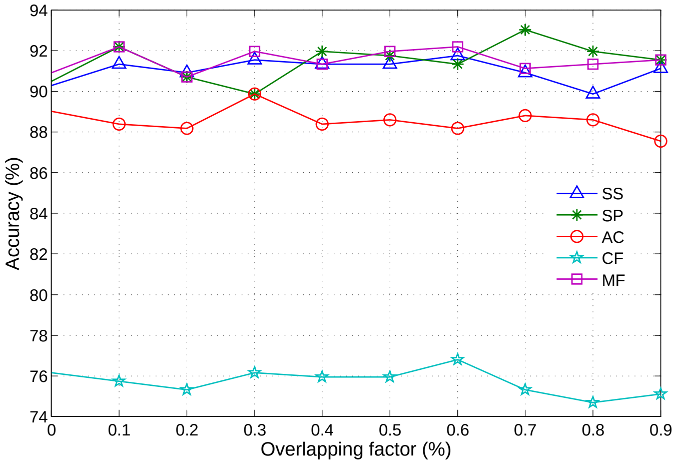


Figure 5.4: On the influence of the window overlap for the TUT test database.

When analyzing the system efficacy on the TUT database, we found that the overlapping parameter presents a rather different behavior. It appears to have no impact on the results and we think this is directly related to the audio signal length (amount of information) available during the decision process. In fact, the largest signal in the ENST database is smaller than the shortest signal in the TUT database. Therefore, even if ρ is small the amount of data available still allows to correctly resolve the instance under analysis.

Just like the window-length parameter, large ρ values bring a loss in adaptability coupled to an increase in the computational complexity. The overlapping factor appears to have a minor influence if a large amount of audio data is available, however it displays a small but consistent positive-influence in the presence of a limited data size, as shown for the ENST database. We decided to fix the value $\rho = 0.6\ell$, since we consider that it provides a "good" trade-off between accuracy and tracking capability.

In the remaining of this chapter the window length (ℓ) and the overlapping (ρ) parameters are respectively fixed to $\ell = 5$ seconds and $\rho = 0.6\ell$.

5.2 Efficacy of the system by musical genre

The system performance by musical genre is presented in Figures 5.5 and 5.6 in the form of bars, showing accuracy *vs.* musical genre for the ENST and TUT databases respectively. These values were computed using the *Accuracy 1* criterion. In addition, Figures 5.7 and 5.8 also present the performance by musical genre for the ENST and TUT databases, but this time using the *Accuracy 2* criterion[3]. Numerical values describing the average performance by periodicity method and accuracy criteria are presented in Table D.5. While inspecting the results obtained under the *Accuracy 1* and *Accuracy 2* criteria, we have found that for those excerpts which were only correctly estimated in the latter case, the periodicity estimation algorithms have a tendency towards estimating faster tempi that those manually annotated. In other words, it is more likely to find the double or triple of the ground-truth than finding one-third or half of its value. This

[3] The numerical values of this four graphs are given in Tables D.6 to D.9 in Appendix D.

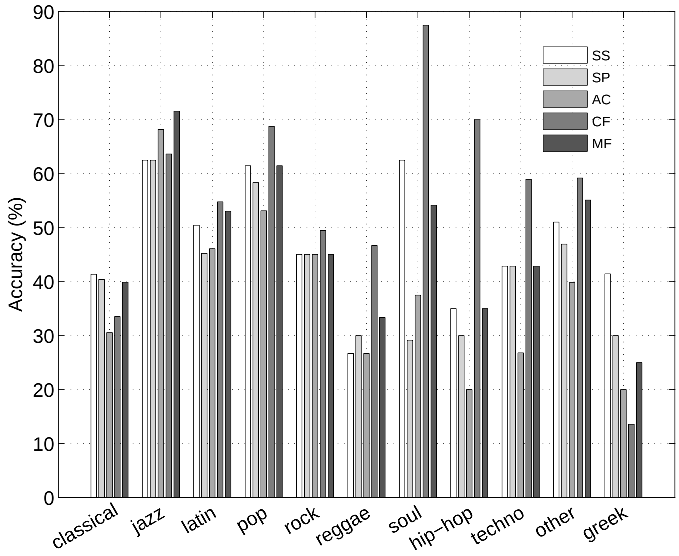


Figure 5.5: Performance under the *Accuracy 1* criterion ENST test database.

phenomenon is particularly more noticeable in those instances containing pulsations at lower (*i.e.*, faster) metrical levels.

For the ENST database the lowest performance was obtained when processing greek music yielding an average score of 76%, and to a lesser extent classical music with an average score of 84%. For the TUT database, the lowest score corresponds to classical music with a method average slightly above 70%. For the other genres in both databases, the average score is close to or above 90%. Moreover, for certain genres like reggae, soul and hip-hop the system attains a success rate of 100%. All these figures were computed under the *Accuracy 2* criterion. Nevertheless, such favorable results must be taken with cautious optimism since these genres are not particularly difficult and their representation in the database is rather limited (see §2.5).

For enhancement purposes, it is more interesting to analyze the instances where the algorithm failed. Perhaps one of the first questions to arise is: what happens with the CF method which exhibits remarkable results in the algorithm developed by Klapuri et al. (2006) and presents meager results in our implementation? This contrasting behavior can also be seen in Table D.5 where the CF method displays the best performance for the *Accuracy 1* criterion and then abruptly falls to the last position under the *Accuracy 2* criterion. After inspecting the instances where this method failed, we found that it has a strong tendency towards selecting tempi that bear an integer-ratio relation to the ground-truth, typical values are $\frac{2}{3}$, $\frac{3}{4}$ and $\frac{4}{3}$. In fact, Klapuri et al. point out this behavior "that all resonators that are in rational-number relations to the period of the impulse train show response to it". The actual system proposed by Klapuri et al. overcomes this issue with a back-end that carries out a joint estimation of three metrical levels (tatum, beat and measure) through probabilistic modeling of their relationships and temporal evolutions. Our implementation of the CF method still requires further work to cope with this problem.

In the case of the spectral methods, the SS displayed a slightly better performance in most of the cases compared the SP. While inspecting some of instances where these

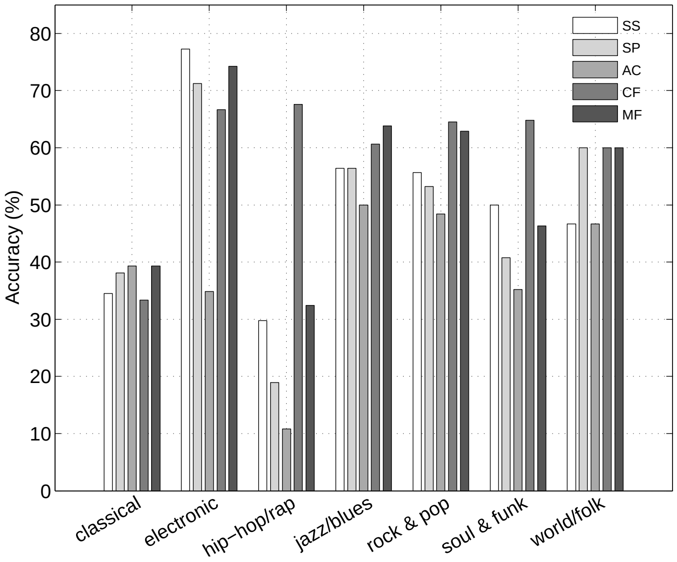


Figure 5.6: Performance under the *Accuracy 1* criterion TUT test database.

methods failed we found that they can be sensitive to the degree of aperiodicity in the detection function PSD, producing results slightly outside the 5% precision window. Just like the CF, spectral methods also suffer from selecting tempi that bear a rational-number relation to the ground-truth. The AC showed a fair performance although this method hardly ever outperformed their spectral counterparts.

In general the MF procedure displayed a performance comparable (although slightly below) to that of the SS[4], however in some particular cases combining the results from all the methods did *payoff*. For example, in both databases under the *Accuracy 1* criterion when processing jazz music the tempi combination was more successful than individual entries. Another example is when processing classical music from the ENST database under the *Accuracy 2* criterion, the MF achieved a higher score of about 3% above that of the SS. Finally, a particularly noteworthy example is when processing world/folk music from the TUT database, in this case the MF outperform the other methods by at least 6%.

It is also interesting to explore by musical genre the instances where the system did not succeed. Undoubtedly the most complicated and challenging case is when processing classical music. In our case, for a large part of these excerpts the SEF algorithm (see §3.4) in charge of computing the musical stress profile merely did not work. In that case, no matter how good the periodicity estimation and path-tracking blocks are, the algorithm is doomed to fail. The problems with classical music are numerous. For example, smooth attacks are very frequent and they are usually produced by bowed-string or wind instruments. Moreover, timing variations are natural in classical interpretations. Another problem is that for some excerpts a wrong metrical level was chosen, *i.e.*, one with a value having rational-number relation to the ground-truth. In the jazz/blues case, most failures are related to poly-rhythmic excerpts where the tactus found by the algorithm differed from the one selected by the annotators. For genres like latin, pop/rock, soul/funk, "other" and traditional greek music, the large majority of the errors are found in ex-

[4] As a reminder for the reader, in case there exists no agreement between periodicity methods, the MF takes the value of the SS.

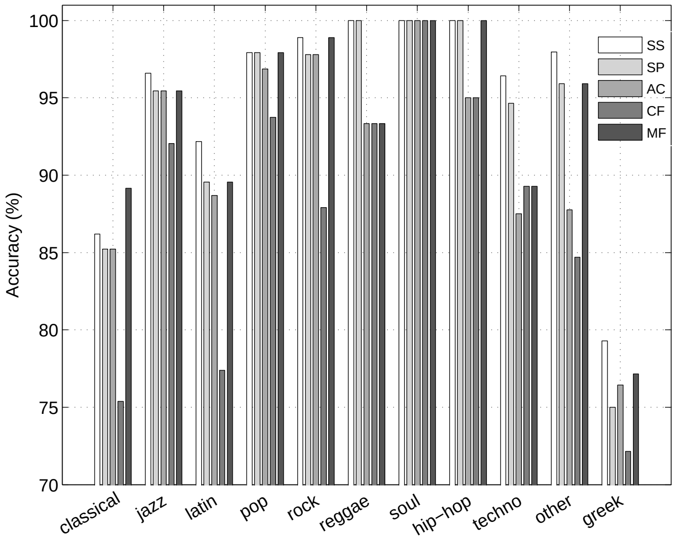


Figure 5.7: Performance under the *Accuracy 2* criterion ENST test database.

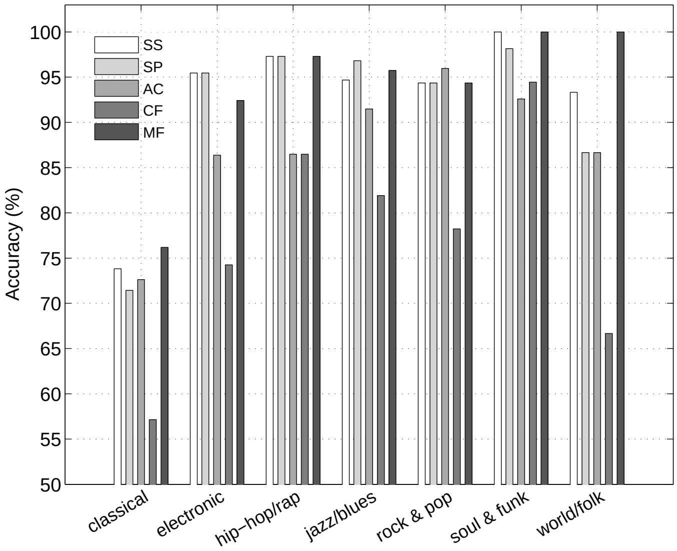


Figure 5.8: Performance under the *Accuracy 2* criterion TUT test database.

cerpts with a strong vocals foreground or having large chorus regions, both incorrectly managed by the SEF algorithm. For the traditional greek music, poly-rhythmic excerpts with a peculiar time-signature are also the cause of failure. In techno/electronic music, we have found that some digital audio effects (chirp-like signals with a non-stationary frequency behavior) lead to false onsets. Most of these issues suggest that the crucial problem in metrical analysis remains the estimation of an accurate musical stress profile, well above other difficulties such as path-tracking and periodicity estimation.

5.3 Influence of the H+N decomposition

A natural question arises when we inquire about the influence on the accuracy directly related to separating and processing the audio signal as harmonic and noise parts. In Chapter 3 were presented two different techniques to carry out the harmonic-plus-noise (H+N) decomposition, the first one is based on the Exponentially Damped Sinusoidal (EDS) model and the second one on the more classical Fourier Transform (FT). In this section, we compare the difference in performance between both of these methods to a third variant of the algorithm that does not use (or bypasses) the H+N block. This evaluation uses the uniform frequency decomposition, additionally, the effect of the causal (C) and non-causal (NC) preprocessing schemes (see §3.2) is also tested.

Figures 5.9 and 5.10 present the results for the ENST and TUT databases respectively[5]. In addition, we also compared the above mentioned system variations to the well-known classical method proposed by Scheirer[6]. A minor modification of his algorithm was carried out, contrary to our implementation it was conceived to produce a set of beat times rather than an overall scalar estimate of the tactus. For this reason, the tempo was computed from the median of the inter-beat intervals.

From Figures 5.9 and 5.10 we can see that for most cases (regardless of the preprocessing scheme used) carrying out the H+N decomposition yielded a small but consistent improvement in the results. The only exception being the AC method in the TUT database.

Precision of the results For the evaluations comparing various different configurations, it is important to include error-bars in the analysis with the intention to find whether the differences between algorithm variants are statistically significant or not. It is also important to notice that while computing this significance test we assume that our databases are *truly* random samples of Western music. Otherwise this significance test may overstate the accuracy of the results, because it implicitly considers the algorithm errors as random, *i.e.*, the test cannot consider biases resulting from a non-random error (a badly selected test corpus). According to Schwartz (1963, Page 47), if a percentage p_o (considered in the $[0,\ 1]$ range) is observed for a sample of size N, we can assign to the unknown true percentage p the 95% confidence interval

$$p_o \pm\ 1.96\sqrt{\frac{p_o q_o}{N}} \tag{5.1}$$

[5] The corresponding numerical values can be found in Tables D.10 and D.11 in Appendix D.

[6] Although the method developed by Scheirer (1998) is free software, the version that we used was ported from the Dec Alpha platform to GNU/Linux by Anssi Klapuri.

