HAL open science

Transcription des signaux percussifs. Application à l'analyse de scènes musicales audiovisuelles

Olivier Gillet

▶ **To cite this version:**

Olivier Gillet. Transcription des signaux percussifs. Application à l'analyse de scènes musicales audiovisuelles. domain_other. Télécom ParisTech, 2007. English. NNT : . pastel-00002805

HAL Id: pastel-00002805

https://pastel.hal.science/pastel-00002805v1

Submitted on 28 Sep 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TELECOM PARIS
école nationale supérieure des télécommunications

Thèse

Présentée pour obtenir le grade de Docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Olivier Gillet

Sujet :

TRANSCRIPTION DES SIGNAUX PERCUSSIFS. APPLICATION À L'ANALYSE DE SCÈNES MUSICALES AUDIOVISUELLES.

M. Bernard MERIALDO	Rapporteur
M. Laurent GIRIN	Rapporteur
Mme. Régine ANDRÉ-OBRECHT	Examinatrice
M. Xavier RODET	Examinateur
M. Dan ELLIS	Examinateur
M. Mark SANDLER	Examinateur
M. Gaël RICHARD	Directeur de thèse

Remerciements

Je tiens d'abord à remercier mon directeur de thèse Gaël Richard pour avoir su faire converger mes motivations et intérêts personnels vers le domaine de l'indexation audio, jusqu'au choix du sujet de cette thèse, vaste, riche, mais aussi parfois déroutant. Il a su à de maintes reprises me soutenir dans des moments difficiles, m'encourager et me conseiller, toujours en accordant une grande valeur à mes opinions et intuitions.

Je remercie les rapporteurs Bernard Merialdo et Laurent Girin, pour l'intérêt qu'ils ont voulu accorder à mes travaux, ainsi que tous les membres du jury : Xavier Rodet, Dan Ellis, Régine André-Obrecht et Mark Sandler.

Le département de Traitement du Signal et des Images de Telecom Paris (et ses membres chaleureux) m'a offert un cadre de travail à la fois agréable et stimulant. Je tiens en particulier à remercier Slim Essid, Miguel Alonso, Roland Badeau, Bertrand David, Nancy Bertin et tous les autres collègues qui m'ont, à un moment ou à un autre, dépanné d'un script Maltab ou sorti d'une ornière technique.

Un grand merci à tous les membres du *Centre for Digital Video Processing* de la *Dublin City University* où j'ai eu la chance de séjourner pendant une semaine, en particulier Noel O'Connor's et Kevin McGuinness. De nombreuses idées développées dans cette thèse sont nées de ce séjour.

Je remercie également les musiciens et ingénieurs du son impliqués dans la tâche d'enregistrement de la base ENST-drums : Louis Cavé, Bertrand Clouard, Frédéric Rottier et Michel Desnoues ; pour leur patience, leur talent, et pour m'avoir rappelé que le son d'une batterie peut être autre chose qu'une suite de vecteurs de paramètres acoustiques.

Aucun des travaux présentés dans cette thèse n'auraient pu être menés si je n'avais eu à ma dispositions les puissants outils que sont Matlab et ses boîtes à outils Auditory, Spider et SimpleSVM ; le langage Python, toujours là pour apaiser ou émerveiller l'informaticien qui sommeille en moi ; et l'excellente bibliothèque C++ de traitement d'images développée au CDVP. Merci à leurs auteurs.

Merci à ma famille pour son soutien constant durant ces quarante mois, en particulier à mes parents pour leur effort de relecture essentiel. Merci également à mes futurs beaux-parents pour supporter le plus dingue des beaux-fils.

Merci enfin à tous ceux dont les contributions se trouvent entre les lignes de cette thèse. À Bablu, Bloby et Ganaël pour, entre autres, les naans de chez Shan, Mariah Carey, les régressions logistiques aux cajoux, *grrrou*, les lapinous, le pur chipop' style, erotikast, la trotinette à la Villette, *Inférence est dans la place*, le Nabaztag, les fausses adresses, *Il est 6h28 dans le Chimboland*, les pizzas de chez Rabbit, le tarot à 3 (partenaires ou heures du mat), les fous rires et les business-plans, *un bon réseau de neurones et on en parle plus* et les longues discussions désabusées dans la cuisine. À Priyanka pour tout le reste.

Table des matières

Remerciements I

Table des matières III

Table des figures VII

Liste des tableaux IX

Acronymes XI

1 Introduction, motivations 1
- 1.1 Indexation et transcription automatique . . . 1
- 1.2 Motivations . . . 2
- 1.3 Définitions, champ d'étude et restrictions . . . 4
- 1.4 Plan d'étude et résumé des contributions . . . 8

I Analyse des signaux audiofréquences percussifs : application à la batterie 11

2 Transcription automatique des signaux percussifs : un état de l'art 13
- 2.1 Analyse du rythme . . . 13
- 2.2 Analyse des signaux percussifs : les trois approches . . . 17
- 2.3 Utilisation des connaissances musicales pour la transcription . . . 29
- 2.4 Applications . . . 30

3 Pré-traitements pour l'accentuation de la piste de batterie 31
- 3.1 Principe et motivations . . . 31
- 3.2 Banc de filtres . . . 35
- 3.3 Séparation et sélection de sources à partir d'enregistrements stéréophoniques 36
- 3.4 Extraction de la composante stochastique . . . 43
- 3.5 Conclusion . . . 48

4 Transcription de la batterie dans un signal de musique 51
- 4.1 Mise en oeuvre de l'approche Segmenter et Reconnaître . . . 51
- 4.2 Détection d'onsets . . . 55
- 4.3 Paramétrisation des signaux . . . 59
- 4.4 Classification des instruments de la batterie . . . 63
- 4.5 Du modèle acoustique au modèle de séquence . . . 68
- 4.6 Résultats expérimentaux . . . 84
- 4.7 Conclusion . . . 99

5 Extraction de la piste de batterie dans un signal de musique **101**
5.1 Bref état de l'art . . . 101
5.2 Filtrage temps/fréquence/sous-espace (TFS) . . . 104
5.3 Filtrage pseudo-Wiener et modèles spectraux . . . 107
5.4 Résultats expérimentaux . . . 112
5.5 Conclusion . . . 115

Conclusion de la partie I **117**

II Transcription audiovisuelle du jeu de la batterie **119**

6 Transcription musicale et multimodalité : état de l'art et problématique **121**
6.1 Spécificité du problème à résoudre et typologie des tâches connexes . . . 121
6.2 État de l'art . . . 122
6.3 Discussion . . . 128

7 Segmentation de scènes de jeu de batterie **133**
7.1 Segmentation des éléments de la batterie dans une scène : cas des images fixes . . . 133
7.2 Segmentation des éléments dans une séquence d'images . . . 142
7.3 Segmentation des baguettes . . . 147
7.4 Conclusion . . . 148

8 Transcription audiovisuelle de séquences de batterie **151**
8.1 Détection des frappes dans une séquence vidéo . . . 151
8.2 Transcription audiovisuelle par fusion tardive . . . 155
8.3 Autres stratégies pour la transcription musicale audiovisuelle . . . 161
8.4 Conclusion . . . 164

Conclusion de la partie II **167**

III Vers l'analyse des documents audiovisuels musicaux **169**

9 Problématique **171**
9.1 État de l'art . . . 171
9.2 Approche proposée . . . 173

10 Détection des changements dans les documents audiovisuels musicaux **177**
10.1 Détection des changements de section dans les signaux de musique . . . 177
10.2 Extraction de la structure des séquences vidéo . . . 191
10.3 Détection d'événements dans une séquence vidéo . . . 194
10.4 Conclusion . . . 196

11 Mesures de corrélation entre flux audio et vidéo **197**
11.1 Mesures de corrélation des flux audio et vidéo structurés . . . 197
11.2 Applications . . . 199
11.3 Conclusion . . . 203

Conclusion de la partie III **205**

12 Perspectives **207**
12.1 Analyse des signaux percussifs 207
12.2 Analyse audiovisuelle du jeu de la batterie 209
12.3 Analyse de documents audiovisuels musicaux 210

IV Annexes - Boîte à outils **213**

A Palette d'attributs **215**
A.1 Paramètres de distribution de l'énergie 215
A.2 Paramètres cepstraux 218
A.3 Paramètres spectraux 220
A.4 Paramètres temporels 221
A.5 Paramètres psychoacoustiques 222

B Machines à vecteurs de support (SVM) **223**
B.1 Principe, primal et dual 223
B.2 Cas non linéairement séparable 228
B.3 SVM à noyaux 232
B.4 Estimation de probabilités a posteriori à partir de SVM 235

V Annexes - Documents complémentaires **237**

C Autres articles **239**

D Corpora utilisés **257**

Bibliographie **263**

Bibliographie de l'auteur **279**

Index **281**

Table des figures

1.1 HAL9000 saurait-il toujours transcrire une partition dans ces circonstances ? 4
1.2 Plan de la thèse et champ d'étude 8

2.1 Architecture typique d'un système d'analyse de surface du rythme 14
2.2 Quelques procédés d'extraction de formes rythmiques 16
2.3 Topologie de HMM pour la reconnaissance et la segmentation de signaux de batterie . . 21
2.4 Détection de grosse caisse par filtrage adapté 24
2.5 L'ISA appliquée à une boucle de batterie 26
2.6 Résultats de la campagne MIREX 2005, transcription de batterie 28

3.1 Intérêt de la décomposition déterministe/stochastique 33
3.2 Architecture du système d'accentuation des instruments percussifs 34
3.3 Banc de filtres en bandes d'octave 35
3.4 Distribution de l'énergie dans les sous-bandes 36
3.5 Réponses en fréquence du banc de filtre et d'un de ses filtres 37
3.6 Panoramique des sources percussives 39
3.7 Séparation d'un enregistrement stéréophonique avec ADRes 40

4.1 Le phénomène musical, et les deux approches de la transcription 53
4.2 Architecture du système de transcription de la piste de batterie pour deux approches : fusion précoce et fusion tardive 56
4.3 Algorithme de détection des onsets 57
4.4 Algorithme de localisation des pics 59
4.5 Exemple de hiérarchie de répétitions dans un accompagnement rythmique 69
4.6 De la liste d'événements à la représentation symbolique 70
4.7 Extraction du tatum pour un rythme de Blues-Rock ternaire 71
4.8 Grille de tatum flexible 73
4.9 Batteries et batteurs dans la base ENST-drums 85
4.10 Protocole de validation emboîtée utilisé 86
4.11 Surfaces de décision 95

5.1 Enveloppes d'amplitudes 105
5.2 Dictionnaires de d.s.p 110
5.3 Fenêtres longues, courtes et de transition utilisées pour l'analyse et la synthèse 111
5.4 Pré-echo dans les signaux séparés 111

6.1 HMM pour la reconnaissance de parole audiovisuelle 126
6.2 Modèle factoriel pour le débruitage audiovisuel de la parole 127
6.3 Architecture du système proposé pour la transcription audiovisuelle du jeu de la batterie 130

7.1 Filtrage bilatéral gaussien 134
7.2 Critère de couleur appris 136
7.3 Segmentation par critère de couleur 137
7.4 Regroupement des contours : critère de proximité, prise en compte de la courbure 139

7.5 Détection d'ellipses 140
7.6 Critère de validité des régions obtenues par segmentation 142
7.7 Fusion d'images pour la segmentation 144
7.8 Masques obtenus par NMF 146
7.9 Régions extraites par segmentation supervisée par l'audio 147
7.10 Segmentation des baguettes 149

8.1 Exemples de paramètres extraits 153
8.2 Modèles de pics $r_B(m)$, $r_{MF}(m)$ et $r_{MC}(m)$ 153
8.3 Compatibilité régions/instruments 156
8.4 Segmentation manuelle détaillée 162

9.1 Structuration et analyse de synchronie dans les documents audiovisuels musicaux 174

10.1 Principe de la segmentation par détection de nouveauté 181
10.2 Séparation par un hyperplan des points sur une hypersphère 183
10.3 Principe de l'algorithme KCD 186
10.4 Fonctions de détection de nouveauté 189
10.5 Comparaison des algorithmes de segmentation 191
10.6 Comparaison des jeux d'attributs pour la segmentation 192
10.7 Segmentation en séquences d'un clip vidéo 194
10.8 Champ de vecteurs de mouvement 195
10.9 Champ de vecteurs de mouvement sur une zone non-texturée 195

11.1 Recherche d'accompagnement musical à partir d'une séquence vidéo : courbes rappel/précision 200
11.2 Matrice de synchronie entre les flux audio et vidéo 201
11.3 Influence du retard entre la musique et l'image sur les mesures de corrélation 203

A.1 Filtres passe-bande adaptés définis par Tanghe et al 216
A.2 Banc de filtre en bandes d'octave utilisé pour le calcul des attributs OBSIR 217
A.3 Banc de filtres en demi-tons 218

B.1 Hyperplans séparateurs 224
B.2 Marge d'un hyperplan séparateur et vecteurs de support 225
B.3 Plus court segment joignant les enveloppes convexes des exemples positifs et négatifs . . 227
B.4 Enveloppes convexes μ-réduites 229
B.5 Projection non-linéaire et séparabilité 233
B.6 Surfaces de décisions pour différents noyaux 234
B.7 Principe de la méthode de Platt 236

Liste des tableaux

3.1 Limites des bandes de fréquence du banc de filtres en bandes d'octave 37
3.2 Performances de l'algorithme ADRes pour la séparation de sources percussives 39
3.3 Pureté des sources extraites dans les signaux de sous-bande 41
3.4 Performances de l'ICA par sous-bande 43
3.5 Paramètres utilisés pour la séparation de la partie stochastique dans chacune des bandes . 48

4.1 Pouvoir descriptif des taxonomies, et nombre de combinaisons d'instruments rencontrées 54
4.2 Performances du module de détection d'onsets 60
4.3 Récapitualtif des 147 attributs utilisés. Leur calcul est détaillé dans l'annexe A 61
4.4 Opérateurs de fusion 68
4.5 Symboles associés aux combinaisons de frappes 73
4.6 Pouvoir prédictif des modèles de séquence 78
4.7 Exemple d'inférence de grammaire par l'algorithme SEQUITUR 81
4.8 Exemple d'inférence de grammaire avec transformations 82
4.9 Exemples de complétion automatique de séquence par minimisation de la complexité . . 84
4.10 Performances des systèmes de transcription 88
4.11 Performances avec et sans pré-traitement 89
4.12 Comparaison des méthodes de fusion tardive 90
4.13 Attributs sélectionnés 91
4.14 Composition des attributs sélectionnés 93
4.15 Paramètres de classification choisis automatiquement 93
4.16 IRMFSP vs RFE 94
4.17 Composition des vecteurs de support : caisse claire 96
4.18 Composition des vecteurs de support : grosse caisse 96
4.19 Comparaison avec d'autres systèmes 97
4.20 Performances de la transcription avec modèle de séquence 98

5.1 Performances des méthodes de séparation évaluées 114

6.1 Quelques problèmes connexes traités dans la littérature 122

7.1 Évaluation des attributs de couleur pour la segmentation 136
7.2 Évaluation de la détection d'ellipses pour la segmentation 141

8.1 Classification cymbales/fûts par critère de couleur 155
8.2 Identification des instruments à partir des régions : performances 159
8.3 Performances de la transcription audiovisuelle 160
8.4 Quelles méthodes de segmentation et de détection choisir ? 165

10.1 Récapitualtif des 70 attributs utilisés pour la segmentation audio. Leur calcul est détaillé dans l'annexe A 178
10.2 Attributs sélectionnés pour la segmentation en sections de signaux de musique 180
10.3 Temps de calcul des fonctions de détection avec et sans résolution adaptative des SVM à 1 classe 187

10.4 F-mesure, avec un seuil $\tau = 1$, pour la tâche de détection de frontières de segments dans la base `Music-100` 190

11.1 Influence du *genre visuel* sur les résultats de l'expérience de recherche de musique par la vidéo 202

A.1 Découpage empirique du spectre et éléments de la batterie associés 217

D.1 Corpus `Music-54` 258
D.2 Corpus `Music-100` 259
D.3 Corpus `Video-100` 260
D.4 Fréquence des combinaisons de frappes dans le corpus ENST-drums 261

Acronymes

La littérature relative à la plupart des thèmes connexes à cette thèse est jeune et rarement traduite. Pour la plupart des acronymes employés, nous avons utilisé la dénomination la plus courante, qui est de fait en langue anglaise.

ADRes discrimination d'Azimuth et Resynthèse – *Azimuth Discrimination and Resynthesis*

BD grosse caisse – *Bass Drum*

BIC critère d'information bayésien – *Bayesian Information Criterion*

BPM Battements Par Minute

d.s.p densité spectrale de puissance

DTW déformation temporelle dynamique – *Dynamic Time Warping*

EDS sinusoïdes modulées exponentiellement – *Exponentially Damped Sinusoids*

EVD décomposition en valeurs propres – *Eigenvalue Decomposition*

GMM modèle(s) de mélanges de gaussiennes – *Gaussian Mixture Model(s)*

HH hi-hat

HMM modèle(s) de Markov caché(s) – *Hidden Markov Model(s)*

ICA analyse en composantes indépendantes – *Independent Component Analysis*

IRMFSP maximisation du rapport d'inertie avec projection sur l'espace des attributs – *Inertia Ratio Maximization using Feature Space Projection*

ISA analyse en sous-espaces indépendants – *Independent Subspace Analysis*

MatAda Mettre en correspondance et Adapter

MFCC coefficients cepstraux en échelle de Mel – *Mel Frequency Cepstrum Coefficients*

NMF factorisation matricielle non-négative – *Nonnegative Matrix Factorizaion*

PCA analyse en composantes principales – *Principal Component Analysis*

PSA analyse en sous-espaces appris – *Prior Subspace Analysis*

RFE-SVM élimination récursive d'attributs par machines à vecteurs de support – *Recursive Feature Elimination with Support Vector Machines*

RKHS espace de Hilbert à noyau reproduisant – *Reproducing Kernel Hilbert Space*

SegRec Segmenter et Reconnaître

SepDet Séparer et Détecter

SAR rapport signal à artefacts – *Signal to Artefact Ratio*

SD caisse claire – *Snare Drum*

SDR rapport signal à distorsion – *Signal to Distortion Ratio*

SEF flux d'énergie spectral – *Spectral Energy Flux*

SIR rapport signal à interférences – *Signal to Interferences Ratio*

SVD décomposition en valeurs singulières – *Singular Value Decomposition*

SVM machine(s) à vecteurs de support – *Support Vector Machine(s)*

SVM1C machine(s) à vecteurs de support à une classe

TFCT transformée de Fourier à Court Terme

TWM mesure de non-coïncidence – *Two-Way Mismatch*

CHAPITRE 1

Introduction, motivations

1.1 Indexation et transcription automatique

1.1.1 Perspective historique

Les premières applications musicales de l'informatique et du traitement de signal ont eu pour but l'imitation, par l'ordinateur, des sons musicaux, et la reproduction d'oeuvres existantes ou nouvelles à partir de ces sonorités de synthèse. Dans les années soixante en effet, seuls les laboratoires des universités disposaient des ressources de calcul nécessaires à l'accomplissement de ces tâches – l'ordinateur servait donc les intérêts des compositeurs les plus inspirés (ou fortunés) souhaitant produire de la musique, et personne n'aurait osé imaginer qu'il jouerait un jour un rôle dans la consommation de cette musique par des particuliers.

La situation a bien changé quatre décennies plus tard. L'avènement de l'internet et la croissance exponentielle des capacités de calcul des microprocesseurs allant de pair avec la diminution du coût des supports de stockage ont fait que désormais, la musique est produite, diffusée et consommée au travers de systèmes informatiques. La recherche a anticipé cette évolution, produisant des méthodes efficaces de synthèse, modification, restitution et codage des signaux musicaux. C'est grâce à ces travaux que nous pouvons stocker aujourd'hui des dizaines de milliers d'oeuvres musicales sur un lecteur multimédia portable.

Cependant, ces nouvelles possibilités soulèvent de nouveaux problèmes : comment organiser de tels volumes de données et permettre un accès facile à l'information ? Comment retrouver dans ma collection personnelle ou dans le catalogue d'une boutique en ligne toutes les reprises de *Light my Fire*, un morceau de Minor Threat qui commence juste par de la basse, tous les instrumentaux de Hip-Hop, ou cet air que je suis en train de siffler ? De façon plus générique, comment extraire automatiquement des descriptions sémantiques à partir de signaux de musique, de manière à faciliter la recherche d'information – ce que nous appelons *indexation* ? Les recherches se sont malheureusement avérées bien moins fructueuses sur cette question. D'abord peut être parce que peu d'attention a été porté au sujet : il y avait d'autres priorités (développer des codeurs efficaces par exemple), et il était d'ailleurs difficile d'imaginer que le problème de l'accès aux données se poserait si vite. Mais surtout parce que ce problème est extrêmement difficile. Des tâches qui peuvent être effectuées aisément par des auditeurs humains ne disposant d'aucune formation musicale – reconnaître les instruments de musique, suivre un rythme, distinguer le Hip-Hop du Death Metal – apparaissent comme incroyablement complexes pour des systèmes informatiques.

Le domaine de l'indexation audio tente de relever ce défi : apprendre aux machines à comprendre et décrire les sons. Dans le cas où ces sons sont musicaux, une description complète et intéressante prendrait la forme d'une partition détaillée, listant tous les événements (notes) avec leurs hauteurs, dynamiques, instants de jeu, instruments utilisés, telle qu'elle peut être stockée dans un fichier MIDI. On parle alors de transcription musicale automatique.

1.1.2 Applications

Une des premières applications de la transcription musicale automatique est bien évidemment l'indexation. Les systèmes de requête par chantonnement (*query by humming*) tels ceux décrits dans [SGM98], [GJCS95] ou [CC98] présupposent ainsi qu'il existe une représentation symbolique (sous forme de partition) de chaque enregistrement dans la base de données. S'il est intéressant de pouvoir effectuer des recherches dans des collections de fichiers au format MIDI, il est bien plus utile de pouvoir faire la même chose sur une collection d'enregistrements musicaux. Une phase préalable de transcription de ces enregistrements musicaux vers un format symbolique s'avère donc nécessaire. Le chantonnement n'est pas la seule modalité de requête possible : on peut également concevoir des systèmes de requête par l'exemple, ou de navigation cartographique dans les collections [PDW03]. La norme MPEG-7 décrit déjà un format de stockage des méta-données associées à des documents multimédia, ainsi que des descripteurs audio [Cas01] et vidéo simples. Les systèmes de transcription musicale permettraient d'en étendre la portée.

Une autre application possible de la transcription musicale automatique est le codage objet ou structuré des signaux de musique. À des très bas débits de transmission, il est en effet plus économe de transmettre non pas le signal de musique (débarrassé de sa redondance), mais une description[1] du contenu musical de ce signal. Le décodeur resynthétise alors le signal de musique à partir de cette description. Cette approche est inclue dans la norme MPEG-4, sous forme des langages *Structured Audio Orchestra Language* (SAOL), *Structured Audio Score Language* (SASL), et *Structured Audio Sample Bank Format* (SASBF) qui décrivent respectivement les procédés de synthèse, les partitions et les échantillons sonores utilisés pour la synthèse [SV99]. Les normes MPEG normalisent les décodeurs et ne se soucient pas du développement des codeurs : ainsi, il n'existe à ce jour aucun codeur capable de produire automatiquement des représentations SAOL/SASL/SASBF à partir d'enregistrements musicaux. Seules des percées dans le domaine de la transcription musicale automatique permettront le développement de tels codeurs.

Par ailleurs, les techniques de transcription musicale automatique, si elles arrivent à se plier à la contrainte du temps réel, offrent aux systèmes informatiques la possibilité d'interagir de façon naturelle avec des musiciens – la musique devenant une modalité d'entrée d'information comme le serait la voix ou le geste. Cela suggère des applications comme l'accompagnement automatique, l'improvisation mêlant interprètes humains et agents informatiques, ou l'aide à l'apprentissage. Les seules modalités d'entrée de données musicales dans l'ordinateur disponibles aujourd'hui font appel à des capteurs, ou nécessitent le jeu sur des surfaces de contrôles (surface sensitives remplaçant les instruments à percussions, claviers MIDI). Ces deux solutions ne sont pas satisfaisantes aussi bien pour le musicien chevronné, qui veut préserver intacts sa technique de jeu et le timbre de son instrument, que pour le débutant qui souhaite apprendre sur un instrument véritable.

Enfin, la transcription musicale automatique trouve une dernière de ses applications dans le domaine des interfaces graphiques. En effet, un système informatique capable de comprendre un signal audio en des termes musicaux peut proposer une interface graphique permettant de modifier le contenu de ce signal en ces termes : effacer une mesure, aligner une interprétation sur une grille temporelle, réarranger la partie rythmique d'un signal audio deviendrait alors aussi facile qu'avec un éditeur de fichiers MIDI.

1.2 Motivations

Cette thèse considère le problème de la transcription musicale sous deux nouveaux angles : la transcription des signaux percussifs, et la transcription de scènes musicales audiovisuelles. Quels procédés de traitement de signal et d'apprentissage doit-on mettre en oeuvre pour extraire une description de la partie rythmique d'un signal de musique ? Comment peut-on par ailleurs tirer parti de l'information visuelle accompagnant un signal de musique pour améliorer cette description, ou l'exploiter autrement ? Nous détaillons ici nos motivations à suivre cette voie.

[1] "code source" ou "recette" pour reprendre l'expression d'Anssi Klapuri dans [Kla04]

1.2.1 Transcription des signaux percussifs

Historiquement, peut-être à cause de la popularité du *Query by Humming*, les premiers travaux en transcription musicale automatique ont privilégié la transcription de la mélodie et l'analyse harmonique, à travers le problème de la détection de fréquences fondamentales multiples [Kla01], et de façon plus modeste l'analyse de la structure rythmique. Le problème de la transcription des signaux percussifs a, lui, été peu considéré.

Or, c'est un problème essentiel pour plusieurs raisons. Tout d'abord, l'accompagnement rythmique joué à la batterie est un élément primordial dans la musique populaire moderne, en particulier, dans les styles contemporains *dance* (House, Techno, Drum'n'Bass, R'n'B, Hip-Hop). Il est aisé de reconnaître le genre musical d'une oeuvre simplement en considérant son accompagnement à la batterie – les systèmes d'indexation effectuant la reconnaissance du genre pourraient donc tirer avantage d'une transcription percussive. De surcroît, certains genres musicaux électroniques (Techno, IDM) sont essentiellement basés sur des structures rythmiques constituées de sons échantillonnés (*samples*). Une description de ces musiques en des termes uniquement harmoniques serait inutile – une représentation plus efficace consisterait en l'extraction de chacun des *samples* utilisés avec leur instant de jeu : précisément le type de représentations qu'est capable de produire un système de transcription des signaux percussifs.

Si l'on considère les applications d'indexation et de recherche par le contenu, là encore, le potentiel de l'analyse des signaux percussifs est grand. D'abord, le chantonnement n'est pas toujours la méthode de requête la plus pratique, en particulier pour les utilisateurs ne sachant pas chanter ! Une alternative intéressante est d'utiliser le contenu rythmique pour effectuer des requêtes par *tapping* (interprétation du rythme en tapotant sur des objets) ou *beatboxing* (interprétation du rythme à l'aide d'onomatopées) [KBT04; NOGH04; CC98; GR05b]. Il existe en outre des bases de données de signaux percussifs qui auraient grandement besoin d'être indexées : les milliers de boucles de batterie (*Drum loops*) fournies avec les logiciels de composition musicale ou vendues sur CD à destination des compositeurs de musiques nouvelles [GR04].

D'un point de vue plus théorique, le problème de la transcription des signaux percussifs est très intéressant par ses différences avec son homologue tonal : contrairement aux signaux des instruments mélodiques ou harmoniques qui peuvent se modéliser simplement par des peignes harmoniques, il n'existe pas de modèle simple des signaux percussifs. De plus, la transcription d'une mélodie utilise une échelle ordonnée (échelle continue de fréquences, éventuellement quantifiée en tons), tandis que la transcription de la batterie utilise des catégories (grosse caisse et caisse claire par exemple). Il y a donc lieu de penser que des outils différents et originaux devront être mis en oeuvre pour effectuer cette dernière.

1.2.2 Transcription musicale et image

Aujourd'hui, une part grandissante de la musique est diffusée accompagnée d'images, qu'il s'agisse de clips vidéos distribués en masse sur l'internet, vendus pour être visionnés sur des lecteurs multimédia portables, ou présents sur DVD en accompagnement d'un album. La popularité de ces documents audiovisuels musicaux étend le problème de la transcription dans de nouvelles directions :

1. Comment peut-on utiliser l'information présente dans les images pour améliorer ou guider la transcription musicale ? Il semble en effet raisonnable de croire que les gestes des musiciens dans un clip vidéo ou une vidéo de concert fournissent une information qui sera complémentaire, ou qui renforcera l'information contenue dans le signal audio.
2. Comment utiliser les outils de transcription musicale et d'analyse vidéo pour découvrir (à des fins d'indexation) les relations liant l'image au son – à quel degré l'image est-elle une illustration de la musique ?

Considérons également les applications de la transcription musicale aux interfaces musicien/machine. Dans les applications où l'on souhaite capturer avec le maximum de précision le jeu d'un musicien,

FIG. 1.1 – HAL9000 saurait-il toujours transcrire une partition dans ces circonstances ?

utiliser des capteurs vidéo apparaît comme une solution intéressante – car de tels capteurs n'interfèrent pas avec l'instrument. Les capteurs vidéos trouveraient de plus tout leur intérêt dans les situations où des capteurs audio seraient mis en défaut (par exemple en présence d'autres musiciens à proximité du musicien dont on veut saisir le jeu). On notera la similarité avec le domaine de la reconnaissance de la parole audiovisuelle.

1.3 Définitions, champ d'étude et restrictions

Dans cette section, nous définissons quelques termes utilisés au fil de cette thèse ; nous précisons également notre champ d'étude : quels types d'enregistrements audio, de séquences vidéos seront considérés ; et quel type d'information en sera extrait ?

1.3.1 Rythme et percussion

Il est communément admis qu'il n'existe pas de définition universelle du rythme – il n'en existe que des définitions pragmatiques propres à une application ou à un problème donné (voir par exemple [Deu82] pour une telle concession). Nous pouvons cependant déjà distinguer deux sens du mot *rythme* dans son usage courant :

1. Le rythme en tant que structure temporelle (horizontale) des événements musicaux, par opposition à la mélodie ou à l'harmonie qui décrivent des structures de hauteur (verticales). Ce sens est le plus fréquent dans des contextes musicaux : *rythme de ska*, *avoir le rythme dans la peau*...
2. Le rythme en tant qu'ensemble de sons produit par des instruments à percussion, par exemple la batterie dans la musique populaire occidentale ou le Tabla dans la musique classique de l'Inde du nord, dans le but de créer ou souligner ces structures temporelles. Ce sens du mot *rythme* se retrouve dans des expressions comme *boîte à rythmes* ou *section rythmique*.

Ces deux définitions renvoient d'une part à un phénomène abstrait (une structure de durées), et d'autre part à la façon dont il peut s'incarner dans un phénomène physique (acoustique). Dans cette thèse, le phénomène abstrait sera désigné par le terme *rythme*, tandis que son incarnation sous la forme de sons sera désignée par l'expression *signal percussif* ou *piste de batterie*, lorsque ces sons sont produits par la batterie. Le terme *piste*, tiré du langage des ingénieurs du son, rappellera constamment notre objectif d'analyser ou de traiter des enregistrements musicaux.

1.3.2 Éléments constitutifs du rythme

La définition que nous venons d'adopter – le rythme est la structure temporelle des événements musicaux – pourrait suggérer que décrire intégralement le rythme d'un enregistrement musical consisterait à extraire la liste de tous les instants auxquels un début de note est perçu (instants désignés par la suite par le terme anglais *onset*). Cette description est cependant insuffisante : une description du rythme ne doit pas se restreindre à une description superficielle sous forme de liste d'onsets, mais doit aussi recenser les formes et structures que les auditeurs percevront.

Parmi ces formes, figure tout d'abord la métrique qui désigne une hiérarchie de pulsations périodiques coïncidant maximalement avec les onsets perçus. Cette structure de pulsations n'est pas explicitement présente dans le rythme (on peut percevoir une pulsation là où il n'y a aucun onset) – mais l'auditeur s'attend à ce que les onsets perçus coïncident avec cette structure. Tout se passe comme si l'auditeur superposait plusieurs horloges ou métronomes internes, dont les périodes sont des multiples entiers les unes des autres, coïncidant avec les onsets perçus. Les niveaux hiérarchiques définissant la métrique sont les suivants : le *tatum*, la plus petite pulsation qui coïncide avec le plus grand nombre d'onsets ; le *tactus* (encore appelé pulsation ou *beat*) qui désigne l'intervalle entre deux battements tels qu'ils pourraient être produits par un auditeur tapant du pied en suivant la musique ; et la mesure – groupement de pulsations aux frontières desquelles sont susceptibles de s'articuler les phrases musicales. Nous soulignons ici que la durée de ces pulsations est subjective : certaines auditeurs tapent du pied avec une période double, ou de moitié, du *tactus* véritable !

Une autre propriété perçue par les auditeurs est l'accent. L'accent désigne le phénomène par lequel certains onsets seront perçus comme plus importants ou significatifs que d'autres. Les indices utilisés pour discriminer les onsets importants des autres sont très divers : il peut s'agir par exemple de l'amplitude, du timbre, de la durée de la note commençant à l'onset considéré (l'alternance cymbale hi-hat ouverte et fermée dans un rythme de charleston par exemple). Les mécanismes par lesquels l'auditeur impose une structure d'accent sur une séquence sont cependant mal compris : par exemple, un accent peut être perçu là où il n'y a qu'une séquence d'événements identiques (le *tic tac* d'une horloge), et une structure d'accent différente peut être perçue en jouant une même séquence musicale à des points de départ différents.

Enfin, les différences perçues entre les durées attendues ou prédites par la métrique, et les onsets réels donnent lieu à la sensation de *swing* ou d'expressivité dans la musique.

Ainsi, décrire le rythme exige à la fois d'extraire les onsets, mais également les différentes sensations liées aux onsets que l'auditeur percevra. Notons que ce point de vue privilégie l'auditeur. De façon duale, on pourrait aussi chercher à décrire le rythme en remontant la chaîne de production musicale, et en expliquant une liste d'onsets comme le produit de différents facteurs : les structures métriques retenues par le compositeur, les valeurs des notes par rapport à ce cadre métrique, le tempo à laquelle l'oeuvre est interprétée, les variations de durées par lesquelles l'interprète nuance son jeu... La *Théorie Générative de la Musique Tonale* [LJ83] propose une formalisation des deux premières étapes.

Champ d'étude et restrictions Nous ne proposons pas dans cette thèse de nouveaux outils d'extraction du rythme, mais nous utilisons des outils existants chaque fois que nécessaire dans le procédé de transcription.

1.3.3 La piste de batterie

1.3.3.1 Les sons de la batterie

Les instruments à percussion les plus courants dans la musique populaire occidentale sont ceux de la batterie. La batterie se compose de deux types d'éléments :

1. Les *membranophones*, constitués d'un fût sur lequel sont fixées deux membranes (peaux). On trouve dans cette catégorie :
   - La caisse claire, dont le diamètre varie entre 25 et 35 cm, la profondeur entre 10 et 20 cm, et dont le son caractéristique est dû au *timbre*, une grille de fils métalliques fixée sur la membrane inférieure.
   - La grosse caisse, dont le diamètre varie entre 45 et 65 cm, et qui se joue essentiellement au pied, à l'aide d'une pédale à laquelle est fixée une tête en caoutchouc.
   - Les toms, présents en plusieurs exemplaires de diamètre variable. Contrairement aux autres éléments de la batterie, ils peuvent être accordés pour produire des notes de hauteur définie. Notons que certains toms ne possèdent qu'une membrane.
2. Les *idiophones*, constitué d'un disque de métal. On distingue :
   - Les cymbales ride, crash, chinoise – qui se distinguent par l'alliage dont elles sont faites et leur diamètre (de 40 à 55 cm).
   - La Hi-hat (ou charleston), qui se compose de deux petites cymbales (de 30 à 40 cm), dont l'une est montée sur un support contrôlé par une pédale. La pédale permet de garder les deux cymbales en contact (hi-hat fermée) ou de les séparer (hi-hat ouverte).