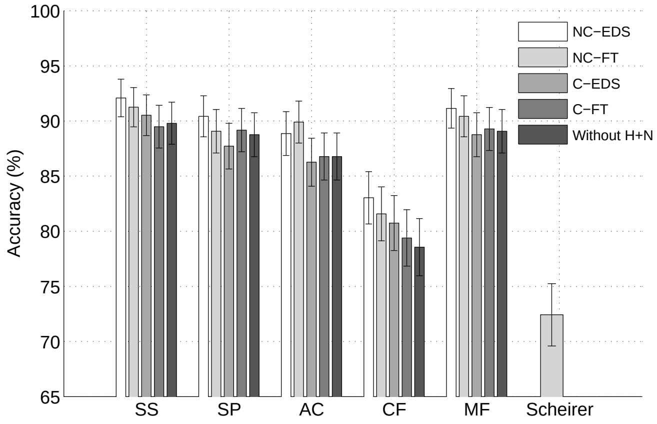


Figure 5.9: Performance comparison between using the causal and non-causal preprocessing (ENST database).

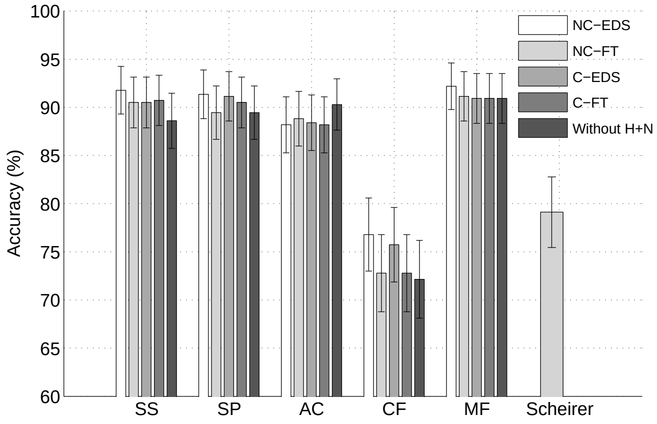


Figure 5.10: Performance comparison between the causal and non-causal preprocessing (TUT database).

where $q_o = 1 - p_o$. This *precision interval*[7] can be presented in the form of error-bars as illustrated by Figures 5.9 and 5.10. Due to the small difference in performance and the size of the test database, these results suggest that carrying out the H+N decomposition is not statistically significant at the 5% level, even if a general trend (slightly above 2% in average) indicating a better performance is perceived. Under the supposition that the tendency exhibited by the results holds as the corpus size increases, it would be necessary to use a database containing at least 2380 samples to guarantee that for the SS (using NC-EDS) the H+N decomposition is statistically significant.

After taking a closer look at the results, we have noticed that the H+N decomposition does not affect in the same way the different musical styles present in the database. For example, when dealing with musical genres like reggae, soul, hip-hop and electronic/techno, we found that the decomposition did not have any noticeable influence at all[8]. For the jazz and rock/pop music, the H+N decomposition had a small and mostly positive influence, although for some periodicity methods the decomposition was slightly counterproductive. The most outstanding enhancement produced by the H+N decomposition concerns the classical music. For this genre, the average improvement was 6.4% (for the EDS decomposition), as illustrated in Figure 5.11. Although such performance still does not guarantee that carrying out the decomposition is statistically significant, we consider these results as satisfactory since this musical genre is particularly difficult and these results exhibit a clear progression. It is important to keep in mind that the statistical significance concept is tightly connected to the number of elements in the database. For the classical music case, due to the sample size reduction the error-bars have more than doubled their span.

In general, we remarked that the improvement brought by the H+N decomposition is mainly formed of excerpts containing weak attacks such as bowed-string and wind instruments, and to a lesser extent of signals with a rather clear rhythm but with a salient speech foreground (vocals). Ironically, when we examined the excerpts for which none of the algorithms succeeded, we practically found the same kind of signals: bowed-strings with large vibratos and weak attacks, orchestral pieces and signals with chorus and/or a strong vocals foreground. As mentioned before, the weakness of the algorithm lies in the musical stress estimation module. This can be seen as a single problem formed of two different facets:

- ⋆ the incapability of detecting soft attacks mainly seen in classical pieces, while visual inspecting the set of detection functions we noticed that true attacks do not surpass the noise level;

- ⋆ the presence of too many false attacks in the detection function, mainly provoked by the appearance of local frequency variations seen in vibratos and speech signals.

Finally, from the results presented in this section it is possible to state that in the context of rhythm analysis the H+N decomposition based on the EDS model generally outperforms its counterpart based on the Fourier Transform.

[7] The precision interval was not computed in the previous section since for some of the genres in the test database it is not possible to satisfy the *large sample size* condition which states that np and $nq \gg 5$

[8] However this is not a concluding remark since such genres a rather underrepresented in our databases.

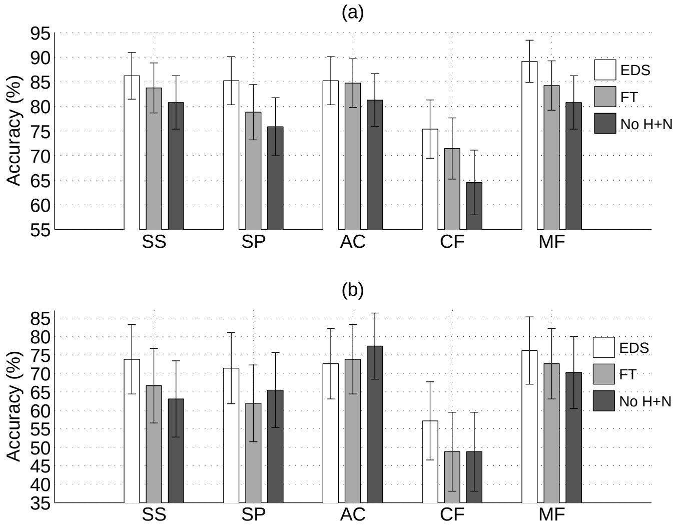


Figure 5.11: Influence of the H+N decomposition on classical music. (a) Results obtained using the ENST database and (b) the TUT dataset, both were computed using the non-causal preprocessing scheme.

5.4 Impact of the harmonic and noise parts in the accuracy

Since we already have access to the harmonic and noise parts of the audio signal, it is interesting to find out which of these constituents has a stronger influence on the accuracy. For that purpose we use the same analysis framework illustrated in Figure 4.1, but during the periodicity induction block only the information coming from the harmonic or noise part was considered. These algorithm variants were compared to the version that does not compute the H+N decomposition. Figures 5.12 and 5.13 present the outcome for the ENST and TUT databases respectively[9]. These results were computed using the non-causal preprocessing scheme.

The afore mentioned results do not provide any compelling evidence supporting the idea (given either a decomposition or periodicity method) of any specific signal part being more influent than the other. A closer inspection shows that for the percussion-driven music (rock/pop/latin, hip-hop and to a lesser extent jazz) the noise part works slightly better, however other musical genres sharing this characteristic (like soul and techno/electronic) do not show this tendency. On the other side, for classical music, the harmonic part exhibits a moderate predominance in the results.

5.5 Influence of the frequency decomposition

For many years, researchers working on musical rhythm processing have wondered if there exists an optimal frequency decomposition for *beat analysis*. For example, Scheirer (1998, page 591) argues that ″empirical studies of the use of various filter banks (...) have demonstrated that″ his ″algorithm is not particularly sensitive to the particular bands or implementations used″, in other words, the key is to decompose the signal in frequency

[9] The corresponding numerical values can be found in Tables D.13 and D.14.

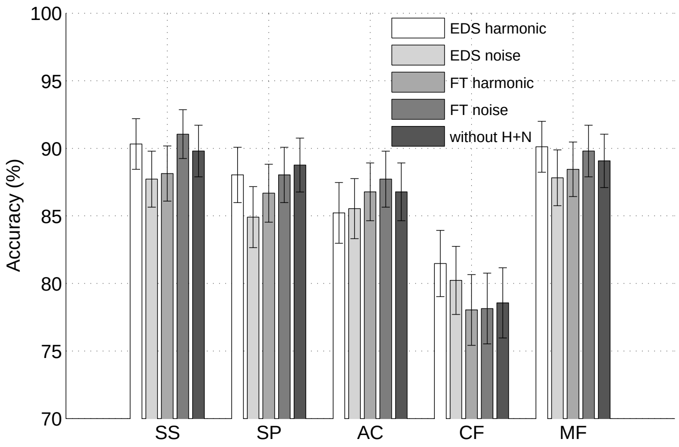


Figure 5.12: On the influence of the signal and noise parts in the analysis (ENST database).

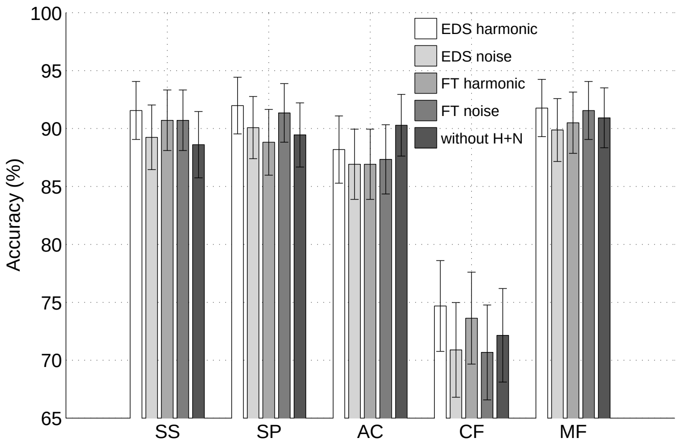


Figure 5.13: On the influence of the signal and noise parts in the analysis (TUT database).

bands regardless of the layout used. On the other side, Gouyon (2005, page 174) advocates for "the superiority of the ERB frequency subband decomposition over others (proposed in the literature) as the basis for the computation of effective energy feature sets". The goal of this section is to investigate the influence of the frequency decomposition on the performance of our system. The analysis is divided in two sections, first §5.5.1 explores the influence of the filter bank together with the H+N decomposition. Then, §5.5.2 drops the H+N part and only considers the influence of the frequency decomposition on the system performance.

5.5.1 Filter bank and H+N decomposition

In Chapter 3 we introduced two different filter banks that can be used together with the H+N decomposition: the eight-band uniform filter bank (U-FB) and the five-band logarithmic filter bank (log-FB) (see §3.3.1.3), but until now we have only considered the former one in our analysis. Figures 5.14 and 5.15 present the performance for both kinds of filter banks for the ENST and the TUT databases respectively, these values were obtained using the non-causal preprocessing scheme. Although this test only considers two distinct filter banks, the results obtained strongly suggest that the overall performance of our algorithm is not particularly sensitive to a specific frequency decomposition. That is, a very similar performance is displayed if the uniform filter bank is replaced by the logarithmic filter bank. During a closer inspection by musical genre we found that the logarithmic frequency decomposition had a slightly higher significance on classical music, this phenomenon is especially noticeable in the TUT database[10], as illustrated in Figure 5.16. Perhaps the most noteworthy case is that of the "EDS log-FB" which almost attained an accuracy of 85% (TUT database) for the MF periodicity procedure, exceeding by more than 8% the uniform filter bank variant. The rest of the genres do not exhibit any relevant behavior directly linked to the filter bank structure.

In order to conduct the H+N decomposition using the EDS model combined with the logarithmic filter bank, the user should keep in mind the non-neglectable increase in computational complexity entailed by the upper bands of the logarithmic frequency decomposition (see §3.3.3). This practical order issue makes the uniform filter bank more attractive.

5.5.2 Frequency decomposition

In this section we investigate the influence of the frequency decomposition without using the H+N decomposition. In addition, we have included in the analysis an entry called "single band" where the Spectral Energy Flux (SEF) algorithm is directly applied to the bulk audio signal (actually after passing it through the non-causal preprocessing block). Figures 5.17 and 5.18 present the results.

As in the previous case, selecting between the uniform or the logarithmic frequency decomposition yields a fairly similar overall performance. That is, classical music still exhibits an improvement, although it is less pronounced than that described above. The rest of the genres do not display any important modifications.

In addition, computing one single musical stress profile for the whole passband proves to be less advantageous. The results obtained suggest that those passages containing a strong singing voice component or other non-stationary constituents causing false onsets

[10] See Table D.17 for numerical values.

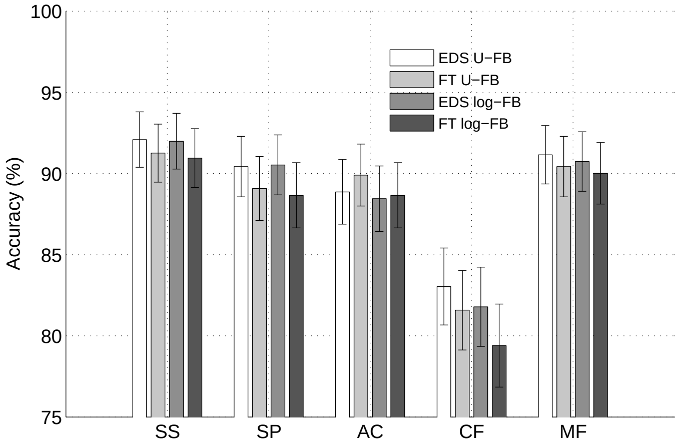


Figure 5.14: Comparison of the filter bank influence on the system's performance together with the H+N decomposition. The first two bars of each periodicity induction method correspond to the uniform filter bank (U-FB) and the last two to correspond to the logarithmic filter bank (log-FB). These results were obtained on the ENST database.

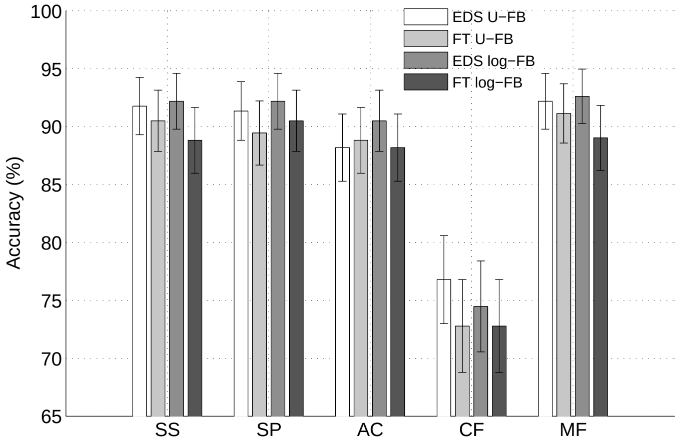


Figure 5.15: Filter bank comparison as in Figure 5.14, but using the TUT database.

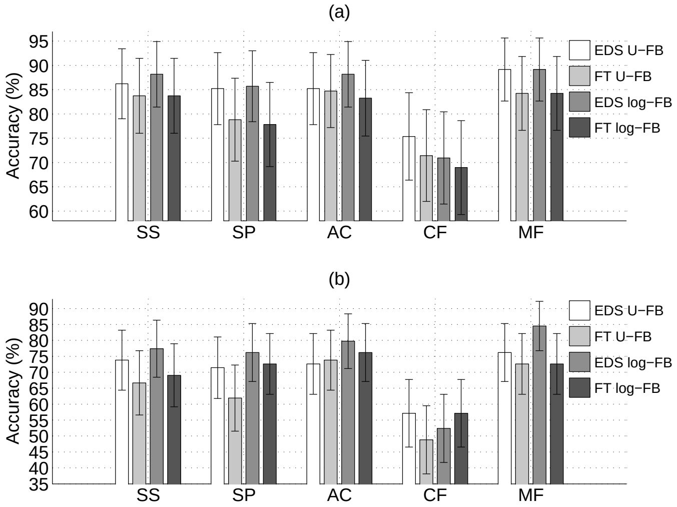


Figure 5.16: Influence of the filter bank on classical music. For (a) the ENST and (b) TUT databases respectively.

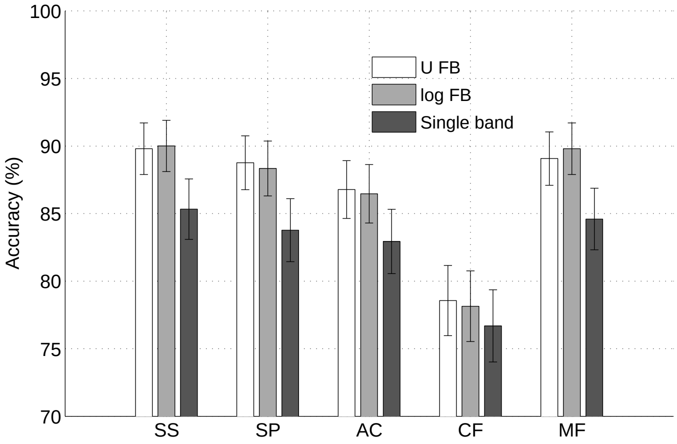


Figure 5.17: On the influence of the frequency decomposition on the system's performance (ENST database). In this case, the H+N decomposition is disabled and three different configurations are studied: the uniform filter bank (U FB), the logarithmic filter bank (log FB) and one single band (*i.e.*, accent detection is performed on the bulk signal).

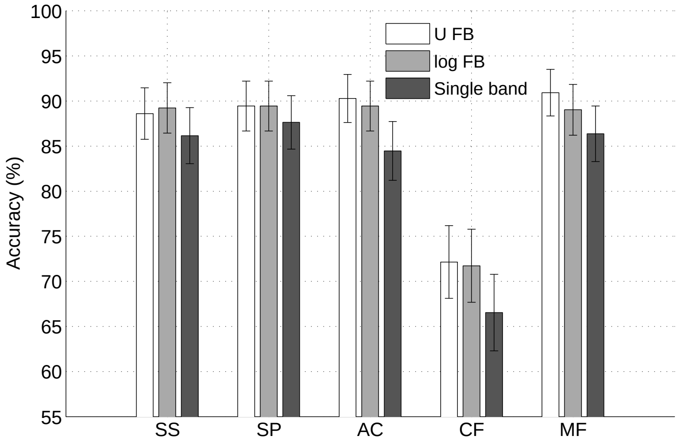


Figure 5.18: Same as Figure 5.17, but using the TUT database.

are affected by this "single band" analysis. By separating the computation of the musical stress profile in frequency bands it is possible to attenuate the propagation of these spurious detections to other spectral regions. On the other hand, Figure 5.18 indicates that computing a single stress profile for long data sequences is close in performance to using a filter bank (uniform or logarithmic) but has a lower computational cost.

5.6 Detection function comparison

In this section we compare our musical stress estimation module to two other reference systems frequently cited in the literature: the Spectral Difference (SD) proposed by Masri (1996) and later by Duxbury et al. (2002), and the Complex Spectral Difference (CSD) developed by Bello et al. (2004). These algorithms[11], as well as ours, share a common general approach based on measuring the rate of change of the power-spectrum. In fact, these methods compute the detection function as a "distance" between successive short-term Fourier spectra, treating them as points in an N-dimensional space, a more detailed description can be found in Appendix E.

To carry out the detection function comparison, the block corresponding to the SEF (see Figure 3.1) was replaced by the SD and CSD algorithms, the rest of the system remains unaltered. The configuration for this test uses the non-causal preprocessing and the uniform filter bank. Figures 5.19 and 5.20 present the performance using the ENST and TUT databases respectively. These results exhibit a contrasting variation for the CSD and SD methods, while for the ENST database their accuracy is somewhat low, for TUT database it considerably improved (in average) by 5.6% and 7.2% respectively. We believe that the difference in signal-length[12] in each base is at the origin of this variation between databases. We consider that the performance of SEF algorithm as satisfactory.

[11] The SD and CSD algorithms were implemented by Pierre Leveau, who generously made his source code available for comparison purposes.

[12] The major difference between test corpora is the length of the signals that they contain, this parameter is much larger for the TUT database, see §2.5.

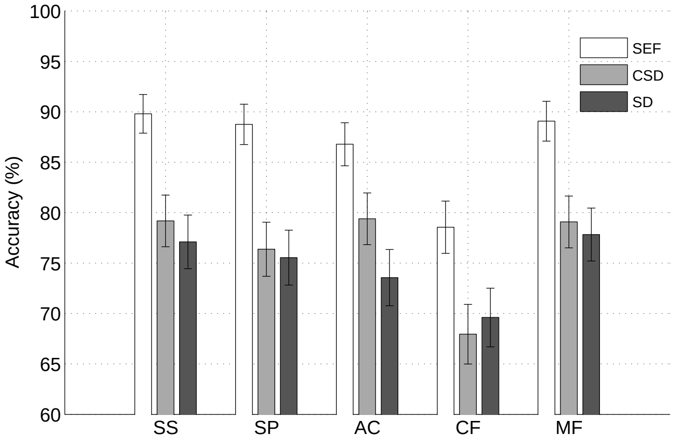


Figure 5.19: Detection function comparison: the Spectral Energy Flux (SEF) method proposed in §3.4, the Complex Spectral Difference developed by Bello et al. (2004) and the traditional Spectral Difference (Masri, 1996; Bello et al., 2005). These values were obtained on the ENST database.

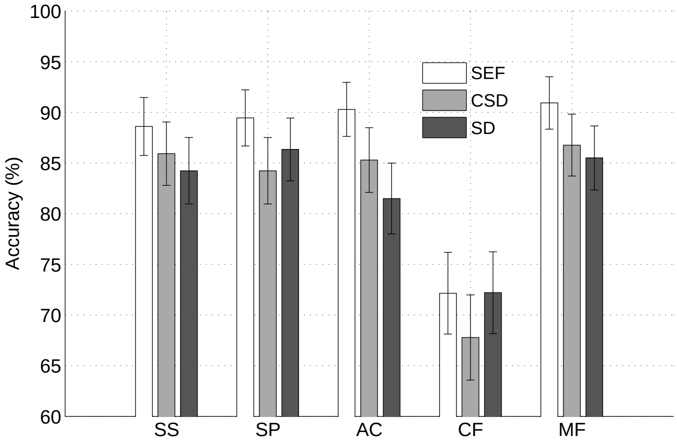


Figure 5.20: Same as Figure 5.19 but using the TUT database.

5.7 Computational complexity

A key attribute of any metrical analysis system is its computational complexity. During this work, we have developed our framework under the Matlab (7.0.1 R14) environment. Since our implementation uses a mixture of various built-in functions as well as an important number of scripts, a meticulous evaluation appears to be rather complicated. For this reason, we adopt again the same principle used in §3.3.3 consisting in measuring the time it takes to each stage of the algorithm to process a 20 s excerpt taken from the test database. Although this approach does not provide the algorithm complexity, it gives a fairly good idea of the computational requirements.

During this test three different system configurations were used. Figure 5.21 presents their respective results in the form of slices showing the time consumption for each stage as a fraction of the total time. From left to right, the system configuration mnemonics stand for ”uniform filter bank using the EDS decomposition model” (U-FB & EDS), ”logarithmic filter bank using the EDS decomposition model” (log-FB & EDS) and ”uniform filter bank using the FT decomposition model”.

For each of the configuration variants, the total computation time for analyzing a 20 s excerpt was 32.5 s, 92.3 s and 24.2 s respectively. In addition, Table D.22 in Appendix D presents the exact values.

The three configurations shown above use the non-causal preprocessing, the spectral sum for periodicity analysis and the same path-tracking parameters. These figures were obtained using a Pentium 4 machine running at 3 GHz with 1 GB of memory under Debian GNU/Linux 3.1 (Sarge) using the ”Profiler” utility of Matlab.

However, the results presented in Figure 5.21 can be somewhat misleading concerning the complexity of estimating the musical stress profile. Taking a closer look at this module, we found that about 85% of the time spent on this block corresponds to the ”reassignment loop”. Actually, this loop sweeps through every single element of the STFT time–frequency matrix used by the Spectral Energy Flux (SEF) algorithm. Moreover, this process is repeated for all subbands used in the analysis. Implementing this function outside the Matlab environment (*i.e.*, coded in a faster and lower level language such as C/C++) should considerably reduce the running time.

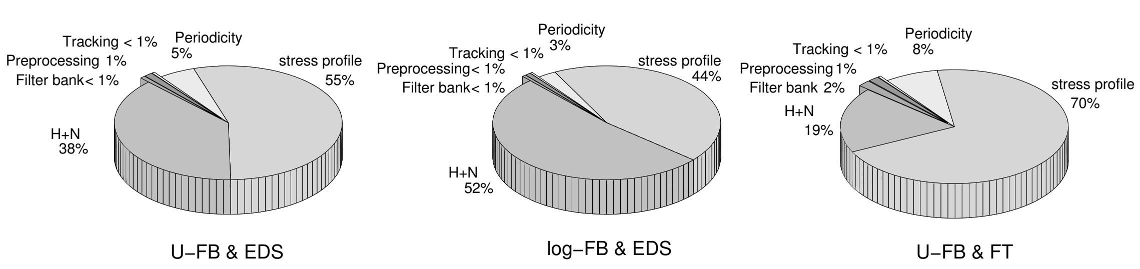


Figure 5.21: Algorithm complexity.

In fact, the most computationally demanding task of our system is the subspace filtering operation, which is particularly expensive for signals with a large bandwidth (*e.g.*, the upper bands of the log-FB & EDS configuration). In this case, a large running-time improvement seems more difficult to obtain since we already use a highly optimized C-coded version[13]. Table D.22 also provides information on how many times each module is executed for a given configuration.

5.8 Beat-phase estimation

In this part we test the performance of the beat location methods introduced in §4.3. The evaluation was carried out using the beat annotations of TUT database. In fact, all the instances in this corpus, the tactus pulses were manually annotated for approximately one-minute long excerpts which were selected to represent each piece. As described in §2.5.2, these annotations were made by a musician who tapped along while listening at each piece. Then, the tapping signal was recorded and the tapped beat times were automatically detected.

Before testing the beat location algorithms, we estimated the tempo for each piece using the system configuration employed in §5.2 under the *Accuracy 2* criterion. Then, beat location was conducted only for those pieces whose tempo was correctly estimated. Table 5.1 shows the composition of the data corpus used in this test.

	Genre							**Total**
	classic	**electronic**	**hip-hop**	**jazz/blues**	**rock/pop**	**soul/funk**	**other**	
number of instances	68	60	36	85	120	52	11	432

Table 5.1: Beat location is conducted only for those pieces whose tempo was correctly estimated. Genre distribution of the test corpus employed during the evaluation.

According to Lerdahl & Jackendoff (1983), finding the beat involves matching a regular grid to the accent structure of a musical piece. The assumption, borne out in traditional Western music, is that beats tend to fall on more accented/salient events. In fact, when humans perform beat-tracking they use knowledge and memory in addition to the pitch and timing information produced by the audio signal. In our implementation, we only have access to the detection function and we rely on the assumption that salient moments appear on this stress profile as prominent peaks.

Our tempo estimation criterion considers as correct not only the ground-truth value, but also other accepted multiples and sub-multiples. In §5.2 we noticed that our system has a stronger tendency towards overestimating the tempo than towards undervaluing it. To avoid putting at disadvantage this faster tempi trend by penalizing unmatched[14] beats, we evaluate the ability of the beat location algorithm to match all the manually annotated beats. That is, we penalize only unmatched beats in the ground-truth, but not unmatched beats in the beat location output. This criterion is more severe regarding tempo underestimations, however this scenario is rather infrequent compared to the total number of correct estimations and overestimations.

The criterion to consider if a correct beat match was found (*i.e.*, to know if a beat in the ground-truth and an estimated beat coincide), is that both of them should lie in-

[13] This function was developed and kindly provided by Olivier Gillet.

[14] By unmatched beats me wean beats located by the algorithm which do not coincide with the ground-truth beats because they occur at a higher rate. For example, if an excerpt is manually annotated at 80 BPM, but tempo estimation is 160 BPM, in the best case only 50% of the beats located by the algorithm will match those in the ground-truth.

side an *acceptance window*. The length of the acceptance window was fixed according to the precision requirement suggested by Goto & Muraoka (1997a). This criterion indicates that a correct beat position must not deviate from the annotated beat by more than 0.175 times (*i.e.*, 17.5%) the annotated period length. According to Klapuri et al. (2006), who also used this specification, inaccuracies in the manually annotated beat times allow meaningful comparison of only up to that precision.

Table 5.2 presents the performance figures for the causal and non-causal beat location methods. Both methods show a moderate performance, with the non-causal algorithm displaying better results. As in most previous tests, the lowest performance was obtained when processing classical music. In fact, this kind of music has the highest rate of false note attacks. We have noticed that such spurious peaks in the detection function considerably complicate the beat detection task for both methods. This problem is especially annoying during the beat initialization process, since the location of the starting beat is only based on the information provided by the autocorrelation function.

Concerning the causal algorithm, if it is properly initialized and the rhythm is fairly stable it displays acceptable results, for example when processing hip-hop or soul/funk music. However, we have noticed that in some cases the algorithm loses the beat synchrony and starts tracking the offbeat. This situation usually occurs during a musical transition, for example when changing from verse to chorus or vice-versa. We have found that in certain cases, the algorithm loses track of the beat if a false attack stronger than the actual beat appears inside the acceptance window. In that case, this artifact is used as new reference producing erroneous beat locations.

The problems found with non-causal algorithm are different. Contrary to the afore mentioned technique, this one is more sensitive to the initialization process. If the starting beat is a valid offbeat, this method will never hook-up. On the other side, since it is based on a rather rigid pulse train, it also has the advantage of not losing beat synchrony. We consider this technique as very promising for beat alignment, however under its current form it is affected by long-term tempo variations. For example, if the estimation algorithm considers that the tempo of a musical piece is 125 BPM, and in the middle of the performance it accelerates or slows down, it will produce some alignment issues. The problem is that the Correlation Optimized Warping algorithm aligns each beat individually and if there are more beats in the pulse train than in the actual signal, some of them will not be properly aligned (or will be aligned to false attacks). On the contrary, if there are more beats in the audio signal than in the pulse train, some of the actual beats will not be detected. A possible solution to this problem is to decompose the audio signal in shorter segments and to align them separately using a locally more precise tempo value.