Différents facteurs expliquent la très grande variabilité de timbres observée dans les sons de la batterie. Premièrement, comme souligné plus haut, les matériaux et tailles de chacun des instruments de la batterie peuvent varier, résultant en une diversité de timbres. Deuxièmement, à l'exception de la grosse caisse presque toujours jouée au pied, les autres éléments peuvent être joués, selon le style musical, aux baguettes, aux fagots (fins rondins de bois liés), aux balais, aux mailloches ou avec les mains. Troisièmement il existe des modes de jeu et variantes spécifiques à certains des instruments. Par exemple, la cymbale ride peut être jouée en la frappant sur le bord, ou en son sommet (dôme). Différents modes de jeu de la caisse claire existent : frappe sur la peau, frappe simultanée de la peau et du cercle du fût (*rim shot*), frappe du bord du fût tandis que l'extrémité de la baguette repose sur le fût (*cross stick*). Enfin, il faut rappeler que les choix de l'ingénieur du son – type de microphone utilisé, égalisation, traitements de dynamique, ajout éventuel de réverbération artificielle – vont déterminer comment "sonne" une batterie.

Cette variabilité des sons de la batterie se manifeste bien entendu entre divers enregistrements, mais également au sein d'un même enregistrement – le batteur pouvant alterner les techniques de jeu pour marquer l'accent, l'ingénieur du son pouvant également appliquer à la piste de batterie des effets variant au cours du temps. Ce tour d'horizon des facteurs de variabilité parmi les sons de la batterie ne serait complet sans évoquer les sons de batterie de synthèse, largement utilisés dans la production musicale contemporaine. On peut les classer en trois catégories :

- Les sons produits par des boîtes à rythmes ou synthétiseurs à base d'échantillonnage. Dans ce cas, les sons produits sont équivalents à ceux des batteries acoustiques, mais n'en possèdent pas la variabilité.
- Les sons produits à l'aide de boucles de batterie échantillonnées (utilisées par exemple dans le hip-hop). Même si ces boucles sont originellement des enregistrements de batterie acoustique, leurs propriétés timbrales sont très particulières, parce qu'elles ont traversé une ou plusieurs chaînes complètes d'enregistrement, mastering, gravure puis lecture sur disque vinyle.
- Les sons produits par synthèse, comme ceux produits par les boîtes à rythmes Roland TR utilisées abondamment dans les musiques Techno et Electro. Ces sons reproduisent de façon très grossière les propriétés des sons naturels.

Terminons enfin par une distinction fondamentale : on peut s'intéresser à la transcription rythmique monophonique (la batterie est jouée seule) ou polyphonique[2] (la batterie joue en accompagnement d'autres instruments).

Champ d'étude et restrictions Les méthodes que nous présentons dans cette thèse s'appliquent à tous les sons de la batterie, et considèrent toutes leurs variations de mode de jeu. Même si l'évaluation n'est réalisée que sur des enregistrements de batterie acoustique, les méthodes que nous introduisons se généralisent aux sons de batterie de synthèse. Les méthodes que nous présentons sont explicitement conçues pour gérer le cas polyphonique. Elles seront également évaluées sur des enregistrements monophoniques.

1.3.3.2 Description d'une piste de batterie

Une forme de transcription rythmique possible consisterait en une liste de couples (t_i, e_i) où t_i est un instant (onset) et e_i est le nom de l'instrument de la batterie (label) joué à l'instant t_i. Notons qu'une telle partition n'est qu'une description de *surface* de la piste de batterie. Une description plus complète pourrait inclure les formes rythmiques soulignées en 1.3.2 et déduites de la liste d'onsets t_i, ainsi que d'autres informations extraites de la suite des labels e_i.

Par exemple, à chaque genre musical sont associés des motifs rythmiques typiques, en particulier dans le jeu de la grosse caisse et de la caisse claire. De tels motifs sont contraints par le genre, mais aussi par des règles de composition ou les limites du musicien. Une description plus complète de la piste de batterie pourrait inclure une telle analyse de haut niveau pour isoler les motifs, et reconnaître à quel genre ils sont associés.

Champ d'étude et restrictions Nous nous restreignons ici à une transcription de surface de l'accompagnement rythmique. Cependant, nous préparons le terrain pour un niveau supérieur de description de la piste de batterie, en portant toute notre attention sur ses éléments les plus courants : la grosse caisse, la caisse-claire, et la hi-hat. Par ailleurs, nous utilisons certaines connaissances sur ses caractéristiques de haut niveau (y compris celles dépendantes du genre) pour améliorer la transcription. Ainsi, même si notre objectif est d'en extraire une transcription de surface, nous n'ignorons pas les propriétés de haut-niveau des accompagnements rythmiques.

1.3.4 Documents audiovisuels musicaux, scènes musicales audiovisuelles

Nous désignons par *document audiovisuel musical*, tout document audiovisuel dont la partie audio contient exclusivement de la musique. Cela inclut par exemple les enregistrements vidéos de concert ou d'opéra, les clips vidéos ou une séquence d'un guide vidéo d'enseignement d'un instrument.

Nous désignons par *scène musicale audiovisuelle* un document audiovisuel montrant un plan fixe d'un ou plusieurs instrumentistes jouant une oeuvre. Une *scène musicale audiovisuelle* peut apparaître dans un *document audiovisuel musical* (par exemple, une retransmission d'un concert peut alterner entre des images du public et des musiciens).

Champ d'étude et restrictions Nous nous intéressons d'abord dans cette thèse au problème de la transcription musicale à partir de *scènes musicales audiovisuelles* – puisque ces documents sont explicitement construits pour documenter et illustrer visuellement le jeu de l'instrument. Cependant la problématique de l'indexation exige de prendre en compte une classe de contenus la plus large possible, nous étudierons donc par la suite quel type d'information peut être extrait des *documents audiovisuels musicaux*, même si dans leur cas la description extraite s'éloigne de la partition.

[2] Polyphonique est ici à prendre au sens de *multi-instrumentale*.

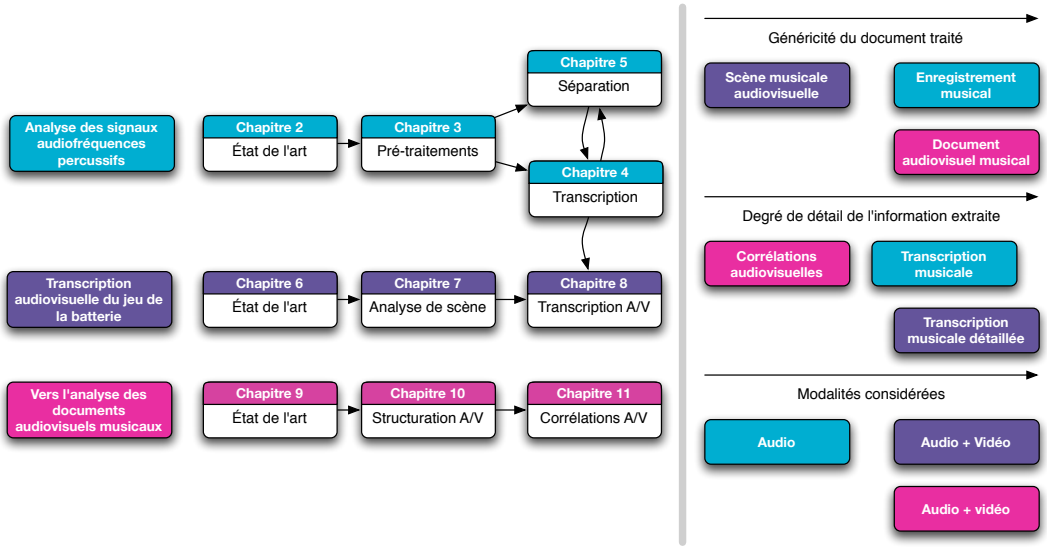


FIG. 1.2 – Plan de la thèse et champ d'étude

1.4 Plan d'étude et résumé des contributions

1.4.1 Plan du document

Nous étudions tout d'abord dans une première partie le problème de la transcription de la piste de batterie d'enregistrements musicaux polyphoniques, à partir de la modalité audio seule. Nous listons dans le chapitre 2 différentes solutions partielles qui ont été proposées dans la littérature à ce problème. Après avoir présenté dans le chapitre 3 des pré-traitements permettant d'accentuer la piste de batterie, nous mettons en oeuvre au chapitre 4 des techniques d'apprentissage statistique (Machines à Vecteurs de Support) sur une large gamme d'attributs pour réaliser la transcription. Nous détaillons ensuite dans le même chapitre deux approches, l'une supervisée, l'autre non, pour améliorer la reconnaissance en prenant en compte le caractère périodique et structuré des accompagnements rythmiques. Nous considérons également au chapitre 5 le problème de l'extraction de la piste de batterie à des fins de remixage, en proposant des améliorations à une méthode de séparation existante (basée sur le filtrage de Wiener), et en proposant une nouvelle approche utilisant un masquage temps/fréquence/sous-espace. Les liens entre les problèmes de transcription et de séparation seront soulignés dans la conclusion de cette première partie.

Dans une seconde partie, nous incorporons au système de transcription précédent l'information visuelle provenant d'une ou plusieurs caméras filmant le batteur. Les similarités et différences entre cette tâche et des problèmes plus classiques de reconnaissance ou de suivi vidéo de gestes sont présentées au chapitre 6. Nous étudions au chapitre 7 le problème de la segmentation de la scène – comment détecter sur l'image les différents éléments de la batterie et éventuellement les associer à des catégories sonores ? – et plusieurs de ses variantes supervisées et non-supervisées, unimodales ou multimodales, correspondant à divers scénarios d'usage. Une fois cette segmentation effectuée, des descripteurs d'intensité de mouvement sont utilisés pour permettre la détection des frappes. Nous évaluons dans le chapitre 8 différentes stratégies de fusion permettant la combinaison des transcriptions audio et vidéo, pour illustrer l'intérêt d'incorporer une information vidéo (si elle est disponible) dans les applications de transcription musicale. Les résultats démontrent la capacité d'une approche

multimodale à résoudre certaines des ambiguïtés propres à la transcription audio, à condition que les conditions de prise de vue soient bien contrôlées. Nous concluons que ces contraintes ne sont pas gênantes pour certaines applications (système d'aide à l'apprentissage de la batterie par exemple), mais ne permettent pas pour l'heure de traiter des documents audiovisuels musicaux commerciaux.

C'est ce type de documents que nous considérons dans la dernière partie. S'il n'est pas possible d'utiliser l'information visuelle qu'ils contiennent pour améliorer la transcription musicale, nous suggérons cependant d'autres applications à la croisée des domaines de l'indexation audio et vidéo. Après avoir présenté, au chapitre 9, quelques problèmes connexes (en particulier des problèmes liés à l'analyse de clips vidéos), nous nous intéressons dans le chapitre 10 au problème consistant à évaluer de quelle façon une musique peut être illustrée par des images. Nous présentons ou introduisons à cet effet de nouvelles méthodes de structuration automatique des flux audio et vidéo – segmentation en notes et sections pour la musique, en mouvements, plans et séquences pour la vidéo. Le chapitre 11 définit des mesures de corrélation sur les structures obtenues : en plus de permettre des applications de recherche de musique par l'image, ces corrélations sont fortement dépendantes du type de document musical (clip vidéo narratif, vidéo des musiciens, danse).

Enfin, le chapitre 12 propose diverses perspectives de recherche, liées aux problèmes de la transcription des signaux percussifs, ou à l'utilisation de la modalité vidéo en indexation audio.

Le plan du document est schématisé dans la figure 1.2.

1.4.2 Résumé des contributions

Nous listons maintenant nos contributions principales :

En transcription automatique de la piste de la batterie

- L'enregistrement et l'annotation de la base de recherche ENST-drums contenant plus de 3h30 de jeu de batterie enregistré en multipiste et filmé sous deux angles. Une telle base, unique en son genre, a permis des expériences jusque là inaccessibles, et est diffusée publiquement à des fins de recherche.
- L'introduction de divers pré-traitements pour l'analyse des signaux percussifs dans un enregistrement musical polyphonique, visant à atténuer les instruments non percussifs.
- L'application de méthodes d'apprentissage statistiques (machine à vecteurs de supports) à la transcription de séquences de batterie – avec un accent particulier sur la sélection d'attributs pour la classification, et l'évaluation de la robustesse de ces attributs en présence d'autres instruments. Un aspect original de notre contribution est d'utiliser à la fois des attributs calculés sur le signal original, et sur une version dans laquelle les instruments non percussifs ont été atténués.
- L'utilisation de modèles de séquences (N-grammes, N-grammes généralisés) pour améliorer la qualité de la transcription. Nous mettons en particulier l'accent sur les limites des méthodes d'apprentissage supervisé des modèles de séquences dans des situations réalistes d'utilisation.
- La présentation d'un critère de complexité mesurant la régularité des transcriptions rythmiques. Minimiser ce critère permet de corriger les erreurs de transcription, de manière non-supervisée.
- L'extension d'une méthode de séparation de sources à un seul capteur basée sur le filtrage de Wiener au problème de la séparation de la piste de batterie.
- L'introduction d'une méthode de séparation de sources spécifique à la batterie, basée sur un masquage temps/fréquence/sous-espace.

En analyse musicale audiovisuelle

- La proposition de différents attributs permettant l'analyse visuelle de scènes de jeu de batterie : segmentation et suivi de mouvement.
- L'évaluation de diverses méthodes de calibration permettant d'associer automatiquement des événements visuels à des classes d'instruments.

- La description et l'évaluation d'un système complet d'analyse audiovisuelle du jeu de la batterie.
- Une discussion de l'intérêt relatif des approches de détection et de classification supervisée pour l'analyse audiovisuelle de scènes musicales.
- Une évaluation de l'apport des méthodes à noyaux pour la segmentation d'enregistrements musicaux.
- Une méthodologie de sélection de variables pour les tâches de segmentations de signaux, et son application au problème de la segmentation d'enregistrements musicaux.
- L'introduction de critères de corrélation entre différents niveaux de structures audio et vidéo, et quelques illustrations de leur intérêt pour l'indexation de documents audiovisuels musicaux.

Première partie

Analyse des signaux audiofréquences percussifs : application à la batterie

CHAPITRE 2

Transcription automatique des signaux percussifs : un état de l'art

Ce chapitre est consacré aux diverses méthodes de traitement de signal proposées dans la littérature pour l'analyse automatique des signaux percussifs, à travers deux problèmes clés : la description du contenu rythmique des signaux de musique, et la transcription de surface des signaux percussifs. Quelques solutions apportées au problème de l'analyse rythmique sont présentées dans la section 2.1. Nous accordons une importance particulière à la détection des onsets (description rythmique de surface) à partir de signaux audio, cette étape étant essentielle pour de nombreuses tâches de transcription automatique et d'indexation. Dans la section 2.2, nous présentons trois grandes familles de systèmes de transcription de signaux percussifs, en insistant sur leurs domaines d'application et leurs limites respectives. Nous terminons cet état de l'art en détaillant dans la section 2.3 quelques unes des approches utilisées pour intégrer des connaissances musicales aux systèmes de transcription de la piste de batterie ; et en passant en revue dans la section 2.4 quelques applications intéressantes de ces systèmes.

2.1 Analyse du rythme

Nous avons distingué à la section 1.3.2 deux niveaux de description du rythme : le niveau superficiel, constitué de la liste des instants auxquels le début d'un événement musical est perçu (onsets) ; et le niveau des formes perçues à partir de cette structure. On peut donc séparer la tâche de description du rythme en deux étapes : l'extraction d'une description de surface à partir d'un signal audiofréquence, présentée en 2.1.1, puis l'extraction des propriétés de métrique ou d'accent à partir de cette description présentée en 2.1.2. Notons que quelle que soit la propriété de haut niveau extraite (métrique, tempo), l'analyse de surface est nécessaire[1] – ce qui explique l'abondance dans la littérature de travaux traitants de la détection d'onsets.

2.1.1 Des signaux aux descriptions de surface

2.1.1.1 Détection sur un critère de variation d'énergie

Principe Les premiers systèmes de détection d'onsets décrits dans la littérature extraient l'enveloppe d'amplitude du signal à considérer et cherchent les maxima de sa dérivée. Par exemple, le système décrit par Schloss dans [Sch85] utilise le maximum de la valeur absolue du signal sur des fenêtres de 10 ms comme estimée de l'enveloppe d'amplitude. Une fenêtre glissante de 4 valeurs de

[1] Sauf dans le cas où nous effectuons une analyse rythmique de haut niveau à partir d'une liste d'onsets enregistrée par des capteurs ou des instruments MIDI.

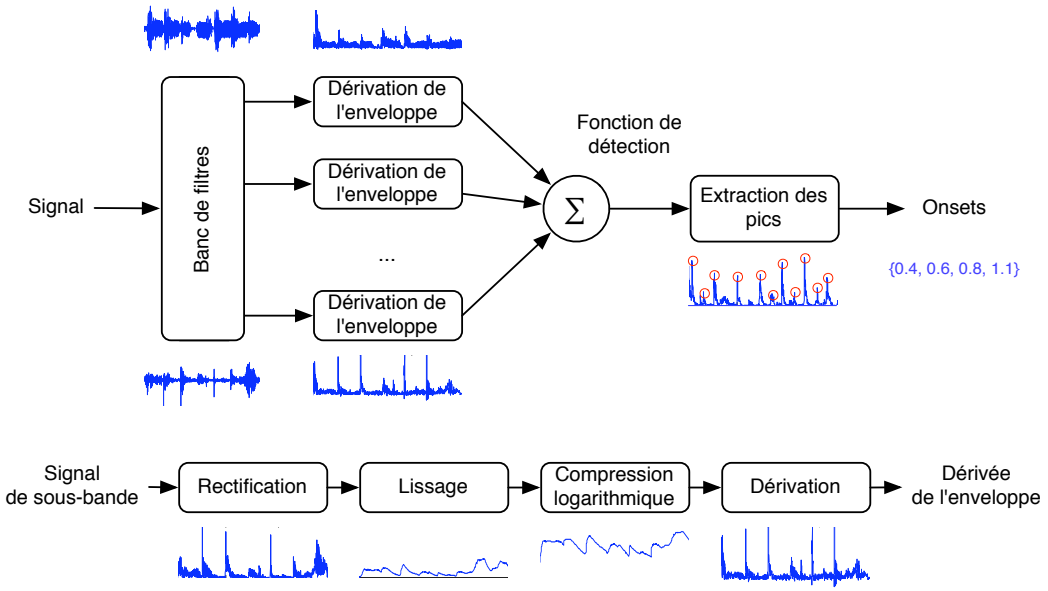


FIG. 2.1 – Architecture typique d'un système d'analyse de surface du rythme

cette estimée est ensuite considérée. Sur cette fenêtre, l'enveloppe d'amplitude est approximée par une droite, permettant ainsi le calcul de la dérivée. Les maxima de la dérivée au dessus d'un certain seuil sont considérés comme des onsets, et une durée minimale est imposée entre onsets consécutifs. Pour l'application de transcription de signaux de congas considérée par Schloss, il est suggéré de pré-traiter le signal par un filtre passe-haut qui atténue la composante résonnante de la note – composantes tonales dont la fréquence est de l'ordre de plusieurs centaines de Hertz – tout en retenant le transitoire produit lors de la frappe de l'instrument – impulsion s'étendant sur toute la largeur du spectre. Cette méthode est reprise par Dixon [Dix01] pour l'analyse d'enregistrements de piano, en utilisant la moyenne de la valeur absolue du signal sur des fenêtres de 20 ms pour estimer l'enveloppe d'amplitude. Nous présentons dans [GR03] un système de transcription du Tabla[2] utilisant une approche similaire.

Ces méthodes ne sont efficaces que pour des signaux monophoniques très impulsifs, et peinent, par exemple, à détecter des attaques lentes (comme celles d'un instrument à cordes frottées) ou noyées dans la partie entretenue d'une note d'un autre instrument. Cet échec a motivé l'apparition de nouvelles méthodes basées sur des bancs de filtres, utilisant des techniques plus robustes de calcul de la dérivée de l'enveloppe, ou employant d'autres critères de détection des onsets.

Détection par sous-bandes L'intérêt des bancs de filtres pour la détection des onsets est multiple. Tout d'abord, ils permettent de minimiser l'impact des composantes tonales – qui ne sont localisées que dans un nombre minoritaire de bandes, tandis que les attaques des notes, – phénomènes impulsifs à spectre large – se manifestant simultanément dans toutes les bandes. Par ailleurs, le choix du banc de filtre peut être motivé par des modèles perceptuels, le processus de détection d'onsets s'attachant alors à reproduire les traitements effectués par l'appareil auditif humain.

Par exemple, le système de détection du tempo présenté par Scheirer dans [Sch98] utilise un banc de filtres logarithmique à 6 voies, dont les limites des bandes sont $0, 200, 400, 800, 1600, 3200, \frac{f_s}{2}$ Hz où f_s est la fréquence d'échantillonnage. L'extraction des enveloppes d'amplitude dans chacune des voies est effectuée en convoluant la partie positive du signal de sous-bande par une demie fenêtre de Hann (cosinus surélevé) longue de 200 ms. Aucun consensus n'existe sur la décomposition

[2]Instrument à percussion de l'Inde du nord se composant de deux tambours.

fréquentielle optimale : Seppänen utilise une variante de cette méthode [Sep01] avec un banc de filtre à 8 bandes, Goto utilise 14 bandes [GM95], Uhle et Herre en utilisent 7 [UH03], leur méthode se distinguant en outre par le choix d'un filtre passe-bas différent pour l'extraction des enveloppes d'amplitude. Dans [ABDR03], Alonso et al. utilisent une décomposition uniforme sur 12 bandes. Le choix du nombre de bandes semble dans tous les cas guidé par des observations empiriques.

Une voie plus originale a été suivie par Klapuri [Kla99], qui motive le choix de chacun des modules de son système par des considérations psychoacoustiques. Le signal musical est d'abord analysé par un banc de filtres à 21 voies – chaque voie correspondant approximativement à une bande critique. La valeur absolue de chaque signal de sous-bande est sous-échantillonnée, et lissée par un filtre de réponse impulsionnelle égale à une demie fenêtre de Hann de 100 ms. Cette intégration de l'énergie est similaire à celle effectuée par l'appareil auditif humain. Klapuri propose ensuite de considérer non pas la dérivée de l'enveloppe, mais la dérivée de son logarithme (dérivée relative) – remarquant que la sensibilité aux variations d'intensité sonore dépend de cette intensité. Les maxima locaux détectés dans chacune des bandes sont ensuite groupés, et un modèle perceptuel d'intensité est utilisé comme critère de détection.

Des travaux plus récents considèrent la transformée de Fourier à Court Terme (TFCT) du signal à analyser en lieu et place d'un banc de filtre. Celle-ci fournit en effet l'équivalent d'une décomposition par un banc de filtre uniforme – efficace à calculer et permettant une analyse sur un grand nombre de voies. Cette approche est retenue par Laroche [Lar01; Lar04]. L'analyse temps-fréquence réalisée par la TFCT permet le calcul du flux d'énergie spectral – *Spectral Energy Flux* (SEF), défini comme la dérivée par rapport au temps de l'énergie dans chacun des canaux de la TFCT. Alonso et al. présentent dans [ARD05] une formulation rigoureuse de cette méthode : le calcul de l'énergie dans chacun des canaux de la TFCT utilise un filtrage passe-bas compatible avec un modèle de réponse du nerf auditif ; tandis que l'opération de dérivation utilisée dans le calcul du SEF est effectuée par un filtre différentiateur optimal. Notons que la faible résolution fréquentielle associée à la TFCT peut être améliorée par l'utilisation de techniques de réallocation [Alo06].

2.1.1.2 Autres critères pour la détection d'onsets

Si les critères basés sur l'énergie ou l'enveloppe des signaux de sous-bande sont les plus courants, d'autres critères leur sont parfois préférés :

Critère de nouveauté Les onsets peuvent être considérés comme les frontières de segments durant lesquels les propriétés du signal restent stables. De telles frontières peuvent alors être détectées en considérant une fenêtre glissante et en comparant ses deux moitiés – si la seconde moitié est "nouvelle" ou "surprenante" comparée à la première, alors le milieu de la fenêtre est un onset. Une telle approche a été utilisée par exemple par Abdallah et Plumbey [AP03], et par Davy et Godsill [DG02]. Notons que le problème plus général de la segmentation de documents multimédia est traité au chapitre 9 – on peut s'y référer pour une présentation plus exhaustive des méthodes de détection de nouveauté.

Critère de déviation de phase Bello et Sandler utilisent dans [BS03] la dérivée seconde de la phase entre trames adjacentes de la TFCT. Ce critère peut être couplé [BDDS04] à un critère d'énergie (ou module), en considérant le module de la différence entre une amplitude complexe prédite et une amplitude complexe observée sur des trames adjacentes de la TFCT.

Critère d'erreur de modélisation Un dernier critère utilisé pour la détection de notes est fondé sur l'observation suivante : les onsets correspondent à des transitoires difficiles à modéliser. Il est donc intéressant de considérer les instants auxquels le résidu de modélisation est maximal, pour un modèle de signal donné. Un modèle couramment utilisé pour les signaux des instruments non-percussifs est le modèle sinusoïdal, utilisé par Duxbury et al. dans [DDS01], ou par Alonso et al. dans [ARD07]. L'apport de cette dernière méthode est cependant limité lorsqu'il s'agit de détecter

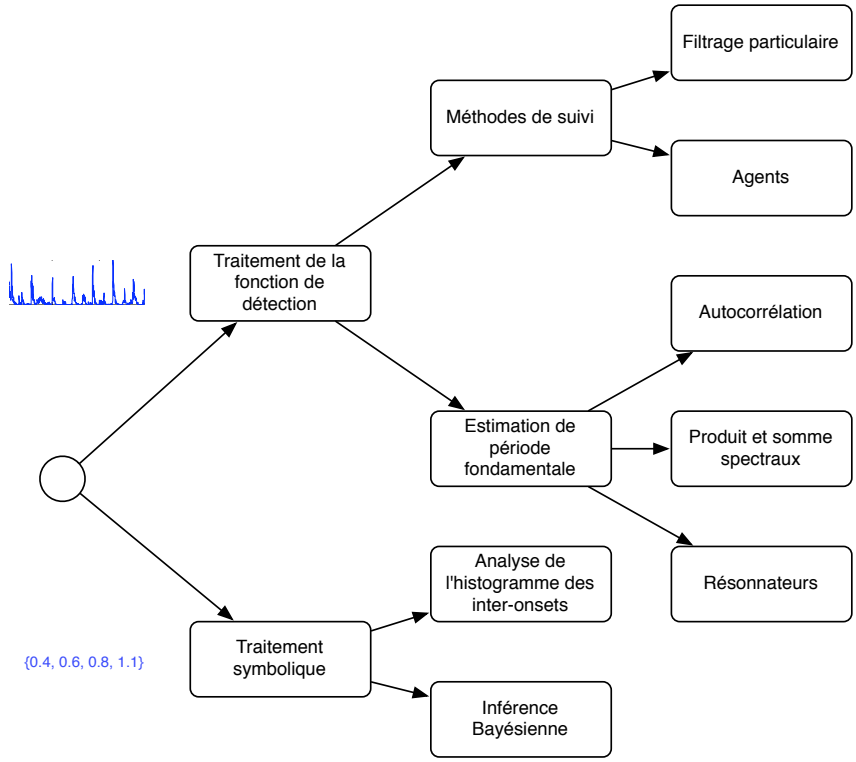


FIG. 2.2 – Quelques procédés d'extraction de formes rythmiques

les onsets associés aux instruments percussifs – en effet, le modèle sinusoïdal n'en fournit pas une représentation pertinente.

Sachant qu'il n'existe pas de modèle exact de ce qu'est un onset et de ses propriétés, une direction de recherche intéressante consiste à considérer plusieurs de ces critères, et à utiliser des approches d'apprentissage statistique supervisé pour classer des trames de signal en classes onset/non onset. Cette voie prometteuse est suivie par Lacoste et Eck [LE07].

2.1.2 Des descriptions de surface aux formes

Nous présentons ici brièvement quelques-unes des méthodes introduites dans la littérature pour l'extraction de descriptions de haut niveau (tempo, métrique) à partir des onsets. Ces méthodes sont schématisées dans la figure 2.2.

2.1.2.1 Analyse de l'histogramme des intervalles inter-onsets

Divers niveaux de périodicité peuvent être déterminés en recherchant les intervalles les plus fréquents entre des onsets consécutifs. Cela suggère de quantifier les durées entre onsets successifs pour en établir l'histogramme – une approche simple introduite par Schloss [Sch85] et suivie par Uhle et Herre [UH03], ou par Gouyon et al. [GHC02]. Cette méthode impose cependant un compromis entre résolution et robustesse. Une amélioration introduite par Dixon dans [Dix01] consiste alors à effectuer un clustering des intervalles inter-onsets, de manière à construire un histogramme dont les classes sont non-uniformes.

2.1.2.2 Analyse de la fonction de détection pour l'extraction de périodicités

Précisons tout d'abord que les systèmes de détection d'onsets produisent une liste des instants où débute une note. Il est cependant plus intéressant de considérer une fonction de détection – fonction continue présentant des maxima locaux aux instants t correspondant aux débuts de note. Le problème de l'estimation métrique consiste alors à chercher une périodicité dans la fonction de détection. Plusieurs méthodes ont été proposées pour cette tâche.

Maxima de l'autocorrélation Les maxima de l'autocorrélation correspondent aux périodicités candidates. Cette méthode est évaluée par exemple par Alonso et al. [ABDR03]. Un estimateur de fréquence fondamentale plus sophistiqué basé sur l'autocorrélation (YIN) est utilisé par Paulus et Klapuri [PK02].

Estimateurs robustes de fréquence fondamentale Le produit et la somme spectraux, deux méthodes robustes d'estimation de période, ont été utilisés par Alonso et al. dans [ABDR03].

Résonateurs La fonction de détection est filtrée en parallèle par plusieurs résonateurs, par exemple des filtres en peigne. À chaque résonateur correspond une période fondamentale, et le *tactus* estimé correspond au résonateur d'excitation maximale. Cette solution est retenue par Scheirer [Sch98].

Agents Cette méthode consiste à maintenir une liste d'hypothèses de périodes (agents). Chaque agent effectue des prédictions quant à l'instant auquel apparaîtra le prochain onset, la qualité de ces prédictions permettant de donner un score à chaque agent. Les agents dont les scores sont faibles sont supprimés, et de nouvelles hypothèses de périodicité peuvent ainsi être introduites. Goto introduit cette méthode dans [GM95], également utilisée par Dixon dans [Dix01]. Cette méthode se veut être une simulation du processus de perception du rythme par un auditeur humain – bien qu'on puisse la considérer également comme une formulation d'un algorithme de recherche en faisceau. Une autre famille de modèles visant à reproduire le processus de formation d'hypothèses de tempo par l'auditeur se base sur le filtrage particulaire [HM03].

2.1.2.3 Analyse bayésienne pour l'extraction conjointe du tempo et des valeurs de notes

Terminons par une dernière famille de méthodes décrites dans la littérature, qui visent à extraire d'une séquence d'inter-onsets à la fois une information de tempo et la valeur des notes correspondantes (mesurée, par exemple, par leur rapport à la valeur d'une noire). Une telle entreprise se heurte à des questions du type suivant : s'agit-il de noires à un tempo de 120 battements par minutes, ou de croches à un tempo deux fois plus lent ? De telles ambiguïtés peuvent être résolues dans un formalisme Bayésien – en proposant un modèle probabiliste des variations de tempo et des successions de valeurs de notes. Raphael propose une telle méthode dans [Rap01]. Une solution similaire traitant en bloc des groupes de notes est proposée par Takeda et al. dans [TNS04]. Des modèles plus réalistes de variation de tempo sont proposés par Filippi dans [Fil06].

2.2 Analyse des signaux percussifs : les trois approches

Nous présentons à présent les trois familles de solutions introduites dans la littérature au problème de la transcription des signaux percussifs : Segmenter et Reconnaître (SegRec), Mettre en correspondance et Adapter (MatAda), Séparer et Détecter (SepDet).

2.2.1 SegRec : Segmenter et reconnaître

2.2.1.1 Principe

Le problème de la transcription de signaux percussifs a été initialement considéré dans sa version monophonique – autrement dit lorsque la batterie (ou l'instrument à percussion considéré) joue seul, sans accompagnement. Une méthode directe pour obtenir une transcription peut consister à :

1. **Segmenter** le signal à transcrire de manière à délimiter chacune des frappes[3], tâche que peuvent effectuer les systèmes de détection d'onsets présentés en 2.1.1.
2. **Reconnaître**, pour chacun des segments, l'instrument ou la combinaison d'instruments qui a été joué. Cette tâche d'étiquetage est une instance particulière du problème général de la reconnaissance des instruments de musique dans un signal audio – on s'intéresse ici à discriminer les différents timbres correspondant à chaque instrument de la batterie (ou de l'instrument à percussion considéré), et à leurs différents modes de jeu.

2.2.1.2 Reconnaissance des instruments de musique

Les premiers travaux en reconnaissance des instruments de musique considèrent des notes isolées, sur toute leur longueur. Le cadre théorique retenu est celui de la reconnaissance des formes : un ensemble d'attributs (*features*) est extrait du signal, et utilisé pour l'apprentissage d'un classifieur. Les différentes méthodes proposées dans la littérature se distinguent par le nombre de classes considérées, le choix des attributs, et les techniques de classification mises en oeuvre. Les premiers travaux privilégient des méthodes de classification simples, comme les k plus proches voisins dans [Kam00; FM00; Ero01], et des ensembles d'attributs motivés par des résultats de psychoacoustique sur les dimensions du timbre. L'amélioration de ces méthodes se fait par la mise en oeuvre de techniques de sélection d'attributs, et l'utilisation de méthodes de classification plus robustes [Pee03].

Une direction plus récente, aux applications pratiques plus nombreuses, consiste à effectuer la reconnaissance non pas sur des notes isolées, mais sur de véritables enregistrements de soli instrumentaux. La tâche s'avère plus difficile car certains attributs (notamment d'enveloppe) ne peuvent plus être extraits – tandis que d'autres attributs perdent leur robustesse en situation polyphonique. Moreno et Marques présentent dans [MM99] un système testé sur des soli, utilisant modèle(s) de mélanges de gaussiennes – *Gaussian Mixture Model(s)* (GMM) et machine(s) à vecteurs de support – *Support Vector Machine(s)* (SVM).

Les travaux d'Essid et al. [ERD06b] prolongent rigoureusement ces recherches : les signaux considérés sont des phrases musicales tirées de soli réels, et des méthodes de sélection d'attributs et de classification éprouvées (SVM) sont utilisées. L'originalité de cette contribution consiste également en l'utilisation d'une stratégie de classification discriminant des paires d'instruments "un contre un", plutôt que des approches plus classiques de type "un contre tous". La sélection des attributs et des paramètres de classification optimaux peut ainsi être effectuée différemment pour chaque paire à discriminer.

Le problème de la reconnaissance d'instruments dans un contexte multi-instrumental a été peu traité : dans [VR04a], Vincent et Rodet décrivent un modèle Bayésien du contenu spectral d'un signal de musique permettant d'inférer la composition de la formation instrumentale (deux instruments parmi cinq) le décrivant le mieux. Le coût important de cette méthode en terme de calculs la rend difficile à généraliser à des sélections d'instruments plus nombreuses. Une approche plus pragmatique est suivie par Essid et al. dans [ERD06a] – elle consiste à utiliser une classification hiérarchique, discriminant différents types de formations musicales.

Ces approches peuvent-elles s'appliquer directement à la reconnaissance des instruments à percussion ? Le cadre théorique de la reconnaissance des formes et les outils de classification sont toujours valides, de même que certains des attributs utilisés. Ainsi, Gouyon et al. [GHD03] utilisent ces

[3]Nous préférerons par la suite le terme *frappe* à *note* pour rappeler que les événements constituant la transcription ne sont pas tonaux, et doivent donc être décrits par une classe plutôt que par une hauteur.

mêmes techniques pour classer des frappes isolées des différents instruments de la batterie (grosse caisse, caisse claire, toms, cymbales crash et ride, hi-hat). Une étude comparative des différents attributs à considérer et de diverses méthodes d'apprentissage statistique est proposée par Herrera et al. dans [HYG02]. Mais il ne s'agit ici que de reconnaissance de frappes isolées : ces résultats sont d'intérêt limité pour les applications de transcription de signaux percussifs. En effet :

- La reconnaissance de combinaisons d'instruments ne peut être ignorée. Un solo de violoncelle ne contient que des notes de violoncelle, tandis qu'un solo de batterie typique contient des combinaisons variées de frappes de chaque instrument.
- Dans les applications d'indexation d'enregistrements multi-instrumentaux, la reconnaissance est rendue encore plus difficile par la présence des autres instruments non-percussifs. Le problème s'apparente alors à un problème de classification de signaux bruités – si ce n'est que le bruit est ici hautement structuré et dépendant du signal à analyser.
- Une dernière difficulté est la longue décroissance de l'enveloppe de certains des instruments percussifs – cymbale crash et toms par exemple. Ces lentes décroissances forment ainsi une "traînée" qui sera superposée aux frappes suivantes. Ce type de situation adverse n'est pas rencontrée sur des frappes isolées.