Method	Genre							**Total**
	classic	**electronic**	**hip-hop**	**jazz/blues**	**rock/pop**	**soul/funk**	**other**	
causal	35.1	78.9	85.4	72.1	77.1	84.1	55.6	70.5
non-causal	42.8	87.6	90.1	78.8	82.4	89.2	67.7	77.1

Table 5.2: Beat location results for both causal and non-causal algorithms. Performance results (in %) are presented by musical genre.

5.9 Tatum estimation

In this part we evaluate the tatum estimation algorithm introduced in §4.4. To be more specific, we only estimate the tatum rate (in BPM), phase information is ignored during our analysis. As a reminder, the term tatum (introduced in Chapter 1) refers to a time *quantum*. It can be defined as the lowest metrical level, *i.e.*, a regular pulse train that a listener intuitively infers from the timing of the musical events. For Bilmes (1993b) and Gouyon et al. (2002), it is roughly equivalent to the time division that most highly coincides with note onsets: a sort of trade-off between how well a regular grid explains the onsets, and how well the onsets fit to that grid.

Since we do not have tatum values for the ENST dataset and only a number of instances of the TUT database were manually annotated at this level, to conduct this test a different corpus (a subset of the TUT data) was used. Its composition by musical genre and size is depicted in Table 5.3.

	Genre							**Total**
	classic	**electronic**	**hip-hop**	**jazz/blues**	**rock/pop**	**soul/funk**	**other**	
number of instances	68	47	22	70	114	42	12	375

Table 5.3: Genre distribution for the tatum database.

For the tatum, estimating the pulse period is difficult. According to Klapuri et al. (2006, page 352) ”this is because the temporally atomic pulse rate typically comes up only occasionally, making temporally stable analysis hard to attain”. In other words, this pulse rate is not always present. Moreover, at any moment a new pulsation at a higher rate (in western music usually the double or triple) might appear.

Table 5.4 presents the performance figures for the tatum rate estimation algorithm. For evaluation purposes we used three different benchmarks. The first one is called *Accuracy 1* and only considers as correct an exact ground-truth period match within a 5% precision window. The second is *Accuracy 2* and takes into account correct values as well as ”correct overestimations”, *i.e.*, erroneous tatum estimations which match 2 or 3 times the ground-truth period. Finally, the *Accuracy 3* is quite similar to its predecessor, but it takes into account ”correct underestimations” , *i.e.*, erroneous tatum estimations which match $\frac{1}{2}$ or $\frac{1}{3}$ of the ground-truth period.

From the first row of Table 5.4, we see that in general (with the exception of rock/pop music) the performance obtained is lower than that obtained for tempo estimation. If the requirements are slacken and correct overvalues are accepted, the performance of the algorithm becomes fairly admissible (not including classical music). In fact, by comparing rows two and three we can see that our algorithm has stronger tendency towards overestimating the tatum. We consider this behavior preferable than undervaluing, since a faster tatum will still capture all the potential events, which is not guaranteed by a lower frequency value. During the evaluation we noticed that overestimations are often related to low tatum values (close to or below 100 BPM). On the other side, underestimations are often associated to a tatum pulsation which is not prevalent enough and rather appears only during sporadic time intervals. Finally, in the presence of a deficient detection function the algorithm totally fails the tatum estimation.

The score obtained under the *Accuracy 1* criterion suggests that there is still a considerable amount of research required on this topic of automatic metric analysis. However, we consider these results encouraging for a technique still in an early stage of development. In addition, these results are comparable in performance to those obtained by the renowned system developed by Klapuri et al. (2006).

Criterion	Genre							Average
	classic	**electronic**	**hip-hop**	**jazz/blues**	**rock/pop**	**soul/funk**	**other**	
Accuracy 1	48.5	74.7	45.5	68.6	79.8	66.7	66.7	**67.5**
Accuracy 2	57.4	85.1	81.8	77.1	92.1	81.0	75.0	**79.7**
Accuracy 3	58.8	83.0	45.5	74.3	83.3	69.1	75.0	**73.1**

Table 5.4: Performance (in %) for the tatum rate estimation algorithm. Three different criteria were tested, for *Accuracy 1* an exact period match is required. *Accuracy 2* contemplates the over-estimation case and 1, 2 and 3 period multiples are accepted. *Accuracy 3* contemplates the under-estimation case and 1, $\frac{1}{2}$ and $\frac{1}{3}$ period multiples are accepted.

5.10 Conclusions

This chapter addressed the task of testing our meter analysis system. Most of it was devoted to evaluating the tempo induction algorithm. In the first part we investigated the influence of the length of the periodicity analysis window (ℓ) and of the overlapping factor (ρ), we fixed these parameters using a trade-off criterion between accuracy and tracking capability.

Then, to provide a more detailed perspective of the system performance, we analyzed the results by musical genre. Our method displays good results for music with a rather clear rhythm, however performance drops when dealing with more challenging material (*e.g.*, from classical music). After examining those instances where the algorithm did not succeed, we conclude that the bottleneck of our system lies in the musical stress estimation module. This block requires further reinforcement to properly process smooth and long attacks (especially those created by bowed-string and wind instruments) and musical passages with a strong vocal (singing voice or chorus) foreground.

Next, we analyzed the influence of the H+N decomposition on the system's performance. We found that this operation does not affect in the same way all the musical genres. In fact, it was encouraging to discover that this decomposition was particularly productive when dealing with difficult instances such as classical music. Our system was also compared to the well-known algorithm proposed by Scheirer (1998).

We also studied the influence of the harmonic and noise components, when processed separately and the results were compared. We did not find any convincing evidence supporting that a specific component is more influent on a particular genre.

Researchers working on automatic rhythm analysis have wondered in many occasions about the influence of the frequency decomposition on the system's performance. To address this question, we tested three different frequency structures: the uniform and logarithmic filter banks proposed in §3.3.1.3 and another variant which proceeds directly to the musical stress estimation without decomposing the signal in subbands. The results obtained suggest that our proposal is not sensitive to a particular frequency decomposition, however we found that splitting the audio signal in subbands yields better results.

For some experiments, the results on both databases were noticeably different which raises the question about the universality of these results. However, we believe that this contrast is merely due to the discrepancy in size between the instances in the TUT and ENST test corpora. As a matter of fact, performance significantly raises when longer and stable musical excerpts are used.

The numerical complexity is a significant characteristic of any computer-based algorithm. In this chapter we have measured the computational load in terms of execution time for three different system configurations. We found that under the current implementation, estimating the stress profile is one of the most time consuming stages. However, the most computationally demanding block is the H+N decomposition when the EDS method is used.

Finally beat location techniques and tatum estimation were tested. Highly encouraging results were obtained in these tasks, but further research is required to obtain more robust estimations.

Chapter 6

Concluding remarks and perspectives

This dissertation was devoted to the development of mechanisms to make computers "understand" certain aspects of musical rhythm. The goal is to create a system who uses as input commercial audio recordings and whose output should be comparable to the response produced by a human listener acquainted with traditional Western music if she/he was asked to *tap* along with a musical passage.

6.1 Conclusions

The first part of this dissertation focused on the presentation of fundamental concepts, namely: a suitable definition of musical rhythm and the notion of metrical structure coupled to the central abstraction of metrical levels.

Then, an introduction to the field of computer based rhythm analysis was provided. We highlighted that the primary goal of this area consists in developing a system to artificially replicate the process by which humans understand musical rhythm. An important contribution of this thesis is the comprehensive survey on the current state of automatic rhythm analysis. We classified the existing approaches according to the nature of their input signal into two broad categories: *symbolic models* and *acoustic models*. A review of many existing methods for both kind of approaches was presented, although more emphasis was given to the second kind of models since they directly concern the objectives of the present work. Table 2.1 gives a highly illustrative panorama. It presents a rather inclusive list of the available acoustic models and evidences how the field has considerably grown during the last years. As a matter of fact, the current trend is towards dealing with audio recordings rather than processing symbolic signals.

The evaluation problem was also addressed and special attention was given to the recent initiative of systematic benchmarking and comparison of tempo induction algorithms headed by a group of researchers working on Music Information Retrieval (see Appendix A). The composition of the evaluation material used in this work was also discussed. Actually this database has considerably evolved during the last years.

We have exhaustively covered the question of measuring the degree of musical accent as a function of time. In our opinion, this is by far the most complex task to accomplish during the development of an automatic rhythm description system. We have proposed a novel method to cope with challenging sounds. It is based on the idea of separating the audio signal in harmonic and noise parts[1] (H+N). The motivation to conduct this separation is to highlight phenomenal accents by separating them from the surrounding and potentially disturbing events. Two different methods were used to carry out this decomposition: one uses a subspace analysis technique that exploits the Exponentially Damped Sinusoidal (EDS) model; the other one is based on a more traditional Fourier-based approach.

In addition, this research work also contributed to the computer music field with a major improvement of a previously existing technique called the Spectral Energy Flux (SEF), which measures the degree of change of the power spectrum as a function of time. We have discovered that obtaining a proper estimation of the power envelope inside the frequency channels is central for energy-based onset detection techniques. For this purpose, a novel low-pass smoothing filter was proposed. The rationale behind this decision consist in using a smoothing filter which mimics the auditory nerve response to a sudden stimulus. Of considerable importance is also the use of a good differentiator filter. Instead of following the traditional approach, which computes a poor approximation of the power envelope derivative using the first order difference, we tested several differentiator filters and we opted for one with remarkable characteristics. The combination of these techniques (H+N coupled to SEF) produces a highly accurate estimation of the degree of musical stress as a function of time and proved to be pretty successful for a wide range of audio material.

A drawback of this method is that it has a somewhat elevated computational burden. However, it is possible to disable some of the components to reduce the complexity at the expense of reducing its efficiency. For example, when processing strong beat music, a fairly accurate detection function can be obtained without using the H+N decomposition. Another way to reduce even more the computational burden consists in using a traditional STFT instead of a reassigned version.

Then we addressed the problem of estimating some metrical aspects of the audio signal from the musical stress profile. To estimate the underlying periodicities four different methods were used: the autocorrelation function, a bank of comb-filter resonators, the spectral sum and the spectral product. To process the output of the periodicity induction stage, an efficient pulse tracking module based on the dynamic programming algorithm was developed. Although this method has already been proposed in context of metrical analysis, we are the first to propose a modified version capable of tracking the most salient periods of the musical stress profile.

At low frequencies (<10 Hz), only part of these tracked periodicities correspond to valid metrical levels, while the others correspond to *aliases* intrinsically created by the periodicity induction algorithms. An open and highly sensitive question consists in selecting from this list of pulsations only those containing significant metrical information and to discard the others. However, this process requires a high-level musical knowledge which is not embedded into the current system.

In the present work we estimated the main tempo by weighting the saliences of the most important periodicity paths using an *a priori* knowledge of the tempo preferred by humans. The principal drawback of this approach is that it does not exploit the hierarchical relationship between metrical levels. Indeed, while periodicity tracking can be done separately for each path, their linking into metrical levels should be carried out jointly.

We proposed two methods to locate the position of beats inside the audio signal.

[1] By ”noise” we refer to all the elements from the original audio signal that cannot be modeled as sinusoidal components.

The first one is based on the idea of predicting beat positions. The second approach was *borrowed* from the field of chemometric analysis (where it is used to conduct pattern matching) and works by aligning an artificial pulse train with the detection function. This last method displayed encouraging results.

During the development of a tatum estimation algorithm we have discovered that the shape of the detection function peaks has a large influence on the behavior of the respective Power Spectral Density (PSD). In fact, we consider that the influence of the peak-shape is even larger than the effect provoked by small fluctuations in the timing of the musical piece. Our tatum estimation algorithm is based on the idea of pruning and modifying the amplitude of the peaks present in the detection function. The results of preliminary tests suggest that this method has a large potential.

After numerous evaluation tests, we consider the overall performance of our metrical analysis system as satisfactory. First, we investigated the influence of the analysis window length and overlapping factor[2]. For the case of our rhythm analysis system, we have shown that medium-size windows ($\approx$5 s) with a relatively important overlapping factor (60%) yields good results. Although using very long analysis windows with a large overlapping factors seems to yield a small gain, it also requires an important increase in computational complexity. Moreover, such configuration might be counter productive for the tracking capabilities of the system since it entails a large rhythmic *inertia*.

Then, a more detailed analysis by musical genre was presented. The proposed system displays highly acceptable results when dealing with ”strong beat” music, however performance drops noticeably when processing classical music.

Concerning the harmonic plus noise (H+N) decomposition, for most cases it does not seem to provide a significant gain in performance to adequately justify the considerable increase in computational complexity. In fact, for percussion-based music this technique does not seem to contribute. However, we believe that the gain displayed when processing classical music (6.4% for the EDS H+N), which is a particularly complicated genre, justifies its use when dealing with more challenging music material not driven by percussive instruments. In addition, we did not find any convincing evidence supporting that a specific signal component (*i.e.*, harmonic or noise) has more influence on the performance.

With respect to the frequency decomposition, we presume that the system developed in the present work is not sensitive to the use of a particular filter bank. As suggested by the results, we consider that the important is to create some sort of redundancy by decomposing the signal in frequency bands. In that case, if artifacts appear in one or more subbands, they will only affect the stress profile computed using information from those frequency regions. Ideally, the nuisance of the artifacts should be canceled when fusing the periodicity information coming from all subbands.

The framework that we have proposed can be seen as modular and scalable system and can be adapted in function of the music material to be processed or in terms of the computational requirements/limitations. If the causal preprocessing scheme is selected, the system described can be used for on-line real-time tempo estimation even at a reasonable computational cost if the H+N block is disabled.

[2] In fact, a number of methods in the literature fix those parameters in a rather arbitrarily manner, thus we wanted to examine their real effect on the performance.

6.2 Perspectives

There are many potential ways of extending the system proposed in this work. In addition, there are some tasks that should be fulfilled.

- ⋆ More research is required to measure the influence of decomposing the signal into harmonic and noise parts. In fact, the two methods currently employed do not completely remove the sinusoids from the noise part. It is possible that this imperfect separation hampers the advantages of this method. Both approaches can be enhanced to improve the separation.

- ⋆ The bottleneck of our framework lies on the musical stress estimation stage. Indeed, this block is the core of the whole system. An erroneous and implicit assumption of the current Spectral Energy Flux (SEF) algorithm is the calculation of the envelope and its derivative only using information from one STFT channel at the time. Due to their non-stationary nature, the frequency components in audio signals hop from one channel to the other at any moment. To take account of this effect, and significantly reduce false detections, the energy flux should be calculated on frequency trajectories rather than on frequency channels. One potential solution to this problem is to use a pitch tracker before computing the flux. Nevertheless, current state-of-the-art in frequency tracking may not allow to satisfactory get around this problem.

- ⋆ Although we are among the first to propose a mechanism to fuse periodicity information into a single periodicity vector (from the subbands and/or the H+N components), the current approach is rather crude. A machine learning perspective can be used to establish if a given subband is informative or if it should be discarded. This selection process could improve the periodicity profile.