2.2.1.3 Application des méthodes de classification supervisée à la transcription des signaux percussifs

Transcription de soli d'instruments percussifs Le premier système à combiner segmentation et classification des frappes est le système de transcription de séquences de Congas proposé par Schloss dans [Sch85]. Pour chaque segment de signal, les attributs extraits sont la constante de temps d'une exponentielle décroissante modélisant l'enveloppe d'amplitude de la frappe, l'énergie dans trois bandes de fréquences empiriquement choisies ($[0, 100]$ Hz, $[100, 1000]$ Hz, $[1000, \frac{f_s}{2}]$ Hz), l'écart type de ces énergies, et la période fondamentale. Les valeurs moyennes de ces paramètres sont estimées sur une séquence de référence jouée par l'instrumentiste au début de l'utilisation du système. Quatre types de frappes sont considérés par Conga, définissant ainsi 8 classes de frappes (les frappes combinées ne sont pas acceptées). La classification s'effectue par recherche du plus proche voisin, en utilisant une distance euclidienne pondérée.

Nous avons présenté [GR03] un système complet de transcription du Tabla dont l'architecture reprend celle proposée par Schloss. Douze attributs sont extraits de chaque segment, correspondant à la fréquence centrale, largeur, et amplitude des 4 pics principaux extraits du spectre. En dehors de cette paramétrisation originale adaptée aux signaux de Tabla, notre principale contribution réside dans l'emploi d'un modèle(s) de Markov caché(s) – *Hidden Markov Model(s)* (HMM) pour modéliser la suite de ces vecteurs de paramètres. L'intérêt de ce modèle est triple :

1. Il permet de prendre en compte une spécificité du système de notation des *bols*[4] – une même frappe peut être nommée par un *bol* différent en fonction de son contexte de jeu.
2. Il modélise certaines séquences de *bols* typiques qui forment des "mots" rythmiques.
3. Les modèles acoustiques associés à chacun de ses états sont contextuels – ainsi il existe un modèle différent de chaque frappe en fonction de son contexte de jeu. Cette approche permet de gérer efficacement les problèmes de "traînées" causées par les frappes longues et résonnantes.

Ces travaux relatifs au Tabla ont été étendus par la suite par Chordia dans [Cho05], où sont considérés une plus vaste palette d'attributs, et différents algorithmes de classification.

Un point commun des systèmes présentés jusqu'ici est que les instruments à percussion pour lesquels ils ont été développés ne possèdent pas de frappes combinées – dans le cas du Tabla, il existe en fait des frappes combinées, mais elles sont notées comme des frappes simples. Par exemple, la superposition de la frappe *Ge* et de la frappe *Na* est notée *Dha* – et constitue donc une catégorie à part.

[4] Syllabes utilisées pour désigner chacune des frappes de l'instrument, permettant aux musiciens de transmettre oralement leurs compositions en les récitant.

Nous avons présenté dans [GR04] une première étude relative à la transcription de la batterie par l'approche SegRec, qui aborde le problème des frappes combinées. Les enregistrements considérés sont des boucles de batterie issues de CD commerciaux, et présentant donc une grande variété de timbres d'instruments et de traitements. La segmentation est effectuée par l'algorithme de détection d'onsets de Klapuri [Kla99]. Pour chaque segment, sont extraits les 4 moments spectraux, la moyenne des 13 MFCC, et 6 paramètres mesurant l'énergie dans 6 bandes de fréquences empiriquement définies. La classification est effectuée par des HMM, ou par des SVM. Deux stratégies sont évaluées pour traiter le cas des frappes combinées : la première consiste en l'apprentissage d'un classifieur binaire par instrument, détectant sa présence ou son absence ; la seconde consiste à apprendre un seul classifieur dont chacune des classes est une combinaison possible de frappes. De manière à mieux appréhender la diversité des signaux de batterie rencontrés dans les boucles, 4 modèles différents sont appris sur 4 sous-ensembles de la base d'apprentissage (boucles jouées sur une batterie acoustique, boucles jouées sur une batterie acoustique avec réverbération et traitements, boucles jouées sur une batterie électronique, et boucles de Hip-Hop enregistrées à partir de disques vinyles). La reconnaissance est effectuée par les 4 classifieurs, et le classifieur donnant le meilleur score de vraisemblance est retenu. Ce processus de classification effectue ainsi indirectement une reconnaissance du type de batterie utilisée dans la boucle, avec une précision de l'ordre de 70%. Ces travaux ont été poursuivis dans [GR05e] pour étendre aux SVM l'emploi de modèles contextuels propres aux HMM.

Généralisation aux enregistrements polyphoniques L'application de l'approche SegRec aux enregistrements polyphoniques est plus récente et ses résultats plus mitigés. Le problème est en effet le suivant : les attributs ne seront plus extraits sur un signal de batterie seul, mais sur un signal de batterie bruité – le bruit provenant des autres instruments. Notons que les caractéristiques de ce bruit diffèrent d'un enregistrement à l'autre (une caisse claire peut être noyée dans un mélange contrebasse/saxophone dans un morceau, ou dans un mélange guitare électrique saturée/basse dans un autre), mais varient aussi au sein d'un enregistrement (une caisse claire peut être jouée en même temps qu'une note de contrebasse, et, quelques pulsations plus loin, en solo). Comment effectuer la classification en tenant compte de ce bruit, qui semble faire preuve de tant de variabilité ?

Une première voie, qu'on pourrait qualifier de pragmatique et d'optimiste, consiste à ignorer le problème du bruit : si l'ensemble d'apprentissage est suffisamment varié, et si l'algorithme de classification a un bon pouvoir de généralisation, il est raisonnable de croire que la classification de signaux bruités sera possible. C'est l'approche retenue par Steelant, Tanghe, Degroeve et al. dans leurs travaux [STD+05; TDB05] : leur algorithme de détection, basé sur 72 attributs classiques, utilise des SVM. Certains des paramètres intervenant dans le calcul des attributs ont été optimisés par recuit similé [DTB+05] pour garantir des performances maximales.

Une seconde voie se base sur l'observation suivante : la classification ne serait-elle pas plus facile si le classifieur avait été appris sur des signaux bruités identiquement aux signaux à reconnaître ? Sandvold et al. [SGH04] proposent un schéma de classification adaptatif. La classification est d'abord effectuée sur l'ensemble de la séquence à transcrire, par un classifieur générique – appris sur une large gamme de signaux. Un sous-ensemble des frappes reconnues est ensuite sélectionné, les frappes sélectionnées étant celles pour lesquelles la classification est la plus fiable. Un classifieur "local" est appris à partir de ce sous-ensemble. Ce classifieur va ainsi apprendre les caractéristiques spécifiques du bruit, et du timbre de la batterie employée dans la séquence. Le classifieur local est enfin appliqué à l'intégralité de la séquence. Sandvold et al. rapportent dans [SGH04] des gains de performance substantiels. Cependant, dans leur étude, la sélection des frappes sur lesquelles doit être appris le modèle local est effectuée manuellement. Sandvold et al. suggèrent qu'un score de vraisemblance pourrait être utilisé comme mesure de fiabilité, et permettre d'effectuer cette sélection automatiquement. Nous avons évalué cette solution [GR05c] et les résultats se sont montrés décevants. En fait, il s'est avéré que les frappes pour lesquelles le score de vraisemblance est le plus grand sont celles sur lesquelles l'influence du bruit est la plus faible – typiquement les frappes jouées dans les soli de batterie, ou jouées sur des temps où la basse ne joue pas. Le classifieur local est ainsi incapable d'apprendre les caractéristiques du bruit.

La dernière voie est celle que nous présentons dans cette thèse : elle consiste à pré-traiter les

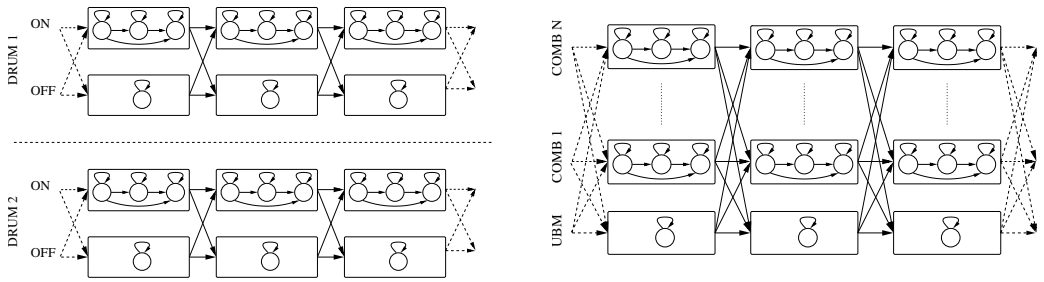


FIG. 2.3 – Topologies de HMM pour la reconnaissance et segmentation simultanée de signaux de batterie, d'après Paulus [Pau06]

signaux à analyser par diverses méthodes d'accentuation de la piste de batterie. Nous nous affranchissons ainsi (dans une certaine limite) du bruit introduit par les autres instruments.

Segmentation et reconnaissance simultanées Les systèmes de reconnaissance de la parole basés sur des HMM ne cherchent pas à segmenter le signal en phonèmes. Au contraire, la segmentation peut être vue comme un sous-produit du processus de reconnaissance. Serait-il possible de faire la même chose pour la transcription de signaux percussifs ?

Nous avons étudié dans [Gil03] l'application directe de techniques de reconnaissance de la parole aux signaux de Tabla. Le signal à transcrire est découpé en trames longues de 46 ms, sur lesquels sont calculés les coefficients cepstraux en échelle de Mel – *Mel Frequency Cepstrum Coefficients* (MFCC). À chaque paire de frappes à reconnaître (par analogie avec les modèles de diphones) est associé un modèle gauche-droit à 3 états (décroissance de la frappe précédente ou silence, attaque, décroissance), la distribution des paramètres acoustiques étant modélisée par un mélange de 4 gaussiennes. Les scores de reconnaissance obtenus avec cette méthode sont inférieurs à ceux présentés dans [GR03] – dans le cas du Tabla, les signaux sont suffisamment impulsionnels pour rendre la segmentation par détection d'onset robuste et préférable.

L'application de cette approche à la batterie a été réalisée par Paulus dans [Pau06]. Les attributs considérés sont variés : MFCC, dérivées des MFCC, moments spectraux, puissances et rapports de puissance en sortie d'un banc de filtre en bandes d'octave. Deux topologies sont proposées pour le HMM : une topologie employant N HMM en parallèle, chaque HMM comportant de 4 états – un état de silence et 3 états associés à un instrument de la batterie ; ou bien une topologie employant un seul HMM, comportant $1 + 3 \times 2^{N-1}$ états – un état de silence et 2^{N-1} groupes de 3 états associés à chaque combinaison d'instruments de la batterie (figure 2.3). Les résultats s'avèrent rarement meilleurs que ceux obtenus avec des méthodes plus classiques.

Le clustering comme alternative à la classification supervisée Précédemment, nous avons souligné la difficulté d'apprendre des classifieurs généraux capable de modéliser à la fois la diversité des timbres de chaque instrument de la batterie, et les différents bruits additifs susceptibles d'être présents dans des enregistrements musicaux polyphoniques. Pourrait-on éviter ce problème en se passant de classifieurs supervisés ?

Cette question est abordée par Gouyon et al. [GHC02], qui suggère l'emploi de méthodes de clustering (k-moyennes, clustering agglomératif) à partir des vecteurs d'attributs extraits sur chaque segment. Cette procédure produit alors une transcription partielle, dans laquelle les événements détectés sont étiquetés par des indices de clusters, et non par les instruments de la batterie correspondant. La tâche d'interprétation consistant à associer à chaque cluster l'instrument ou la combinaison d'instruments lui correspondant incombe à l'utilisateur – rendant cette solution réalisable uniquement dans des contextes où l'intervention d'un opérateur humain est possible.

Paulus et Klapuri suivent une approche similaire [PK03b], mais proposent une méthode pour associer automatiquement à chaque classe l'instrument correspondant : parmi toutes les associations possibles, doit être choisie celle qui est la plus probable selon un modèle de séquence rythmique. Par exemple, si à l'issue du clustering, la séquence de batterie est transcrite en :

$$C_1, C_2, C_3, C_2, C_1, C_2, C_3, C_3 \tag{2.1}$$

Où C_i sont les clusters obtenus, l'association $C_1 \rightarrow$ grosse caisse, $C_2 \rightarrow$ hi-hat, $C_3 \rightarrow$ caisse claire est la plus probable, et permet donc de déduire une transcription.

La tâche d'interprétation peut également être effectuée selon des critères acoustiques. Ravelli et al. proposent [RBS06] d'extraire, par la méthode des k-moyennes, 3 clusters à partir des frappes détectées. Le contenu spectral du centroïde de chaque cluster est considéré, et permet d'associer à chaque cluster une des 3 classes suivantes : bas (grosse caisse), medium (caisse claire, clap, rim shot, cross sticks), et haut (hi-hat, cymbale).

L'efficacité et la simplicité apparente de ces méthodes de clustering ne doit pas faire oublier leurs défauts. Tout d'abord, elles ne produisent que des descriptions extrêmement simplifiées, basées sur des taxonomies limitées à deux ou trois classes – insuffisantes pour certaines applications de transcription musicale. Ensuite, elles ne s'appliquent malheureusement pas au problème de la transcription polyphonique. En effet, dans un enregistrement polyphonique, un même instrument de la batterie est susceptible d'être joué superposé à des instruments différents. Ainsi, une classification non supervisée risque de placer différentes frappes d'un même instrument dans des groupes différents. À notre connaissance, aucune étude n'a été réalisée sur l'emploi de méthodes de clustering pour la transcription de batterie sur des signaux polyphoniques, et il est raisonnable de croire qu'une telle entreprise serait vouée à l'échec.

2.2.2 MatAda : Mettre en correspondance et adapter

Une deuxième famille de solutions proposées au problème de la transcription des signaux percussifs consiste à définir pour chaque instrument à identifier un modèle[5], et à rechercher les occurrences de ce modèle dans le signal à transcrire.

Dans [GM94], Goto et Muraoka proposent d'utiliser comme modèles les spectrogrammes $|\hat{X}_i(m,k)|$ de chacun des instruments à détecter, où $m \in \{1, \ldots, M\}$ est un indice de trame et $k \in \{1, \ldots, K\}$ un indice de bande de fréquence. Une mesure de distance est ensuite définie pour permettre la comparaison de ce modèle à une portion donnée du spectrogramme $|\hat{X}(n+m,k)|$ du signal à transcrire, produisant pour chaque instrument la fonction :

$$s_i(n) = \sqrt{\sum_{m=1}^{M} \sum_{k=1}^{K} \left(|\hat{X}_i(m,k)| - |\hat{X}(n+m,k)| \right)^2} \tag{2.2}$$

dans la cas où une distance euclidienne est utilisée. Les minima locaux de $s_i(n)$ en dessous d'un certain seuil traduisent une occurrence de l'instrument i à l'instant n. Cette méthode est appliquée avec succès à la transcription de soli de batterie. Notons sa complexité prohibitive en $\mathcal{O}(MKN)$, où $N + M$ est le nombre de trames du signal à transcrire.

Sillänpaa et al. [SKSV00] apportent plusieurs raffinements à cette méthode. Tout d'abord, pour contourner le coût prohibitif de la comparaison du modèle à toutes les positions possibles $n \in \{1, \ldots, N\}$, les modèles ne sont comparés qu'aux instants n correspondant à des onsets. Ensuite, plutôt que de considérer le spectogramme $|\hat{X}(m,k)|$ en échelle temporelles et fréquentielles linéaires, Sillänpaa propose d'appliquer un groupement des fréquences en bandes logarithmiquement espacées (correspondant à l'échelle de Bark), et une distorsion similaire de l'échelle temporelle. Enfin, la distance proposée pour la comparaison est pondérée :

[5] Le terme modèle doit être vu ici comme un synonyme de *prototype* ou *gabarit* (*template* en anglais) – il ne s'agit pas de modèle au sens statistique du terme.

$$s_i(n) = \sqrt{\sum_{m=1}^{M}\sum_{k=1}^{K} |\tilde{X}_i(m,k)| \left(|\tilde{X}_i(m,k)| - |\tilde{X}(n+m,k)|\right)^2} \tag{2.3}$$

Où $\tilde{X}(n+m,k)$ désigne le spectrogramme en échelles de temps et de fréquence non-linéaires.

Un modèle peut également être défini plus simplement dans le domaine temporel. Dans ce cas, la détection est effectuée en recherchant les maxima de la corrélation croisée entre le signal à transcrire et les modèles – cette opération pouvant aussi être vue comme un filtrage du signal à transcrire par le filtre adapté associé à chaque exemple. Cette approche est suivie par Jørgensen dans [Jør02] et utilisée sur des soli. Elle demande cependant que le modèle utilisé pour la détection soit produit par le même instrument que celui utilisé dans le signal à analyser. Zils et al. [ZPDG02] proposent une extension au cas polyphonique. Dans un premier temps, la détection est effectuée avec des modèles extrêmement génériques, correspondant en fait à des réponses impulsionnelles de filtres passe-bas (pour la grosse caisse) et passe-bande (pour la caisse claire). Les instances correspondant aux maxima locaux de la corrélation croisée entre le signal à analyser et les modèles sont évaluées selon :

- Leur proximité à un onset.
- La valeur de ce maxima local.
- La valeur moyenne de la corrélation croisée au voisinage du maxima local.

Les instances les plus fiables sont ensuite moyennées pour former un nouveau modèle, cette fois-ci adapté au timbre de l'instrument percussif utilisé dans le signal à traiter. Lors de l'addition des instances détectées pour former un nouveau modèle (phase d'adaptation), il est suggéré dans [ZPDG02] de décaler dans le temps chacune des instances sommées pour maximiser leur corrélation, de manière à synchroniser leurs phases. Nos expériences suggèrent que l'intérêt de cette étape est discutable : ajouter les instances détectées de façon désynchronisée est un moyen efficace d'annuler les contributions d'instruments non-percussifs (par exemple la basse), tandis que l'addition avec resynchronisation accentue ces contributions – la figure 2.4 présente un exemple de cette situation.

Les performances limitées de cette méthode (moins de 50% des transcriptions obtenues sont considérées satisfaisantes) s'expliquent surtout par la faible robustesse du modèle temporel – l'idée d'adaptation du modèle reste par contre valide.

La combinaison de ce principe d'adaptation avec des modèles temps-fréquence (spectrogrammes) a été réalisée par Yoshii et al. [YGO04a; YGO04b]. Après une étape de détection utilisant des modèles génériques (appris en moyennant les spectrogrammes de différents signaux de chacun des instruments considérés), une fraction (10%) des instances détectées les plus proches du modèle sont considérées. Leurs spectrogrammes sont combinés – la médiane est utilisée pour effectuer cette combinaison. La détection est réitérée en utilisant cette fois-ci les modèles adaptés. Une des contributions originales des travaux de Yoshii et al. est la définition d'une distance spectrale autorisant des déformations locales du modèle. Les performances rapportées pour la détection de grosse caisse et caisse claire sont très satisfaisantes.

2.2.3 SepDet : Séparer et détecter

Une dernière famille de solutions consiste à utiliser des méthodes de séparation de sources pour extraire un ensemble de signaux où chaque instrument de la batterie à transcrire est joué isolément. Les méthodes aveugles n'utilisent aucun a priori quant aux propriétés spectrales des signaux à séparer – d'autres méthodes supposent que le profil spectral de la source à extraire est connu.

2.2.3.1 Séparation aveugle

L'application directe de méthodes de séparation de sources comme l'analyse en composantes indépendantes – *Independent Component Analysis* (ICA) [HO00] n'est pas possible : ces méthodes ne permettent d'extraire N sources que de $M \geq N$ signaux – leur application à la transcription de

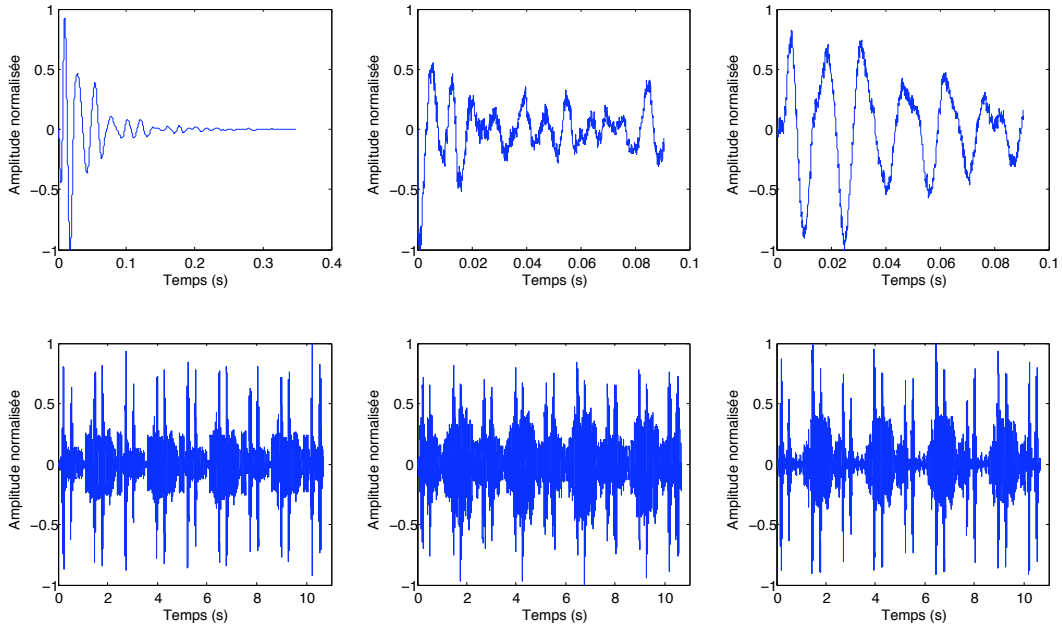


FIG. 2.4 – Détection de grosse caisse par filtrage adapté : Modèle initial (générique), modèle adapté obtenu par sommation des instances détectées, modèle adapté obtenu par sommation et resynchronisation des instances détectées ; Sortie du filtre adapté dans chacun des cas. Signal : Beats International – *Dub be good to me*

signaux percussifs se limite donc à la situation rare où seulement deux classes d'instruments sont utilisées dans un enregistrement stéréophonique – situation étudiée par Riskedal [Ris02].

Une voie plus prometteuse applicable à la séparation de sources avec un seul capteur a été proposée par Casey et Westner : l'analyse en sous-espaces indépendants – *Independent Subspace Analysis* (ISA) [CW00]. Elle se base sur l'hypothèse suivante : une source sonore peut être décrite entièrement par un profil spectral $\mathbf{F_i}$ (représenté par un vecteur de taille $K \times 1$) et par une enveloppe temporelle $\mathbf{T_i}$ (représentée par un vecteur de taille $M \times 1$). Dans ce cas, le module de la TFCT de cette source (représenté dans la matrice $\mathbf{X_i}$ de taille $K \times M$) peut s'écrire comme :

$$\mathbf{X_i} = \mathbf{F_i}\mathbf{T_i}^T \tag{2.4}$$

Si l'on suppose que les sources ont des supports fréquentiels ou temporels disjoints, le module de la TFCT de la somme de N sources peut s'écrire sous la forme :

$$\mathbf{X} = \sum_{i=1}^{N} \mathbf{F}_i\mathbf{T}_i^T = \mathbf{F}\mathbf{T}^T \tag{2.5}$$

Où $\mathbf{F} = [\mathbf{F}_1 \dots \mathbf{F}_N]$ et $\mathbf{T} = [\mathbf{T}_1 \dots \mathbf{T}_N]$. L'analyse en sous-espaces indépendants vise, à partir d'une observation de $\mathbf{X}$, à extraire des composantes $\mathbf{T}_i$ et $\mathbf{F}_i$. Tout d'abord, une analyse en composantes principales – *Principal Component Analysis* (PCA) est appliquée à la matrice $\mathbf{X}$, par le biais d'une décomposition en valeurs singulières – produisant ainsi une approximation de $\mathbf{X}$ sous forme de N produits impliquant les N valeurs singulières principales :

$$\mathbf{X} \overset{PCA}{=} \mathbf{USV}^T \tag{2.6}$$

$$\approx \sum_{i=1}^{N} \mathbf{U_i}(s_i \mathbf{V}_i^T) \tag{2.7}$$

Par identification, cette décomposition fournit déjà une famille de profils spectraux $(\mathbf{U}_i)$ et d'enveloppes $(s_i \mathbf{V}_i^T)$. Notons qu'une variante de cette méthode utilisant une autre méthode de réduction de dimensionnalité (Local Linear Embedding) à la place de la PCA est présentée dans [FL03]. L'étape suivante vise à rendre ces profils spectraux ou ces enveloppes indépendants, en effectuant une ICA des N profils spectraux ou N enveloppes. Par exemple, l'application d'une ICA aux profils spectraux produit une matrice de démixage $\mathbf{W}$ et des profils spectraux indépendants :

$$\mathbf{F} \overset{ICA}{=} \mathbf{WU} \tag{2.8}$$

Les enveloppes spectrales correspondantes se déduisent par :

$$\mathbf{T} = \mathbf{F}^{\dagger}\mathbf{X} \tag{2.9}$$

Où $\mathbf{F}^{\dagger}$ désigne la pseudo-inverse $\mathbf{F}$.

L'application de cette méthode à la transcription de signaux percussifs semble directe : une ISA est appliquée au signal à transcrire, avec N égal au nombre d'instruments à transcrire. Les maxima locaux des enveloppes $\mathbf{T}_i$ permettent de détecter les instants auxquels chacune des sources est active. La procédure est illustrée dans la figure 2.5, dans des circonstances d'utilisation idéales : le signal est une boucle de batterie n'utilisant que trois instruments mixés également. Trois problèmes restent à résoudre pour appliquer l'ISA dans des conditions plus réalistes :

1. Comment gérer une situation fréquente où deux instruments joués toujours simultanément se retrouvent dans une même source (sous-séparation), tandis qu'un même instrument se retrouve extrait dans deux sources distinctes (sur-séparation) ?
2. Comment gérer le cas polyphonique, où les autres instruments accompagnant la batterie vont produire des composantes superflues ?
3. Comment identifier, parmi les sources extraites, celles correspondant à un instrument donné ?

Dans un contexte où un opérateur humain peut ajuster le nombre de sources extraites, et identifier chaque instrument parmi les sources extraites, ces problèmes ne sont pas gênants – par exemple, le système de séparation décrit par Orife [Ori01] est utilisé dans un tel contexte. Ces problèmes doivent cependant être résolus dans des applications de transcription automatique.

FitzGerald et al. [FCL02], ont étudié en détail l'application de l'ISA à la transcription de signaux percussifs, et proposent une solution aux deux premiers problèmes : effectuer l'analyse du signal sur deux sous-bandes, pour éviter les problèmes de sous- et sur- séparation, et assigner à chaque composante une catégorie d'instruments sur un critère de centroïde spectral extrait à partir des profils F_i.

Une autre solution proposée par Uhle et al. dans [UDS03] consiste à sur-séparer, et à classer ensuite les composantes extraites pour identifier à quel instrument elles sont associées. Les composantes associées au même instrument sont ensuite regroupées. Dans [UDS03], cette classification reste cependant sommaire, et consiste juste à séparer les composantes associées à des instruments percussifs aux composantes associées aux instruments non-percussifs – une classification complète des sources selon chacun des instruments de la batterie, ainsi qu'une procédure d'adaptation semblable à celle de Yoshii et al. est effectuée dans [UD04b].

Un des défauts de l'ISA est qu'elle fait appel à deux décompositions, la PCA et l'ICA produisant des matrices pouvant prendre des valeurs négatives. Ces valeurs négatives n'ont pas d'interprétation évidente, car les enveloppes $\mathbf{T}$, les profils spectraux $\mathbf{F}$, et le module de la TFCT $\mathbf{X}$ sont par définition positifs ou nuls.

Une approximation de la forme $\mathbf{X} = \sum_{i=1}^{N} \mathbf{F}_i \mathbf{T}_i^T$ sous contraintes $\mathbf{X} \geq 0$, $\mathbf{F} \geq 0$ et $\mathbf{T} \geq 0$ peut être obtenue par factorisation matricielle non-négative – *Nonnegative Matrix Factorizaion* (NMF)

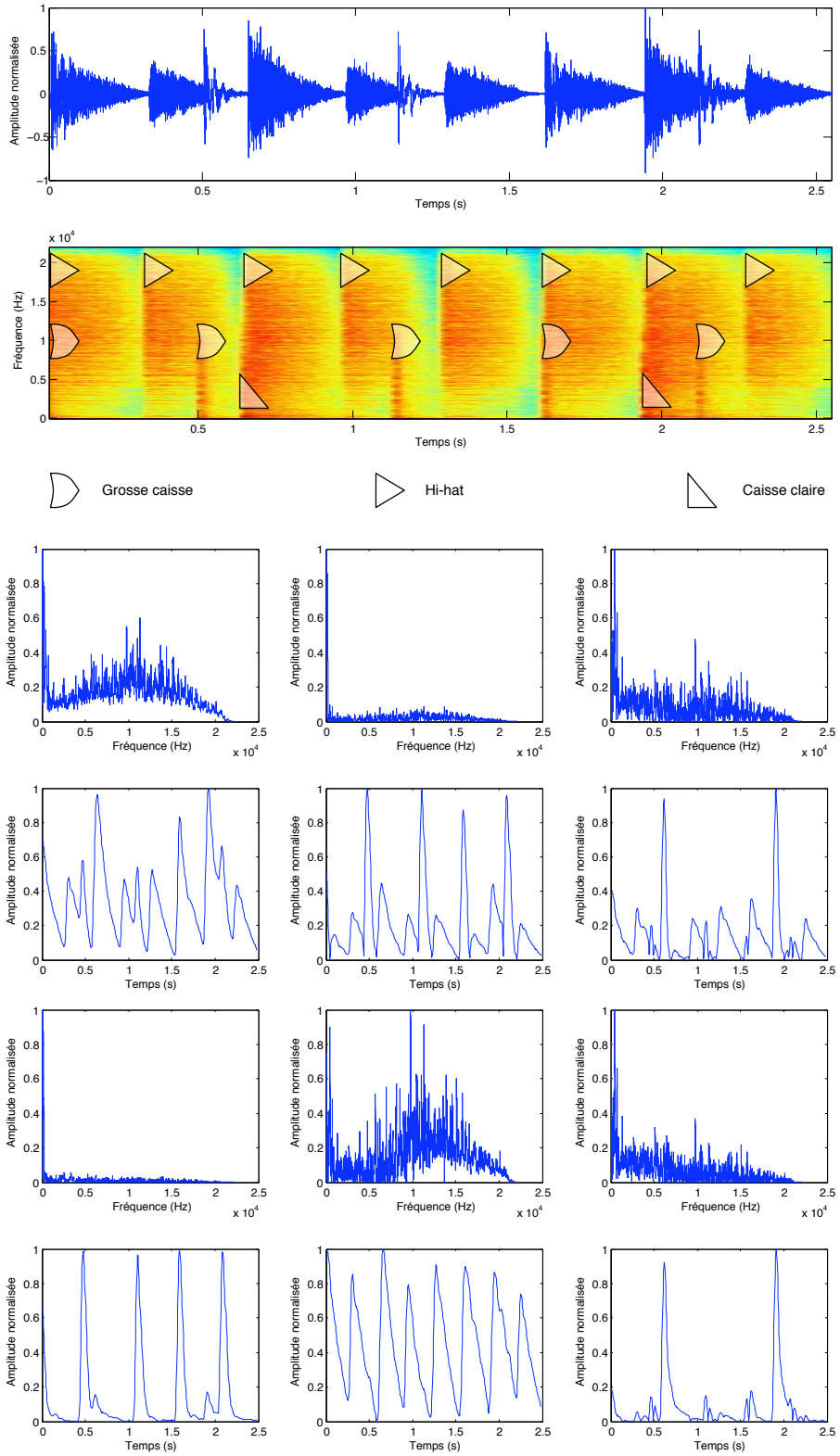


FIG. 2.5 – Représentations temporelles et temps/fréquence (annotée) d'une boucle de batterie ; Profils spectraux et enveloppes extraites par PCA ; Profils spectraux et enveloppes après ICA

[LS01]. Cette décomposition ne garantit pas l'indépendance des colonnes de $\mathbf{F}$ ou de $\mathbf{T}$. Cependant, une contrainte de parcimonie [AP04] peut être imposée lors de la décomposition – contrainte toute aussi pertinente musicalement puisque les profils spectraux extraits sont présumés avoir un support compact, et les sources ne sont pas supposées être actives en permanence. Dans les applications de transcription de signaux percussifs, la NMF s'utilise de la même manière que l'ISA, et pose les mêmes problèmes : compromis entre sur- et sous- séparation, et identification des sources. Un exemple de mise en oeuvre de la NMF pour l'analyse de signaux percussifs est donné dans [HV05] : Helén et Virtanen y utilisent des SVM pour discriminer les sources tonales et percussives parmi les composantes extraites.

2.2.3.2 Séparation avec information a priori

Le problème de l'identification des sources et de la sous- et sur- séparation ont conduit FitzGerald et al. à proposer une nouvelle méthode de séparation appelée l'analyse en sous-espaces appris – *Prior Subspace Analysis* (PSA). Cette approche requiert la définition, pour chaque instrument à transcrire, d'un profil spectral générique $\mathbf{F}_i$ – un tel profil peut par exemple être obtenu en moyennant les spectres de plusieurs instances de signaux de l'instrument considéré. L'étape de réduction de dimensionnalité est remplacée par une projection sur ce sous-espace. Les enveloppes obtenues sont ensuites rendues indépendantes par ICA, permettant d'estimer un nouvel ensemble de profils spectraux $\mathbf{F}'$, cette fois-ci spécifiques au signal considéré :

$$\mathbf{T} = \mathbf{F}^{\dagger}\mathbf{X} \text{ (projection)} \tag{2.10}$$

$$\mathbf{T}' \overset{ICA}{=} \mathbf{W}\mathbf{T} \text{ (séparation des enveloppes par ICA)} \tag{2.11}$$

$$\mathbf{F}' = \mathbf{X}\mathbf{T}'^{\dagger} \text{ (estimation de nouveaux profils spectraux)} \tag{2.12}$$

Cette méthode est présentée dans [FLC03a] et [FLC03b]. Notons qu'elle peut également être appliquée dans le cadre d'une décomposition par NMF. Dans ce cas, la NMF est utilisée pour trouver une approximation du spectrogramme observé $\mathbf{X}$ de la forme $\mathbf{X} = \mathbf{F}\mathbf{T}^T$ sous contraintes de non-négativité, où $\mathbf{F_i}$ est appris sur des signaux de référence de chaque instrument isolé. La détection des instants auxquels l'instrument i joue est alors possible en recherchant les pics dans $\mathbf{T}_i$. Cette solution est évaluée par Paulus et Virtanen dans [PV05] – il est rapporté que pour une tâche de transcription de soli de batterie, les performances de cette méthode sont semblables à celles offertes par une des méthodes de type SegRec évaluées [GR04].

2.2.4 Comparaison des méthodes

Jusqu'à récemment, il n'existait pas de base de données librement distribuée de signaux percussifs entièrement annotés – empêchant la comparaison des méthodes de transcription sur le même corpus. Le comparatif le plus complet à ce jour est issu de la campagne d'évaluation MIREX 2005 [MIR]. Des systèmes représentatifs des 3 approches sont évalués : Les systèmes de Tanghe et al. [TDB05], Gillet et Richard [GR05c], et Paulus [Pau06] sont dans la famille SegRec, le système de Yoshii et al. [YGO04b] suit l'approche MatAda, le système de Dittmar et Uhle [UD04b] suit l'approche SepDet. Les trois bases de données utilisées dans l'évaluation ont été fournies par respectivement Tanghe et al, Yoshii et al, et Dittmar et Uhle – les systèmes proposés par ces trois équipes sont donc avantagés, puisqu'entraînés sur les mêmes données que celles de l'évaluation. Les résultats sont donnés dans la figure 2.6. Trois systèmes représentatifs de chacune des familles présentées ci-dessus figurent alternativement à la première place lorsqu'ils sont évalués sur le jeu de données proposé par leurs concepteurs – résultat qui suggère la grande sensibilité de ces systèmes aux réglages de leurs paramètres.

Si ce critère de performance ne nous apprend rien, il est cependant possible de comparer les mérites de chaque méthode sur d'autres critères :

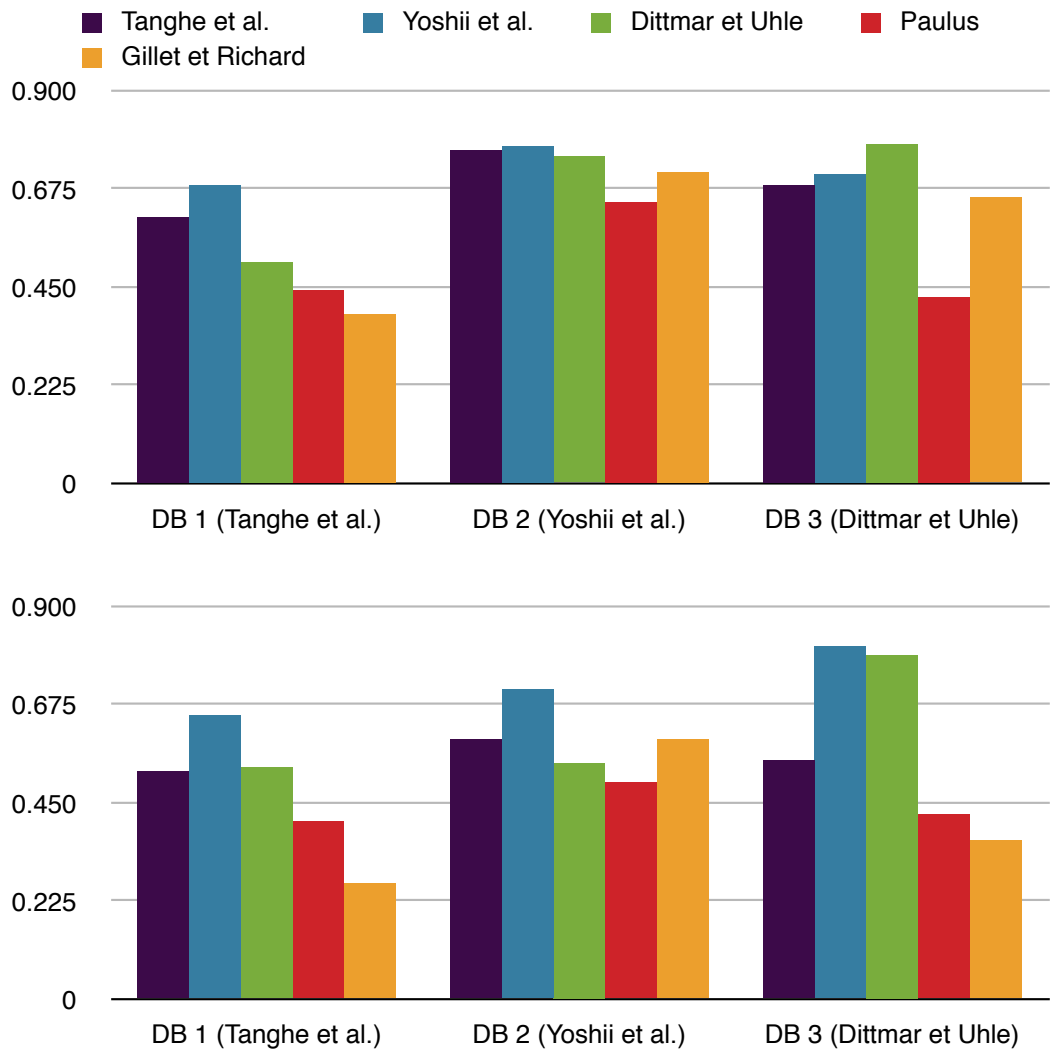


FIG. 2.6 – Résultats de la campagne d'évaluation MIREX 2005 des algorithmes de transcription de batterie. Détection de frappes de grosse caisse (F-mesure donnée en haut), et détection de frappes de caisse claire (F-mesure donnée en bas)

Critère de causalité Un inconvénient des systèmes de type SepDet ou MatAda est leur non-causalité – l'intégralité du signal doit être connue pour permettre la PSA ou l'adaptation du modèle. Dans les applications d'interaction musicien-machine, seule l'approche SegRec peut être utilisée, avec une latence modérée de l'ordre de 100 ms (latence de l'implémentation de Tanghe et al.).

Critère de robustesse à la diversité entre signaux Les méthodes SepDet et MatAda ne permettent de définir qu'un seul profil spectral ou modèle par classe d'instruments à reconnaître. Ce modèle doit ainsi être le plus générique possible, et ne peut donc pas représenter la diversité des timbres de l'instrument considéré (par exemple, ce modèle ne peut représenter à la fois la caisse claire jouée aux balais et aux baguettes). Les méthodes d'apprentissage statistique utilisées par les approches de type SegRec peuvent apprendre cette diversité.

Critère de robustesse à la diversité au sein d'un même signal Les procédures d'adaptation et d'extraction du profil spectral des approches SepDet et MatAda supposent que le timbre de toutes les frappes d'un instrument sont similaires sur la durée du signal à traiter. Cette hypothèse n'est pas valide dans le cas où le batteur alterne entre plusieurs modes de jeu (par exemple, couplet joué en cross sticks, refrain joué en frappes normales), ou dans la situation où des effets sont appliqués à la piste de batterie (modulation de la fréquence de coupure d'un filtre passe-bas résonnant dans les musiques électroniques par exemple). Plus couramment, des frappes douces (par exemple, des *ghost notes*) peuvent sonner très différemment de frappes fortes. Les méthodes de type SegRec, lorsqu'elles effectuent une classification supervisée, permettent de traiter cette variabilité.

Critère d'exploitation de l'information disponible En contrepartie, les méthodes de type SegRec sont incapables de tirer parti de la similarité de timbre entre toutes les frappes au sein d'un morceau, lorsque cette similarité est forte (par exemple dans les morceaux utilisant des batteries synthétiques ou des boucles).

2.3 Utilisation des connaissances musicales pour la transcription

La plupart des méthodes de transcription présentées ici analysent exclusivement l'information présente dans le signal audio. Une source d'information complémentaire pour guider la transcription consiste à considérer des connaissances musicales sur la structure ou les règles de composition des motifs rythmiques à transcrire.

Sillänpaa et al. proposent [SKSV00] de prendre en compte deux types de connaissances musicales : les fréquences d'utilisation de chacun des instruments de la batterie dans les motifs rythmiques, et le caractère périodique de la partie jouée par chacun des instruments de la batterie (il existe pour chaque instrument une périodicité τ tel que si l'instrument est joué à t, il sera joué également à $t + \tau$). Nous mettons en oeuvre cette méthode dans [GR05c], en agrégeant les probabilités fournies par le modèle acoustique à t, $t - \tau$ et $t + \tau$, où τ est la durée d'une mesure, pour effectuer la classification d'une frappe jouée à l'instant t. Un critère similaire de périodicité est utilisé par Yoshii et al. [YGK+06] : l'autocorrélation de la somme des fonctions de détection de chacun des instruments est ici utilisée pour extraire une périodicité τ ; la décision de détecter une frappe à l'instant t prend en compte les résultats des détections aux instants $\{t + k\tau\}, k \in \{-2, -1, 1, 2\}$.

Les deux types de connaissances musicales proposées par Sillänpaa peuvent être unifiées dans le cadre du modèle de N-grammes périodiques introduit par Paulus et Klapuri [PK03a]. Ce modèle définit la probabilité d'apparition d'un symbole rythmique en fonction des symboles rythmiques joués aux mesures précédentes. Paulus et Klapuri considèrent différents contextes d'observation, et deux types de modèles qui consistent ou bien à modéliser individuellement la partie jouée par chaque instrument (modèle de symboles) ou bien à modéliser une seule séquence de symboles combinés (modèle de mots). Les gains de performance offerts par les modèles à N-grammes sont modérés par rapport au simple emploi de probabilités a priori pour chaque symbole. Mais dans tous les

cas, les gains de performance sont substantiels par rapport au modèle acoustique seul. Une application plus convaincante de ces modèles est proposée dans [PK03b], où ils sont utilisés pour trouver l'association la plus vraisemblable entre clusters et classes d'instruments percussifs. Nous avons évalué l'emploi de modèles de N-grammes classiques pour les applications de transcription de Tabla, où ils s'avèrent nécessaires pour modéliser certaines propriétés du système de notation musicale sous-jacent [GR03], et pour la transcription de boucles de batterie dans [GR04]. Nous présentons ultérieurement dans ce document (section 4.5) une généralisation de ces méthodes.

2.4 Applications

Nous terminons cet état de l'art par quelques applications intéressantes des systèmes de transcription de signaux de batterie.

Une première application consiste en l'indexation de bases de données de signaux rythmiques, afin de permettre la recherche par le contenu. Nous présentons dans [GR05b] et [GR05e] un système complet pour la gestion de collections de boucles de batterie. Une base de données stocke les transcriptions de chacune des boucles de la collection. Les requêtes peuvent être formulées en utilisant des onomatopées (requête par *beatboxing*) – auquel cas un système de reconnaissance vocale indépendant du locuteur en assure la transcription – ou jouées sur un clavier MIDI. Nous proposons un modèle statistique d'interprétation des rythmes, permettant de calculer un score de similarité entre une requête et chacune des boucles contenues dans la base. Un système similaire est décrit par Nakano et al. [NOGH04] – le critère utilisé pour mesurer la similarité entre documents et requête est ici plus simple, et n'est en particulier pas robuste à l'ajout ou à la suppression d'éléments. De tels systèmes peuvent être améliorés par l'emploi de meilleurs modules de reconnaissance de rythmes interprétés à la voix – tâche pour laquelle sont proposées à la fois des méthodes issues de la reconnaissance vocale [NOGH04; GR05b] ou s'inspirant de la transcription de signaux de batterie [Haz05].

Tzanetakis et Cook ont montré [TC02] l'importance des caractéristiques rythmiques pour l'identification du genre musical – bien que les attributs rythmiques utilisés dans leur étude sont simplement de nature métrique. Uhle et Dittmar utilisent dans [UD04a] le résultat d'une transcription de la piste de batterie pour l'identification du genre. Dans [EA04], Ellis et Arroyo proposent de projeter une représentation symbolique d'un motif rythmique de batterie sur une base de "rythmes propres" (*Eigenrhythms*). Les coefficients de cette projection pourraient être utilisés comme attributs pour la classification de rythmes, après une étape de transcription.

La transcription extraite peut faciliter la manipulation ou le remixage des signaux de batterie. Ravelli et al. proposent dans [RBS07] un système de *morphing* de boucles de batterie, réorganisant des segments d'une boucle de batterie pour que sa transcription soit identique à celle d'une boucle de référence. Un tel système est bien plus flexible que les outils de *Drum replacement* [Dru03; Dig01] utilisés dans la production musicale contemporaine (en particulier pour le Metal) qui effectuent une détection d'onsets sur des signaux de batterie enregistrés en pistes séparées (une piste par instrument) afin de remplacer chaque frappe détectée par un échantillon tiré d'une table d'ondes.

Terminons enfin sur les liens très étroits entre les problèmes de transcription et de séparation de la piste de batterie. Les systèmes de type SepDet et MatAda extraient conjointement du signal à la fois des informations sur le timbre des instruments utilisés (modèles, profils spectraux) et sur les instants auxquels ils sont joués. Ainsi, les systèmes de transcriptions présentés dans [YGO05] et [FLC03a] permettent une resynthèse de la piste de batterie du signal original, en utilisant dans le premier cas le modèle adapté extrait pour chaque instrument, et dans le second cas en resynthétisant un signal dont le spectrogramme est $\mathbf{F}'\mathbf{T}'^T$ – produit des profils spectraux et enveloppes produites par la PSA. Nous reparlerons de cette application au chapitre 5.

CHAPITRE 3

Pré-traitements pour l'accentuation de la piste de batterie

Dans ce chapitre sont présentés deux traitements complémentaires permettant l'accentuation de la piste de batterie dans des signaux de musique polyphonique. Ces traitements peuvent être inclus dans un système de transcription de la batterie (comme étudié au chapitre suivant), ou peuvent être considérés comme des procédés élémentaires de séparation de sources dédiés à la batterie. Le premier traitement – décrit dans la section 3.3 – produit, à partir d'un signal stéréophonique, un signal monophonique dans lequel les instruments non-percussifs sont atténués. Le second traitement, introduit en 3.4 – tire parti du caractère non-harmonique et bruité des signaux percussifs, en estimant et soustrayant les composantes déterministes stables du signal à traiter. Au préalable, diverses observations justifiant ces deux méthodes sont données dans la section 3.1. Les deux méthodes nécessitent une décomposition du signal à traiter en signaux de sous-bande, discutée en 3.2.

3.1 Principe et motivations

3.1.1 Analyse d'enregistrements stéréophoniques

La plupart des systèmes d'analyse de la piste de batterie présentés au chapitre 2 ne considèrent que des enregistrements monophoniques (mono-canaux). Cependant, la majorité des enregistrements de musique populaire produits durant les dernières décennies sont stéréophoniques (bi-canaux). Classiquement, les canaux droite et gauche de tels enregistrements sont moyennés avant tout traitement – un traitement en apparence bénin puisqu'il préserve les propriétés de haut-niveau (rythme, tempo, genre) des signaux considérés. Toutefois, il serait certainement plus avantageux d'exploiter toute l'information contenue dans ces deux canaux.

Nous nous proposons ainsi, à partir de la paire de signaux observée de :

1. Séparer les sources mono-instrumentales dont elle se compose.
2. Sélectionner, parmi ces sources mono-instrumentales, celles associées à des instruments à percussion.

Nous insistons sur le fait que cette approche sélectionne *a posteriori* les sources percussives – l'étape de séparation n'utilise aucun modèle décrivant les sources à extraire.

3.1.2 Séparation harmonique / bruit

La plupart des sons produits par la batterie peuvent difficilement être décrits par un mélange de composantes sinusoïdales lentement modulées en amplitude ou en fréquence[1].

Cymbales Les cymbales peuvent être vues comme une surface rigide dont les bords peuvent vibrer librement [Hal01]. Des observations suggèrent plusieurs dizaines de modes de vibration [Ros01], tous excités simultanément au moment de la frappe : le nombre de partiels à considérer est très grand. De plus, des comportements chaotiques (bifurcations) dus à des phénomènes non-linéaires ont également été rapportés [CTT05]. Il en résulte que les nombreux partiels inharmoniques dont se compose un signal de cymbale sont fortement modulés et difficilement modélisables.

Grosse caisse et toms L'observation de signaux de grosse caisse (ou de toms) révèle qu'ils sont quasi-harmoniques. Cependant la variation de la tension de la peau au moment de la frappe modifie les modes de vibration. Il en résulte une augmentation rapide de la fréquence fondamentale perçue au début de la frappe, décroissant ensuite lentement.

Caisse claire La caisse claire sans timbre peut être modélisée de façon semblable à la grosse caisse, si ce n'est qu'il existe un couplage entre les modes des peaux supérieures et inférieures. Modéliser le comportement du timbre est plus difficile, car les transferts d'énergie entre la peau inférieure et le timbre sont non-linéaires (les deux sont parfois en contact, parfois non). La composante associée au timbre peut donc être considérée comme entièrement stochastique.

Baguettes Indépendamment de l'instrument frappé, le choc de la baguette (ou de la mailloche) sur la peau ou la cymbale produit une composante très courte et impulsive. Quant au frottement du balai sur la caisse claire (shuffle), il produit un signal clairement stochastique.

Par contraste, la partie stable des signaux d'instruments non-percussifs se modélise bien par un modèle sinusoïdal, popularisé par les travaux de Serra et Smith [SS90]. Un modèle plus récent tout aussi efficace, le modèle de sinusoïdes modulées exponentiellement – *Exponentially Damped Sinusoids* (EDS), est proposé par Badeau et al. dans [BBD02]. Notons que le modèle EDS ne décrit pas entièrement les signaux des instruments non-percussifs – les composantes transitoires de ces signaux (choc de la corde sur le marteau dans un signal de piano, souffle du flûtiste) sont tout aussi difficiles à modéliser que les signaux percussifs.

Ces observations suggèrent l'approche suivante pour la séparation des sources percussives et non-percussives dans un signal de musique : les paramètres du modèle EDS décrivant le mieux le signal considéré sont estimés ; Cette partie déterministe, expliquée par le modèle, est attribuée aux instruments non-percussifs. La partie stochastique, non expliquée par le modèle, est attribuée aux instruments percussifs. Cela suppose que :

1. Les composantes sinusoïdales stables présentes dans les signaux d'instruments à percussion peuvent être négligées. C'est évidemment le cas pour les cymbales et la composante de la caisse claire due au timbre. Pour les fûts, cette hypothèse reste vraie à condition que l'étape d'estimation des paramètres du modèle EDS ne soit pas robuste aux modulations de fréquence décrites.
2. Les composantes transitoires, non-harmoniques, des signaux non-percussifs peuvent être négligées. La méthode de séparation que nous venons de décrire extraira ainsi les bruits mécaniques, souffles ou frottements produits par les instruments non-percussifs. Cependant, dans les enregistrements de musique populaire, la place prédominante accordée à la batterie laisse supposer que ces composantes non voulues seront de faible puissance.

[1] Les premiers confrontés à cette difficulté ont été les constructeurs de synthétiseurs et de boîtes à rythmes – Quelques circuits typiques utilisés dans les synthétiseurs analogiques sont discutés et modélisés dans [Cla]. Presque tous ces modèles empiriques emploient des générateurs de bruit.

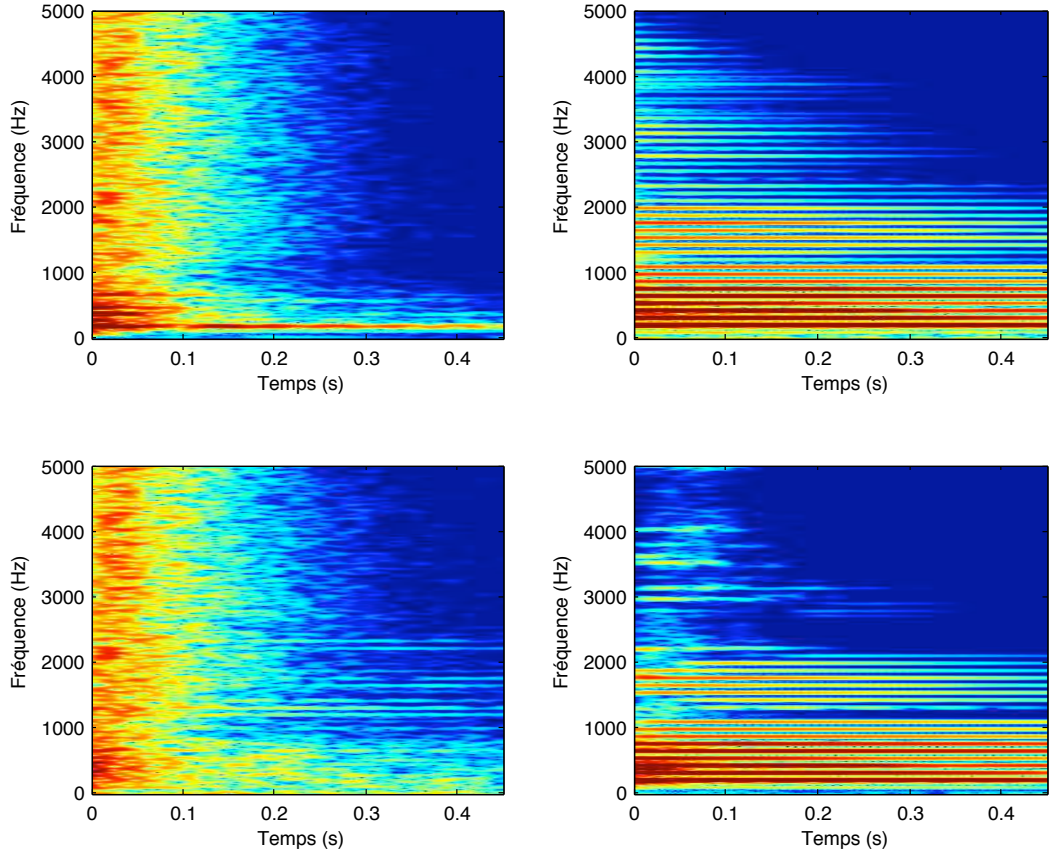


FIG. 3.1 – Spectrogrammes d'une frappe de caisse claire et d'une note de guitare (en haut) ; parties stochastiques et harmoniques de la somme de ces deux signaux (en bas)

Pour illustrer cette discussion, nous considérons la somme d'un signal de caisse claire et d'un signal de guitare. Une somme de 20 sinusoïdes modulées en amplitude est estimée à partir de ce mélange, définissant sa partie déterministe. Le résidu de modélisation forme la partie stochastique. Sont présentés dans la figure 3.1 les spectrogrammes des signaux originaux, et des composantes stochastiques et déterministes du mélange. La composante déterministe contient les harmoniques principales de la note de guitare, ainsi qu'une composante harmonique issue de la caisse claire. La composante stochastique provient presque exclusivement de la caisse claire. Elle contient aussi le pincement de la corde de la guitare, de faible puissance et très localisé dans le temps, et quelques harmoniques de la note de guitare qui n'ont pas été prises en compte par le modèle. Ne figure pas dans la composante stochastique la composante harmonique quasi-stable (modulée en fréquence) principale de la caisse claire.

Précisons enfin qu'une telle décomposition harmonique/bruit a déjà été utilisée par Alonso et al. [ARD07; Alo06] pour améliorer la détection de tempo dans des signaux de musique peu percussifs – par exemple pour des enregistrements de musique de chambre. En effet, pour de tels signaux, les indices les plus robustes permettant la détection des onsets sont tantôt les bruits mécaniques (marteau frappant la corde dans le cas du piano), tantôt les composantes sinusoïdales – la partie stochastique gênant au contraire la détection (cas d'un frottement d'archet).

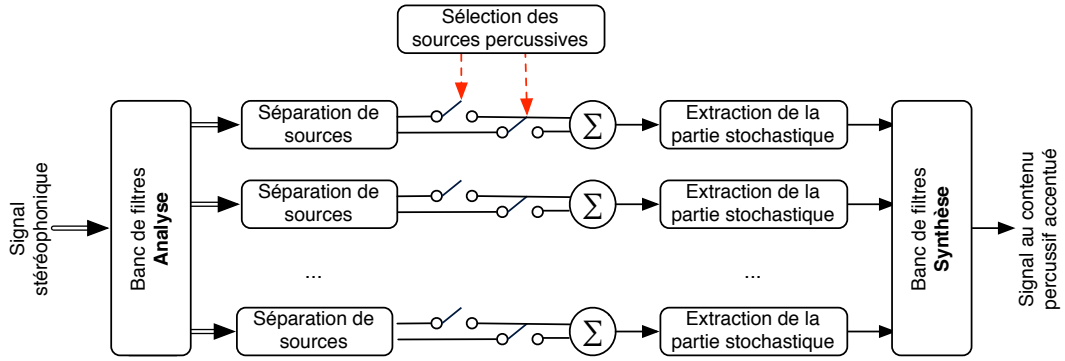


FIG. 3.2 – Architecture du système d'accentuation des instruments percussifs

3.1.3 Nécessité d'un traitement par bande

Les deux traitements que nous venons de présenter pourraient être effectués sur l'intégralité du signal à traiter. Il est cependant préférable de séparer le signal en différents signaux de sous-bande à l'aide d'un banc de filtres, et d'effectuer ces traitements sur chacun des signaux de sous-bande. Nos motivations sont les suivantes :

La distribution fréquentielle de l'énergie de chacun des instruments à percussion est différente Les centroïdes spectraux (voir annexe A.3) de la grosse caisse, de la caisse claire, et de la hi-hat sont respectivement de l'ordre de 150 Hz, 2.5 kHz, et 8 kHz. Il est ainsi possible de concevoir un banc de filtres tel que chaque instrument de la batterie soit prédominant dans chacune de ses sous-bandes. De façon similaire, dans un enregistrement musical multi-instrumental, chacun des instruments utilisés couvre une bande de fréquences qui lui est propre – une propriété accentuée à l'égalisation par l'ingénieur du son pour améliorer la "lisibilité" du mixage. En conséquence, si les sous-bandes sont suffisamment étroites, un nombre limité de sources seront prédominantes dans chacune des sous-bandes.

La séparation harmonique/bruit est plus aisée sur des signaux à bande étroite La méthode d'estimation de la partie harmonique que nous avons retenue et que nous présenterons dans la section 3.4 nécessite que le bruit présent dans les signaux à traiter soit blanc. Cette contrainte peut être satisfaite en traitant le signal par bandes, avec des bandes suffisamment étroites pour que la densité spectrale de puissance (d.s.p) du bruit dans chacune des bandes puisse être considérée comme uniforme. Par ailleurs, l'extraction de la partie harmonique nécessite de définir le nombre de sinusoïdes à estimer. Effectuer cette estimation par bande permet de n'avoir à extraire qu'un nombre restreint de sinusoïdes, et d'utiliser un ordre de modélisation différent dans chacune des bandes. En imposant un ordre de modélisation à chacune des sous-bandes il est ainsi possible de "structurer" le modèle estimé. Enfin, en décimant chacun des signaux de sous-bande, le coût en calculs de l'opération d'estimation de la partie harmonique est réduit. En effet, la complexité de cette opération est $\mathcal{O}(nr^2)$, où r est le nombre de composantes sinusoïdales à estimer et n le nombre d'échantillons considérés. L'apport d'un traitement par bande réduisant à la fois le nombre d'échantillons à traiter et le nombre de composantes à estimer est donc substantiel.

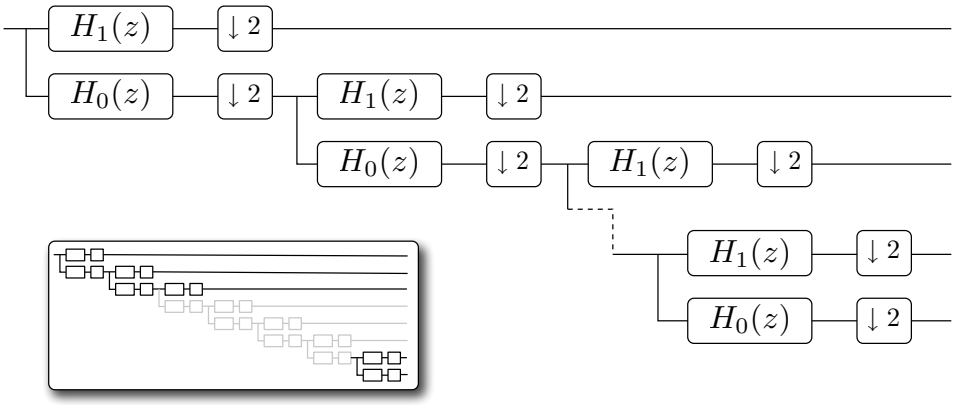


FIG. 3.3 – Banc de filtres en bandes d'octave

3.1.4 Architecture retenue

L'architecture retenue pour notre système de séparation est illustrée dans la figure 3.2. Des sources monophoniques sont d'abord extraites de chacune des paires de signaux stéréophoniques de sous-bande. Parmi ces sources, uniquement celles associées aux percussions sont retenues. La partie stochastique du signal obtenu à cette étape est extraite. Enfin, un signal pleine-bande est produit à partir des signaux de sous-bandes. Nous détaillons chacun des composants de ce système dans les sections suivantes.

3.2 Banc de filtres

Supposons d'abord qu'un banc de filtres uniforme soit ici utilisé. Pour séparer dans des voies différentes la grosse caisse et la caisse claire, dont 90% de l'énergie est concentrée respectivement dans les bandes $[78, 104]$ Hz et $[330, 8240]$ Hz[2], la largeur des bandes doit être de l'ordre de 100 Hz. Si l'on suppose que les signaux à traiter sont de qualité CD, cela impose l'utilisation de près de 200 bandes. Cette solution n'est pas réalisable pratiquement pour les raisons suivantes :

1. Les filtres devront être extrêmement sélectifs donc longs et coûteux en calculs.
2. L'ajustement du nombre de sinusoïdes extraites dans chacune des bandes est délicat. En effet, certaines de ces bandes ne contiendront vraisemblablement aucune sinusoïde.
3. Une résolution fréquentielle aussi fine n'est intéressante que pour les basses fréquences.

Ces problèmes peuvent être évités par l'emploi d'une analyse multi-résolution. Nous proposons ainsi l'emploi d'un banc de filtres en bandes d'octave, implémentant une transformée en ondelettes dyadique (figure 3.3). La largeur des bandes décroît avec leur fréquence centrale : ainsi cette décomposition permet de disposer d'une résolution fréquentielle suffisante dans les basses fréquences, même avec un nombre limité de bandes (8 bandes suffisent pour atteindre la résolution voulue). De plus, elle est adaptée à la distribution de l'énergie dans les signaux audio : la figure 3.4 donne la valeur relative de l'énergie mesurée dans chacune des sous-bandes d'un banc de filtres uniforme et d'un banc de filtres en bandes d'octave[3]. À chaque bande du banc de filtres en bandes d'octave correspond une fraction quasiment identique de l'énergie du signal original.

[2] Valeurs calculées sur l'ensemble des frappes isolées de la base ENST-drums.

[3] Valeurs mesurées sur le corpus `Music-54`, constitué des 54 extraits musicaux longs de 15 secondes référencés dans l'annexe D.1.

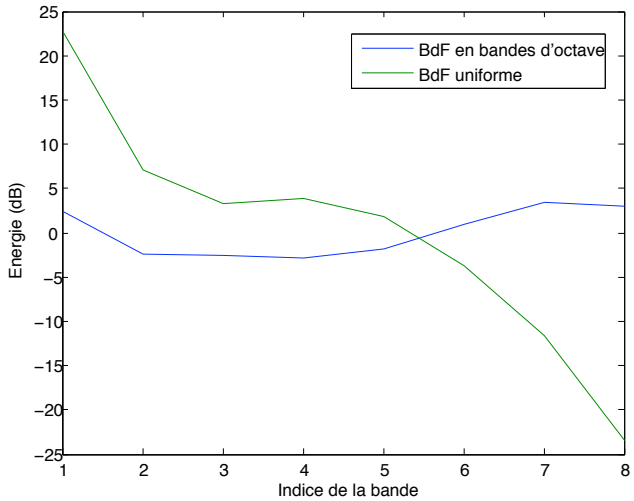


FIG. 3.4 – Distribution de l'énergie dans les sous-bandes

Chaque noeud du banc de filtre retenu consiste en un banc de filtres modulés en cosinus [Vai93], utilisant un filtre prototype de longueur $N = 128$. Les réponses des filtres sont données dans la figure 3.5. Le banc de filtres comporte $M = 8$ bandes, les limites des bandes correspondantes (pour des signaux échantillonnés à 44.1 kHz) étant listées dans la table 3.1.

Précisons qu'une alternative aux bancs de filtres uniformes est discutée par Badeau dans [Bad05] et Alonso dans [Alo06]. Elle consiste à utiliser un banc de filtres uniforme (par exemple des filtres modulés en cosinus), et à en regrouper les bandes adjacentes. Cette approche n'est cependant pas applicable ici, car elle n'offre pas une résolution suffisante dans les basses fréquences.

3.3 Séparation et sélection de sources à partir d'enregistrements stéréophoniques

Nous détaillons à présent l'étape de séparation de sources monophoniques à partir de signaux stéréophoniques, et de sélection des sources percussives. Nous commençons en 3.3.1.1 par présenter quelques résultats expérimentaux obtenus avec l'algorithme de discrimination d'Azimuth et Resynthèse – *Azimuth Discrimination and Resynthesis* (ADRes) proposé par Barry et al. [BLC04], et concluons quant à la nécessité d'une autre approche présentée dans 3.3.1.2. Nous explicitons ensuite la procédure de sélection des sources percussives en 3.3.2.

3.3.1 Séparation

3.3.1.1 Présentation critique de l'algorithme ADRes

Principe Différentes méthodes de séparation de sources à partir d'enregistrements stéréophoniques ont été proposées dans la littérature, fondées sur une variété d'hypothèses quant aux propriétés statistiques des sources à extraire, et quant à la nature de la fonction de mixage (liant les signaux observés sur les canaux droit et gauche aux signaux des sources monophoniques mixées). La méthode ADRes, proposée par Barry et al. [BLC04] ne repose que sur une hypothèse simple et réaliste : la paire de

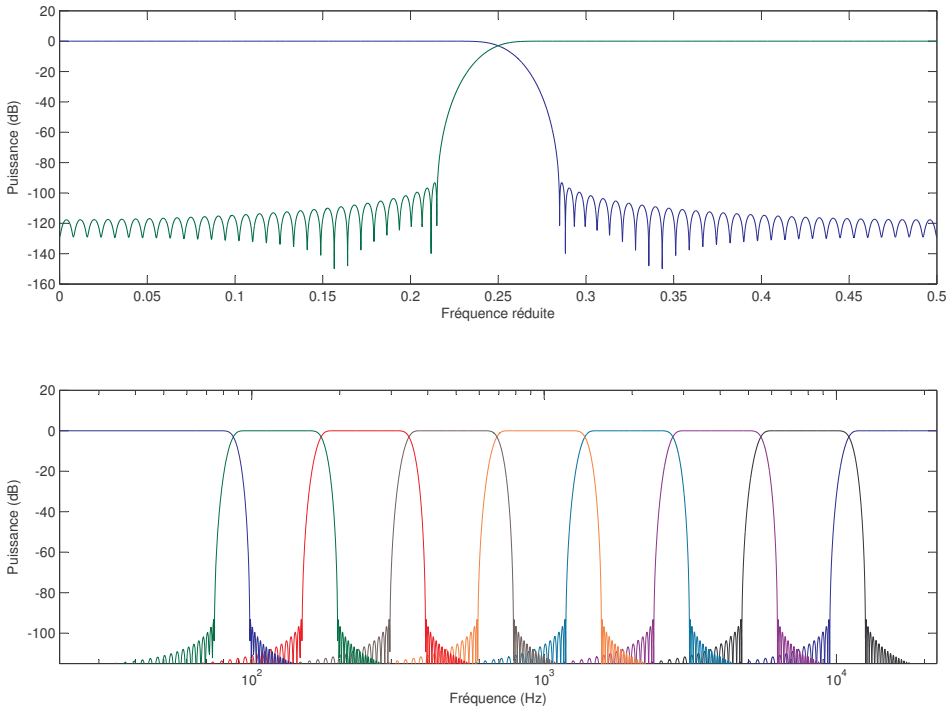


FIG. 3.5 – Réponse en fréquence des deux filtres utilisés à chaque noeud ; Réponse en fréquence du banc de filtres complet

Indice de la bande	1	2	3	4
Fréquences (Hz)	0–172	172–344	344–689	689–1378
Indice de la bande	5	6	7	8
Fréquences (kHz)	1.38–2.76	2.76–5.51	5.51–11.02	11.02–22.05

TAB. 3.1 – Limites des bandes de fréquence du banc de filtres en bandes d'octave

signaux considérée est produite par mixage *panoramique* ; et sur une approximation : les sources ont des représentations temps/fréquence à supports disjoints (une même approximation est faite dans la formulation de l'ISA). Le mixage *panoramique*, popularisé dans les années 60 en même temps que les premiers systèmes hi-fi stéréophoniques, consiste à enregistrer chaque source sonore $s_i(t)$ à l'aide d'un seul microphone, et à "doser" différemment chaque source dans les canaux droite et gauche en lui appliquant respectivement des gains γ_i et $1-\gamma_i$, $\gamma_i \in [0,1]$:

$$d(t) = \sum_{i=1}^{M} \gamma_i s_i(t) \tag{3.1}$$

$$g(t) = \sum_{i=1}^{M} (1-\gamma_i) s_i(t) \tag{3.2}$$

Dans ce cas, la contribution de la source s_i est annulée dans le signal $\Delta_\alpha(t) = \alpha d(t) - g(t)$ si

et seulement si $\alpha = \frac{1-\gamma_i}{\gamma_i}$. En particulier, si (t, f) est dans le support de $TFCT\{s_i\}$:

$$\frac{1-\gamma_i}{\gamma_i} = \arg\min_{\alpha} |TFCT\{\Delta_\alpha\}(t, f)| \tag{3.3}$$

Ainsi, pour une valeur de α donnée, tous les points (t, f) vérifiant $|TFCT\{\Delta_\alpha\}(t, f)| = 0$ sont associés à une même source. La discrimination d'Azimuth consiste à considérer une famille de valeurs $(\alpha_i)_{i\in\{1,\ldots,R\}}$, et à former pour chaque valeur de α_i une source $\hat{s}_i$ dont le module de la TFCT est :

$$|TFCT\{\hat{s}_i\}(t, f)| = \begin{cases} (1+\alpha_i)|TFCT\{d\}(t, f)| & \text{si } \alpha_i = \arg\min_\alpha |TFCT\{\Delta_\alpha\}(t, f)| \\ 0 & \text{sinon} \end{cases} \tag{3.4}$$

La reconstruction du signal $\hat{s}_i$ à partir de $|TFCT\{\hat{s}_i\}(t, f)|$ est possible par un processus itératif décrit dans [HHLO83]. La discrimination d'Azimuth fournit ainsi, pour une famille de réels[4] positifs (α_i), une famille de sources $\hat{s}_i$.