- ⋆ High-level musical knowledge must be embedded into the system. So far we have barely exploited all the information provided by the tracking algorithm. By adding new stages making use of this periodicity information combined with *a priori* knowledge, it would be possible to extend the current system capabilities. For example, to find the appropriate beat/meter subdivision, to jointly estimate the metrical levels and so forth.

- ⋆ The use of high-level musical knowledge combined with statistical classification methods can be used to categorize music instances according to their specific rhythmic patterns. This would allow to conduct rhythm-adapted metrical analysis.

- ⋆ The beat location stage could also exploit certain aspects of high-level musical knowledge such as pitch or harmony changes. As already suggested in the literature by Goto & Muraoka (1997b) and Dixon & Cambouropoulos (2000), such information would help to avoid tracking the offbeat. In addition, the beat alignment algorithm based on the Correlation Optimized Warping method should be implemented to work on shorter windows rather than on the whole signal.

- ⋆ It is practically impossible to know the effectiveness of any rhythm analysis systems without a large and well annotated test corpus. Research advancement in the field of automatic rhythm analysis (and in its applications, such as MIR) is clearly limited by the lack of public ground-truth data. However, the annotation of musical excerpts is a painstaking job that should be repeated by many people in order to have highly reliable information. In the particular case of this research, some of the issues that could be immediately addressed with new and/or better annotations are the following.

- ◇ The criterion defined as *Accuracy 2* should be refined to take into account the beat/meter subdivision (*i.e.*, simple duple, compound duple, simple triple and compound triple). In that case, the current system performance would surely be reduced, however the results would be more realistic.
- ◇ The automatic rhythm analysis community should "move on" from processing short signal segments (tens of seconds) to process entire songs of several minutes. This action would approach current research a step closer to real-life applications. In addition, we have seen that the amount of data available during the analysis might play a significant role in system's performance. This problem has not been address in the literature and could reveal interesting results.

Appendix A

Methodological benchmarking of tempo extraction algorithms

The first step towards establishing a methodology of evaluation and comparison for tempo induction algorithms, along with several other Music Information Retrieval (MIR) tasks, was taken in 2004 by the steering committee of the International Conference on Music Information Retrieval (ISMIR). Although called ”Audio Description Contest” at that time, the year after it officially became the ”Music Information Retrieval Evaluation eXchange”, or more succinctly MIREX[1]. This is a contest run in conjunction with the ISMIR Conference and it emerged as a response to the intention of the MIR community to establish formal evaluation frameworks and metrics with which researchers could scientifically compare and contrast their wide variety of approaches to solving MIR tasks (Downie et al., 2005).

A.1 Tempo induction competition during the first Audio Description Contest

The first ”tempo extraction contest” was organized by Fabien Gouyon during ISMIR 2004, held at the University Pompeu Fabra in Barcelona, Spain, in October 2004. Specific details and information are provided in (Gouyon et al., 2006). The goal of this contest was to evaluate some state-of-the-art algorithms in the task of inducing the basic tempo (as a scalar, in beats per minute) from musical audio signals. It was not required to find individual beat positions or any other rhythmic description.

Participants were invited to submit algorithms to the contest organizer. No training data was provided. A total of 12 algorithms (representing the work of seven research teams) were evaluated. We submitted two closely related entries described in (Alonso et al., 2004). The test database contained 3199 annotated instances[2] in three different data sets: 2036 sound library drum loops, 698 excerpts of ballroom dance music (styles like cha-cha, rumba, samba, tango) and 465 song excerps containing various music genres.

Table A.1 presents the results for this contest ordered by rank.

[1] Further information on MIREX is available at http://www.music-ir.org/mirexwiki/index.php.

[2] Part of this data is available for free download at http://ismir2004.ismir.net/ISMIR_Contest.html.

Rank	Participant	Score (in %)
1	Klapuri et al. (2006)	85.0
2	Dixon et al. (2003)	82.3
3	*Anonymous*	81.2
4	Uhle et al. (2004)	76.1
5	Dixon (2001)	74.3
6	Dixon (2001)	73.6
7	**Alonso et al. (2004)**	**69.8**
8	Scheirer (1998)	68.1
9	**Alonso et al. (2004)**	**57.9**
10	Tzanetakis & Cook (2002)	55.5
11	Tzanetakis & Cook (2002)	54.7
12	Tzanetakis & Cook (2002)	50.7

Table A.1: Results of the first tempo extraction contest.

A.2 MIREX'05: Audio tempo extraction contest

In the second edition of the contest, officially known as MIREX, the organizers of the tempo extraction contest was organized by Martin McKinney & Dirk Moelants[3].

Contrary to the previous contest, in this edition there was a distinction between notated tempo and perceptual tempo. The goal was to test the different methods for the extraction of the perceptual tempo.

If the notated tempo is available (e.g., from the score) it is straightforward to attach a tempo annotation to an excerpt and run a contest for algorithms to predict this notated tempo. For excerpts for which there is no "official" tempo annotation, it is possible to annotate the "perceived" tempo. This is not a straightforward task and needs to be done carefully. If someone asks a group of listeners (including skilled musicians) to annotate the tempo of music excerpts, they can provide different answers (they tap at different metrical levels) if they are unfamiliar with the piece. For some excerpts the perceived pulse or tempo is less ambiguous and everyone taps at the same metrical level, but for other excerpts the tempo can be quite ambiguous and it is possible to get a complete split across listeners.

There are several reasons to examine the perceptual tempo, either in place of or in addition to the notated tempo. For many applications of automatic tempo extractors, the perceived tempo of the music is more relevant than the notated tempo. An automatic playlist generator or music navigator, for instance, might allow listeners to select or filter music by its (automatically extracted) tempo. In this case, the "feel", or perceptual tempo may be more relevant than the notated tempo. An automatic DJ apparatus might also perform better with a representation of perceived tempo rather than notated tempo. A more pragmatic reason for using perceptual tempo rather than notated tempo as a ground truth for our contest is that we simply do not have the notated tempo of our test set. If we notate it by having a panel of expert listeners tap along and label the excerpts, we are by default dealing with the perceived tempo. The handling of this data as ground truth must be done with care.

During this contest, the test database was much smaller and only contained 140 song

[3] A comprehensive description of the methodology of this contest can be found on the WWW at the address `http://www.music-ir.org/mirex2005/index.php/Audio_Tempo_Extraction`

excerpts. On the other side, the ground-truth was highly reliable since it was obtained after averaging the annotations carried about by dozens of listeners.

The competition consisted in four different tasks:

i) extraction of the most salient perceptual tempo, T_1 (scalar in BPMs),

ii) extraction the second most salient perceptual tempo, T_2 (scalar in BPMs),

iii) temporal location of the first beat for T_1,

iv) temporal location of the first beat for T_2.

Table A.2 presents the results for this contest ordered by rank[4]. Algorithm information for those participants without reference can be consulted on the contest web page.

Rank	Participant	Score
1	**Alonso et al. (2005b)**	**0.689**
2	Uhle, C.	0.675
3	Uhle, C.	0.675
4	Gouyon & Dixon (1)	0.670
5	Peeters (2005)	0.656
6	Gouyon & and Dixon (2)	0.649
7	Gouyon & Dixon (4)	0.645
8	Eck & Casagrande (2005)	0.644
9	Davies & Brossier	0.628
10	Gouyon & Dixon (3)	0.607
11	Sethares	0.597
12	Brossier	0.583
13	Tzanetakis	0.538

Table A.2: Results of the second tempo extraction contest.

[4] A comprehensive description of the results can be found on the WWW at the address: `http://www.music-ir.org/evaluation/mirex-results/audio-tempo/index.html`.

Appendix B

Formulae of numerical differentiation

In solving many mathematical and physical problems by means of numerical methods one is often challenged to seek derivatives of various functions given in discrete points. In such cases, when it is difficult or impossible to take derivative of a function analytically one resorts to numerical differentiation. This appendix briefly resumes the so-called "formulae of numerical differentiation" developed by Dvornikov (2003).

Without loss of generality we suppose that the derivative is taken in the zero point, *i.e.*, $x_0 = 0$. Let us consider the discrete function $f(x)$ given in equidistant points $x_m = \pm mh$, where $m = 0, \dots, n$ and h is a constant value. It is possible to pass a polynomial of order $2n$ through these points

$$P_{2n}(x) = \sum_{k=0}^{2n} c_k x^k, \tag{B.1}$$

where the values of the function coincide with the interpolation points: $P_{2n}(x_m) = f_m$ and $f_m = f(x_m)$. Let us define as d_m the differences of the values of the function $f(x)$ in diametrically opposite points x_m and x_{-m}, *i.e.*, $d_m = f_m - f_{-m}$. We can write d_m in the form

$$d_m = 2 \sum_{k=0}^{n-1} c_{2k+1} h^{2k+1} m^{2k+1}. \tag{B.2}$$

To find the coefficients c_{2k+1}, $k = 0, \dots, n-1$, we must solve a system of inhomogeneous linear equations using the terms d_m. Instead of solving Eq. (B.2) directly, we proceed as follows

$$c_{2k+1} = \frac{1}{2h^{2k+1}} \sum_{m=1}^{n} d_m \alpha_m^{(2k+1)}(n), \tag{B.3}$$

where the $\alpha_m^{(2k+1)}(n)$ are the undetermined coefficients satisfying the condition

$$\sum_{m=1}^{n} \alpha_m^{(2l+1)}(n) m^{2k+1} = \delta_{lk}, \tag{B.4}$$

where $l, k = 0, \dots, n-1$. The system of Eq. (B.4) is equivalent to that of Eq. (B.2), but simpler to solve, in which for each $k = 0, \dots, n-1$ it is necessary to find the coefficients

$\alpha_m^{(2l+1)}(n)$. This system can be solved according to Cramer's rule:

$$\alpha_m^{(2l+1)}(n) = \frac{\Delta_m^{(2l+1)}(n)}{\Delta_0(n)} \tag{B.5}$$

where

$$\Delta_0(n) = \begin{vmatrix} 1 & 2 & \cdots & n \\ 1 & 2^3 & \cdots & n^3 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{2n-1} & \cdots & n^{2n-1} \end{vmatrix} = n! \prod_{1\le i\le j\le n} (j^2 - i^2) \neq 0 \tag{B.6}$$

and

$$\Delta_m^{(2l+1)}(n) = \begin{vmatrix} 1 & 2 & \cdots & m-1 & 0 & m+1 & \cdots & n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{2l+1} & \cdots & (m-1)^{2l+1} & 0 & (m+1)^{2l+1} & \cdots & n^{2l+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{2n-1} & \cdots & (m-1)^{2n+-1} & 0 & (m+1)^{2n-1} & \cdots & n^{2n-1} \end{vmatrix}. \tag{B.7}$$

In Eqs. (B.6) and (B.7) we used the formula for the calculation of the Vandermonde determinant. The simplest expression for $\Delta_m^{(2l+1)}(n)$ is obtained for $l = 0$, which corresponds to a calculation of the first-order derivative

$$\Delta_m^{(1)}(n) = (-1)^{m+1} \left(\frac{n!}{m}\right) \prod_{\substack{1\le i\le j\le n \\ i,j\neq m}} (j^2 - i^2). \tag{B.8}$$

From Eq. (B.5) and using Eqs. (B.6)–(B.8) we obtain the expression for the coefficients

$$\alpha_m^{(1)}(n) = \frac{1}{m\pi_m(n)}, \tag{B.9}$$

where

$$\pi_m(n) = \prod_{\substack{k=1 \\ k\neq m}}^{n} \left(1 - \frac{m^2}{k^2}\right). \tag{B.10}$$

Finally, using Eqs. (B.1)–(B.3) we get the formula for the first derivative of the discrete function $f(x)$

$$f'(0) \approx P'_{2n}(0) = \frac{1}{2h} \sum_{m=1}^{n} \alpha_m^{(1)}(n)(f_m - f_{-m}). \tag{B.11}$$

Appendix C

Perceptual weighting filter: ITU-R ARM

It has been known for years that human hearing does not have the same response at all frequencies in the audible range (Fletcher & Munson, 1933). The purpose of weighting filters is to account for this issue. In fact, this kind of filters are designed to *weight* or give more attention to the upper midrange frequency region, where hearing is more sensitive. The goal is to obtain measurements that correlate well with the subjective perception of sound intensity[1].

For the present work, instead of using the widespread and more familiar A-weighting curve (IEC268-1, 1968) we opted for a more recent and perceptually plausible weighting curve usually known as the "ITU-R ARM" (average response meter) derived from the standard ITU-R468 (1986). According to its developers (Dolby et al., 1979) this weighting curve has better agreement with subjective assessments and is widely employed in professional and commercial audio equipment. The electrical circuit to physically implement this filter is presented in Figure C.1. After computing the respective transfer function we obtained the following equation:

$$\mathcal{W}(s) = \frac{N_1 s}{(s^2 + D_1 s + D_2)(s^2 + D_3 s + D_4)(s^2 + D_5 s + D_6)} \tag{C.1}$$

where $s = \jmath 2\pi f$ represents the complex frequency, and the values of the constants are:

$$\begin{array}{lll} N_1 = 1.05973883e24 & D_1 = 8.85788709e04 & D_2 = 1.62351886e09 \\ D_3 = 4.72310705e04 & D_4 = 1.88119128e09 & D_5 = 3.74874935e04 \\ D_6 = 4.25259918e09 & & \end{array}$$

The frequency shape of this weighting filter can seen on page 89.

[1] However, this sensitivity variation is not only frequency dependent, but also depends on sound intensity. For this reason, some researchers consider that the idea that a single filter can represent this phenomenon at all intesity levels is wrong.

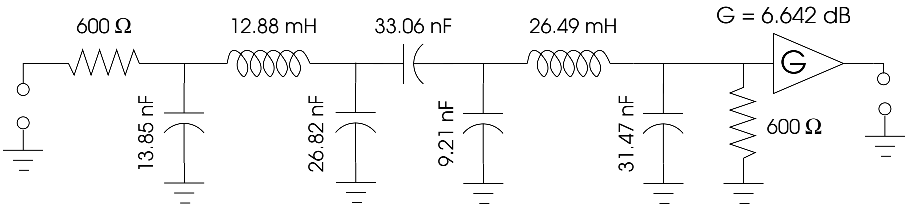


Figure C.1: Circuit of the Weighting Filter ITU-R ARM.

Appendix D

Numerical values of the results given in Chapter 5

This Appendix presents the numerical values of the results presented in Chapter 5. Unless otherwise stated, the figure that appear the tables in of this Appendix were computed using the *Accuracy 2* criterion.