Une des difficultés rencontrées dans la mise en oeuvre de cet algorithme est le choix de l'ensemble des valeurs α à considérer. Un ensemble de valeurs trop proches les unes des autres résulte en une sur-séparation – une même source se retrouve dispersée sur plusieurs sources reconstruites $\hat{s}_i$. Un ensemble de valeurs trop distantes ne permet pas de s'approcher des valeurs $\frac{1-\gamma_i}{\gamma_i}$ annulant la source s_i. Barry et al. suggèrent deux solutions : utiliser leur méthode de façon interactive – dans ce cas, l'utilisateur explore lui même l'espace des valeurs α de manière à sélectionner la source voulue ; et sur-séparer, quitte à regrouper par la suite les sources correspondant à des valeurs α adjacentes.

Résultats expérimentaux Dans cette expérience, nous considérons 54 enregistrements musicaux commerciaux stéréophoniques de styles variés (corpus `Music54` décrit dans l'annexe D.1), de durées égales à 15 secondes. Chaque enregistrement est séparé à l'aide de la méthode ADRes, en utilisant $\alpha \in \{0, \frac{1}{8}, \ldots, \frac{7}{8}, 1\}$. 17 sources monophoniques sont ainsi produites. Parmi ces sources, celles contenant des instruments à percussion sont retenues pour former un signal monophonique. La table 3.2 et la figure 3.6 résument nos observations. La plupart des sources percussives sont placées au centre du champ stéréo (gain identique pour les canaux droit et gauche). Ainsi, en pratique, les sources supprimées seront celles localisées aux extrémités du champ stéréo. Dans 74% des signaux considérés, de telles sources étaient présentes et ont pu être supprimées. Malheureusement, il existe presque toujours (96% des cas), des sources mixées avec le même panoramique que les percussions. Ces sources ne peuvent pas dans ce cas être séparées.

D'autres difficultés ont été rencontrées :

1. La procédure d'association des points temps-fréquence aux sources est très sensible aux perturbations. En particulier, un même point fréquentiel peut être associé, au cours du temps, à deux sources différentes et "sauter" d'une source à l'autre. Cela se traduit par du bruit musical désagréable, et parfois impulsif (donc susceptible de perturber la sélection des sources). De tels phénomènes sont visibles dans l'exemple donné en figure 3.7 : les stries et tâches dans le spectrogramme de l'orgue correspondent à des fréquences dont l'affectation à une des sources est instable.
2. Comme toutes les méthodes basées sur la TFCT, ADRes produit des signaux dont les phases sont inexactes. En particulier, dans le cas où les traitements décrits dans cette section sont utilisés à des fins de remixage de la batterie dans un signal de musique, le signal extrait ne peut pas être soustrait ou superposé au signal original, car leurs phases ne correspondent pas.

Ces deux difficultés nous ont poussé à considérer une autre méthode, plus conservative – dans le sens où elle permet de préserver l'information de phase du signal original, et où le procédé de reconstruction des sources ne produit pas les discontinuités et artefacts observés.

[4]En permutant le rôle des canaux droite et gauche et en considérant $\frac{1}{\alpha_i}$ au lieu de α_i, la source extraite est la même – cela permet de traiter les cas où $\gamma_i = 0$.

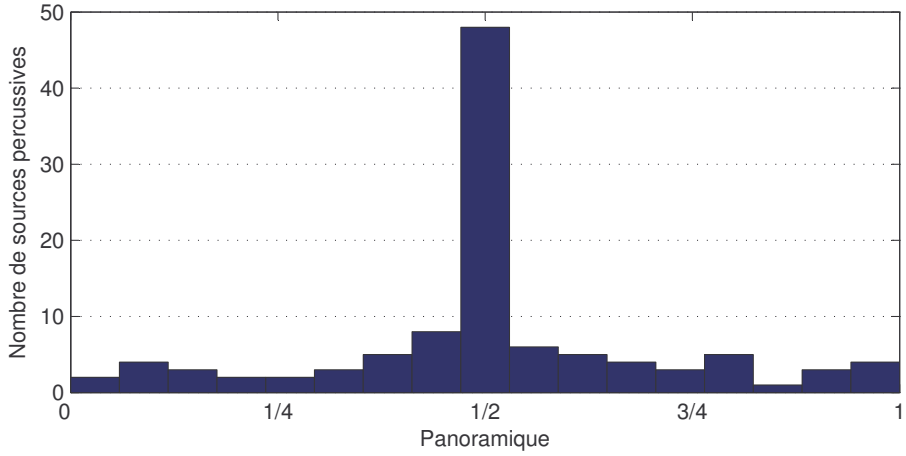


FIG. 3.6 – Panoramique des sources percussives

Nombre de sources non-percussives soustraites	
Aucune	26 %
Une	33 %
Deux ou plus	41 %
Nombre de sources non-percussives restantes	
Aucune	4 %
Une	17 %
Deux ou plus	79 %

TAB. 3.2 – Performances de l'algorithme ADRes pour la séparation de sources percussives

3.3.1.2 ICA par sous-bande

Principe L'approche retenue consiste à décomposer les signaux droite et gauche $d(n)$ et $g(n)$ à traiter par le banc de filtres décrit en 3.2. Soient $d_k(n)$ et $g_k(n)$ les signaux de sous-bande produits. L'application d'une ICA [HO00] à la matrice :

$$\mathbf{S_k} = \begin{bmatrix} d_k(0) & \dots & d_k(L-1) \\ g_k(0) & \dots & g_k(L-1) \end{bmatrix} \tag{3.5}$$

produit une matrice de démixage $\mathbf{W_k}$ et une matrice $\mathbf{Y_k}$ telles que :

$$\mathbf{Y_k} \overset{ICA}{=} \mathbf{W_k}\mathbf{S_k} \tag{3.6}$$

avec :

$$\mathbf{Y_k} = \begin{bmatrix} \hat{s}_{k,1}(0) & \dots & \hat{s}_{k,1}(L-1) \\ \hat{s}_{k,2}(0) & \dots & \hat{s}_{k,2}(L-1) \end{bmatrix} \tag{3.7}$$

$\hat{s}_{k,1}$, $\hat{s}_{k,2}$ étant deux sources indépendantes vérifiant :

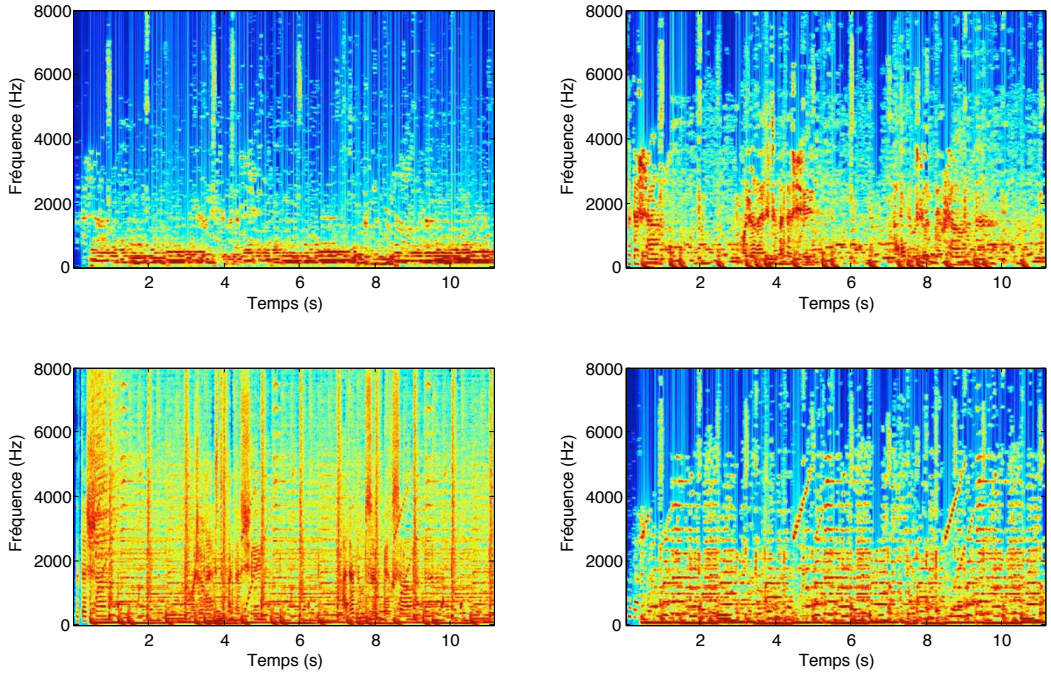


FIG. 3.7 – Séparation d'un enregistrement stéréophonique (Czerkinsky – *Natacha*) avec la méthode ADRes. Sources extraites, de gauche à droite et de haut en bas : orgue ($\gamma = \frac{1}{16}$), voix et grosse caisse ($\gamma = \frac{3}{8}$), caisse claire, voix et cuivres ($\gamma = \frac{1}{2}$), basse et effets ($\gamma = \frac{11}{16}$)

$$p(\hat{s}_{k,1}(n) = x, \hat{s}_{k,2}(n) = y) = p(\hat{s}_{k,1}(n) = x)p(\hat{s}_{k,2}(n) = y) \tag{3.8}$$

Sous les deux hypothèses suivantes : a) Le signal considéré est produit par mixage panoramique, b) Dans chaque bande k du banc de filtres, ne sont non-nuls que les signaux de sous bandes provenant de deux sources indépendantes $s_{k,i}(n)$ et $s_{k,j}(n)$; on peut identifier $\hat{s}_{k,1}$ et $\hat{s}_{k,2}$ à $s_{k,i}(n)$ et $s_{k,j}(n)$ à permutation et gain près. La première de ces hypothèses a déjà été discutée au début de ce chapitre – elle peut être considérée comme valide sur des enregistrements commerciaux de musique populaire. L'hypothèse d'indépendance des échantillons des sources est également valide. La présence de deux sources par sous-bande est par contre discutable, puisque, dans chacune des sous-bandes k, plusieurs sources peuvent être actives. Cependant, une hypothèse moins forte peut être formulée : dans chaque sous-bande, une ou deux sources sont prédominantes. Dans ce cas, le critère d'indépendance utilisé dans l'ICA favorise la séparation de cette ou de ces deux sources prédominantes. Cette propriété de l'ICA est vérifiée expérimentalement selon le protocole suivant :

1. N sources s_i sont tirées aléatoirement parmi une collection de 22 signaux monophoniques, correspondant à diverses parties et variations d'un arrangement construit sur une même grille d'accords, jouées sur différents instruments. Même si le mixage produit est synthétique, les signaux ne sont pas musicalement indépendants.
2. Un mélange panoramique de ces N sources est réalisé, avec des valeurs de panoramique aléatoires. Nous distinguons trois cas :
   - Dans une première série d'expériences, un des gains est à 0 dB, les autres sont à $-12dB$.
   - Dans une seconde série d'expériences, deux des gains sont à 0 dB, les autres sont à $-12dB$.
   - Dans une troisième série d'expériences, tous les gains sont à 0 dB.

	$N = 2$		$N = 3$		$N = 4$		$N = 6$		$N = 8$	
Bande	SIR_1	SIR_2	SIR_1	SIR_2	SIR_1	SIR_2	SIR_1	SIR_2	SIR_1	SIR_2
Une source prédominante										
1	79	37	55	19	44	13	32	7	27	5
2	46	29	29	11	22	7	14	3	10	1
3	51	31	32	12	22	8	15	3	13	2
4	55	35	32	14	21	8	13	3	9	0
5	57	39	35	15	29	9	22	4	17	2
6	64	44	41	21	33	11	24	6	18	3
7	83	62	62	44	53	29	40	16	32	13
8	101	94	83	76	68	59	49	39	37	26
Deux sources prédominantes										
1	78	38	60	20	44	13	33	9	27	5
2	47	28	30	13	22	8	16	4	11	2
3	52	28	30	13	22	7	17	4	11	1
4	56	30	33	13	22	7	14	2	10	0
5	58	35	36	16	28	8	22	5	17	3
6	66	36	44	20	32	13	24	7	19	3
7	87	57	67	42	51	24	43	18	37	14
8	111	103	84	70	69	54	56	42	46	31
Sources également mixées										
1	78	38	56	15	46	11	33	8	29	6
2	47	28	30	13	22	7	15	4	8	1
3	52	28	28	11	23	6	17	4	11	1
4	56	30	27	12	19	6	14	2	7	-1
5	58	35	35	14	26	7	23	4	14	1
6	66	36	44	19	31	10	24	6	15	2
7	87	57	68	26	52	19	42	16	33	9
8	111	103	86	69	70	52	55	36	47	23

TAB. 3.3 – SIR (dB) des deux sources extraites par ICA dans les signaux de sous-bande, à partir d'enregistrements stéréophoniques

3. Dans chaque sous-bande, deux sources sont extraites par ICA $\hat{s}_{k,1}$ et $\hat{s}_{k,2}$. L'implémentation de l'ICA choisie est FastICA [Hyv99]. Ces sources sont projetées sur les signaux de sous-bandes des sources originales, permettant le calcul d'un critère de pureté des sources extraites. Ce critère est le rapport signal à interférences – *Signal to Interferences Ratio* (SIR), rapport de puissance entre la source prédominante extraite et les autres sources présentes :

$$SIR_j = \log_{10} \frac{||\langle \hat{s}_{k,j} s_{k,m} \rangle s_{k,m}||^2}{||\sum_{i \neq m} \langle \hat{s}_{k,j} s_{k,i} \rangle s_{k,i}||^2} \tag{3.9}$$

Où $m = \text{argmax}_{m \in \{1,\ldots,N\}} ||\langle \hat{s}_{k,j} s_{k,m} \rangle s_{k,m}||^2$ (m représente l'indice de la source prédominante dans $\hat{s}_{k,j}$).

Les résultats sont donnés dans la table 3.3. Dans le cas où le signal original est constitué de sources mixées avec le même gain, une des sources extraites par l'ICA parmi les signaux de sous-bandes est toujours "pure", au sens où elle se compose majoritairement d'une des sources originales. Par contre, l'autre source extraite par ICA est plus fréquemment composite, en particulier pour de grandes valeurs de N. Nous observons également que la pureté des sources extraites varie en fonction de l'indice de la bande. Les bandes 2, 3, 4 et 5, correspondant à l'intervalle de fréquences $[172, 2760]$ Hz, sont les bandes dans lesquelles les sources extraites sont les moins pures – cette

région du spectre est la plus remplie par les partiels des instruments jouant les parties harmoniques et mélodiques. Dans les hautes fréquences, les sources extraites sont extrêmement pures – une explication possible est que cette région du spectre contient principalement les composantes bruitées provenant des cymbales ou de la caisse claire.

L'ICA sur les signaux de sous-bandes est ainsi retenue comme méthode d'extraction de sources à partir de signaux stéréophoniques. L'accentuation de la piste de batterie peut alors se faire en ne retenant, parmi les sources extraites, que celles associées aux instruments percussifs.

3.3.2 Critères de percussivité pour la sélection des sources

Les sources extraites correspondent soit à des sources harmoniques pures (à rejeter), soit à des sources percussives pures (à garder), soit à des mélanges de sources harmoniques et percussives (à garder). La classification des sources en classes "source à retenir" et "source à rejeter" est effectuée par une C-SVM avec noyau gaussien et sorties probabilistes (se référer à l'annexe B pour une présentation en détail des SVM). À cet effet, divers attributs sont calculés à partir de chaque source $s_{k,j}$ extraite, en particulier à partir de son enveloppe d'amplitude $e_{k,j} = |s_{k,j}| * h$, où h est un filtre passe-bas, et de sa dérivée relative $\partial e_{k,j} = \log(1 + |s_{k,j}| * h) * \Delta$ où Δ est un filtre dérivateur. Les attributs utilisés sont rapidement listés ici, et dérivent en partie de ceux utilisés par Helén et Virtanen dans [HV05] pour sélectionner les sources percussives parmi des profils spectraux et temporels produits par NMF.

Asymétrie (skewness) et platitude (kurtosis) Calculés sur le signal de sous-bande $s_{k,j}$ et de son enveloppe d'amplitude $e_{k,j}$. La platitude est particulièrement intéressante car elle fournit une bonne mesure de l'impulsivité d'un signal.

Facteur de crête Défini comme le rapport entre la puissance RMS (*Root Mean Square*) d'un signal et son maximum. Le facteur de crête est calculé à la fois sur $s_{k,j}$ et son enveloppe.

Platitude de l'enveloppe Définie comme le rapport entre la moyenne géométrique et arithmétique des valeurs prises par $e_{k,j}$.

Moyenne et variance de la vitesse des attaques Les attaques correspondent aux échantillons n pour lesquels $(e_{k,j} * \Delta)(n) > 0$. La vitesse de l'attaque est alors mesurée par $(e_{k,j} * \Delta)(n)$.

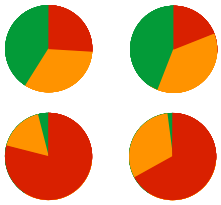
Périodicité. La périodicité de la fonction de détection $\partial e_{k,j}$ est mesurée à l'aide de la valeur du maximum de son autocorrélation dans l'intervalle de délais correspondant à des tempi variant entre 35 à 240 Battements Par Minute (BPM).

Corrélation avec un modèle empirique d'enveloppe de source percussive. Ce critère proposé par Uhle et al. [UDS03] est défini comme la corrélation croisée entre l'enveloppe d'amplitude observée et l'enveloppe d'amplitude qu'aurait une source percussive typique, construite en convoluant un train d'impulsions marquant chaque début de note, par une exponentielle décroissante de constante de temps égale à 100 ms.

Un attribut supplémentaire est considéré : l'indice de la bande k duquel la source est extraite. Les 12 attributs ainsi définis permettent l'apprentissage d'une SVM. Une C-SVM linéaire a été utilisée, avec comme paramètre de régularisation $C = 10$. Les résultats ne dépendent que très peu de C et du noyau utilisé. Le corpus d'apprentissage, dont les sources de sous-bandes ont été annotées manuellement, est le corpus `Music54`. Lors de la classification, la SVM apprise fournit, à partir des paramètres extraits pour chaque source, la probabilité a posteriori $p_{k,j}$ que la j-ième source extraite dans la sous-bande k contienne des composantes percussives.

Ainsi la procédure de séparation consiste à reconstruire un signal à partir des signaux de sous-bandes :

	ADRes	ICA par sous-bandes
Nombre de sources non-percussives soustraites		
Aucune	26 %	19 %
Une	33 %	**37** %
Deux et plus	41 %	**44** %
Nombre de sources non-percussives restantes		
Aucune	**4** %	2%
Une	17 %	**31** %
Deux et plus	79 %	67 %



TAB. 3.4 – Performances de l'ICA par sous-bande suivie d'une sélection de sources pour la séparation de sources percussives

$$\mathbf{S'_k} = \begin{bmatrix} d'_k(0) & \dots & d'_k(L-1) \\ g'_k(0) & \dots & g'_k(L-1) \end{bmatrix} \tag{3.10}$$

avec :

$$\mathbf{S'_k} = (\mathbf{A_k}\mathbf{P_k})\mathbf{Y_k} \tag{3.11}$$

Où $\mathbf{Y_k}$ contient les signaux indépendants produits par ICA, $\mathbf{A_k}$ est la matrice de mixage correspondante, et $\mathbf{P_k}$ est une matrice de sélection de source telle que

$$\mathbf{P_k}_{ij} = \begin{cases} 1 & \text{si } p_{k,j} > \frac{1}{2} \text{ et } i = j \\ 0 & \text{sinon} \end{cases} \tag{3.12}$$

Il est possible de modifier la constante $\frac{1}{2}$ de manière à privilégier soit les faux rejets, soit les fausses acceptations. Dans nos expériences de transcriptions, nous utilisons comme condition $p_{k,j} > \frac{1}{3}$ de manière à éviter les faux rejets de sources percussives. En effet, un faux rejet aboutira vraisemblablement à une erreur de transcription (frappe ou ensemble de frappes non transcrites), suggérant l'usage d'un seuil de décision inférieur à $\frac{1}{2}$. Inversement, pour des applications de séparation et remixage, les fausses acceptations sont moins souhaitables que les faux rejets.

3.3.3 Résultats expérimentaux

Dans cette expérience, le procédé de séparation et sélection automatique des sources décrit est appliqué aux 54 enregistrements musicaux utilisés précédemment (Corpus `Music-54`). La SVM utilisée pour la classification des sources extraites d'un enregistrement a été apprise sur les 53 autres enregistrements, selon le protocole dit *leave one out*. Les résultats sont donnés dans la table 3.4, et sont comparés à ceux obtenus avec ADRes (précisons que dans le cas d'ADRes, la sélection des sources était effectuée manuellement). Ces résultats montrent que l'ICA par sous-bandes est plus apte à supprimer du signal stéréophonique des sources non-percussives. Cependant, le nombre de sources non-percussives restantes dans le signal est supérieur à deux dans 67% des cas observés. Ainsi, cette méthode, utilisée seule, ne peut permettre de séparer efficacement la piste de batterie.

3.4 Extraction de la composante stochastique

Cette section présente la méthode retenue pour l'extraction de la composante stochastique (bruit) d'un signal de musique. Dans une première partie, nous présentons un modèle de la partie déterministe

(harmonique) du signal et une méthode d'estimation de ses paramètres et d'obtention de la composante stochastique. Dans une seconde partie, nous discuterons de la mise en oeuvre de cette méthode pour l'accentuation de la batterie dans les signaux de musique.

Mais avant tout, soulignons que la méthode que nous présentons n'est pas la seule voie possible. Dans [Alo06], Alonso présente une méthode d'extraction de la partie stochastique d'un signal, basée sur la méthode analyse-transformation-synthèse et sur un estimateur spectral non-paramétrique insensible à la présence de pics dans le périodogramme. Son application à la détection d'onsets sur des signaux de piano est traitée par Filippi dans [Fil06]. Nous n'avons cependant pas retenu cette solution, qui malgré son très faible coût en calcul, détruit l'information de phase dans le signal original – une propriété gênante pour des applications de remixage où la composante stochastique extraite doit être rajoutée ou superposée au signal original.

3.4.1 Présentation théorique

3.4.1.1 Modèle EDS

Le modèle retenu pour la modélisation de la partie déterministe du signal est le modèle sinusoïdes modulées exponentiellement – *Exponentially Damped Sinusoids* (EDS). Ce modèle présente l'avantage d'être à la fois pertinent pour les signaux d'instruments de musique, et d'avoir été suffisamment étudié pour disposer de méthodes d'estimation robustes et efficaces. En particulier, Badeau présente dans [Bad05] une large gamme de résultats quant à la convergence et la complexité de ces méthodes d'estimation. Les méthodes présentées ici sont dites à *haute résolution*, car elles ne souffrent pas du compromis résolution temporelle/résolution fréquentielle propre à l'analyse de Fourier.

La partie déterministe $s(n)$ du signal observé est décrite par une somme de r sinusoïdes de pulsations ω_m, phases ϕ_m, amplitudes a_m, dont l'amplitude est modulée par une exponentielle de constante de temps $-\frac{1}{\delta_m}$:

$$s(n) = \sum_{m=1}^{r} a_m e^{\delta_m n} cos(\phi_m + \omega_m n) \tag{3.13}$$

En posant $\alpha_m = a_m e^{j\phi_m}$ (amplitudes complexes) et $z_m = e^{j\omega_m + \delta_m}$ (pôles complexes), on a :

$$s(n) = \Re(\sum_{m=1}^{r} (\alpha_m z_m^n)) \tag{3.14}$$

$$= \sum_{m=1}^{r} \alpha_m z_m^n + \alpha_m^\star z_m^{\star n} \tag{3.15}$$

3.4.1.2 Méthodes d'estimation

Principe de l'analyse en sous-espaces Si l'on considère un vecteur constitué de l échantillons consécutifs de s :

$$\mathbf{s} = [s(n) \dots s(n+l-1)]^T \tag{3.16}$$

Alors ce vecteur appartient au sous-espace de dimension $2r$, dont une base est donnée par la matrice :

$$\mathbf{Z} = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ z_1 & z_1^* & \dots & z_r & z_r^* \\ \vdots & \vdots & & \vdots & \vdots \\ z_1^{l-1} & z_1^{*l-1} & \dots & z_r^{l-1} & z_r^{*l-1} \end{bmatrix} \tag{3.17}$$

Considérons la matrice de Hankel formée à partir de $2l-1$ échantillons successifs de s, avec $l >> 2r$:

$$\mathbf{H}_s = \begin{bmatrix} s(0) & s(1) & \dots & s(l-2) & s(l-1) \\ s(1) & s(2) & \dots & s(l-1) & s(l) \\ \vdots & \vdots & & \vdots & \vdots \\ s(l-1) & s(l) & \dots & s(2l-3) & s(2l-2) \end{bmatrix} \tag{3.18}$$

Toutes les colonnes de $\mathbf{H}_s$ appartiennent au même sous-espace de dimension $2r$ engendré par $\mathbf{Z}$ – autrement dit, $\mathbf{H}_s$ est de rang égal à $2r$. Une décomposition en valeurs singulières – *Singular Value Decomposition* (SVD) de $\mathbf{H}_s$ fournit :

$$\mathbf{H}_s \overset{SVD}{=} \mathbf{USV}^H \tag{3.19}$$

Où $\mathbf{S}$ est une matrice diagonale dont seulement $2r$ éléments sont non-nuls. Les colonnes de $\mathbf{U}$ correspondant aux éléments non-nuls de $\mathbf{S}$ forment ainsi une base de l'espace signal engendré par $\mathbf{Z}$.

Notons que si l'on considère la matrice de covariance empirique de s, définie par $\hat{\mathbf{R}}_{ss} = \frac{1}{l}\mathbf{H}_s\mathbf{H}_s^H$, on a :

$$\hat{\mathbf{R}}_{ss} = \frac{1}{l}\mathbf{USV}^H\mathbf{VSU}^H \tag{3.20}$$

$$= \mathbf{U\Lambda U}^H \tag{3.21}$$

Ainsi, une décomposition en valeurs propres – *Eigenvalue Decomposition* (EVD) de $\hat{\mathbf{R}}_{ss}$ fournit également une base $\mathbf{U}$ de l'espace signal.

Supposons désormais que l'on observe un signal $x(n) = s(n) + w(n)$ où $w(n)$ est un bruit blanc gaussien de puissance σ^2. La matrice d'autocovariance observée sera alors $\mathbf{R}_{xx} = \mathbf{R}_{ss} + \mathbf{I}\sigma^2$. Soit (v, λ) un vecteur propre de $\mathbf{R}_{ss}$ et sa valeur propre associée. Puisque $\mathbf{R}_{xx}v = (\lambda+\sigma^2)v$, les vecteurs propres de $\mathbf{R}_{ss}$ sont des vecteurs propres de $\mathbf{R}_{xx}$, et les valeurs propres associées sont augmentées de σ^2. Nous en déduisons que dans le cas où $s(n)$ est bruité, les $2r$ valeurs propres principales sont associées à des vecteurs propres engendrant l'espace signal. Notons $\mathbf{W}$ la matrice contenant ces vecteurs. Les $l-2r$ autres valeurs propres sont égales à σ^2, et associées à des vecteurs propres qui définissent une base $\mathbf{W}_\perp$ On appelle span $\mathbf{W}_\perp$ l'espace bruit, et span $\mathbf{W}$ l'espace signal. Ces deux espaces sont orthogonaux : span $\mathbf{W}_\perp \perp$ span $\mathbf{W}$. Comme nous venons de le voir, des bases de ces deux sous-espaces peuvent être obtenues par décomposition de $\mathbf{H_x}$ en valeurs singulières, ou de $\mathbf{R}_{xx}$ en valeurs propres.

Calcul rapide de l'espace signal Les décompositions en valeurs propres et singulières sont des opérations coûteuses en calcul (typiquement $\mathcal{O}(l^3)$). Trois optimisations sont mentionnées dans [Bad05] pour accélérer le calcul de la décomposition en valeurs propres :

1. Puisque seulement les $2r$ valeurs propres principales de $\mathbf{R}_{xx}$ (ou de $\mathbf{C}_{xx} = \mathbf{H}_x\mathbf{H}_x^H$) sont nécessaires, un algorithme itératif dit d'itération orthogonale peut être utilisé. Dans ce cas, $\mathbf{W}_0$ est initialisé aléatoirement et mis à jour selon la règle :

$$\mathbf{W}_{k+1}\mathbf{R} \overset{QR}{=} \mathbf{C}_{xx}\mathbf{W}_k \tag{3.22}$$

   Où $\overset{QR}{=}$ dénote une factorisation $\mathbf{QR}$, et k est l'indice d'itération. Notons qu'il n'est pas nécessaire de calculer $\mathbf{C}_{xx}$ dans la pratique, car le terme de droite $\mathbf{C}_{xx}\mathbf{W}_k$ se réécrit en $\left(\mathbf{H}_x\mathbf{H}_x^H\right)\mathbf{W}_k = \mathbf{H}_x\left(\mathbf{H}_x^H\mathbf{W}_k\right)$. Cette optimisation remplace ainsi une EVD par plusieurs itérations d'une factorisation $\mathbf{QR}$ de complexité $\mathcal{O}(lr^2)$ précédée de deux produits matriciels de complexité $\mathcal{O}(rl^2)$.

2. Les calculs font intervenir deux produits par la matrice des observations $\mathbf{H}_x$ (ou sa transposée), de structure Hankel. Ainsi, le produit de $\mathbf{H}_x$ par un vecteur colonne v contient les valeurs de $x * v$. Un tel produit de convolution peut être calculé rapidement par deux transformées de Fourier rapides de v et de x, un produit terme à terme, et une transformée de Fourier inverse. En appliquant cette méthode à chacune des $2r$ colonne de $\mathbf{W}_k$, les produits intervenant dans $\mathbf{H}_x(\mathbf{H}_x^H\mathbf{W}_k)$ peuvent être effectués par un algorithme de complexité $\mathcal{O}(rl\log l)$.

Suivi de l'espace signal L'estimation de l'espace signal ne s'est faite jusqu'ici que sur une fenêtre d'observation de longueur $2l$. Les paramètres des signaux de musique variant au cours du temps – de telles variations sont dues à des phénomènes aussi divers que les apparitions et disparitions de notes ou les vibratos et trémolos – l'estimation doit se faire successivement sur des fenêtres de longueur suffisamment courtes pour que le signal x y soit considéré stationnaire. Il est alors possible d'utiliser l'espace signal obtenu à la fenêtre précédente pour initialiser l'algorithme d'itération orthogonale. Badeau et al. rapportent dans [Bad05] qu'avec cette approche, la convergence est atteinte en une seule itération. Si $\mathbf{W}_n$ dénote l'espace signal estimé sur la n-ième fenêtre d'observation, on a la récurrence suivante :

$$\mathbf{C}_{xx} \overset{EVD}{=} \mathbf{W}_0\mathbf{\Lambda}\mathbf{W}_0^H \tag{3.23}$$

$$\mathbf{W}_{n+1}\mathbf{R} \overset{QR}{=} \mathbf{H}_x\left(\mathbf{H}_x^H\mathbf{W}_n\right) \tag{3.24}$$

Nous avons réalisé une implémentation d'une bibliothèque en langage `C` dédiée au suivi de l'espace signal (et plus généralement à l'estimation des paramètres du modèle EDS), utilisant `LAPACK`[5] pour les opérations matricielles et `FFTW` pour les transformées de Fourier rapide. Cette implémentation permet le suivi de l'espace signal de dimension $2r = 50$ en temps réel sur des signaux audio échantillonnés à 44.1 kHz, avec une machine équipée d'un processeur Core Duo cadencé à 2 GHz.

Extraction de la composante stochastique Il serait possible d'estimer les pôles complexes z à partir de l'espace signal, puis les amplitudes complexes α, afin de resynthétiser le signal $s(n)$ et d'en déduire $w(n) = x(n) - s(n)$. Cette solution se montrerait trop coûteuse en calculs. Une approche plus économe consiste à projeter les observations du signal à décomposer x sur l'espace bruit. Si l'on note :

$$\mathbf{x} = [x(n)\dots x(n+l-1)]^T \tag{3.25}$$

$$\mathbf{w} = [w(n)\dots w(n+l-1)]^T \tag{3.26}$$

Alors :

$$\hat{\mathbf{w}} = \left(\mathbf{W}_\perp{\mathbf{W}_\perp}^H\right)\mathbf{x} = \left(\mathbf{I} - \mathbf{W}\mathbf{W}^H\right)\mathbf{x} \tag{3.27}$$

Notons que cette approche est un cas particulier de filtrage en sous-espace [WYC04; HW04]. Un filtre en sous-espace est spécifié par L réels $0 \leq (\gamma_i)_{i\in\{1,\dots,l\}} \leq 1$, formant une matrice diagonale $\Gamma = diag(\gamma_1,\dots,\gamma_l)$, et s'applique à un signal selon :

$$\hat{s}_\Gamma = (\Gamma\mathbf{U})^H(\Gamma\mathbf{U})\mathbf{x} \tag{3.28}$$

Dans le cas de l'extraction de la composante stochastique, si les colonnes de $\mathbf{U}$ sont rangées par valeurs propres décroissantes, le filtre en sous-espace correspondant est défini par $\gamma_i = \begin{cases} 0 & \text{si } i \leq 2r \\ 1 & \text{sinon} \end{cases}$

Précisons également que ces méthodes de filtrage en sous-espace ne réalisent qu'une estimation approximative des composantes déterministes et stochastiques du signal. Sur les spectrogrammes de la figure 3.1 (obtenus par filtrage en sous-espace), on distingue par exemple, dans la composante déterministe, du bruit en dehors des raies harmoniques.

[5] L'implémentation utilisée est celle du Framework Accelerate, inclus dans Mac OS X, qui tire efficacement parti des systèmes multi-processeurs.

3.4.2 Mise en oeuvre

La mise en oeuvre de cette méthode pour l'extraction de la partie stochastique des signaux de musique suppose d'abord l'ajustement de deux paramètres : la taille l des fenêtres d'observation et le nombre r de sinusoïdes à extraire.

Fenêtres d'observation Puisque les signaux de sous-bande à traiter sont sous-échantillonnés, la taille l est variable, et est donnée dans la table 3.5. Dans les bandes supérieures, cette taille correspond à des fenêtres d'observation longues de 23 ms – durée sur laquelle les signaux audio peuvent être considérés comme stationnaires. Dans les bandes inférieures, la taille des fenêtres est limitée à une valeur supérieure à 32 pour deux raisons : d'une part, l'estimation des sinusoïdes n'est robuste que sur des fenêtres d'observation suffisamment longues, il n'est donc pas possible d'utiliser des fenêtres plus courtes. D'autre part, utiliser une fenêtre longue (96 ms dans les bandes les plus basses) permet de favoriser le suivi de composantes sinusoïdales stables – les composantes sinusoïdales fortement modulées en fréquence, ou disparaissant rapidement ne seront pas prises en compte par le modèle. Nous avons vu que les composantes harmoniques de la caisse claire et de la grosse caisse ont ces propriétés. À des fins d'accentuation de la piste de batterie, il est donc souhaitable que ces composantes ne soient pas modélisées dans la composante déterministe.

Enfin, précisons que de manière à éviter les discontinuités entre fenêtres successives, la projection sur l'espace bruit est effectuée sur des fenêtres se recouvrant à 75%. La partie stochastique du signal à traiter est alors obtenue par addition recouvrement, utilisant une fenêtre de Hann.

Blanchiment du bruit Nous avons supposé dans les développements précédents que le bruit $w(n)$ était blanc. Ce n'est pas le cas dans la pratique. Nous nous rapprochons cependant de cet idéal par deux moyens. Tout d'abord, en effectuant l'analyse sur des signaux de sous-bandes. Dans ces signaux de sous-bande, la d.s.p du bruit peut être considérée comme plus "plate" que sur l'intégralité du signal. Ensuite, en blanchissant les signaux de sous-bandes avant leur analyse. À cet effet, nous estimons d'abord la d.s.p du bruit à l'aide d'un estimateur spectral non-paramétrique insensible à la présence de pics dans le périodogramme[6]. Nous en déduisons sa fonction d'autocorrélation, puis les coefficients d'un filtre tout-zéros de blanchiment par prédiction linéaire – un filtre d'ordre 5 étant utilisé. Ce filtre est appliqué au signal $x(n)$ avant l'étape de suivi de l'espace signal.

Nombre de sinusoïdes Différentes méthodes de sélection d'ordre pour les modèles sinusoïdaux ont été proposées dans la littérature – par exemple, le critère ESTER défini par Badeau et al. dans [BDR05]. Cependant, ces critères ne sont pas adaptés à notre application. Premièrement, ils sont coûteux en termes de calculs, puisqu'ils demandent d'estimer un espace signal de dimension R, où R est un nombre maximal de sinusoïdes à extraire, avant de n'en retenir qu'un sous-espace. Autrement dit, la sélection de l'ordre se fait a posteriori. Deuxièmement, ces critères n'ont été définis que sur des signaux stationnaires. Dans le cas de signaux de musique polyphonique, il est clair que l'ordre de modélisation doit pouvoir varier au cours du temps, pour accommoder l'arrivée ou la disparition de notes. Dès lors, deux problèmes se posent.

Premièrement, les méthodes de suivi, par exemple la méthode des puissances itérées, supposent que l'ordre ne varie pas. Elles perdent leur efficacité lorsque l'ordre peut varier d'une fenêtre à l'autre. Une méthode de suivi de sous-espace de dimension variable coûteuse en calculs consiste à effectuer pour chaque fenêtre plusieurs itérations orthogonales (au lieu d'une), initialisées avec l'espace signal obtenu à la fenêtre précédente, éventuellement augmenté de vecteurs aléatoires $\mathbf{X}$ (ou tronqué) en cas de changement de l'ordre :

$$\mathbf{W}_{n,0} = \left[\mathbf{W}_{n-1,K_{n-1}} \;\; \mathbf{X}\right] \tag{3.29}$$

$$\mathbf{W}_{n,k+1}\mathbf{R} \overset{QR}{=} \mathbf{H}_x \left(\mathbf{H}_x^H \mathbf{W}_{n,k}\right) \tag{3.30}$$

[6]Les pics dans le périodogramme sont simplement lissés par un filtre de rang.

Indice de la bande	1	2	3	4
Fréquences (Hz)	0–172	172–344	344–689	689–1378
Fenêtre d'observation l	32	32	32	32
Durée correspondante (ms)	93	93	46	23
Sinusoïdes extraites r	2	4	6	6
Indice de la bande	5	6	7	8
Fréquences (kHz)	1.38–2.76	2.76–5.51	5.51–11.02	11.02–22.05
Fenêtre d'observation l	64	128	256	512
Durée correspondante (ms)	23	23	23	23
Sinusoïdes extraites r	12	12	16	0

TAB. 3.5 – Paramètres utilisés pour la séparation de la partie stochastique dans chacune des bandes

K_n désignant ici le nombre d'itérations orthogonales effectuées lors de l'analyse de la fenêtre n. Il n'existe à notre connaissance aucune étude de la convergence et de l'efficacité de cette méthode.

Deuxièmement, nous avons observé sur une large gamme de signaux de musique des variations à court terme de l'ordre estimé par le critère ESTER. Ces variations produisent des composantes sinusoïdales apparaissant et disparaissant rapidement, nuisibles à la qualité du signal extrait.

Nous avons donc sélectionné un ordre fixe pour chacune des bandes. Ce choix a été effectué empiriquement, en considérant le corpus `Music54` (Annexe D.1). Pour chaque extrait musical du corpus, nous avons progressivement augmenté l'ordre dans chaque bande par pas de deux sinusoïdes jusqu'à ce que ce changement n'ait aucun effet perceptible dans le résiduel, ou bien que l'ajout de sinusoïdes élimine une des composantes harmoniques d'un des instruments percussifs. La médiane des valeurs obtenues pour chacun des 54 extraits a été gardée. Les valeurs choisies, listées en 3.5, sont comparables à celles utilisées par Alonso dans [Alo06], bien que légèrement plus faibles, dû à notre choix de ne pas surestimer le nombre de sinusoïdes, et donc de ne pas éliminer des composantes issues des instruments percussifs. Particulièrement, dans la bande la plus basse, où ne jouent typiquement que la grosse caisse et la basse, le nombre de composantes a été fixé à une faible valeur. Précisons également qu'aucune sinusoïde n'est extraite dans la bande la plus haute – l'intégralité du signal est considérée comme stochastique dans cette bande. Cette approximation permet des gains substantiels en termes de temps de calcul, la bande la plus haute ayant la fréquence d'échantillonnage la plus élevée.

3.5 Conclusion

Dans ce chapitre, nous avons présenté deux traitements permettant d'accentuer la piste de batterie dans des enregistrements musicaux polyphoniques. Le premier traitement, propre aux enregistrements stéréophoniques produits par mixage panoramique, extrait des sources monophoniques à l'aide d'une ICA et élimine celles considérées comme non-percussives. La décision est effectuée à l'aide d'une SVM, utilisant comme attributs des mesures d'impulsivité et de périodicité de l'enveloppe d'amplitude. Dans 81% des cas, au moins une source non-percussive peut ainsi être soustraite. Le second traitement consiste en l'estimation de la composante stochastique du signal, à l'aide de méthodes de filtrage en sous-espace – traitement pouvant aussi être vu comme la soustraction de la partie déterministe modélisée par une somme de sinusoïdes modulées en amplitude par une exponentielle.

Une question n'a pas été abordée : qu'apportent ces traitements pour des applications de transcription de la piste de batterie, ou de séparation et remixage ? Nous y répondrons dans les chapitres suivants, en décrivant, au chapitre 4 un système de transcription de la piste de batterie utilisant des at-

tributs calculés sur le signal dans lequel la piste de batterie a été accentuée ; et en évaluant au chapitre 5 ces pré-traitements, ainsi qu'une méthode plus complète les étendant, sur la tâche de séparation et de remixage de la piste de batterie.

Publications liées à ce chapitre

Nos premiers travaux décrivant l'application des méthodes de séparation harmonique/bruit à l'extraction de la piste de batterie dans des enregistrements musicaux sont décrits dans [GR05d]. Les plus récents développements sont décrits dans [GR07].

CHAPITRE 4

Transcription de la batterie dans un signal de musique

Ce chapitre, qui forme le coeur de la première partie de cette thèse, décrit un système de transcription de la piste de batterie des enregistrements musicaux polyphoniques. Ce système suit l'approche Segmenter et Reconnaître. Nous présentons et discutons son architecture dans la section 4.1. Le module de segmentation, qui consiste en un détecteur d'onsets classique, est brièvement décrit en 4.2. Nous explicitons ensuite en 4.3 la procédure de calcul des paramètres acoustiques sur chacun des segments extraits. La tâche de reconnaissance des frappes de la batterie est abordée en 4.4, en présentant les classifieurs et les méthodes de sélection des attributs employés. Jusqu'ici n'ont été considérées pour la transcription que des observations acoustiques. Nous mettons en oeuvre dans la section 4.5 deux stratégies pour inclure des connaissances musicales : une stratégie supervisée utilisant des modèles de N-grammes et plusieurs de ses variantes, et une stratégie non-supervisée de minimisation de la complexité de la transcription. Nous concluons en évaluant ce système de transcription dans la section 4.6. Cette évaluation illustrera l'apport de nos contributions, mais soulignera aussi quelques unes de leurs limites.

4.1 Mise en oeuvre de l'approche Segmenter et Reconnaître

4.1.1 Motivations

Les mérites relatifs des différentes approches proposées dans la littérature pour la transcription des signaux percussifs ont déjà été évoqués au chapitre 2. Nos conclusions étaient alors les suivantes : si les méthodes de type Segmenter et Reconnaître sont incapables de tirer partie de la similarité entre chaque frappe d'un même instrument percussif au sein du morceau (une similarité pas nécessairement présente), ce sont les plus robustes face à la diversité des timbres d'un même élément de la batterie, telles qu'ils peuvent être observés entre différents morceaux.

L'obstacle majeur à la mise en oeuvre de l'approche Segmenter et Reconnaître dans les situations multi-instrumentales est le bruit provenant des instruments non-percussifs. Notre contribution principale consiste à utiliser les méthodes d'accentuation de la piste de batterie introduites au chapitre précédent pour cette tâche de transcription. S'agit-il alors simplement de pré-accentuer la batterie par ces méthodes avant d'effectuer la transcription ? Nous proposons une solution plus complète consistant à effectuer la classification en utilisant à la fois des paramètres acoustiques extraits du signal original, et du signal dont la piste de batterie a été accentuée (signal pré-traité). En effet, d'un côté, certains des attributs du signal original sont très sensibles aux interférences créées par les autres instruments non-percussifs – par exemple, le centroïde spectral d'une frappe de grosse caisse peut être décalé vers le haut si une note aigüe de piano y est superposée. D'un autre côté, d'autres attributs peuvent être plus sensibles aux artefacts introduits par le procédé d'accentuation

de la piste de batterie – par exemple, un attribut mesurant la puissance du signal dans une bande de fréquences où ne jouerait que la grosse caisse est robuste à l'ajout d'autres instruments, mais pas à un pré-traitement qui éliminerait une composante sinusoïdale de la frappe de grosse caisse. Notre solution cherche à combiner ces deux jeux d'attributs de manière à disposer d'attributs les plus robustes possibles. Il serait possible, mais difficile, d'étudier la robustesse des attributs que nous considérons à l'ajout d'autres instruments non-percussifs, ou au procédé d'accentuation de la piste de batterie décrit au chapitre précédent. Comment dès lors déterminer, pour chaque attribut, son instance la plus robuste ? Nous nous proposons de résoudre cette question sans aucun préjugé quant à la robustesse d'un attribut à un traitement donné, par le biais de méthodes statistiques de sélection d'attributs.

Deux démarches se profilent alors :

1. Pour chacun des signaux considérés (signal original, signal avec batterie accentuée), il s'agit de déterminer quels sont les attributs les plus robustes qu'il est possible d'en extraire. Suivre cette approche impose d'utiliser deux systèmes de classification différents pour chaque signal disponible (original, batterie accentuée), chaque système utilisant les attributs les plus robustes pour le signal considéré. La mise en commun des informations fournies par ces classifieurs se présentant alors comme un problème de fusion tardive.
2. Les attributs sont calculés sur tous les signaux considérés, et les plus pertinents d'entre eux sont employés dans un seul système de classification. L'étape de sélection d'attributs peut alors être vue comme un moyen d'accomplir une fusion précoce de l'information.

Les performances offertes par ces deux architectures seront comparées lors de l'évaluation du système dans la section 4.6.

Pour clore cette liste de motivations sur une note plus personnelle, nous espérons que les performances satisfaisantes offertes par notre système constituera un (modeste) argument supplémentaire en faveur des approches guidées par les données, par rapport aux approches guidées par les modèles, pour le traitement de signal. Le débat entre ces deux approches est présenté dans une perspective historique, quasi épistémologique, par Breiman dans [Bre01]. Des deux côtés, qu'il s'agisse de mettre en oeuvre des algorithmes d'apprentissage, ou de proposer un modèle génératif du phénomène observé, des approximations et compromis sont en jeu. Compromis entre bonne généralisation et bon apprentissage d'un côté, ou compromis entre véracité du modèle et tractabilité de la procédure d'estimation de l'autre. Nous pensons que dans le cadre de la transcription des signaux de batterie, signaux pour lesquels il est difficile de dériver un modèle mathématique à la fois expressif et solvable, une approche guidée par les données est préférable. Nous suggérons également que le problème de l'analyse du contenu musical peut être résolu par deux chemins : ou bien en estimant les paramètres des instruments ayant produit les signaux ; ou bien en modélisant le processus de perception d'un auditeur humain (voir figure 4.1). La première voie correspond exactement à ce que nous faisons lorsque nous formulons des modèles génératifs et en inférons les paramètres à partir d'observations. La deuxième voie nécessite de modéliser le processus de perception humaine, dont on sait peu au delà de quelques étapes d'extraction d'attributs et de représentations. À défaut donc, nous pouvons l'approximer par une boîte noire, apprise sur des couples d'entrées et de sorties. Ces "boîtes noires", produites par les algorithmes d'apprentissage, ne doivent pas être perçues comme une marque d'impuissance ou de faiblesse, mais doivent plutôt être vues comme une forme d'approximation d'une autre boîte noire – les étages supérieurs de la cognition musicale.

4.1.2 Quels classifieurs pour quelles taxonomies ?

Une difficulté survenant lors de la mise en oeuvre de l'approche Segmenter et Reconnaître est la reconnaissance des frappes simultanées. Deux solutions sont proposées dans la littérature : ou bien considérer chaque combinaison de frappes possible comme une classe distincte, et utiliser un seul classifieur multi-classes [GR04; SGH04], ou bien considérer autant de classifieurs binaires qu'il existe d'instruments à reconnaître, chaque classifieur binaire détectant la présence ou l'absence d'un des instruments (voir par exemple [TDB05], ainsi que toutes les méthodes représentatives des approches SepDet et MatAda, qui réalisent une détection par élément). La solution la plus adaptée

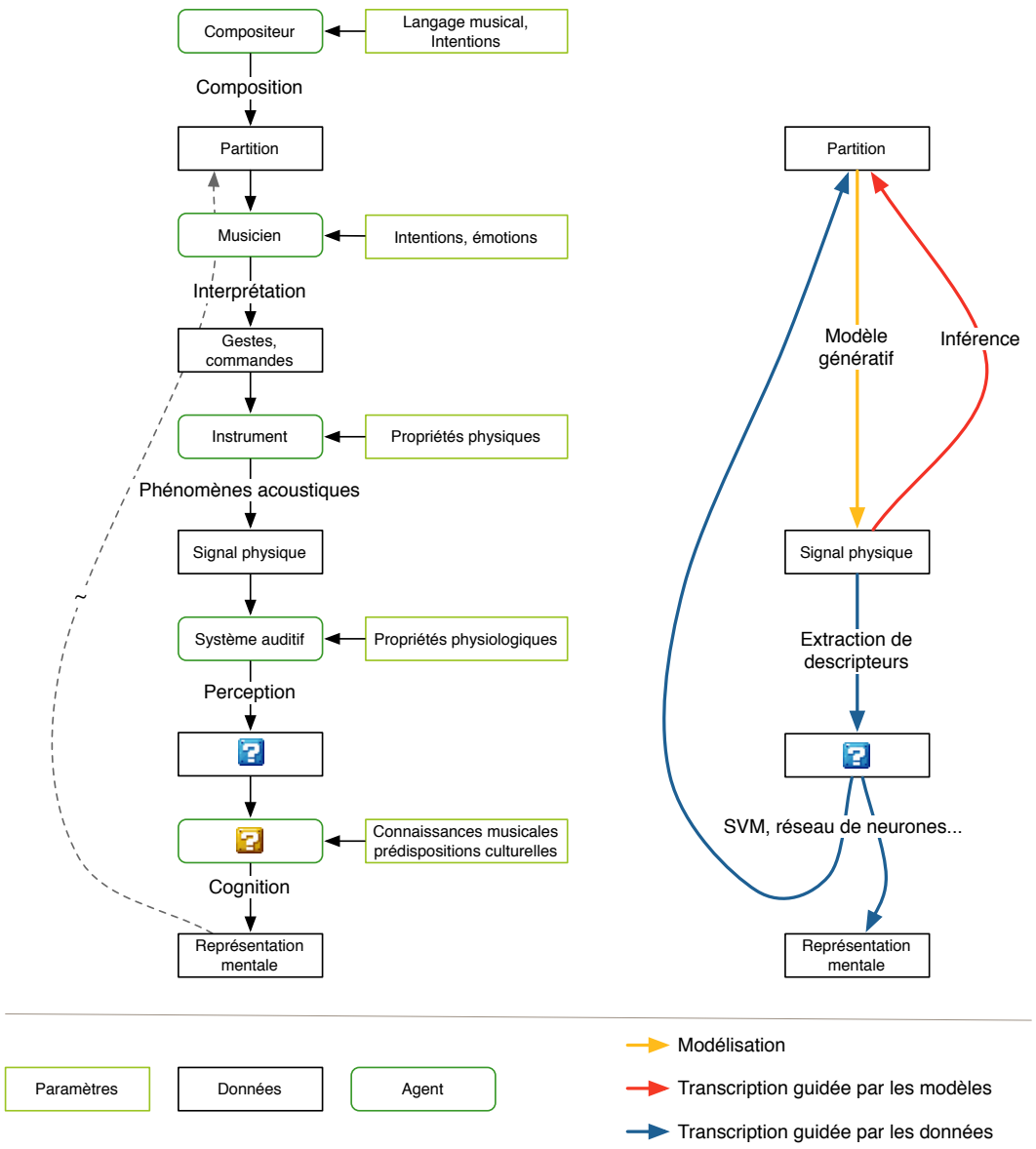


FIG. 4.1 – Le phénomène musical, et les deux approches de la transcription

			Combinaisons		
Taxonomie	**Catégories**	**Couverture**	**95%**	**99%**	**100%**
$\{bd, sd\}$	2	28.8%	3	3	3
$\{bd, sd, hh\}$	3	75.5%	6	7	7
$\{bd, sd, hh, tom\}$	4	79.9%	7	11	15
$\{bd, sd, hh, cym\}$	4	92.1%	10	13	15
$\{bd, sd, hh, tom, cym\}$	5	100.0%	13	20	29
ENST-drums	36	100.0%	68	165	355

TAB. 4.1 – Pouvoir descriptif des taxonomies, et nombre de combinaisons d'instruments rencontrées

dépend de la taxonomie – en particulier, pour une taxonomie détaillée, le nombre de combinaisons à prendre en compte pourrait croître rapidement.

Choix d'une taxonomie pour la transcription de la piste de batterie De manière à déterminer une taxonomie et une stratégie de classification optimale, nous avons évalué la fréquence de jeu des frappes et de leurs combinaisons à partir du corpus ENST-drums (décrit dans l'article [GR06b] reproduit en annexe C). Ce corpus contient 79615 événements, correspondant à 27407 frappes simples et 22545 frappes simultanées. Les fréquences des frappes et de leurs combinaisons les plus communes sont listées, pour diverses taxonomies, dans la table D.4 donnée en annexe. Les acronymes utilisés sont bd pour grosse caisse, sd pour caisse claire, hh pour hi-hat, tom pour les toms, et cym pour les autres cymbales. Dans la table 4.1 sont donnés, pour chaque taxonomie :

1. La couverture, c'est à dire la proportion de frappes pouvant être exactement décrites par les symboles utilisés dans la taxonomie. Par exemple, dans la taxonomie utilisant les catégories $\{bd, sd\}$, la frappe $\{bd, hh\}$ ne peut pas être décrite exactement, la description la plus proche étant $\{bd\}$.
2. Le plus petit nombre de frappes simples ou combinées couvrant respectivement, 95%, 99% et 100% du corpus.

Nous observons d'abord que l'utilisation des deux catégories grosse caisse et caisse claire, fournit une description insuffisante (ne couvrant que 28.8% des frappes). L'ajout de la catégorie hi-hat augmente le pouvoir descriptif de la taxonomie. Dans ce cas, 6 des combinaisons d'instruments possibles (parmi 7) permettent de décrire 95 % des frappes observées. La meilleure taxonomie à 4 éléments est celle incluant, en plus, la cymbale. Dans ce cas, 10 combinaisons (parmi 15) permettent de décrire 95 % des frappes observées. Une taxonomie complète mais grossière – ne faisant pas la distinction entre les diverses variétés de toms et de cymbales – ne fait majoritairement appel qu'à 13 des 31 combinaisons possibles. Notons enfin que la taxonomie détaillée originale du corpus ENST-drums, utilisant des classes différentes pour chaque tom et chaque cymbale, n'emploie majoritairement que 68 combinaisons, parmi les $2^{36} - 1$ combinaisons possibles.

Dans ce chapitre, nous utiliserons une taxonomie à trois éléments – grosse caisse, caisse claire et hi-hat. Cette taxonomie fournit une description acceptable du contenu rythmique, aussi bien pour la recherche par le contenu et l'analyse du genre (tâches pour lesquelles grosse caisse et caisse claire sont les catégories les plus importantes), que pour les applications de resynthèse ou de transcription automatique, où la hi-hat vient compléter et remplir les motifs rythmiques. Par ailleurs, cette taxonomie est celle ayant été retenue pour la campagne d'évaluation MIREX 2005 [MIR] : elle permettra donc de comparer nos performances à d'autres systèmes dont les implémentations logicielles sont disponibles.

De la taxonomie à la stratégie de classification... Explicitons maintenant la stratégie de classification à retenir. Dans le cas où une taxonomie très détaillée est utilisée (taxonomie à 4

éléments ou plus), seulement une fraction des combinaisons possibles est effectivement majoritairement représentée dans le corpus. Il n'est pas souhaitable, dans ce cas, d'utiliser une famille de classifieurs binaires, puisqu'une grande partie de ses sorties possibles représenteront des combinaisons non-existantes. Par exemple, avec la taxonomie $\{bd, sd, hh, tom, cym\}$, une famille de classifieurs binaires est capable de produire la combinaison $\{tom, cym, sd\}$, pourtant impossible à jouer par un batteur.

Par contre, dans le cas où une taxonomie à trois éléments est retenue, quasiment toutes les combinaisons possibles sont représentées dans le corpus. Cela n'exclut donc pas l'utilisation d'une famille de classifieurs binaires, puisque chacune de ses sorties possibles représentent une combinaison significative. Un autre critère entre alors en jeu : le volume de données disponibles pour l'apprentissage. Dans l'exemple d'une taxonomie à trois éléments $\{bd, sd, hh\}$, si l'on utilise un seul classifieur à 7 classes, le nombre d'exemples disponibles pour l'apprentissage sera trop faible pour les combinaisons les moins fréquentes $\{bd, sd, hh\}$ et $\{bd, sd\}$. Au contraire, si l'on utilise 3 classifieurs binaires, les ensembles d'apprentissage pour les classes positives et négatives de chaque classifieur seront équilibrés : 48.2 % des combinaisons observées incluent la caisse claire, 41.6 % la grosse caisse, 58.2 % la hi-hat. Notons cependant que ces classes sont moins homogènes – les exemples positifs pour le classifieur détectant la présence d'une grosse caisse incluront par exemple à la fois des frappes de grosse caisse, et des frappes simultanées grosse caisse + caisse claire.

Traitons maintenant le cas de la combinaison $\emptyset$. Si l'on utilise une famille de classifieurs binaires détectant la présence de chacun des instruments de la batterie, il se peut que tous ces classifieurs renvoient une réponse négative. Dans le cas de signaux de batterie sans accompagnement, cette réponse n'a pas de sens, puisque toute note jouée provient nécessairement de la batterie. Dans une situation polyphonique, une telle réponse a du sens, les classifieurs signifiant simplement que l'événement détecté n'est pas attribué à la batterie, mais à un des autres instruments d'accompagnement. Autrement dit, l'utilisation d'une famille de classifieur binaires peut produire des sorties inconsistantes dans le cas monophonique, alors qu'elle fournit, dans le cas multi-instrumental, une solution élégante à la reconnaissance et au rejet des événements non percussifs détectés.

Pour toutes les raisons évoquées ici, nous affirmons que pour la taxonomie considérée et des enregistrements polyphoniques, la stratégie de classification optimale consiste à utiliser 3 classifieurs binaires détectant la présence ou l'absence de chacun des instruments grosse caisse, caisse claire et hi-hat.[1]

4.1.3 Architecture du système

Nous donnons dans la figure 4.2 un diagramme résumant l'architecture de notre système de transcription de la piste de batterie, sous ses deux variantes : fusion précoce, et fusion tardive.

4.2 Détection d'onsets

4.2.1 Principe de la méthode choisie

La détection d'onsets est effectuée au moyen de l'algorithme de détection proposé par Alonso et al. dans [ARD05].

Tout d'abord, une représentation temps-fréquence du signal considéré est obtenue par TFCT. Notons $\hat{X}(m, k)$ cette représentation – k désigne l'indice d'une bande de fréquence, m l'indice

[1] Notons que ces discussions nous permettent également de mieux comprendre les résultats d'expériences de transcription de la batterie sur des enregistrements monophoniques (boucles de batterie) réalisées en [GR04]. Pour quasiment toutes les méthodes de classification testées (HMM, SVM), l'emploi d'un seul classifieur multi-classes offrait des performances supérieures à une famille de classifieurs binaires. La différence était cependant moindre pour une taxonomie détaillée, et pouvait en grande partie être expliquée par la combinaison $\emptyset$, qui n'a pas de sens en contexte monophonique.

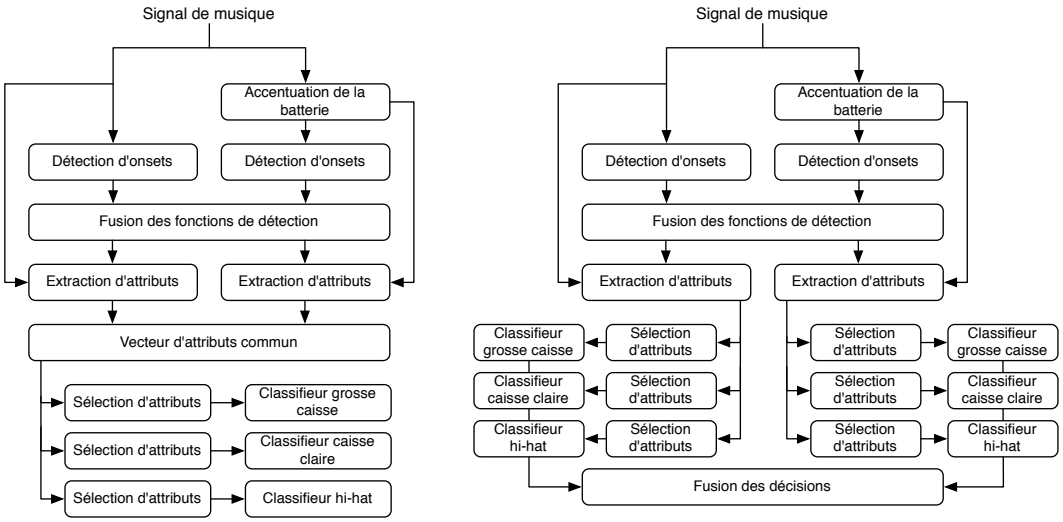


FIG. 4.2 – Architecture du système de transcription de la piste de batterie pour deux approches : fusion précoce et fusion tardive

d'une fenêtre d'analyse. 512 bandes de fréquences sont considérées, les trames d'analyses ayant une longueur de 12ms et se recouvrant de 50%.

Dans chaque bande d'indice k donné, le signal $|\hat{X}(m,k)|$ est filtré par un filtre passe-bas modélisant l'intégration de l'énergie par le système auditif humain, de manière à obtenir une représentation perceptuelle plausible de son enveloppe d'amplitude. Le filtre passe-bas à réponse impulsionnelle infinie utilisé a pour fonction de transfert [Alo06] :

$$H(z) = \frac{(a+b) - (ae^{-\frac{1}{T_2}} - be^{-\frac{1}{\tau_1}})z^{-1}}{1 - (e^{-\frac{1}{\tau_1}} + e^{-\frac{1}{\tau_2}})z^{-1} + e^{-\frac{1}{\tau_1}}e^{-\frac{1}{\tau_2}}z^{-2}} \tag{4.1}$$

Sa réponse impulsionnelle correspond à la somme de deux exponentielles décroissantes $ae^{-\frac{t}{\tau_1}} + be^{-\frac{t}{\tau_2}}$, avec $a = 5$, $b = 1$, $\tau_1 = 75$ ms, $\tau_2 = 15$ ms. Il présente l'avantage d'un coût en calcul moindre par rapport aux filtres à réponse impulsionnelle finie réalisant la même fonction d'intégration de l'énergie sur des longueurs caractéristiques équivalentes – par exemple, les fenêtres de Hann uilisées par Klapuri [Kla99].

La partie positive de la dérivée relative (dérivée du logarithme) de l'enveloppe d'amplitude obtenue est ensuite calculée. À cet effet, le filtre dérivateur optimal décrit dans [Alo06] est utilisé. Il réalise une interpolation polynomiale du signal sur une fenêtre glissante de 11 points pour en calculer la dérivée. Est ainsi obtenue, pour chaque bande, une estimation du flux d'énergie spectral. Une fonction de détection $d_0(m)$ est obtenue en sommant le flux d'énergie spectral sur l'ensemble des canaux de la TFCT. Une dernière étape de filtrage par une demie fenêtre de Hann permet d'en élargir les pics, produisant la fonction de détection finale $d(m)$. Cette fonction possède des pics très prononcés aux instants correspondant aux attaques des notes. La figure 4.3 illustre étape par étape le processus de détection des onsets. Les onsets sont traditionnellement détectés aux instants où la fonction de détection vérifie $d(m) > \tau(m)$, où $\tau(m)$ est un seuil dynamique, obtenu par exemple par filtrage médian de $d(m)$.

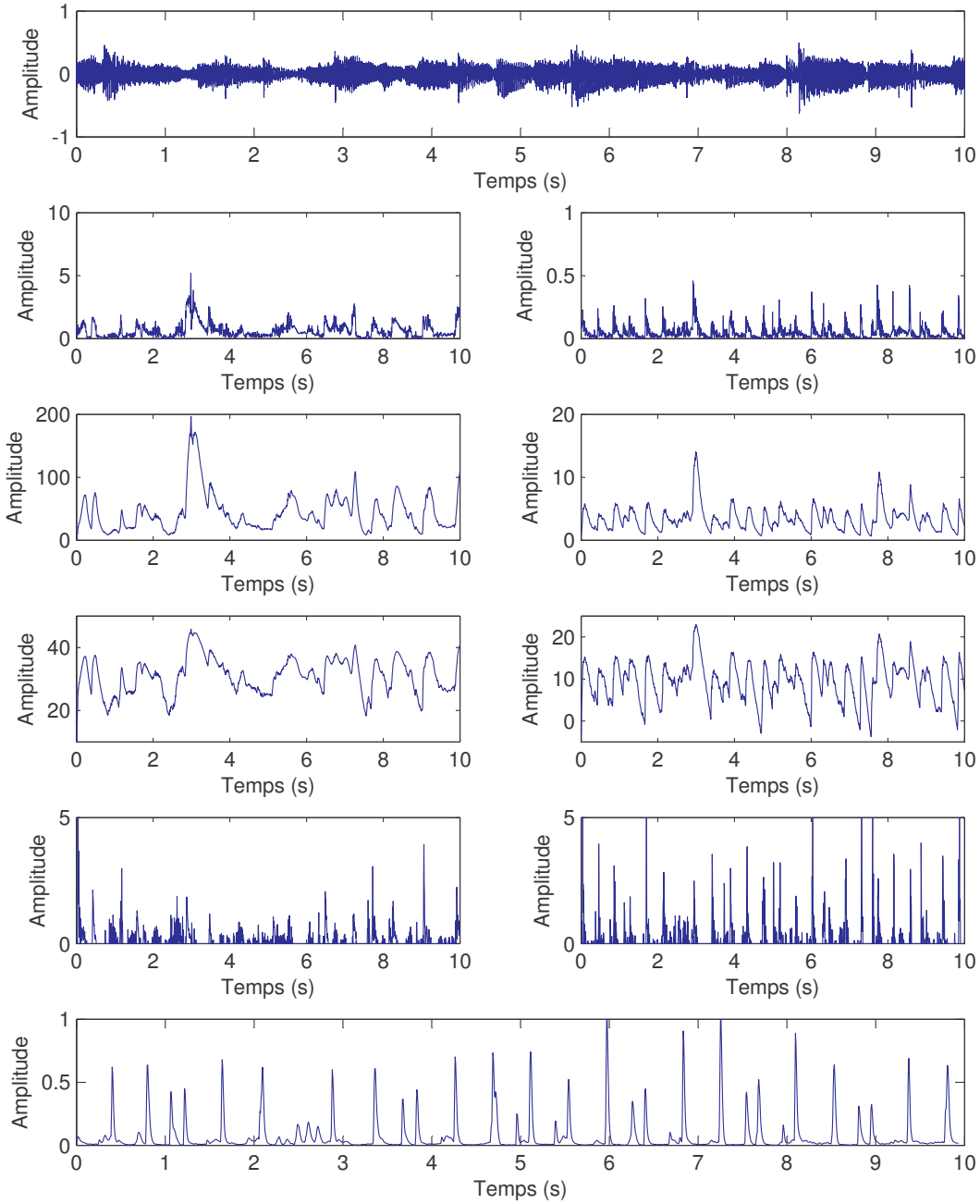


FIG. 4.3 – Détection des onsets : signal original ; module de la TFCT dans les canaux d'indices 20 et 200, intégration de l'énergie, compression de la dynamique, partie positive de la dérivée ; et fonction de détection

4.2.2 Filtres non-linéaire pour la sélection de pics

Nous avons retenu une procédure de sélection de pics s'inspirant de traitements non-linéaires utilisés en traitement d'image, illustrée dans la figure 4.4, et décrite en détails ici :

1. La fonction de détection est filtrée par un filtre médian, selon :

$$d_m(m) = \text{median}[d(m - W_l), \ldots, d(m-1), d(m), d(m+1), \ldots, d(m + W_l)] \quad (4.2)$$

   Une fonction de détection centrée d_c est formée en considérant $d_c(m) = d(m) - d_m(m)$.
2. Une mesure d'échelle (écart-type) est calculée sur la fonction de détection centrée :

$$d_s(m) = \text{std}[d_c(m - W_l), \ldots, d_c(m-1), d_c(m), d_c(m+1), \ldots, d_c(m + W_l)] \quad (4.3)$$

   Une fonction de détection normalisée d_n est formée en considérant $d_n(m) = \frac{d_c(m)}{d_s(m)}$.
3. Les maxima locaux au dessus d'un certain seuil τ sont recherchés dans la fonction de détection mise à l'échelle :

$$d_t(m) = \max[d_n(m - W_s), \ldots, d_n(m-1), d_n(m), d_n(m+1), \ldots, d_n(m + W_s), \tau] \quad (4.4)$$

   Nous avons utilisé la valeur $\tau = 0.5$.
4. Un onset est détecté aux instants où ces maxima locaux sont atteints, c'est à dire aux instants m vérifiant $d_t(m) = d_n(m)$.

Notons que ce procédé de détection garantit deux propriétés. Tout d'abord, puisque la fonction de détection est normalisée sur des durées caractéristiques $T_l = 2W_l + 1$, il ne peut exister de segment long de $2W_l + 1$ durant lequel aucun onset ne sera détecté. Ensuite, puisque les maxima locaux sont calculés sur des fenêtres d'observation de longueur $T_s = 2W_s + 1$, les onsets détectés sont au moins séparés d'une durée T_s. Ces deux observations guident notre choix des paramètres W_l et W_s : nous les avons choisis de façon à avoir $T_s = 100$ ms et $T_l = 10$ s.

4.2.3 Fusion des détecteurs

Nous effectuons ici la détection à partir de deux signaux – le signal original et le signal pré-traité. Il est donc nécessaire, avant même de localiser les onsets, de fusionner les fonctions de détection $d(m)$ et $d_a(m)$ obtenues à partir de ces deux signaux. Plusieurs opérateurs ont été testés pour cette fusion : produit, maximum, minimum, et somme.

Nous donnons dans la table 4.2 les performances du système pour la tâche de détection des onsets des frappes de la batterie, évaluées sur un sous-ensemble du corpus ENST-drums pour différents mixages de la batterie et de l'accompagnement (voir section 4.6). Les performances sont mesurées par le taux de rappel – proportion d'onsets dans le signal original qui ont été effectivement détectés – et de précision – proportion d'onsets valides parmi les onsets détectés. Nous précisons ici que nous nous intéressons à la détection des onsets des frappes de la batterie : un onset associé à une note d'un instrument non-percussif sera considéré comme invalide. Nous avons ajusté les seuils de détection (paramètre τ) de manière à limiter le nombre d'erreurs de type I (onsets non détectés), au prix d'un nombre important d'erreurs de type II (faux onsets), autrement dit, nous avons privilégié le rappel par rapport à la précision. En effet, détecter de faux onsets, ou des onsets associés à des instruments non-percussifs n'est pas gênant, car de tels événements peuvent être par la suite reconnus comme tels lors de la classification. Ces résultats doivent donc être analysés avec précaution.