On the impact of the window length parameter

ℓ / Method	**3 s**	**4 s**	**5 s**	**6 s**	**7 s**	**8 s**
SS	89.9	91.5	91.3	91.6	90.7	91.1
SP	88.0	89.5	90.4	90.8	90.0	90.6
AC	89.3	88.9	89.1	89.4	87.6	88.1
CF	83.8	83.0	82.9	82.4	82.1	82.5
MF	90.5	91.1	91.2	91.5	90.1	90.6

Table D.1: Accuracies (in %) by periodicity method depending on the window length (ℓ) value (in seconds) for the ENST database.

ℓ / Method	**3 s**	**4 s**	**5 s**	**6 s**	**7 s**	**8 s**
SS	88.2	90.9	91.4	91.6	92.0	91.8
SP	86.5	90.7	91.8	92.0	92.4	93.9
AC	89.5	88.4	88.6	88.4	89.9	89.2
CF	74.7	75.7	75.9	75.9	75.9	75.7
MF	89.2	90.7	92.0	91.6	92.2	92.6

Table D.2: Accuracies (in %) by periodicity method depending on the window length (ℓ) value (in seconds) for the TUT database.

On the impact of the overlapping parameter

ρ / Method	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
SS	91.1	91.7	90.9	91.5	91.5	91.3	92.1	91.9	92.1	91.8
SP	88.9	89.4	89.2	89.9	90.1	90.4	90.4	90.8	91.7	91.5
AC	88.8	88.8	89.0	88.6	89.1	89.1	88.9	88.8	89.3	88.4
CF	82.4	82.0	82.1	82.1	82.3	82.9	83.0	83.2	83.9	83.5
MF	90.5	91.4	90.5	91.3	90.7	91.2	91.2	91.2	92.0	91.2

Table D.3: Accuracies (in %) by periodicity method depending on the overlapping factor ρ as a function of the window length (figures computed for $\ell = 5$). Values obtained on the ENST database.

ρ / Method	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
SS	90.3	91.4	90.9	91.6	91.4	91.4	91.8	90.9	89.9	91.1
SP	90.5	92.2	90.7	89.9	92.0	91.8	91.4	93.0	92.0	91.6
AC	89.0	88.4	88.2	89.9	88.4	88.6	88.2	88.8	88.6	87.6
CF	76.2	75.7	75.3	76.2	75.9	75.9	76.8	75.3	74.7	75.1
MF	90.9	92.2	90.7	92.0	91.4	92.0	92.2	91.1	91.4	91.6

Table D.4: Accuracies (in %) by periodicity method depending on the overlapping factor ρ as a function of the window length (figures computed for $\ell = 5$). Values obtained on the TUT database.

Database	**Method**					**Criterion**
	SS	SP	AC	CF	MF	
ENST	47.8	43.7	38.5	47.6	46.6	*Accuracy 1*
TUT	52.1	49.8	40.7	58.7	56.1	
ENST	92.1	90.4	88.9	83.0	91.2	*Accuracy 2*
TUT	91.8	91.4	88.2	76.8	92.1	

Table D.5: Performance (in %) by periodicity method using both criteria.

Efficacy of the system by musical genre

Method	Genre										
	classical	**jazz**	**latin**	**pop**	**rock**	**reggae**	**soul**	**hip-hop**	**techno**	**other**	**greek**
SS	41.4	62.5	50.4	61.5	45.1	26.7	62.5	35.0	42.9	51.0	41.4
SP	40.4	62.5	45.2	58.3	45.1	30.0	29.2	30.0	42.9	46.9	30.0
AC	30.5	68.2	46.1	53.1	45.1	26.7	37.5	20.0	26.8	39.8	20.0
CF	33.5	63.6	54.8	68.8	49.5	46.7	87.5	70.0	58.9	59.2	13.6
MF	39.9	71.6	53.0	61.5	45.1	33.3	54.2	35.0	42.9	55.1	25.0

Table D.6: Performance (in %) by genre using the *Accuracy 1* criterion. Results obtained on the ENST database.

Method	Genre						
	classical	**electronic**	**hip-hop**	**jazz/blues**	**rock/pop**	**soul/funk**	**other**
SS	34.5	77.3	29.7	56.4	55.6	50.0	46.7
SP	38.1	71.2	18.9	56.4	53.2	40.7	60.0
AC	39.3	34.8	10.8	50.0	48.4	35.2	46.7
CF	33.3	66.7	67.6	60.6	64.5	64.8	60.0
MF	39.3	74.2	32.4	63.8	62.9	46.3	60.0

Table D.7: Performance (in %) by genre using the *Accuracy 1* criterion. Results obtained on the TUT database.

Method	Genre										
	classical	**jazz**	**latin**	**pop**	**rock**	**reggae**	**soul**	**hip-hop**	**techno**	**other**	**greek**
SS	86.2	96.6	92.2	97.9	98.9	100.0	100.0	100.0	96.4	98.0	79.3
SP	85.2	95.5	89.6	97.9	97.8	100.0	100.0	100.0	94.6	95.9	75.0
AC	85.2	95.5	88.7	96.9	97.8	93.3	100.0	95.0	87.5	87.8	76.4
CF	75.4	92.0	77.4	93.8	87.9	93.3	100.0	95.0	89.3	84.7	72.1
MF	89.2	95.5	89.6	97.9	98.9	93.3	100.0	100.0	89.3	95.9	77.1

Table D.8: Performance (in %) by genre using the *Accuracy 2* criterion. Results obtained on the ENST database.

Method	Genre						
	classical	**electronic**	**hip-hop**	**jazz/blues**	**rock/pop**	**soul/funk**	**other**
SS	73.8	95.5	97.3	94.7	94.4	100.0	93.3
SP	71.4	95.5	97.3	96.8	94.4	98.1	86.7
AC	72.6	86.4	86.5	91.5	96.0	92.6	86.7
CF	57.1	74.2	86.5	81.9	78.2	94.4	66.7
MF	76.2	92.4	97.3	95.7	94.4	100.0	100.0

Table D.9: Performance (in %) by genre using the *Accuracy 2* criterion. Results obtained on the TUT database.

Influence of the pre-processing and the H+N decomposition method

The meaning of the initials in the tables shown below is the following: the names NC and C refer to the pre-processing scheme (see §3.2) and stand for "non-causal" and "causal" respectively. The names EDS and FT stand for "Exponetially Damped Sinusoid" and "Fourier Transform" respectively and refer to harmonic-plus-noise (H+N) decomposition method (see §3.3).

Configuration	Method					
	SS	**SP**	**AC**	**CF**	**MF**	**Scheirer**
NC-EDS	92.1±1.7	90.4±1.9	88.9±2.0	83.0±2.4	91.2±1.8	72.4±2.8
NC-FT	91.3±1.8	89.1±2.0	89.9±1.9	81.6±2.5	90.4±1.9	
C-EDS	90.5±1.9	87.7±2.1	86.3±2.2	80.7±2.5	88.8±2.0	
C-FT	89.5±1.9	89.2±2.0	86.8±2.1	79.4±2.6	89.3±2.0	
Without H+N	89.8±1.9	88.8±2.0	86.8±2.1	78.6±2.6	89.1±2.0	

Table D.10: On the impact in the performance (in %) of different pre-processing and H+N decomposition configurations. The success rate of Scheirer's algorithm is also presented. Figures computed on the ENST database.

Configuration	Method					
	SS	**SP**	**AC**	**CF**	**MF**	**Scheirer**
NC-EDS	91.8±2.5	91.4±2.5	88.2±2.9	76.8±3.8	92.2±2.4	79.1±3.7
NC-FT	90.5±2.6	89.5±2.8	88.8±2.8	72.8±4.0	91.1±2.6	
C-EDS	90.5±2.6	91.1±2.6	88.4±2.9	75.7±3.9	90.9±2.6	
C-FT	90.7±2.6	90.5±2.6	88.2±2.9	72.8±4.0	90.9±2.6	
Without H+N	88.6±2.7	89.5±2.8	90.3±2.7	72.2±4.0	90.9±2.6	

Table D.11: Similar as Table D.10, but using the TUT database.

Database	Config.	SS	SP	AC	CF	MF
ENST	EDS	86.2±4.7	85.2±4.9	85.2±4.9	75.4±5.9	89.2±4 .3
	FT	83.7±5.1	78.8±5.6	84.7±4.9	71.4±6.2	84.2±5.0
	No H+N	80.8±5.4	75.9±5.9	81.3±5.4	64.5±6.6	80.8±5.4
TUT	EDS	73.8±9.4	71.4±9.7	72.6±9.5	57.1±10.6	76.2±9.1
	FT	66.7±10.1	61.9±10.4	73.8±9.4	48.8±10.7	72.6±9.5
	No H+N	63.1±10.3	65.5±10.2	77.4±8.9	48.8±10.7	70.2±9.8

Table D.12: Influence of the H+N decomposition on classical music. These figures were computed using the non-causal preprocessing.

Configuration	Method				
	SS	SP	AC	CF	MF
EDS signal	90.3±1.9	88.0±2.1	85.2±2.2	81.5±2.5	90.1±1.9
EDS noise	87.7±2.1	84.9±2.3	85.5±2.2	80.2±2.5	87.8±2.1
FT signal	88.1±2.0	86.7±2.1	86.8±2.1	78.0±2.6	88.4±2.0
FT noise	91.1±1.8	88.0±2.1	87.7±2.1	78.1±2.6	89.8±1.9
Without H+N	89.8±1.9	88.8±2.0	86.8±2.1	78.6±2.6	89.1±2.0

Table D.13: On the influence (in %) of the signal and noise parts for both H+N decomposition methods. Figures obtained on the ENST database.

Configuration	Method				
	SS	SP	AC	CF	MF
EDS signal	91.6±2.5	92.0±2.4	88.2±2.9	74.7±3.9	91.8±2.5
EDS noise	89.2±2.8	90.1±2.7	86.9±3.0	70.9±4.1	89.9±2.7
FT signal	90.7±2.6	88.8±2.8	86.9±3.0	73.6±4.0	90.5±2.6
FT noise	90.7±2.6	91.4±2.5	87.3±3.0	70.7±4.1	91.6±2.5
Without H+N	88.6±2.9	89.5±2.8	90.3±2.7	72.2±4.0	90.9±2.6

Table D.14: Same as Table D.13 but using the TUT database.

Configuration	Method				
	SS	SP	AC	CF	MF
EDS UFB	92.1±1.7	90.4±1.9	88.9±2.0	83.0±2.4	91.2±1.8
FT UFB	91.3±1.8	89.1±2.0	89.9±1.9	81.6±2.5	90.4±1.9
EDS logFB	92.0±1.7	90.5±1.9	88.4±2.0	81.8±2.4	90.7±1.8
FT logFB	90.9±1.8	88.7±2.0	88.7±2.0	79.4±2.6	90.0±1.9

Table D.15: On the influence of the filter bank and the H+N decomposition method. Figures obtained using the ENST database.

Configuration	Method				
	SS	SP	AC	CF	MF
EDS UFB	91.8±2.5	91.4±2.5	88.2±2.9	76.8±3.8	92.2±2.4
FT UFB	90.5±2.6	89.5±2.8	88.8±2.8	72.8±4.0	91.1±2.6
EDS logFB	92.2±2.4	92.2±2.4	90.5±2.6	74.5±3.9	92.6±2.4
FT logFB	88.8±2.8	90.5±2.6	88.2±2.9	72.8±4.0	89.0±2.8

Table D.16: Same as Table D.15 but using the TUT database.

Database	Config.	SS	SP	AC	CF	MF
ENST	**EDS U-FB**	86.2±7.2	85.2±7.4	85.2±7.4	75.4±9.0	89.2±6.5
	FT U-FB	83.7±7.7	78.8±8.5	84.7±7.5	71.4±9.4	84.2±7.6
	EDS log-FB	88.2±6.7	85.7±7.3	88.2±6.7	70.9±9.5	89.2±6.5
	FT log-FB	83.7±7.7	77.8±8.7	83.3±7.8	69.0±9.7	84.2±7.6
TUT	**EDS U-FB**	73.8±9.4	71.4±9.7	72.6±9.5	57.1±10.6	76.2±9.1
	FT U-FB	66.7±10.1	61.9±10.4	73.8±9.4	48.8±10.7	72.6±9.5
	EDS log-FB	77.4±8.9	76.2±9.1	79.8±8.6	52.4±10.7	84.5±7.7
	FT log-FB	69.0±9.9	72.6±9.5	76.2±9.1	57.1±10.6	72.6±9.5

Table D.17: Influence of the filter bank on classical music.

Configuration	Method				
	SS	**SP**	**AC**	**CF**	**MF**
UFB	89.8±1.9	88.8±2.0	86.8±2.1	78.6±2.6	89.1±2.0
logFB	90.0±1.9	88.3±2.0	86.5±2.2	78.1±2.6	89.8±1.9
Single-band	85.3±2.2	83.8±2.3	82.9±2.4	76.7±2.7	84.6±2.3

Table D.18: On the influence of the filter bank without H+N method. Figures obtained using the ENST database.

Configuration	Method				
	SS	**SP**	**AC**	**CF**	**MF**
UFB	88.6±2.9	89.5±2.8	90.3±2.7	72.2±4.0	90.9±2.6
logFB	89.2±2.8	89.5±2.8	89.5±2.8	71.7±4.1	89.0±2.8
Single-band	86.2±3.1	87.6±3.0	84.5±3.3	66.5±4.2	86.4±3.1

Table D.19: Same as Table D.18 but using the TUT database.

Configuration	Method				
	SS	**SP**	**AC**	**CF**	**MF**
SEF	89.8±1.9	88.8±2.0	86.8±2.1	78.6±2.6	89.1±2.0
CSD	79.2±2.6	76.4±2.7	79.4±2.6	68.0±3.0	79.1±2.6
SD	77.1±2.7	75.5±2.7	73.6±2.8	69.6±2.9	77.8±2.6

Table D.20: Comparing algorithms to compute the musical stress profiles. The Spectral Energy Flux (SEF), the Complex Spectral Difference (CSD) and the Spectral Difference (SD).Figures obtained using the ENST database.

Configuration	Method				
	SS	**SP**	**AC**	**CF**	**MF**
SEF	88.6±2.9	89.5±2.8	90.3±2.7	72.2±4.0	90.9±2.6
CSD	85.9±3.2	84.2±3.4	85.3±3.3	67.8±4.2	86.8±3.1
SD	84.2±3.3	86.3±3.1	81.5±3.5	72.2±4.0	85.5±3.2

Table D.21: Same as Table D.20, but using the TUT database.

Stage	Configuration					
	UB-EDS		LB-EDS		UB-FT	
	iterations	time (s)	iterations	time (s)	iterations	time (s)
Preprocess	×1	1.2	×1	0.8	×1	1.5
Filter bank	×1	0.7	×1	0.5	×2	1.8
H+N	×8	38.0	×5	52.0	×1	18.6
Musical stress	×16	54.6	×10	43.8	×16	67.5
Periodicity	×16	5.1	×10	2.6	×16	7.6
Tracking	×1	0.4	×1	0.3	×1	0.4

Table D.22: Computation time.