Les performances obtenues sont très voisines, bien que légèrement meilleures pour l'opérateur somme si l'on utilise une mesure effectuant un compromis entre rappel et précision (comme la

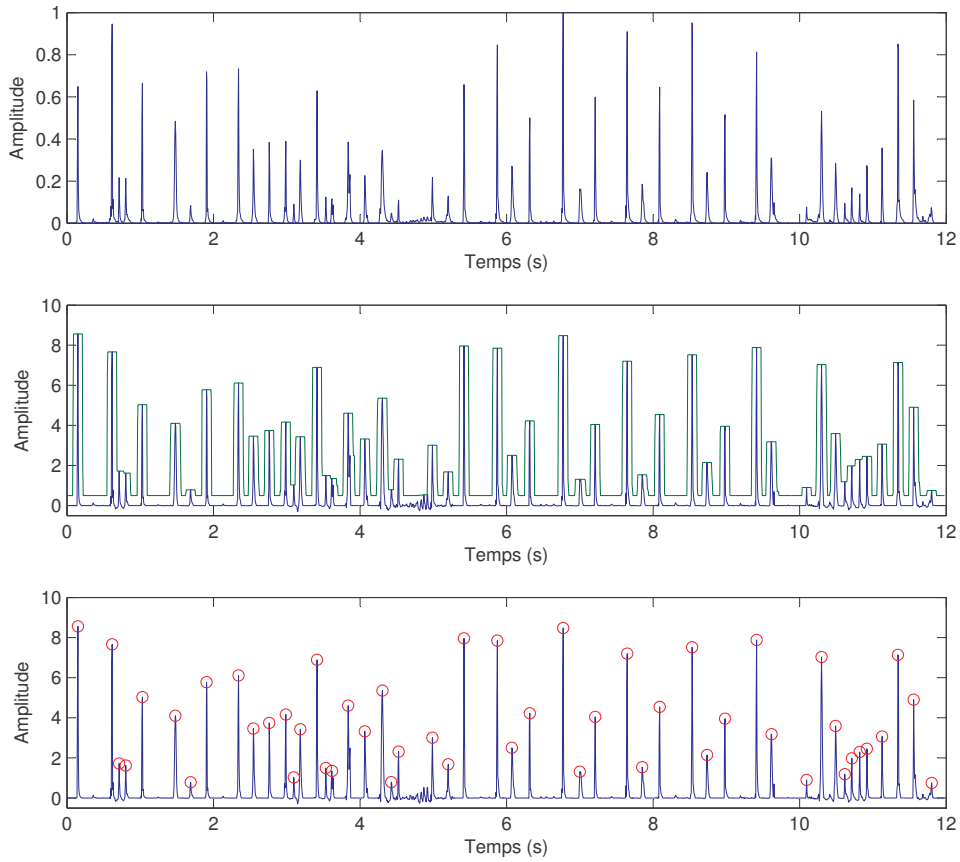


FIG. 4.4 – Localisation des pics dans la fonction de détection : fonction de détection originale $d(m)$, normalisée $d_n(m)$ et maxima locaux $d_t(m)$, onsets détectés

F-mesure). Ce résultat peut s'expliquer par le fait que la méthode d'accentuation de la piste de batterie préserve une partie des transitoires des instruments non-percusifs, et que, plus généralement, la méthode de détection d'onsets choisie est particulièrement efficace sur les signaux impulsionnels aux attaques très marquées, tels que les percussions : $d(m)$ et $d_a(m)$ sont ainsi très voisines.

4.3 Paramétrisation des signaux

4.3.1 Calcul des attributs

Il n'existe aucun consensus quant aux paramètres acoustiques à utiliser pour la reconnaissance des différentes classes d'instruments de la batterie. Dans le contexte monophonique, différents attributs sont décrits dans [GR04] ou [GHD03]. Il serait cependant hasardeux d'appliquer tels quels ces résultats au cas polyphonique. Une étude du cas polyphonique est effectuée par Tanghe et al. dans [TDB05], où sont utilisés différents attributs relativement peu coûteux à calculer et supposés robustes à l'ajout de bruit provenant d'autres instruments de musique (banc de filtres adaptés), ainsi que des attributs plus communs comme les MFCC. Certains de ces attributs ont une interprétation perceptuelle ou acoustique directe (par exemple, les MFCC expriment la forme de l'enveloppe spec-

	Rappel (%)	Précision (%)
Accompagnement $-\infty$ dB		
maximum	94.7	87.8
minimum	94.5	87.9
somme	94.6	88.1
produit	94.4	87.9
Accompagnement -6 dB		
maximum	87.4	82.0
minimum	88.2	83.0
somme	88.0	83.5
produit	88.0	83.1
Accompagnement $+0$ dB		
maximum	85.8	79.5
minimum	86.5	80.3
somme	86.2	81.1
produit	86.6	80.2
Accompagnement $+6$ dB		
maximum	83.7	76.6
minimum	84.6	77.5
somme	84.4	78.5
produit	84.7	78.0

TAB. 4.2 – Performances du module de détection d'onsets, pour divers opérateurs de fusion

trale), qui justifient leur intérêt pour la tâche de classification considérée. D'autres attributs n'offrent pas de telles interprétations, mais ont un fort pouvoir discriminant. Nous choisirons ici de mettre l'accent sur le pouvoir discriminant des attributs considérés, plutôt que sur leur interprétation perceptuelle ou acoustique. Ainsi, nous considérons un ensemble d'attributs candidats particulièrement grand, sans nous soucier pour l'instant de leur robustesse et pertinence, et nous sélectionnons par la suite les plus efficaces d'entre eux par des techniques d'apprentissage statistique. Cette approche, qui troque l'interprétabilité des classifieurs, au profit de leur efficacité, a été appliquée avec succès par Essid et al. [ERD06b] pour le problème de la reconnaissance des instruments de musique.

Il n'existe pas non plus de consensus sur la taille des fenêtres d'observation à considérer pour le calcul des paramètres acoustiques. Dans [TDB05], Tanghe et al. utilisent une durée fixe (180 ms pour le détecteur de grosse caisse, 100 ms pour le détecteur de caisse claire, 140 ms pour le détecteur de hi-hat), tandis que dans [GH01], Gouyon et al. considèrent l'intervalle entre deux pulsations de *tatum*. Dans [GR04], nous utilisons comme fenêtre d'analyse l'intégralité de l'intervalle entre deux onsets successifs. Ce choix améliore la robustesse de l'extraction des paramètres – par exemple, l'estimation de l'enveloppe d'amplitude ou de la densité spectrale de puissance est effectuée à partir d'un plus grand nombre d'échantillons. Cependant, cela augmente également la variabilité des attributs extraits, puisqu'un même attribut peut être tantôt calculé sur l'attaque seule d'une frappe (en cas de frappes très rapprochées dans le temps), ou sur l'intégralité de sa durée (en cas de frappes très espacées dans le temps). De manière à assurer la robustesse du processus d'extraction, tout en minimisant la variabilité des attributs extraits, nous avons décidé d'utiliser pour le calcul des paramètres acoustiques le plus grand nombre possible d'échantillons dans une limite de 200 ms. Ainsi, les paramètres acoustiques associés à l'onset t_i sont calculés sur la fenêtre $[t_i, \min\{t_i + 0.2, t_{i+1}\}]$.

Les différents attributs utilisés sont répertoriés dans le tableau 4.3. L'annexe A offre une définition détaillée de chacun de ces attributs.

Catégorie	Notation	Dimension	Description
D	$lRMS_t$	1	Puissance totale
D	$lRMS_{bd}, lRMS_{sd}, lRMS_{hh}$	3	Puissance en sortie de filtres adaptés [TDB05]
D	$lRMSrel_{bd}, lRMSrel_{sd}, lRMSrel_{hh}$	3	Puissance relative en sortie de filtres adaptés [TDB05]
D	$lRMSrel_{bd,sd}, lRMSrel_{sd,hh}, lRMSrel_{hh,bd}$	3	Puissances comparées en sortie de filtres adaptés [TDB05]
D	$lRMS_{gband,i}$	8	Puissance en sortie d'un b.d.f. adapté à la batterie [GR04]
D	$OBSIR_i$	7	Rapports d'énergie dans un b.d.f. en bandes d'octaves [ERD06b]
D		**25**	Attributs de distribution d'énergie
C	$\mu MFCC_k$	13	Moyenne des MFCC
C	$\sigma MFCC_k$	13	Écart-type des MFCC
C	$\mu\Delta MFCC_k$	13	Moyenne des Δ MFCC
C	$\sigma\Delta MFCC_k$	13	Écart-type des Δ MFCC
C	$\mu\Delta^2 MFCC_k$	13	Moyenne des Δ^2 MFCC
C	$\sigma\Delta^2 MFCC_k$	13	Écart-type des Δ^2 MFCC
C		**78**	Attributs cepstraux
S	$S_{cntr}, S_{sprd}, S_{skew}, S_{kurt}$	4	Moments spectraux [GR04]
S	S_{flat}	1	Platitude spectrale [Pee04]
S	F_c	1	Fréquence de coupure
S	AR_i	6	Coefficients de prédiction linéaire
S		**12**	Attributs spectraux
T	$Crest$	1	Facteur de crête
T	T_{cntr}	1	Centroïde temporel
T	ZCR, ZCR_r	2	Taux de passage par zéro classique/robuste
T	T_A, T_B	2	Paramètres d'enveloppe
T		**6**	Attributs temporels
P	Ldr_i	24	Sonie spécifique relative [Pee04]
P	Acu	1	Acuité [Pee04; Zwi77]
P	Et	1	Étendue [Pee04]
P		**26**	Attributs psychoacoustiques

TAB. 4.3 – Récapitualtif des 147 attributs utilisés. Leur calcul est détaillé dans l'annexe A

4.3.2 Transformation des attributs

4.3.2.1 Normalisation

Les attributs calculés précédemment occupent des échelles et intervalles variés. De manière à disposer d'une échelle commune et commensurable, chaque attribut est transformé de manière à ce que sa moyenne soit nulle et sa variance soit unitaire. Les paramètres de cette transformation affine sont calculés sur la base d'apprentissage, en utilisant des estimateurs empiriques de la moyenne et de la variance.

Une autre méthode de normalisation est fréquemment rencontrée dans la littérature – elle est par exemple utilisée dans [TDB05]. Elle consiste à appliquer une transformation linéaire telle que les valeurs minimales et maximales de chaque attribut sur la base d'apprentissage soient respectivement -1 et 1. Nous n'avons pas appliqué cette méthode, trop sensible à la présence de valeurs extrêmes ou aberrantes.

4.3.2.2 Autres transformations

Nous présentons ici quelques autres transformations des paramètres communément rencontrées dans la littérature, et nous expliquons pourquoi nous ne les avons pas retenues.

Gaussianisation des données Peeters utilise dans [Pee03] une transformation de Box-Cox de paramètre λ définie par :

$$f_\lambda(x) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{si } \lambda \neq 0 \\ \log x & \text{sinon} \end{cases} \tag{4.5}$$

L'intérêt de cette transformation est de rapprocher la distribution de l'attribut x d'une distribution gaussienne. À cet effet, pour chaque attribut, un paramètre λ optimal est choisi, maximisant un critère de gaussianité. Une telle transformation n'a que peu d'intérêt dans notre cas, puisque les méthodes de classification que nous utilisons par la suite ne font pas d'hypothèse de gaussianité des données (une telle transformation aurait plus de sens, par exemple, si la distribution des paramètres associés à chaque classe avait été modélisée par une gaussienne).

Décorrélation des attributs L'analyse en composantes principales – *Principal Component Analysis* (PCA) est une méthode courante d'analyse de données permettant de transformer les vecteurs d'attributs, de manière à extraire de nouveaux attributs à la fois décorrélés, et concentrant un maximum de variance. Si l'on appelle $\mathbf{x}$ les vecteurs d'attributs observés, et $\mathbf{R_{xx}}$ leur matrice de covariance, alors une EVD de $\mathbf{R_{xx}}$ fournit :

$$\mathbf{R_{xx}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{4.6}$$

La diagonale de $\mathbf{\Lambda}$ contient les valeurs propres par ordre décroissant de valeur absolue. Si l'on transforme un vecteur d'attributs $\mathbf{x}$ selon :

$$\mathbf{y} = \mathbf{U}^T\mathbf{x} \tag{4.7}$$

Alors la matrice de covariance des vecteurs transformés est :

$$\mathbf{R_{yy}} = \mathbf{U}^T\mathbf{R_{xx}}\mathbf{U} = \mathbf{\Lambda} \tag{4.8}$$

On en déduit les deux propriétés suivantes :

1. Les composantes de $\mathbf{y}$ sont décorrélées ($\mathbf{R_{yy}}$ est diagonale).

2. Les premières composantes de $\mathbf{y}$ concentrent la variance. En particulier, il est courant de tronquer $\mathbf{y}$ à ses premières composantes (par exemple, celles comportant 95 % de la variance totale), dites composantes principales.

En dépit de sa popularité, nous n'employons pas cette méthode pour différentes raisons. Tout d'abord, les attributs transformés sont une combinaison linéaire de tous les attributs originaux. Or, nous aimerions utiliser par la suite des méthodes de sélection d'attributs afin de sélectionner un ensemble réduit d'attributs pertinents, et éviter ainsi le calcul systématique (et coûteux) de tous les attributs. La PCA est incompatible avec ce but. En outre, il est difficile d'interpréter les attributs transformés $\mathbf{y}$: que serait-il possible de conclure si un algorithme de sélection d'attributs indiquait que l'attribut le plus discriminant est $0.7OBSI_4 - 0.1MFCC_6 + 0.9ZCR_r - 0.2T_A$? Par ailleurs, une motivation fréquente à utiliser une PCA pour décorréler les attributs, est qu'elle rend plus plausible, par la suite, l'usage d'un modèle gaussien avec matrice de covariance diagonale. Puisque nous n'utilisons pas de tels modèles, cet argument ne pèse pas. Terminons enfin par un argument plus pragmatique : nous n'avons observé durant des expériences préliminaires de classification aucun gain notable de performances.

Précisons pour conclure qu'il a également été suggéré d'extraire une matrice de transformation $\mathbf{W}$ rendant statistiquement indépendantes (et non plus seulement décorrélées) les composantes de $\mathbf{y}$, à l'aide d'une ICA. Les gains de performance observés avec cette méthode lors d'études préliminaires ont été négligeables. Il semblerait que les gains de performances rapportés dans la littérature [Ero03] lorsque l'ICA est utilisée comme méthode de réduction de dimensionnalité sont principalement dus à la PCA qui la précède !

4.4 Classification des instruments de la batterie

Dans les développements qui suivent, nous notons $\mathbf{x}_i$ le vecteur d'attributs normalisés extrait dans le segment suivant l'onset t_i. Comme discuté en 4.1.2, nous aimerions disposer de trois classifieurs permettant de détecter si la grosse caisse, la caisse claire, et la hi-hat ont été jouées à l'instant t_i. Notons $\mathbf{y}_{ij}$ la variable égale à -1 si l'instrument j n'est pas joué et à 1 s'il est joué à l'instant t_i.

4.4.1 Expliquer ou discriminer ?

Plusieurs formalismes d'apprentissage sont possibles pour construire de tels classifieurs à partir d'une base d'exemples annotés $(\mathbf{x}_i, y_{ij})_{i\in\{1,\ldots,N\}}$. Deux de ces approches sont dites explicatives (ou génératives), au sens où elles cherchent à obtenir des modèles décrivant (ou pouvant servir à générer) les paramètres acoustiques observés pour chaque classe d'instrument, dont on déduira une règle de décision. La dernière de ces méthodes est dite discriminative, au sens où elle ne cherche pas à extraire d'information quant à la distribution des paramètres acoustiques, mais cherche plutôt à formuler directement une règle de décision optimale.

Approche explicative Une telle approche consiste à construire des modèles décrivant la distribution des vecteurs de paramètres acoustiques $\mathbf{x}$ associés aux frappes incluant la grosse caisse, la caisse claire, ou la hi-hat. Pour chaque instrument j considéré (grosse caisse, caisse claire, hi-hat) :

1. On extrait le sous-ensemble $A_j^+ = \{\mathbf{x}_i, y_{ij} = +1\}$ de la base d'apprentissage contenant les frappes incluant l'instrument j considéré.
2. On modélise la distribution des paramètres acoustiques observés sur A_j^+, de manière à obtenir une estimée de $p(\mathbf{x}|y_j = 1)$. Cette étape peut tout aussi bien utiliser des modèles paramétriques de la densité (modèle de mélange de gaussiennes par exemple) dont les paramètres sont estimés au maximum de vraisemblance, que des estimateurs non-paramétriques (fenêtres de Parzen, SVM à 1 classe).
3. Étant donné un vecteur de paramètres acoustiques $\mathbf{x}$, on détecte la présence de l'instrument j si $p(\mathbf{x}|y_j = 1) > \tau$, où τ est un seuil de décision.

Approche explicative avec "modèle du monde" Cette approche, correspondant au formalisme Bayésien classique de l'apprentissage, consiste à mettre en compétition, pour chaque instrument j à reconnaître, deux modèles : un modèle décrivant la distribution des vecteurs de paramètres acoustiques $\mathbf{x}$ associés aux frappes incluant cet instrument, et un modèle décrivant la distribution des vecteurs de paramètres $\mathbf{x}$ associés aux frappes n'incluant pas cet instrument. Pour chaque instrument j considéré :

1. On extrait le sous-ensemble A_j^+ de la base d'apprentissage contenant les frappes incluant l'instrument j considéré, et son complémentaire A_j^-.
2. On modélise la distribution des paramètres acoustiques observés sur A_j^+, de manière à obtenir une estimée de $p(\mathbf{x}|y_j = 1)$. La même opération est effectuée sur A_j^-, de manière à obtenir une estimée de $p(\mathbf{x}|y_j = -1)$. Par analogie avec le vocabulaire des systèmes de vérification du locuteur, ce second modèle, porte le nom de "modèle du monde".
3. Étant donné un vecteur de paramètres acoustiques $\mathbf{x}$, on détecte la présence de l'instrument j si :

$$\frac{p(\mathbf{x}|y_j = 1)}{p(\mathbf{x}|y_j = -1)} > \tau \tag{4.9}$$

   Où le seuil de décision τ dépend à la fois de la répartition des classes, et du coût associé aux erreurs de classification de type I et II. Dans le cas où on associe un coût identique à ces erreurs, et où $p(y_j = 1) = p(y_j = -1)$, $\tau = 1$.

Approche discriminative Cette approche consiste à directement déterminer une règle de classification (ou une estimée de la probabilité a posteriori $p(y_j|\mathbf{x})$), sous la forme d'une fonction $f_{j,\theta}(\mathbf{x})$, dont le paramètre $\theta \in \Theta$ est choisi pour minimiser un critère, qui peut intégrer à la fois un terme de risque (par exemple, une mesure de l'erreur de classification sur l'ensemble d'apprentissage), et de marge ou de régularité (on impose que la fonction de décision $f_{j,\theta}(\mathbf{x})$ prenne des valeurs "contrastées" selon que $\mathbf{x}$ inclue ou non une frappe de l'instrument j, tout en restant lisse). Ces approches se présentent ainsi traditionnellement sous forme de problèmes d'optimisation – descente de gradient pour les réseaux de neurones artificiels, optimisation quadratique sous contrainte pour les SVM[2].

Nous suivons dans la suite de ce travail une approche discriminative, en privilégiant comme classe de fonctions de décision les machines à vecteurs de support. Ce choix s'explique par notre volonté de ne pas imposer aux données observées un modèle qui s'avérerait inadéquat (modèle de mélange de gaussiennes par exemple), et de résoudre directement le problème de classification sans chercher à résoudre un problème plus général – celui de la formulation d'un modèle des données. Et encore une fois, de façon plus pragmatique, les résultats obtenus dans des études précédentes [GR04] ou préliminaires suggèrent la supériorité des approches discriminatives.

Une présentation détaillée des SVM est effectuée dans l'annexe B. Nous invitons le lecteur, même familier avec cette méthode de classification, à la consulter, ne serait-ce que pour se familiariser avec les notations utilisées par la suite à diverses reprises.

4.4.2 Sélection d'attributs pour la classification

Nous ne souhaitons pas entraîner des classifieurs directement sur les 147 attributs décrits en 4.3 (ou sur les 147×2 attributs extraits du signal original, et du signal dont la piste de batterie a été accentuée dans le cas où l'on utilise une fusion précoce). En effet, certains de ces attributs sont bruités, redondants les uns avec les autres, ou n'ont aucun pouvoir discriminant pour la taxonomie

[2] Précisons que la frontière entre les approches génératives et discriminatives n'est pas toujours aussi prononcée que cette présentation peut le laisser croire. En particulier, l'estimation des paramètres de modèles génératifs au maximum de vraisemblance peut être remplacée par des méthodes d'estimation dites discriminatives ou informatives. De telles méthodes sont utilisées avec succès en reconnaissance de la parole, pour l'apprentissage des paramètres des HMM [BYB04].

considérée. D'autre part, l'extraction systématique de l'intégralité des attributs, tout comme le calcul de produits scalaires ou noyaux sur des vecteurs de grandes dimensions durant l'apprentissage et la classification sont des opérations coûteuses.

La sélection d'attributs consiste à extraire un sous ensemble de d attributs parmi l'ensemble des D attributs candidats, le sous ensemble choisi contenant les attributs les plus *efficaces*. Les méthodes de sélection d'attributs proposées dans la littérature (voir [GE03] pour une introduction au sujet) se distinguent par les méthodes de recherche qu'elles emploient pour explorer l'espace des $2^D - 1$ sous-ensembles d'attributs candidats : algorithmes évolutionnaires, algorithmes grimpeurs (*Hill-climbing*) avec redémarrage, ou simple recherche gloutonne ; et par les critères qu'elles utilisent pour évaluer l'efficacité d'un sous-ensemble d'attributs candidats. Trois familles de méthodes de sélection d'attributs peuvent être définies, en fonction du critère d'efficacité qu'elles emploient :

- Les méthodes en boucle fermée (dites *wrapper*) mesurent l'utilité d'un sous-ensemble d'attributs en évaluant ses performances dans l'étape d'apprentissage et d'évaluation qui suivent la sélection d'attributs : l'ensemble d'attributs sélectionné dépend ainsi des outils d'apprentissage statistiques mis en oeuvre pour la classification. De telles méthodes sont enclines au surapprentissage. Par exemple, dans [FF06], Fiebrink et Fujinaga rapportent le faible pouvoir de généralisation obtenus avec des classifieurs pour lesquels les jeux d'attributs optimaux ont été choisis en boucle fermée.
- Les filtres (*filters*) mesurent l'efficacité d'un attribut indépendamment de l'algorithme d'apprentissage retenu : l'efficacité d'un attribut est mesuré selon sa redondance ou similarité [MMP02] par rapport aux autres attributs sélectionnés, et en mesurant son pouvoir prédictif par rapport aux classes.
- Enfin, les méthodes embarquées (*embedded*) commencent par apprendre un classifieur, et en analysent la fonction de décision pour déterminer les poids et la contribution de chacun des attributs [GWBV02].

Deux algorithmes, l'un représentatif des filtres, l'autre des méthodes embarquées, sont présentés dans la section suivante.

4.4.2.1 Sélection d'attributs par l'algorithme IRMFSP

Considérons un problème de classification à deux classes. Soient N^+ (resp. N^-) le nombre d'exemples $\mathbf{x}_i$ vérifiant $y_i = +1$ (resp. $y_i = -1$) ; le nombre total d'exemples étant N. Si $S = \{s_1, \ldots, s_n\}$ est un ensemble d'entiers distincts, avec $s_1 < s_2 < \ldots < s_n$, on note :

$$\mathbf{x}|_S = [x_{s_1}, x_{s_2}, \ldots, x_{s_n}] \tag{4.10}$$

On notera également $\mathbf{x}|^S$ le vecteur dont la i-ème composante est x_j si $i = s_j$, 0 sinon.

Les centroïdes $\mathbf{m}^+(S)$ et $\mathbf{m}^-(S)$ des deux classes, et le centroïde global $\mathbf{m}(S)$ se calculent selon :

$$\mathbf{m}^+(S) = \frac{1}{N^+} \sum_{i=1, y_i=+1}^{N} \mathbf{x}_i|_S \tag{4.11}$$

$$\mathbf{m}^-(S) = \frac{1}{N^-} \sum_{i=1, y_i=-1}^{N} \mathbf{x}_i|_S \tag{4.12}$$

$$\mathbf{m}(S) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i|_S \tag{4.13}$$

Le rapport entre la dispersion inter-classes B et la dispersion intra-classes W est donné par[3] :

[3] Peeters et Rodet utilisent dans [Pee03] le rapport entre la dispersion inter-classes et la dispersion totale $T = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}_i|_S - \mathbf{m}(S)||^2$. Nous avons choisi d'utiliser le rapport $\frac{B}{W}$ afin de souligner la similarité avec l'analyse discriminante de Fisher. Puisque $T = B + W$, maximiser l'un des critères est équivalent à maximiser l'autre.

$$r(S) = \frac{\frac{N^+}{N}||\mathbf{m}^+(S) - \mathbf{m}(S)||^2 + \frac{N^-}{N}||\mathbf{m}^-(S) - \mathbf{m}(S)||^2}{\frac{1}{N^+}\sum_{i=1,y_i=+1}^{N}||\mathbf{x}_i|_S - \mathbf{m}^+(S)||^2 + \frac{1}{N^-}\sum_{i=1,y_i=-1}^{N}||\mathbf{x}_i|_S - \mathbf{m}^-(S)||} \tag{4.14}$$

Nous observons que dans le cas où S ne contient qu'un attribut, et où les classes sont également représentées, $r(S)$ est égal au critère de Fisher dans la direction associée à cet attribut. Une grande valeur de r assure une bonne discrimination des deux classes.

L'algorithme de maximisation du rapport d'inertie avec projection sur l'espace des attributs – *Inertia Ratio Maximization using Feature Space Projection* (IRMFSP) [Pee03] construit de façon gloutonne un ensemble d'attributs optimal, en deux étapes itérées : une étape rajoutant à l'ensemble des attributs sélectionnés l'attribut c pour lequel le critère de Fisher est maximal, et une étape soustrayant aux attributs restant leur projection sur le sous-espace engendré par les observations de l'attribut nouvellement sélectionné.

Algorithme 1 : IRMFSP

entrées : $\mathbf{x}, \mathbf{y}$, d si il est connu, ϵ sinon
$S \leftarrow \emptyset$
$C \leftarrow \{1, \ldots, D\}$
$i \leftarrow 0$
tant que $i < d$ *(ou, si d n'est pas connu* $\frac{r_i}{r_1} > \epsilon$*)* **faire**
  // Choix de l'attribut au pouvoir discriminant le plus fort
  $s_i \leftarrow \text{argmax}_{c \in C}\, r(\{c\})$
  $r_i \leftarrow \max_{c \in C}\, r(\{c\})$
  $S \leftarrow S \cup s_i$
  $C \leftarrow C \backslash s_i$
  pour $c \in C$ **faire**
    // Projection des attributs restants
    $\mathbf{x}|_{\{c\}} \leftarrow \mathbf{x}|_{\{c\}} - \frac{\mathbf{x}|_{\{c\}} \cdot \mathbf{x}|_{\{s_i\}}}{\mathbf{x}|_{\{s_i\}} \cdot \mathbf{x}|_{\{s_i\}}} \mathbf{x}|_{\{s_i\}}$
  fin
  $i \leftarrow i + 1$
fin
$d \leftarrow i$
sorties : S, $(s_0, \ldots, s_{d-1})$, d

Cette deuxième étape assure que les attributs sélectionnés aux itérations suivantes seront décorrélés avec l'attribut nouvellement sélectionné (et par récurrence, avec tous les attributs sélectionnés jusqu'ici). La soustraction itérative des projections peut être vue en effet comme l'application d'une procédure de Gram-Schmidt pour orthogonaliser les colonnes de la matrice :

$$\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_N]^T \tag{4.15}$$

Dans sa formulation originale [Pee03], le critère d'arrêt utilisé $\frac{r_i}{r_1} > \epsilon$ permet de trouver le nombre optimal d'attributs à utiliser. Dans notre cas, on souhaite simplement obtenir les d meilleurs attributs classés par ordre de pertinence.

4.4.2.2 Sélection d'attributs par l'algorithme RFE-SVM

L'algorithme d'élimination récursive d'attributs par machines à vecteurs de support – *Recursive Feature Elimination with Support Vector Machines* (RFE-SVM), introduit dans [GWBV02], supprime itérativement le ou les attributs dont la contribution à la fonction de décision d'une SVM est minimale.

Soulignons que cet algorithme ne peut utiliser que des SVM linéaires, pour lesquels la contribution d'un attribut $\mathbf{x}_k$ à la fonction de décision est un terme linéaire $w_k\mathbf{x}_k$. Dans les cas où la

Algorithme 2 : RFE-SVM

entrées : $\mathbf{x}, \mathbf{y}, \epsilon$
$R \leftarrow \{1, \ldots, D\}$
tant que $\#R \neq d$ **faire**
  // Apprentissage d'une SVM
  $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \mathbf{x} \cdot \mathbf{x}_i|_R \leftarrow$ C-SVM entraîné sur $(\mathbf{x}_i|_R, y_i)$
  // Calcul des poids
  $\mathbf{w} \leftarrow \sum_{i=1}^{N} \alpha_i \mathbf{x}_i|_R$
  $\mathbf{w} \leftarrow \mathbf{w}|_R$
  // Élimination de l'attribut de poids minimal
  $e \leftarrow \operatorname{argmin}_{\{k \in R\}} w_k^2$
  $R \leftarrow R \backslash \{e\}$
fin
sorties : R

surface de décision est non-linéaire, la pertinence d'un attribut peut dépendre de la région dans laquelle se trouve $\mathbf{x}$, ce qui exclut l'utilisation des SVM non-linéaires à des fins de sélection d'attributs globalement pertinents[4].

L'étape d'apprentissage du C-SVM pouvant être coûteuse en calculs, en particulier pour les itérations initiales où le nombre d'attributs utilisés est grand, plusieurs attributs peuvent être éliminés simultanément en une itération - il s'agit dans ce cas de ceux ayant les poids les plus faibles. Dans nos expériences, nous éliminons 25% des attributs restant à chaque itération, jusqu'à ce que 32 attributs restent. Par la suite, les attributs sont éliminés un par un.

4.4.3 Choix des paramètres de classification et de sélection d'attributs

Nous résumons dans cette section tous les paramètres intervenant dans le processus de sélection des attributs et d'apprentissage (voir annexe annexe B). La valeur optimale de ces paramètres sera sélectionnée par validation croisée, ou plus exactement par une de ses variantes adaptée à la structure de notre base de données.

Sélection d'attributs Les valeurs candidates du nombre d'attributs à sélectionner sont $\mathcal{D}(d) = \{4, 8, 16, 32, 64\}$. Les algorithmes RFE-SVM et IRMFSP sont tous deux considérés.

Paramètre de compromis apprentissage/généralisation C La valeur par défaut fixée dans diverses implémentations logicielles [CL01; Joa98] est :

$$C = \left(\frac{1}{N} \sum_{i=1}^{N} K(\mathbf{x}_i, \mathbf{x}_i) \right)^{-1} \tag{4.17}$$

[4]Dans les cas où la sélection d'attributs est effectuée à des fins explicatives, il est intéressant de connaître les attributs les plus pertinents sur des régions restreintes de l'espace $\mathbb{R}^d$ des attributs. Par exemple, dans les applications Marketing où l'on cherche à prédire quelle marque de soda un consommateur achètera, il est intéressant pour un décideur de connaître quelles variables auront le plus d'influence sur les consommateurs proches de la surface de décision, c'est à dire les plus susceptibles de passer d'une marque à une autre. Dans un travail mené en collaboration avec Ganaël Bascoul [BGL07], nous utilisons des SVM non-linéaires et des régresseurs logistiques à noyaux pour mesurer l'effet d'une variable sur une région $\mathcal{B}$ bordant la surface de décision, en utilisant comme poids :

$$w_k = \int_{\mathcal{B}} \left(\frac{\partial f}{\partial x_k}(\mathbf{x}) \right)^2 d\mathbf{x} \tag{4.16}$$

Afin d'approximer les dérivées partielles $\frac{\partial f}{\partial x_k}$, une approximation polynomiale de la fonction de décision du SVM est utilisée. Ses coefficients sont obtenus par intégration de Monte-Carlo.

Nom	Expression
Produit	$p(y\|\mathbf{x}) = p_1(y\|\mathbf{x})p_2(y\|\mathbf{x})$
Somme pondérée	$p(y\|\mathbf{x}) = \alpha p_1(y\|\mathbf{x}) + (1-\alpha)p_2(y\|\mathbf{x})$
Maximum	$p(y\|\mathbf{x}) = \max\{p_1(y\|\mathbf{x}), p_2(y\|\mathbf{x})\}$
Minimum	$p(y\|\mathbf{x}) = \min\{p_1(y\|\mathbf{x}), p_2(y\|\mathbf{x})\}$
Plus confiant	$p(y\|\mathbf{x}) = \begin{cases} p_1(y\|\mathbf{x}) & \text{si } \|p_1(y\|\mathbf{x}) - 0.5\| > \|p_2(y\|\mathbf{x}) - 0.5\| \\ p_2(y\|\mathbf{x}) & \text{sinon} \end{cases}$

TAB. 4.4 – Opérateurs de fusion

Pour le noyau utilisé (Gaussien), cette valeur est égale à 1. Rien ne garantit cependant que cette valeur empirique est optimale. Une pratique courante consiste à rechercher par validation croisée la valeur de C optimale parmi un ensemble de valeurs exponentiellement espacées. Nous avons retenu pour ce paramètre l'ensemble des valeurs possibles suivantes : $\mathcal{D}(C) = \{2, 16, 128, 1024\}$. Nous n'avons pas inclus dans cet ensemble la valeur $C = 1$ uniquement en raison de limitations de l'implémentation logicielle utilisée : en plusieurs circonstances, même avec une tolérance faible, la procédure d'optimisation ne converge pas, ou ne converge qu'au bout de durées jugées trop longues (de l'ordre de 7h pour certains problèmes, tandis que la résolution pour $C = 2$ prend environ 15 secondes).

Paramètre du noyau gaussien σ Nous avons utilisé un noyau gaussien normalisé par la longueur moyenne du vecteur d'attributs $\mathbf{x}$, qui est ici égale à d (conséquence de la procédure de normalisation des attributs par leur moyenne et écart-type) :

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2d\sigma^2}\right) \tag{4.18}$$

La plage de variation du paramètre σ retenue est $\mathcal{D}(\sigma) = \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\}$. La figure B.6 donnée en annexe montre que des valeurs plus faibles de σ conduisent à un surapprentissage, avec une surfaces de décision entourant exactement chaque exemple d'apprentissage, tandis que des valeurs plus élevées conduisent à des surfaces de décision quasiment linéaires.

4.4.4 Fusion des classifieurs

Comme décrit dans la section 4.1, deux méthodes de fusion sont considérées pour prendre en compte à la fois les attributs extraits du signal original, et ceux extraits du signal dont la piste de batterie a été accentuée.

La fusion précoce consiste à joindre les vecteurs d'attributs extraits des deux signaux. Dans ce cas, c'est la procédure de sélection d'attributs qui effectue la fusion en retenant les attributs les plus fiables à partir de ces deux sources.

La fusion tardive consiste à entraîner deux classifieurs pour chaque jeu d'attributs, et d'agréger les probabilités a posteriori qu'ils fournissent. Les opérateurs de fusion [Blo94] considérés sont donnés dans la table 4.4.

4.5 Du modèle acoustique au modèle de séquence

Le système de transcription de la piste de batterie tel que nous l'avons décrit jusqu'ici n'exploite que l'information contenue dans les paramètres acoustiques, en traitant les observations (frappes) indépendamment les unes des autres.

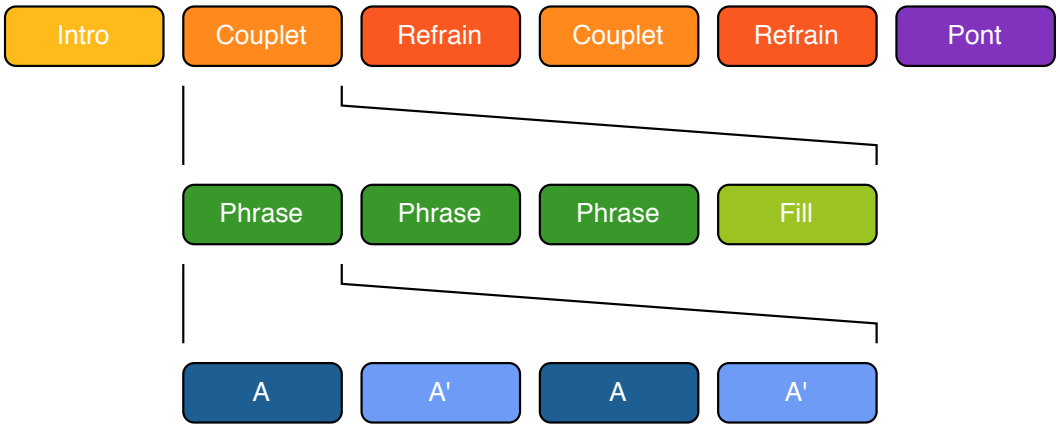


FIG. 4.5 – Exemple de hiérarchie de répétitions dans un accompagnement rythmique

Cependant, de la même façon qu'une succession de phonèmes aléatoires ne constitue pas une phrase syntaxiquement correcte, une succession de frappes de batterie ne constitue pas nécessairement un rythme musicalement intéressant. Par analogie avec les systèmes de reconnaissance vocale qui utilisent à la fois des critères acoustiques, mais aussi un modèle de la langue cible, nous aimerions guider la transcription, ou tout du moins corriger ses erreurs ou ambiguïtés, en tenant compte de certaines spécificités structurelles des rythmes joués à la batterie. Quelques-unes de ces spécificités sont données ici :

Toutes les combinaisons simultanées de sons ne sont pas utilisées Soit ces combinaisons ne sont pas musicalement pertinentes, soit il est impossible à un batteur de les jouer – un batteur pouvant au maximum frapper deux éléments supérieurs (fûts ou cymbales) avec les baguettes, tout en fermant la pédale charleston et frappant la pédale de grosse caisse.