Appendix E

SD and CSD algorithms to compute the musical stress profile

In this section we present a brief description of the Spectral Difference (SD) and Complex Spectral Difference (CSD) algorithms used in §5.6. Both of them share a common general approach based on measuring the rate of change of the power-spectrum. In fact, these methods compute the detection function as a ”distance” between successive short-term Fourier spectra, treating them as points in an N-dimensional space.

E.3 Spectral Difference

The Spectral Difference procedure has been proposed by Masri (1996) and later also by Duxbury et al. (2002). This method relies on the assumption that the introduction of a new acoustic event leads to an increase in the energy of the signal. This energy method has proved to be popular since it is straightforward and it has a low computational cost. Besides, it is efficient to detect percussive notes. Let us consider the discrete time signal $x(n)$, its short time Fourier transform (STFT) is given by

$$X_k(m) = \sum_{n=0}^{N-1} g(n)x(n+m)e^{-\jmath\frac{2\pi}{N}kn} \tag{E.1}$$

where $g(n)$ is an analysis window, m is the time-frame index and k the frequency bin index. The SD algorithm is defined as:

$$d(m) = \sum_{k=0}^{N-1} \left(|X_k(m)| - |X_k(m-1)|\right)^2 \tag{E.2}$$

E.4 Complex Spectral Difference

The Complex Spectral Difference algorithm was developed by Bello et al. (2004). According to them, energy-based onset detection schemes perform well for pitched and nonpitched music with significant percussive content. On the other hand, phase-based onset detection approaches provide better results for strongly pitched signals (even for ”softer” onsets), while being less robust to distortions in the frequency content and to noise. In the complex domain, both phase and amplitude information work together, offering a generally more robust onset detection scheme. This method is also based on the STFT of $x(n)$.

Let us assume that during steady state regions of the audio signal, the magnitude spectrum should remain approximately constant. Then we can predict that the magnitude spectrum $\hat{R}_k$ at frame m is given by the magnitude of the previous frame

$$\hat{R}_k(m) = R_k(m-1) \tag{E.3}$$

where $R_k(m) = |X_k(m)|$. In addition we apply an assumption about the properties of the phase spectrum, that during steady state regions the phase velocity at the k^{th} frequency bin should ideally be constant.

$$\tilde{\phi}_k(m) - \tilde{\phi}_k(m-1) \approx \tilde{\phi}_k(m-1) - \tilde{\phi}_k(m-2) \tag{E.4}$$

We adopt a short-hand notation for the left hand side of equation $\Delta\tilde{\phi}_k(m) = \tilde{\phi}_k(m) - \tilde{\phi}_k(m-1)$. Then, rearranging terms in Eq. E.4, we can predict the phase of the k^{th} frequency bin for frame m given the observation of the two previous frames:

$$\hat{\phi}_k(m) = \text{princarg}[\tilde{\phi}_k(m-1) + \Delta\tilde{\phi}_k(m-1)] \tag{E.5}$$

where "princarg" unwraps the phase value, mapping it into the range $[-\pi, \pi]$. The predictions of the magnitude spectrum $\hat{R}_k(m)$ and the phase spectrum $\hat{\phi}_k(m)$ can be represented in polar form to give a spectral prediction $\hat{X}_k(m)$ in the complex domain

$$\hat{X}_k(m) = \hat{R}_k(m)e^{\jmath\hat{\phi}_k(m)} \tag{E.6}$$

which we compare to the observed complex spectrum $X_k(m)$. To derive the complex spectral difference (CSD) detection function $d(m)$ at we compute the sum of the Euclidean distance between the predicted and observed spectra for all k frequency bins, *i.e.*,

$$d(m) = \sum_{k=1}^{N-1} |X_k(m) - \hat{X}_k(m)|^2. \tag{E.7}$$

Appendix F

Publications

Conference papers

- ⋆ Alonso M., Richard G. and David B., ”Extracting note onsets from audio recordings”, IEEE International Conference on Multimedia & Expo (ICME), Amsterdam, The Netherlands. July 2005.
- ⋆ Alonso M., David B. and Richard G. ”Tempo and beat estimation of music signals”, Proc. of International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain. October 2004.
- ⋆ Alonso M., Badeau R., David B. and Richard G., ”Musical tempo estimation using noise subspace projections”, IEEE Workshop on applications of signal processing to audio and acoustics (WASPAA), New Paltz, New York. October 2003.
- ⋆ Alonso M., David B. and Richard G., ”A study of tempo tracking algorithms from polyphonic music signals”, 4-th European Cooperation in the field of Scientific and Technical Research (COST) Workshop, Bordeaux, France. March 2003.

Journal papers

- ⋆ Alonso M., Richard G., and David B., ”Accurate tempo estimation based on harmonic+noise decomposition”, *EURASIP Journal on Advances in Signal Processing (JASP)*, 2006 (in press).

  .
- ⋆ Gouyon F., Klapuri A., Dixon S., Alonso M., Tzanetakis G., Uhle C., and Cano P., ”An experimental comparison of audio tempo induction algorithms”, *IEEE transactions in Speech and Audio Processing*, Vol. 14(5), pp. 1832–1844, 2006.

Distinctions

- ⋆ Winner algorithm in the ”Audio Tempo Extraction” category during the *Music Information Retrieval Evaluation eXchange* (Mirex) contest. September 2005.

Bibliography

Abdallah, S. & Plumbley, M. D. (2003). Probability as metadata: event detection in music using ica as a conditional density model, *Proc. Int. Symp. Independent Component Analysis and Signal Separation (ICA)*.

Allen, P. E. & Dannenberg, R. B. (1990). Tracking musical beats in real time, *Proc. Int. Computer Music Conference (ICMC)*, Glasgow, Scotland, pp. 140–143.

Alonso, M., Badeau, R., David, B. & Richard, G. (2003a). Musical tempo estimation using noise subspace projections, *Proc. IEEE Workshop on App. Signal Proc. to Audio and Acoust. (WASPAA)*.

Alonso, M., David, B. & Richard, G. (2003b). Tempo tracking algorithms from polyphonic music signals, *4th COST 276 Workshop*.

Alonso, M., David, B. & Richard, G. (2004). Tempo and beat estimation of music signals, *Proc. Int. Symposium on Music Inf. Retriev (ISMIR)*.

Alonso, M., David, B. & Richard, G. (2005a). Tempo extraction for audio recordings, *Proc. Mirex*. `http://www.music-ir.org/evaluation/mirex-results/audio-tempo/index.html`.

Alonso, M., Richard, G. & David, B. (2005b). Accurate tempo estimation based on harmonic+noise decomposition, *EURASIP Journal on Applied Signal Processing*. in Press.

Alonso, M., Richard, G. & David, B. (2005c). Extracting note onsets from musical recordings, *Proc. IEEE Int. Conf. on Multimedia & Expo (ICME)*.

André-Obrecht, R. (1988). A new statistical approach for the automatic segmentation of continuous speech signals, *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP)* **36**(1): 29–40.

Auger, F. & Flandrin, P. (1995). Improving the readability of time–frequency and time–scale representations by the reassignment method, *IEEE Trans. on Signal Processing* **43**(5): 1068–1089.

Badeau, R. (2005). *Méthodes à haute résolution pour l'estimation et le suivi de sinusoïdes modulées. Application aux signaux de musique*, PhD thesis, ENST Télécom-Paris, Paris, France. in French.

Badeau, R., Boyer, R. & David, B. (2002). EDS parametric modeling and tracking of audio signals, *Proc. Int. Conference on Digital Audio Effects (DAFx)*.

Badeau, R., David, B. & Richard, G. (2006). A new perturbation analysis for signal enumeration in rotational invariance techniques, *EEE Transactions on Signal Processing* **54**(2): 450–458.

Bellman, R. E. & Dreyfus, S. E. (1962). *Applied Dynamic Programming*, Princeton University Press, Princeton, NJ.

Bello, J. (2003). *Towards the automated analysis of simple polyphonic music: A knowledge based approach*, PhD thesis, Queen Mary University of London, London, UK.

Bello, J. & Sandler, M. (2003). Phase-based note onset detection for music signals, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M. & Sandler, M. (2005). A tutorial on onset detection in music signals, *IEEE Trans. on Speech and Audio Processing* **13**(5): 1035–1047.

Bello, J. P., Duxbury, C., Davies, M. E. & Sandler, M. B. (2004). On the use of the phase and energy for musical onset detection, *IEEE Signal Processing Letters* **11**(6): 553–556.

Bilmes, J. A. (1993a). Techniques to foster drum machine expressivity, *Proc. Int. Computer Music Conference (ICMC)*.

Bilmes, J. A. (1993b). *Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressivetiming in percussive rhythm*, Master's thesis, Massachusetts Institute of Technology, Boston, MA., USA.

Bovik, A. C., Huang, T. S. & Munson, D. C. (1983). A generalization of median filtering using linear combinations of order statistics, *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP)* **31**(6): 1342–1350.

Carterette, E. C. & Kendall, R. A. (1999). Comparative music perception and cognition, *The psychology of music*, 2nd edition edn, Academic Press, pp. 725–791.

Cemgil, A., Kappen, B., Desain, P. & Honing, H. (2001). On tempo tracking:tempogram representation and Kalman filtering, *Journal of New Music Research* **28**(4): 259–273.

Cemgil, A. T. & Kappen, B. (2003). Monte Carlo methods for tempo tracking and rhythm quantization, *Journal of Artificial Intelligence Research* **18**: 45–81.

Cheveigné, A. & Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music, *Journal of the Acoustical Society of America (JASA)*.

Chua, B. Y. & Lu, G. (2004). Determination of perceptual tempo of music, *Lecture Notes in Computer Science* **3310**: 61–70.

Chua, B. Y. & Lu, G. (2005). Improved perceptual tempo detection of music, *Proc. IEEE Int. Conf. on Multimedia Modeling*.

Clarke, E. F. (1982). Timing in the performance of erik satie's 'vexations', *Acta Psychologica* **52**: 1–19.

Collins, N. (2005a). A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions, *Proc. AES 118th Convention*, Barcelona, Spain.

Collins, N. (2005b). Drumtrack: Beat induction from an acoustic drum kit with synchronised scheduling, *Proc. International Computer Music Conference (ICMC)*.

Cormen, T. H., Leiserson, C. E. & Rivest, R. L. (2001). *Introduction to Algorithms*, 2nd edition edn, MIT Press.

Dannenberg, R. B. (2005). Toward automated holistic beat tracking, music analysis, and understanding, *Proc. International Conference on Music Information Retrieval (ISMIR)*, London, UK.

Daudet, L. (2001). Transients modelling by pruned wavelet trees, *Proc. Int. Computer Music Conference (ICMC)*.

Davies, M. & Plumbley, M. D. (2006). Context-dependent beat tracking of musical audio., *Technical report*, Queen Mary University of London.

Davies, M. E. P. & Plumbley, M. D. (2004). Causal tempo tracking of audio, *Proc. Int. Conf. on Music Information Retrieval*, Barcelone, Spain.

Davies, M. E. P. & Plumbley, M. D. (2005). Beat tracking with a two state model, *Proc. Int. Conf. on Speech and Audio Processing*.

Davy, M. & Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines and application to audio signal segmentation, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

De Moor, B. (1993). The singular value decomposition and long and short spaces of noisy matrices, *IEEE Trans. on Signal Processing* **41**(9): 2826–2838.

Desain, P. & Honing, H. (1999). Computational models of beat induction: The rule-base approach, *Journal of New Music Research* **28**(1): 29–42.

Desobry, F., Davy, M. & Doncarli, C. (2005). An online kernel change detection algorithm, *IEEE Trans. on Signal Processing* **8**(53): 2961–2974.

Di Francesco, R. (1990). Real-time speech segmentation using pitch and convexity jump models: application to variable rate speech coding, *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP)* **38**(5): 741–748.

Dixon, S. (2001). Automatic extraction of tempo and beat from expressive performances, *Journal of New Music Research* **30**(1): 39–58.

Dixon, S. & Cambouropoulos, E. (2000). Beat tracking with musical knowledge, *Proc. European Conf. on Articial Intelligence (ECAI)*.

Dixon, S., Pampalk, E. & Widmer, G. (2003). Classification of dance music by periodicity patterns, *Proc. International Conference on Music Information Retrieval (ISMIR)*.

Dolby, R., Robinson, D. & Gundry, K. (1979). Ccir/arm: A practical noise-measurement method, *Journal of the Audio Engineering Society* **27**(3): 149–157.

Downie, J. S., West, K., Ehmann, A. & Vincent, E. (2005). The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview, *Proc. International Conference on Music Information Retrieval (ISMIR)*, London, UK., pp. 320–323.

Drake, C., Penel, A. & Bigand, E. (2000). *Rhythm perception and production*, Swets and Zeitlinger, chapter Why musicians tap slower than nonmusicians.

Duxbury, C., Sandler, M. & Davies, M. (2002). A hybrid approach to musical note onset detection, *Proc. Int. Conference on Digital Audio Effects (DAFx)*.

Dvornikov, M. (2003). Formulæ of numerical differentiation. e-print. Available from: `http://arxiv.org/abs/math/0306092`.

Eck, D. & Casagrande, N. (2005). A tempo-extraction algorithm using an autocorrelation phase matrix and shannon entropy, *Proc. International Conference on Music Information Retrieval (ISMIR)*, London, UK., pp. 504–509.

Ellis, D. & Rosenthal, D. (1995). Mid-level representations for computational auditory scene analysis, *International Joint Conference on Artificial Intelligence (IJCAI)*.

Ephraim, Y. & Van Trees, H. L. (1995). A signal subspace approach for speech enhancement, *IEEE Trans. on Speech and Audio Processing* **3**(4): 251–266.

Fletcher, H. & Munson, W. A. (1933). Loudness, its definition, measurement and calculation, *Journal of the Acoustical Society of America (JASA)* **5**(2): 82–108.

Foote, J. (2000). Automatic audio segmentation using a measure of audio novelty, *Proc. IEEE International Conference on Multimedia and Expo*, Vol. II, pp. 452–455.

Foote, J. & Uchihashi, S. (2001). The beat spectrum: A new approach to rhythm analysis, *Proc. International Conference on Multimedia and Expo (ICME)*.

Fraisse, P. (1982). Rhythm and tempo, *in* D. Deutsch (ed.), *The Psychology of Music*, Academic Press, pp. 149–182.

Gordon, J. W. (1987). The perceptual attack time of musical tones, *Journal of the Acoustical Society of America (JASA)* **82**(1): 88–105.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drums, *Journal of New Music Research* **30**(2): 159–171.

Goto, M. & Muraoka, Y. (1994). A beat tracking system for acoustic signals of music, *Proc. Association for Computing Machinery (ACM) Multimedia*, San Francisco, CA., USA, pp. 365–372.

Goto, M. & Muraoka, Y. (1997a). Issues in evaluating beat tracking systems, *Proc. IJCAI-97 Workshop on Issues in AI and Music - Evaluation and Assessment*, pp. 9–16.

Goto, M. & Muraoka, Y. (1997b). Real-time rhythm tracking for drumless audio signals –cord change detection for musical decisions–, *Proc. IJCAI-97 Workshop on Computational Auditory Scene Analysis*, pp. 135–144.

Goto, M., Hashiguchi, H., Nishimura, T. & Oka, R. (2003). Rwc music database: Music genre database and musical instrument sound database, *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 229–230.

Gouyon, F. (2005). *A computational approach to rhythm description*, PhD thesis, Pompeu Fabra University, Barcelone, Spain.

Gouyon, F. & Dixon, S. (2005). A review of automatic rhythm description systems, *Computer Music Journal* **29**(1): 34–54.

Gouyon, F., Fabig, L. & Bonada, J. (2003). Rhythmic expressiveness transformations of audio recordings: swing modifications, *Proc. 6th International Conference on Digital Audio Effects (DAFX)*, London, UK.

Gouyon, F., Herrera, P. & Cano, P. (2002). Pulse-dependent analyses of percussive music, *Proc. AES 22nd. Int. Conf. on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland.

Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C. & Cano, P. (2006). An experimental comparison of audio tempo induction algorithms, *IEEE Trans. on Speech and Audio Processing*.

Hainsworth, S. (2003). *Techniques for the automated analysis of musical audio*, PhD thesis, University of Cambridge, Cambridge, UK.

Hainsworth, S. & Macleod, M. (2003a). Beat tracking with particle filtering algorithms, *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY., USA.

Hainsworth, S. & Macleod, M. (2003b). Onset detection in musical audio signals, *Proc. Int. Computer Music Conference (ICMC)*.

Hainsworth, S. W. & Wolfe, P. J. (2001). Time-frequency reassignment for musical analysis, *Proc. Int. Computer Music Conference (ICMC)*.

Handel, S. (1993). The effect of tempo and tone duration on rhythmic discrimination, *Perception. & Psychophysics* **54**(3): 370–382.

Hayes, M. H. (1996). *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons.

Hermus, K. & Wambacq, P. (2004). Assessement of signal subspace based speech enhancement for noise robust speech recognition, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

Hlawatsch, F. & Auger, F. (eds) (2005). *Temps-fréquence: concepts et outils*, Traitement du signal et de l'image, Lavoisier. In French.

IEC268-1 (1968). Sound system equipment, International Electrotecnical Comission (IEC) Standard 268-1. Sound System Equipment.

ITU-R468 (1986). Measurement of audio-frequency noise voltage level in sound broadcasting, International Telecommunications Union (ITU) Recommendation 468.

Jehan, T. (1997). Musical signal parameter estimation, *Technical report*, University of California, Berkeley, CA.

Jehan, T. (2004). Event-synchronous music analysis/synthesis, *Proc. Int. Conference on Digital Audio Effects (DAFx)*, Naples, Italy.

Jehan, T. (2005a). *Creating Music by Listening*, PhD thesis, Massachusetts Institute of Technology.

Jehan, T. (2005b). Downbeat prediction by listening and learning, *Proc. IEEE Workshop on Applications to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA.

Jensen, K. & Andersen, T. H. (2003). Beat estimation on the beat, *Proc. IEEE Workshop of Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY., USA, pp. 87–90.

Jones, M. (1987). Dynamic pattern structure in music: Recent theory and research, *Research. Perception & Psychophysics* **41**(6): 621–634.

Kapanci, E. & Pfeffer, A. (2004). A hierarchical approach to onset detection, *Proc. International Computer Music Conference (ICMC)*.

Kavanagh, R. C. (2001). Fir differentiators for quantized signals, *IEEE Trans. on Signal Processing* **49**(11): 2713–2720.

Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

Klapuri, A. (2003). Musical meter estimation and music transcription, *Proc. Cambridge Music Processing Colloquium*, Cambridge University, UK.

Klapuri, A. (2004). *Signal Processing Methods for the Automatic Transcription of Music*, PhD thesis, Tampere University of Technology, Tampere, Finland.

Klapuri, A. & Davy, M. (2006). *Signal Processing Methods for Music Transcription*, Springer.

Klapuri, A., Eronen, A. & Astola, J. (2006). Analysis of the meter of acoustic music signals, *IEEE Trans. on Speech and Audio Processing* **14**(1): 342–355.

Kodera, K., Gendrin, R. & de Villedary, C. (1978). Analysis of time-varying signals with small bt values, *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP)* **26**(1): 64–76.

Lacoste, A. & Eck, D. (2006). A supervised classification algorithm for note onset detection, *EURASIP Journal on Applied Signal Processing*. *Submitted*.

Large, E. W. & Kolen, J. F. (1994). Resonance and the perception of musical meter, *Connection science* **6**(1): 177–208.

Laroche, J. (2001). Estimating tempo, swing and beat locations in audio recordings, *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 135–138.

Laroche, J. (2003). Efficient tempo and beat tracking in audio recordings, *Journal of the Audio Engineering Society* **51**(4): 226–233.

Lee, C. S. (1991). The perception of metrical structure: Experimental evidence and a model, *in* P. Howell, R. West & I. Cross (eds), *Representing musical structure*, Academic Press, pp. 59–127.

Lerdahl, F. & Jackendoff, R. (1983). *A generative theory of tonal music*, MIT Press, Cambridge, MA., USA.

Longuet-Higgins, H. C. & Lee, C. S. (1982). The perception of musical rhythms, *Perception* **11**(2): 115–128.

Longuet-Higgins, H. C. & Lee, C. S. (1984). The rhythmic interpretation of monophonic music, *Music Perception* **1**(4): 424–441.

Maher, J. & Beauchamp, J. (1994). Fundamental frequency estimation of musical signals using a two-way mismatch procedure, *Journal of the Acoustical Society of America* **95**(4): 2254–2263.

Masri, P. (1996). *Computer Modeling of Sound for Transformation and Synthesis of Musical Signal*, PhD thesis, University of Bristol, Bristol, UK.

Mayor, O. (2001). An adaptative real-time beat tracking system for polyphonic pieces of audio using multiple hypotheses, *Proc. MOSART Workshop on Current Research Directions in Computer Music*, Barcelona, Spain.

McAuley, R. J. & Quatieri, T. F. (1986). Speech analysis/synthesis based on a sinusoidal representation, *IEEE Trans. on Acoustics, Speech and Signal Processing* **34**(4): 744–754.

McKinney, M. F. & Moelants, D. (2004). Deviations from the resonance theory of tempo induction, *Proc. Conference on Interdisciplinary Musicology*.

Meddis, R. (1988). Simulation of auditory-neural transduction: Further studies, *Journal of the Acoustical Society of America (JASA)* **83**(3): 1056–1063.

Meudic, B. (2004). *Détermination automatique de la pulsation, de la metrique, et des motifs musicaux dans des interprétations à tempo variable d'œuvres polyphoniques*, PhD thesis, Université Pierre et Marie Curie, Paris 6, Paris, France. in French.

Moelants, D. (2002). Preferred tempo reconsidered, *Proc. Int. Conf. on Music Perception and Cognition*.

Moore, B. C. (1997). *An introduction to the physiology of hearing*, Academic Press.

Moore, B. C. (ed.) (1995). *Hearing: Handbook of Perception and Cognition*, Academic Press.

Nielsen, N., Carstensen, J. M. & Smedsgaard, J. (1998). Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping, *Journal of Chromatography A* **805**: 17–35.

Noll, A. M. (1970). Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum and a maximum likelihood estimate, *Proc. Symposium on Computer Processing in Communications*, Vol. XIX of *Polytechnic Institute of Brooklin*, Polytechnic Press, N.Y. USA, pp. 779–797.

Papoulis, A. (1962). *The Fourier integral and its applications*, Electronic Sciences Series, McGraw-Hill.

Parncutt, R. (1994). A perceptual model of pulse salience and metrical accent in musical rhythms, *Music Perception* **11**(4): 409–464.

Parncutt, R. & Drake, C. (2001). Psychology: Rhythm, *New Grove Dictionary of Music and Musicians*, Macmillan Publishers, London, UK, pp. 535–538, 542–553.

Paulus, J. & Klapuri, A. (2002). Measuring the similarity of rhythmic patterns, *Proc. Int. Conference on Music Information Retrieval*, Paris, France.

Peeters, G. (2005). Time variable tempo detection and beat marking, *Proc. International Computer Music Conference (ICMC)*, Barcelona, Spain.

Portnoff, M. R. (1980). Time–frequency representation of digital signals and systems based on short-time fourier analysis, *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP)* **28**(1): 55–69.

Povel, D. & Essens, P. (1985). Perception of temporal patterns, *Music Perception* **2**(4): 411–440.

Proakis, J. (2000). *Digital Communications*, 4th edition edn, McGraw Hill.

Proakis, J. G. & Manolakis, D. G. (1996). *Digital Signal Processing: Principles, Algorithms and Applications*, 3rd edition edn, Prentice Hall.

Rabiner, L. & Juang, B. (1993). *Fundamentals of Speech Recognition*, Prentice Hall PTR.

Raphael, C. (2001). Automated rhythm transcription, *Proc. International Conference on Music Information Retrieval (ISMIR)*, Bloomington, IN., USA, pp. 99–107.

Röbel, A. (2003). A new approach to transient processing in the phase vocoder, *Proc. Int. Conference on Digital Audio Effects (DAFx)*.

Röbel, A. (2005). Onset detection in polyphonic signals by means of transient peak classification, *Proc. Mirex*. Available from: www.music-ir.org/evaluation/mirex-results/articles/onset/.

Rosenthal, D. (1992). Emulation of human rhythm perception, *Computer Music Journal* **16**(1): 64–76.

Scharf, L. L. (1991). *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*, Prentice Hall.

Scheirer, E. (1998). Tempo and beat analysis of acoustic music signals, *Journal of the Acoustical Society of America* **103**(1): 588–601.

Scheirer, E. (2000). *Music-listening systems*, PhD thesis, MIT, Cambridge, MA., USA.

Schwartz, D. (1963). *Méthodes statistiques à l'usage des médecins et des biologistes*, third edn, Flammarion medecine series. In French.

Seppänen, J. (2001a). *Computational models of musical meter recognition*, Master's thesis, Tampere University of Technology, Tampere, Finland.

Seppänen, J. (2001b). Tatum grid analysis of music signals, *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY., USA.

Serra, X. (1989). *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition*, PhD thesis, Stanford University, CA., USA.

Serra, X. & Smith, J. O. (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition, *Computer Music Journal* **14**(4): 12–24.

Sethares, W. A. & Staley, T. W. (1999). Periodicity transforms, *IEEE Trans. on Signal Processing* **11**(47): 2953–2964.

Sethares, W. A. & Staley, T. W. (2001). Meter and periodicity in musical performance, *Journal of New Music Research* **22**(5): 149–158.

Sethares, W. A., Morris, R. D. & Sethares, J. C. (2005). Beat tracking of musical performances using low-level audio features, *IEEE Trans. on Speech and Audio Processing* **13**(2): 275–285.

Silverman, H. F. & Morgan, D. P. (1990). The application of dynamic programming to connected speech recognition, *Acoustics, Speech and Signal Processing (ASSP) Magazine* **7**(3): 6–25.

Simon, M. K., Hinedi, S. M. & Lindsey, W. C. (1995). *Digital Communication Techniques: Signal Design and Detection*, Prentice Hall.

Smith, L. S. & Fraser, D. S. (2004). Robust sound onset detection using leaky integrate-and-fire neurons with depressing synapses, *IEEE Trans. on Neural Networks* **15**(5): 1125–1134.

Steedman, M. J. (1977). The perception of musical rhythm and metre, *Perception* **6**(5): 555–569.

Stoica, P. & Moses, R. L. (1997). *Introduction to Spectral Analysis*, Prentice Hall.

Tagare, H. & Figueiredo, R. (1985). Order filters, *Proceedings of the IEEE* **73**(1): 163–165.

Temperley, D. (2004). An evaluation system for metrical models, *Computer Music Journal* **28**(3): 28–44.

Temperley, D. & Sleator, D. (1999). Modeling meter and harmony:A preference-rule approach, *Computer Music Journal* **23**(1): 10–27.

Thornburg, H. & Gouyon, F. (2000). A flexible analysis-synthesis method for transients, *Proc. Int. Computer Music Conference (ICMC)*.

Todd, N. P. (1994). The auditory primal sketch: a multiscale model of rhythmic grouping, *Journal of New Music Research* **23**(1): 25–70.

Todd, N. P. (1999). A sensory-motor theory of rhythm, time perception and beat induction, *Journal of New Music Research* **28**(1): 5–28.

Toiviainen, P. (1998). An interactive MIDI accompanist, *Computer Music Journal* **22**(4): 63–75.

Tomasi, G., van den Berg, F. & Andersson, C. (2004). Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data, *Journal of Chemometrics* **18**: 231–241.

Tzanetakis, G. & Cook, P. (2002). Musical genre classification of audio signals, *IEEE Trans. on Speech and Audio Processing* **10**(5): 293–302.

Uhle, C. & Herre, J. (2003). Estimation of tempo, micro time and time signature from percussive music, *Proc. Int. Conference on Digital Audio Effects (DAFx)*.

Uhle, C., Rohden, J., Cremer, M. & Herre, J. (2004). Low complexity musical meter estimation from polyphonic music, *Proc. Audio Engineering Society 25th International Conference*.

Vaidyanathan, P. (1992). *Multirate systems and filter banks*, Prentice-Hall PTR.

van Noorden, L. & Moelants, D. (1999). Resonance in the perception of musical pulse, *Journal of New Music Research* **28**(1): 43–66.

Wang, J. F., Yang, C.-H. & Chang, K.-H. (2004). Subspace tracking for speech enhancement in car noise environments, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

Wang, Y. & Vilermo, M. (2001). A compressed domain beat detector using MP3 audio bitstreams, *Proc. 9th ACM Int. Conference on Multimedia*, Ottawa, Canada., pp. 194–202.

Weisstein, E. W. (1999). Correlation coefficient, on-line. Available from: `http://mathworld.wolfram.com/CorrelationCoefficient.html`.

Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms, *IEEE Trans. Audio and Electroacoust.* **AU–15**: 70–73.

Widmer, G., Dixon, S., Knees, P., Pampalk, E. & Pohle, T. (2005). From sound to "sense" via feature extraction and machine learning: Deriving high-level descriptors for characterising music, *in* M. Leman & D. Cirotteau (eds), *Sound to Sense: Sense to Sound: A State-of-the-Art*, S2S2 Consortium, Florence, Italy.

Win, M. Z. (1998). On the power spectral density of digital pulse streams generated by m-ary ciclostationary sequences in the presence of stationary timing jitter, *IEEE Trans. on Communications* **46**(9): 1135–1145.

Wise, J. D., Caprio, J. R. & Parks, T. W. (1976). Maximum likelihood pitch estimation, *IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP)* **24**(5): 418–423.