Il existe des motifs rythmiques récurrents, indépendamment du style. Les roulements de toms ou de caisse claire suivis d'une frappe sur la cymbale *crash* sont de tels exemples de *mots rythmiques* utilisés fréquemment dans les séquences de batterie.

Chaque style utilise des mots rythmiques qui lui sont propres. Par exemple, le disco est caractérisé par la présence de la grosse caisse sur tous les temps ; le reggae par la présence de la caisse claire sur le troisième temps. Au sein d'un genre donné, le placement des instruments rythmiques sur chacun des temps est ainsi restreint, donnant lieu à des motifs typiques de durée égale à celle d'une mesure.

Une séquence de batterie est susceptible de contenir des répétitions, sur plusieurs niveaux hiérarchiques. En accompagnement, le rôle de la batterie est de fournir un squelette rythmique stable sur lequel se basent les autres instrumentistes. Il en résulte des répétitions à plusieurs échelles. L'accompagnement peut se construire tout d'abord en assemblant des variations et répétitions d'un motif rythmique de base (typiquement long d'une mesure), donnant lieu à des motifs de type $M = AA'AA'$ ou $M = AAAA'$, où A est un motif élémentaire répété et A' une de ses variations. Au sein d'une section d'un morceau (par exemple, le couplet ou le refrain), plusieurs de ces "paragraphes" rythmiques sont susceptibles d'être répétés. Enfin, à l'échelle d'un morceau entier, le jeu de la batterie pourra suivre l'évolution de la structure du morceau en termes de refrain ou de couplets. On pourrait ainsi avoir, par exemple, un motif $M = AA'AA'$ utilisé au long du refrain,

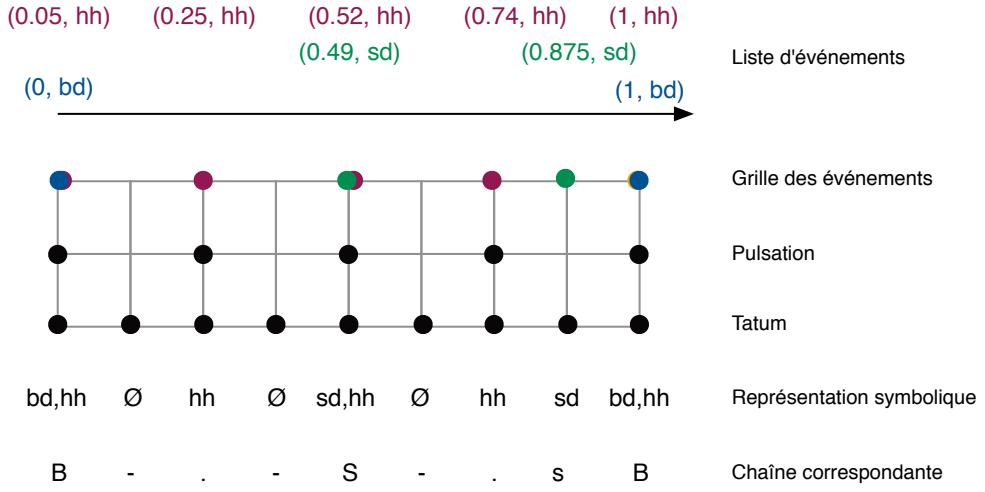


FIG. 4.6 – De la liste d'événements à la représentation symbolique

puis un autre motif $M' = BBBB'$ utilisé pour le couplet. Cette structure hiérarchique, représentée dans la figure 4.5 est plus particulièrement exploitée dans la section 4.5.3.

De telles règles peuvent être prises en compte de deux manières : soit en les incorporant dans un modèle génératif (4.5.2), soit par une procédure d'optimisation modifiant la séquence de façon à maximiser un critère de symétrie et de répétitivité (4.5.3). Ces deux approches ont pour point commun d'opérer sur une représentation symbolique de la séquence, qu'il est d'abord nécessaire de définir.

4.5.1 Extraction d'une représentation symbolique

La détection d'événements effectuée en 4.2 et leur classification en 4.4 produit une représentation de type $(t_i, e_i)_{1 \leq i \leq N}$, où e_i désigne le ou les éléments de la batterie joués à l'instant t_i.

Cette représentation a l'inconvénient de ne pas être synchrone – les instants t_i ne sont pas alignés sur une grille temporelle régulière, et l'intervalle entre deux de ces instants n'est pas constant. De plus, il est possible que deux événements perçus comme simultanés soient représentés sur deux couples consécutifs – par exemple, $(0.502, \text{bd}), (0.500, \text{sd})$ correspond à une frappe simultanée sur la grosse caisse et la caisse claire, mais détectée comme deux événements individuels distants de 2 ms.

Pour résoudre ces deux problèmes, il est d'abord nécessaire d'extraire une base de temps sur laquelle aligner les événements détectés. Il faut ensuite représenter chaque combinaison possible d'événements par un symbole unique, tout en préservant les informations de probabilités fournies par les classifieurs (4.4). Ce procédé est illustré dans la figure 4.6.

Choix d'une base de temps Une base de temps idéale pour l'alignement des événements est le *tatum*. Introduit par Bilmes [Bil93], le tatum peut être défini comme la pulsation qui coïncide avec le plus grand nombre d'événements rythmiques – c'est le plus petit niveau dans la hiérarchie des pulsations rythmiques. Des méthodes d'extraction du tatum à partir d'un signal audio ou d'une liste d'événements sont décrites par Klapuri dans [Kla03], Uhle et Herre dans [UH03] ou Gouyon et al. dans [GHC02]. Nous avons utilisé ici une variante de ces deux dernières méthodes pour estimer la grille de tatum à partir des instants t_i.

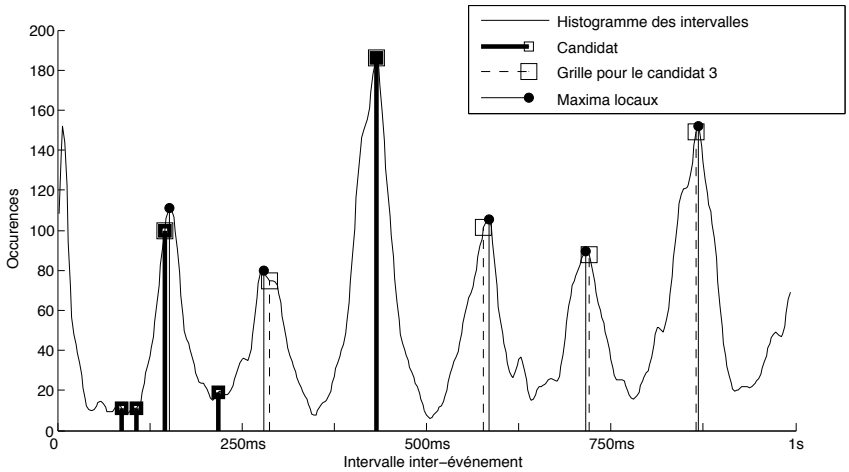


FIG. 4.7 – Extraction du tatum pour un rythme de Blues-Rock ternaire

Deux paramètres interviennent dans cet algorithme : la résolution temporelle q, et la durée maximale considérée entre les événements T. Tout d'abord, un histogramme à T/q classes des valeurs de $t_i - t_j, \forall 1 \leq i < j \leq N$ est extrait. Cet histogramme est lissé par convolution par une fenêtre gaussienne de largeur égale à 9 ms. Les maxima locaux $(m_k)_{1 \leq k \leq K}$ sont extraits de cet histogramme, ainsi que le mode M, correspondant à la durée inter-événement la plus fréquemment rencontrée. Les *tatums* candidats sont les fractions de M, $C_i = (\frac{M}{i})_{1 \leq i \leq 10}$ [5].

Pour chaque candidat C_i, une grille $\mathcal{G}(C_i) = \{kC_i, 1 \leq k \leq [\frac{T}{C_i}]\}$ est générée, et son alignement avec les maxima locaux est mesuré à l'aide de la mesure de non-coïncidence – *Two-Way Mismatch* (TWM) définie comme suit :

$$d(\mathcal{G}(C_i), m) = \sum_k \min_j |m_k - jC_i| + \sum_k \min_j |m_j - kC_i| \tag{4.19}$$

Intuitivement, cette distance pénalise la non-coïncidence entre les multiples entiers du tatum (la grille $\mathcal{G}(C_i)$) et les durées inter-événement les plus fréquentes (les pics m_k de l'histogramme). Le candidat C_i pour lequel $d(\mathcal{G}(C_i), m)$ est minimal est choisi comme tatum. Le tatum τ obtenu par cette procédure est un multiple entier de la résolution q. L'estimation de l'histogramme est d'autant plus robuste que q est grand. En conséquence, un compromis doit être fait entre la robustesse de l'estimation, et la précision à laquelle q sera obtenue. Nous avons choisi ici une résolution $q = 1$ ms, et une durée maximale $T = 1$ s.

Cette procédure est illustrée dans la figure 4.7 pour un rythme de Blues-Rock ternaire à 139 BPM. L'intervalle inter-événement le plus fréquent correspond à 432 ms, soit une pulsation. Les autres intervalles inter-événement les plus fréquents sont représentés par des barres en traits pleins. Les candidats, qui sont des fractions de la pulsation, sont représentés en traits gras. La grille générée pour le troisième de ces candidats, représentée en pointillés, coïncide particulièrement bien avec les maxima locaux – ce troisième candidat s'avère être le tatum.

Une fois le tatum obtenu, Uhle et Herre proposent dans [UH03] de quantifier les événements rythmiques sur une grille $G(\phi) = \{\phi + i\tau, 0 \leq i \leq \frac{L}{\tau}\}$, L étant la durée totale de la séquence. Le paramètre de phase ϕ est à estimer, il est choisi en sorte à minimiser la TWM entre les événements à quantifier et la grille. Cette solution n'est satisfaisante que sur de courts extraits – pour des extraits plus longs, l'erreur d'estimation, de l'ordre de q se propage. On observe typiquement des décalages

[5] Dans [GHC02] ne sont considérées que les fractions $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{9}$ correspondant aux divisions binaires et ternaires les plus couramment rencontrées en musique

entre la grille de tatum et les événements au bout d'une durée de l'ordre de $\frac{\tau^2}{2q}$. Par ailleurs, même lorsque le tatum est constant – autrement dit, quand la structure métrique et le tempo du morceau ne changent pas – il peut être nécessaire d'ajuster la grille pour tenir compte d'un éventuel "décrochage" du batteur, ou plus fréquemment du *swing* – déséquilibre entre les durées de chacune des pulsations. Pour adapter la grille de tatum à ces variations, nous proposons l'algorithme de suivi détaillé dans l'algorithme 3. Après une phase d'initialisation où la phase ϕ_0 est estimée sur une première fenêtre de longueur W, la grille est construite par groupe de K événements. Pour chacun des groupes, la phase est réajustée par un décalage dans l'intervalle $[(\alpha-1)\tau, (1-\alpha)\tau]$, de manière à maximiser la coïncidence entre les événements observés et la grille. Nous avons ici utilisé $K = 4$ et $\alpha = 0.97$.

Algorithme 3 : Grille de tatum flexible

entrées : $(t_i)_{1\leq i\leq N}, \tau, K, \alpha$
$W \leftarrow \frac{\tau^2}{2q}$
$obs \leftarrow \{t_i, 0 \leq t_i \leq W\}$
$\phi_0 \leftarrow \text{argmin}_{\phi\in[0,\tau]} TWM(obs, \{\phi + i\tau, 0 \leq i \leq \left[\frac{W}{\tau}\right]\})$
$grille \leftarrow \emptyset$
$courant \leftarrow \phi_0$
tant que $courant < \max t_i$ **faire**
  $grille \leftarrow grille \cup \{courant + k\tau, 0 \leq k < K-1\}$
  $dernier \leftarrow courant + (K-1)\tau$
  $obs \leftarrow \{t_i, dernier + \tau \leq t_i \leq dernier + W + \tau\}$
  $decalage \leftarrow \text{argmin}_{\beta\in[\alpha,2-\alpha]} TWM(obs, \{dernier + \beta\tau + k\tau, 0 \leq k < \left[\frac{W}{q}\right]\})$
  $courant \leftarrow dernier + \tau decalage$
fin
sorties : grille

Un exemple sur une séquence d'accompagnement de Twist est donné dans la figure 4.8. Au début du morceau (colonne de gauche), la grille rigide (en haut) $\phi_0 + i\tau$ et la grille flexible coïncident avec les événements détectés. Sur le milieu du morceau (colonne de droite), la grille rigide est déphasée par rapport aux événements, à cause de la propagation de l'erreur d'estimation de τ. La grille flexible coïncide toujours.

L'alphabet rythmique $\mathcal{A}$ Si l'on se restreint aux trois classes d'instruments suivantes : grosse caisse (bd), caisse claire (sd) et hi-hat (hh), chaque combinaison d'événements possible à un instant donné peut être représentée par un unique symbole $s \in \mathcal{A}$, ou par un vecteur à 3 composantes, appelé l'indicatrice I.

Alignement temporel et agrégation des probabilités On souhaite représenter la séquence rythmique sous la forme d'une suite de symboles s_n, où le symbole s_n désigne la combinaison d'instruments rythmiques jouée au n-ième point de la grille de tatum τ_n. Chaque symbole s_n est vu comme la réalisation d'une variable aléatoire S_n. On s'intéresse tout d'abord au calcul de $P(S_n = s)$, où $s \in \mathcal{A}$ est un symbole rythmique, à partir de la sortie du système de classification décrit dans les sections précédentes. La sortie de ce système consiste en une suite d'instants (t_i) et de probabilités a posteriori (π_{ij}), où $\pi_{ij} = p(y_{ij} = +1|\mathbf{x}_i)$ est la probabilité que l'instrument j ait été joué à l'instant t_i. Notons $\bar{\pi}_{ij} = 1 - \pi_{ij}$ la probabilité que l'instrument j n'ait pas été joué à l'instant t_i.

Une première étape consiste à associer à chaque instant t_i son plus proche voisin sur la grille de tatum τ_n. On définit ainsi T_n comme étant l'ensemble des indices des événements t_i dont le plus proche voisin est le noeud τ_n, autrement dit $T_n = \{i, n = \text{argmin}_k |\tau_k - t_i|\}$. T_n décrit, intuitivement, quels onsets seront quantifiés en τ_n.

Soit $s \in \mathcal{A}$ un symbole rythmique, d'indicatrice I. À partir du résultat produit par le système de classification, nous pouvons alors calculer $P(S_n = s|t, \pi)$:

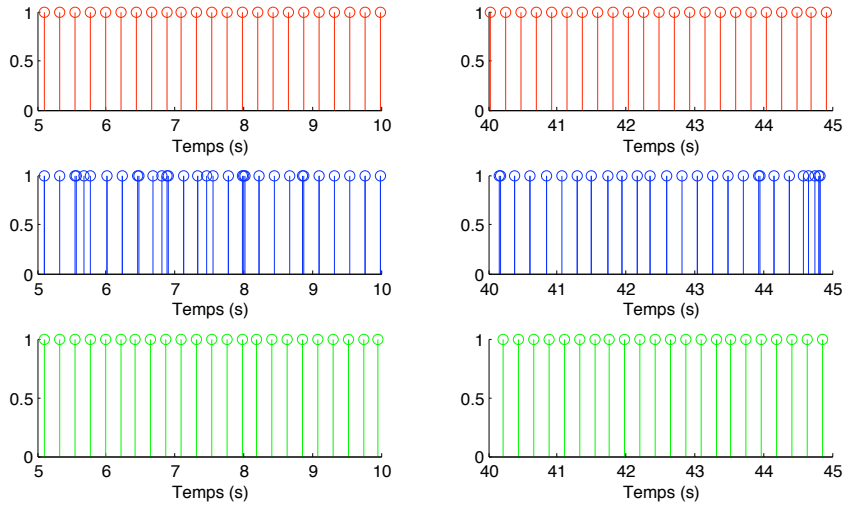


FIG. 4.8 – De haut en bas : grille de tatum rigide, position des événements, grille de tatum flexible extraite par l'algorithme 3. À gauche, au début du morceau, à droite, en milieu de morceau.

Combinaison de frappes	Symbole s	Indicatrice I
$\emptyset$	-	[0 0 0]
$\{hh\}$	.	[0 0 1]
$\{sd\}$	s	[0 1 0]
$\{sd, hh\}$	S	[0 1 1]
$\{bd\}$	b	[1 0 0]
$\{bd, hh\}$	B	[1 0 1]
$\{bd, sd\}$	d	[1 1 0]
$\{bd, sd, hh\}$	D	[1 1 1]

TAB. 4.5 – Symboles associés aux combinaisons de frappes

$$P(S_n = s|t, \pi) = \prod_{j \in \{0,1,2\}} \begin{cases} 1 - \prod_{i \in T_n} 1 - \pi_{ij} & \text{si } I_j = 1 \\ \prod_{i \in T_n} 1 - \pi_{ij} & \text{si } I_j = 0 \end{cases} \tag{4.20}$$

Par exemple, la probabilité que le symbole B, dénotant la combinaison $\{bd, hh\}$, ait été joué à l'instant τ_n est calculée comme la probabilité qu'au moins une frappe de grosse caisse et une frappe de hi-hat aient été jouées dans l'intervalle de temps lié à τ_n, et qu'aucune frappe de caisse claire n'ait été jouée dans ce même intervalle.

Les deux sections suivantes proposent deux approches pour modéliser les séquences s_n. En conciliant de tels modèles avec les informations fournies par le classifieur sous la forme des probabilités $P(S_n = s|t, \pi)$, nous espérons améliorer la qualité de la transcription.

4.5.2 Une approche supervisée : Modèles à N-grammes et ses variantes

4.5.2.1 Présentation des modèles

Modèle à N-grammes classique Nous faisons ici l'hypothèse qu'il existe une dépendance entre les symboles consécutifs s_n des séquences à transcrire. Plus précisément, les séquences de symboles vérifient la propriété de Markov[6] d'ordre $N-1$:

$$P(s_n|s_{n-1}\dots s_{n-N+1}) = P(s_n|s_{n-1}\dots s_1) \tag{4.21}$$

Un symbole est ainsi déterminé conditionnellement à ses $N-1$ symboles précédents. La probabilité d'observer une séquence $(s_n)_{1\leq n\leq L}$ est donc égale à :

$$P(s) = \prod_{1\leq n\leq L} P(s_n|s_{n-1}\dots s_{n-N+1}) \tag{4.22}$$

Nous constatons ainsi que le modèle est déterminé par $(A+1)^N$ probabilités, qui correspondent aux probabilités de retrouver chacun des A symboles de $\mathcal{A}$ dans un contexte à gauche de longueur $N-1$ donné. La croissance exponentielle du nombre de paramètres du modèle avec l'ordre N restreint dans la pratique le choix de N, qui dépasse rarement 4.

De tels modèles sont couramment utilisés en reconnaissance de la parole ou en analyse syntaxique partielle (Shallow Parsing – [Mér95]). Nous les avons appliqués avec succès à la transcription de séquences de Tabla dans [GR03], et de boucles de batterie dans [GR04]. L'intérêt de ces modèles provient de leur capacité à modéliser des dépendances à court terme entre symboles – dépendances dues à la présence de motifs stéréotypés comme des roulements de toms, ou des phénomènes comme l'alternance entre frappes de grosse caisse et de caisse claire.

Modèle à N-grammes périodiques Ce modèle introduit par Paulus et Klapuri dans [PK03a] vise à prendre en compte le caractère répétitif des motifs rythmiques à l'échelle d'une mesure. Il consiste à introduire une dépendance non plus entre des symboles consécutifs, mais entre des symboles distants de M où M est la durée d'une mesure. Ainsi, à l'ordre N, l'expression de la probabilité d'observer une séquence $(s_n)_{1\leq n\leq L}$ se calcule comme :

$$P(s) = \prod_{1\leq n\leq L} P(s_n|s_{n-M}\dots s_{n-(N-1)M}) \tag{4.23}$$

Généralisation des N-grammes Nous nous proposons de généraliser ces approches pour inclure des informations rythmiques à diverses échelles. Un modèle à $N+1$-grammes généralisé est défini par une suite finie strictement croissante de N entiers positifs $\mathcal{S}$ que nous appellerons support. De manière intuitive, le support définit le "crible" au travers duquel nous observons les symboles précédents. Selon ce modèle, la dépendance entre les symboles consécutifs vérifie la propriété de Markov à l'ordre $\mathcal{S}_N$, ainsi que la propriété suivante plus forte :

$$P(s_n|s_{n-\mathcal{S}_1}\dots s_{n-\mathcal{S}_N}) = P(s_n|s_{n-\mathcal{S}_1}\dots s_{n-\mathcal{S}_N}s_{n-\mathcal{S}_N-1}\dots s_1) \tag{4.24}$$

La probabilité d'observer une séquence $(s_n)_{1\leq n\leq L}$ est ainsi :

$$P(s) = \prod_{1\leq n\leq L} P(s_n|s_{n-\mathcal{S}_1}\dots s_{n-\mathcal{S}_N}) \tag{4.25}$$

Les N-grammes classiques sont un cas particulier avec $\mathcal{S} = (1, 2, \dots, N-1)$; les N-grammes périodiques sont un cas particulier avec $\mathcal{S} = (M, 2M, \dots, (N-1)M)$. Le choix de $\mathcal{S}$ permet de réaliser un compromis entre l'horizon d'observation et le nombre de probabilités à estimer. Par

[6] Il s'agit d'une notation simplifiée, qui devrait être plus formellement $P(S_n = s_n|S_{n-1} = s_{n-1}\dots S_{n-N+1} = s_{n-N+1}) = P(S_n = s_n|S_{n-1} = s_{n-1}\dots S_1 = s_1)$

exemple, dans le cas où le tatum correspond à une double croche, avec une mesure $\binom{4}{4}$, le choix $\mathcal{S} = (1, 4, 16)$ permet l'apprentissage de dépendances au niveau de la mesure, de la pulsation, et des symboles successifs, tout en limitant le nombre de probabilités à estimer à $(A + 1)^4$.

4.5.2.2 Apprentissage

La procédure d'apprentissage consiste en l'estimation des probabilités d'observer un symbole s_n connaissant son contexte. Ces probabilités peuvent être estimées par comptage à partir d'un corpus de séquences. Dans le cas des N-grammes classiques, on a par exemple :

$$\hat{P}(s_n|s_{n-1} \dots s_{n-N+1}) = \frac{C(s_{n-N+1} \dots s_{n-1}s_n)}{C(s_{n-N+1} \dots s_{n-1})} \tag{4.26}$$

Où $C(abc)$ désigne le nombre d'occurrences de la sous-séquence abc dans le corpus d'apprentissage.

Dans le cas des N-grammes, généralisés, on a :

$$\hat{P}(s_n|s_{n-\mathcal{S}_1} \dots s_{n-\mathcal{S}_N}) = \frac{C_{\mathcal{S}}(s_{n-\mathcal{S}_N} \dots s_{n-\mathcal{S}_1}s_n)}{\sum_{a\in\mathcal{A}} C_{\mathcal{S}}(s_{n-\mathcal{S}_N} \dots s_{n-\mathcal{S}_1}a)} \tag{4.27}$$

$C_{\mathcal{S}}(c_1 \dots c_N a)$ désigne une opération de comptage comptant les sous-séquences vues au travers du crible défini par $\mathcal{S}$. Plus précisément, $C_{\mathcal{S}}(c_1 \dots c_N a)$ compte dans le corpus d'apprentissage le nombre de sous-séquences de la forme $s_1 \dots s_{\mathcal{S}_N} a$ vérifiant $s_{\mathcal{S}_N+1-\mathcal{S}_n} = c_{N+1-n}, \forall 1 \leq n \leq N$.

Nous simplifierons par la suite cette expression en l'écrivant :

$$\hat{P}(e|\text{txetnoc}) = \frac{C_{\mathcal{S}}(\text{context } e)}{\sum_{a\in\mathcal{A}} C_{\mathcal{S}}(\text{context } a)} \tag{4.28}$$

Cette estimateur simple affecte une probabilité nulle aux sous-séquences absentes du corpus, et des estimations imprécises aux sous-séquences peu fréquentes. Des solutions typiques à ce problème consistent :

- À supposer que le corpus contient au moins un exemplaire de chaque sous-séquence, et à normaliser les probabilités en conséquence (Lissage de Laplace).
- À faire intervenir un terme d'ordre inférieur en écrivant :

$$\hat{P}_{smooth}(s_n|s_{n-\mathcal{S}_1} \dots s_{n-\mathcal{S}_N}) = (1-\alpha)\hat{P}(s_n|s_{n-\mathcal{S}_1} \dots s_{n-\mathcal{S}_N}) + \alpha\hat{P}(s_n|s_{n-\mathcal{S}_1} \dots s_{n-\mathcal{S}_{N-1}}) \tag{4.29}$$

(Ou, plus familièrement $\hat{P}_{smooth}(e|\text{txetnoc}) = (1 - \alpha)\hat{P}(e|\text{txetnoc}) + \alpha\hat{P}(e|\text{txetno})$)

Dans le cas du lissage de Witten-Bell [WB91] le coefficient α prend la forme :

$$\alpha = 1 - \frac{\#\{a \in \mathcal{A}, C_{\mathcal{S}}(\text{context } a) > 0\}}{\#\{a \in \mathcal{A}, C_{\mathcal{S}}(\text{context } a) > 0\} + \sum_{a\in\mathcal{A}} C_{\mathcal{S}}(\text{context } a)} \tag{4.30}$$

C'est cette méthode de lissage, qui a précédemment été utilisée dans [PK03a], que nous avons retenue.

4.5.2.3 Reconnaissance

On se propose de déterminer la séquence de symboles s la plus probable connaissant les instants t et les probabilités π issues des phases de détection des événements et de classification, et un modèle $N + 1$-grammes généralisé de support $\mathcal{S}$ de la séquence :

$$\underset{s}{\text{argmax}} \prod_{1\leq n\leq L} P(S_n = s_n|t, \pi)P(S_n = s_n|S_{n-\mathcal{S}_1} = s_{n-\mathcal{S}_1} \dots S_{n-\mathcal{S}_N} = s_{n-\mathcal{S}_N}) \tag{4.31}$$

L'espace de recherche comporte A^L séquences, rendant une exploration de toutes les combinaisons impossible. Il est cependant possible de construire la séquence optimale de proche en proche par un algorithme de programmation dynamique : l'algorithme de Viterbi [For73] que l'on présente rapidement ici.

Algorithme de Viterbi dans le cas des bigrammes Supposons que l'on connaisse pour un instant n donné et pour tout symbole rythmique b la sous-séquence $s_n^*(b)$ la plus probable, se terminant par le symbole rythmique b à l'instant n. On appelle $H_n(b)$ sa probabilité. Il est alors possible d'exprimer $H_{n+1}(a), \forall a \in \mathcal{A}$:

$$H_{n+1}(a) = \max_{b \in \mathcal{A}} [H_n(b) P(S_{n+1} = a | S_n = s_n^*(b))] P(S_{n+1} = a | t, \pi) \tag{4.32}$$

De la même façon, les sous-séquences les plus-probables peuvent être étendues par :

$$s_{n+1}^*(a) = \operatorname*{argmax}_{b \in \mathcal{A}} H_n(b) P(S_{n+1} = a | S_n = s_n^*(b)) \tag{4.33}$$

La séquence la plus probable est finalement $s_L^*(a^*)$ où $a^* = \operatorname{argmax}_a H_L(a)$. La complexité de cet algorithme est $\mathcal{O}(LA^2)$. Notons que les premiers éléments de la séquence la plus probable ne sont connus qu'à la fin de cette opération de décodage – cet algorithme n'est donc pas causal.

Application aux N-grammes généralisés L'algorithme précédent peut être adapté en :

$$H_{n+1}(a) = \max_{b \in \mathcal{A}} [H_n(b) P(S_{n+1} = a | S_{n+1-\mathcal{S}_1} = s_{n+1-\mathcal{S}_1}^*(b) \dots S_{n+1-\mathcal{S}_N} = s_{n+1-\mathcal{S}_N}^*(b))] P(S_{n+1} = a | t, \pi)$$

$$s_{n+1}^*(a) = \operatorname*{argmax}_{b \in \mathcal{A}} H_n(b) P(S_{n+1} = a | S_{n+1-\mathcal{S}_1} = s_{n+1-\mathcal{S}_1}^*(b) \dots S_{n+1-\mathcal{S}_N} = s_{n+1-\mathcal{S}_N}^*(b))$$

La complexité est toujours $\mathcal{O}(LA^2)$, mais il n'est pas garanti que la séquence optimale soit trouvée – l'algorithme de Viterbi exige en effet que la séquence vérifie une propriété de Markov d'ordre 1, ce qui n'est pas le cas ici. Cependant, on observe que si l'on note $\bar{s}_n^{\mathcal{S}_N} = s_{n-\mathcal{S}_n} \dots s_{n-1}$, la séquence $(\bar{s}_n^{\mathcal{S}_N})$ vérifie la propriété de Markov d'ordre 1. Il est ainsi possible d'utiliser l'algorithme de Viterbi pour trouver la séquence $(\bar{s}_n^{\mathcal{S}_N})$ optimale, et d'en déduire la séquence (s_n) optimale correspondante. La complexité de cette approche "un état par contexte" est $\mathcal{O}(LA^{\mathcal{S}_N+1})$. Cette approche peut donc s'avérer prohibitive pour de longs contextes d'observation.

Décision gloutonne Nous pouvons effectuer également une recherche gloutonne, de proche en proche, de la séquence optimale :

$$s_{n+1}^* = \operatorname*{argmax}_{a \in \mathcal{A}} P(S_{n+1} = a | S_{n+1-\mathcal{S}_1} = s_{n+1-\mathcal{S}_1}^* \dots S_{n+1-\mathcal{S}_N} = s_{n+1-\mathcal{S}_N}^*)] P(S_{n+1} = a | t, \pi) \tag{4.34}$$

Bien qu'elle ne produit pas toujours la séquence de probabilité maximale, cette approche possède deux avantages : sa complexité en $\mathcal{O}(LA)$, et sa causalité, essentielle dans des applications de type contrôle d'instrument MIDI ou suivi de partition. Dans notre cas, les temps de calcul requis par une recherche de Viterbi complète sont négligeables (moins d'une seconde pour une séquence de 250 symboles) ; et nous n'avons aucune contrainte de causalité. L'utilité de la méthode gloutonne est donc limitée. Elle a néanmoins été utilisée dans [PK03a].

4.5.2.4 Du supervisé au non-supervisé : Qu'apprendre ?

La procédure d'apprentissage décrite en 4.5.2.2 nécessite un corpus de séquences de référence. Nous abordons ici la question du choix du corpus. S'il est déjà possible d'affirmer que ce corpus doit être le plus volumineux possible, de manière à garantir la robustesse de l'estimation des probabilités, le choix de son contenu déterminera les connaissances musicales apprises ou modélisées par le modèle de séquence. Dès lors, qu'apprendre ? Plusieurs options sont détaillées ici.

Modèle générique. Une première possibilité consiste à utiliser comme corpus d'apprentissage un ensemble de séquences hétérogènes de différents styles de jeu, issues de différents batteurs. Cette méthode est la plus facile à mettre en oeuvre – le modèle est appris une fois pour toutes, et peut être appliqué à des données inconnues. Cependant, on peut s'interroger quant à l'utilité d'une telle méthode. Quel serait le pouvoir prédictif – ou les connaissances apprises – d'un modèle entraîné sur des séquences aussi variées ? Nous allons par la suite tenter de répondre quantitativement à cette question.

Modèle générique par batteur. Pour certaines applications (enseignement de la batterie assisté par ordinateur, contrôle d'instrument MIDI), on pourrait envisager d'utiliser un ensemble de séquences de référence, de styles hétérogènes, jouées par le même batteur que les séquences à reconnaître. Cette méthode permettrait de modéliser les stéréotypes de jeu du musicien, ainsi que son degré de maîtrise technique (les seules successions de frappes qu'il lui est possible de jouer par exemple). Elle est cependant peu pratique à mettre en oeuvre, puisque chaque utilisateur du système de transcription devra d'abord jouer ou annoter des séquences de référence.

Modèle par style. Une approche plus intéressante consisterait à classer les séquences de la base d'apprentissage selon leurs styles, et à apprendre un modèle de séquence distinct pour chacun de ces N styles. Dans ce cas, la procédure de reconnaissance consiste à calculer en parallèle, pour chacun des N modèles, la séquence optimale et la vraisemblance du modèle associé, puis à choisir parmi les N séquences celle produite par le modèle de vraisemblance maximale. Notons que cette méthode effectue, comme sous-produit, une classification par style de la séquence qui a été jouée.

Modèle par style avec oracle. L'apprentissage est effectué de la même façon que précédemment, produisant N modèles de séquence par style. L'étape de reconnaissance consiste à identifier a priori le style de la séquence, par un classifieur qu'on suppose parfait (par exemple un utilisateur expert humain), puis à effectuer la reconnaissance avec le modèle de séquence correspondant au style reconnu.

Modèle "oracle" de la séquence à transcrire. Si la séquence qui doit être transcrite est connue à l'avance, on peut apprendre un modèle de séquence spécifique à cette séquence. En dehors des applications de suivi de partition ou d'accompagnement existant, cette méthode n'a aucun intérêt pratique. Elle permet en revanche d'illustrer les limites des modèles de séquence, en évaluant leurs performances dans une situation idéale.

Modèle local. Une variante de la méthode précédente utilisable dans la pratique consiste à :

1. Effectuer la reconnaissance sans modèle de séquence, ou avec un modèle de séquence générique.
2. Apprendre le modèle de séquence sur la séquence reconnue. On suppose ici que les erreurs introduites par la transcription sont indépendantes du contexte, autrement dit que les probabilités estimées sur la séquence erronée, issue de la transcription, sont suffisamment proches de celles qui auraient été estimées sur la séquence correcte.
3. Utiliser un tel modèle *local* pour la reconnaissance.
4. Eventuellement, itérer les deux étapes précédentes.

Nous nous proposons maintenant d'évaluer le pouvoir prédictif des modèles de séquence appris selon chacune de ces approches. La mesure que nous retenons est l'information mutuelle entre un symbole et son contexte.

$$I(\text{context}\,, e) = \sum_{\text{context}\in\mathcal{A}^{N-1}} \sum_{e\in\mathcal{A}} P(\text{context}\; e) \log_A \frac{P(\text{context}\; e)}{P(\text{context})P(e)} \tag{4.35}$$

En remarquant que $I(\text{context}, e) = H(e) - H(e|\text{context})$, l'information mutuelle mesure, à une constante additive près, la certitude avec laquelle un symbole est déterminé, connaissant son

Support	Corp. universel	Corp. par batteur	Corp. par style	Séq. individuelles
Bigrammes généralisés				
-1	0.026	0.083	0.134	0.171
-2	0.084	0.128	0.187	0.208
-4	0.106	0.150	0.192	0.209
-8	**0.153**	**0.193**	**0.215**	**0.226**
-16	0.144	0.182	0.206	0.216
Trigrammes généralisés				
-2,-1	0.153	0.237	0.357	0.405
-4,-1	0.157	0.237	0.347	0.396
-8,-1	0.192	0.262	0.359	0.403
-16,-1	0.185	0.254	0.348	0.391
-4,-2	0.179	0.253	0.356	0.398
-8,-2	0.204	0.265	0.353	0.390
-16,-2	0.213	0.273	**0.370**	**0.407**
-8,-4	0.219	0.279	0.354	0.392
-16,-4	0.196	0.254	0.344	0.380
-16,-8	**0.229**	**0.283**	0.348	0.379
-32,-16	0.208	0.264	0.325	0.361
Quadrigrammes généralisés				
-3,-2,-1	0.281	0.414	0.523	0.552
-4,-2,-1	0.297	**0.429**	0.528	0.555
-8,-2,-1	0.307	**0.429**	**0.531**	**0.558**
-16,-8,-1	0.311	0.423	0.517	0.546
-8,-4,-2	0.318	0.428	0.515	0.540
-16,-4,-2	0.308	0.418	0.525	0.551
-16,-8,-2	**0.322**	0.423	0.514	0.541
-16,-8,-4	0.312	0.408	0.500	0.526
-48,-32,-16	0.309	0.403	0.470	0.504

TAB. 4.6 – Pouvoir prédictif du modèle de séquence, mesuré par l'information mutuelle entre un symbole et son contexte $I(\text{context } e)$, pour divers corpus et divers supports

contexte. Une information mutuelle nulle implique que le contexte d'apparition d'un symbole n'a aucun pouvoir prédictif sur ce symbole.

Les résultats sont donnés dans la table 4.6. Ils montrent d'abord l'apport (modeste) des modèles par batteur, par rapport à un modèle universel. Ces modèles ont cependant un pouvoir prédictif plus faible que les modèles par style, plus faciles à mettre en oeuvre – nous ne considérerons donc pas par la suite, dans nos expériences, les modèles par batteur. Ces résultats illustrent également l'intérêt limité des modèles de séquences individuelles par rapport aux modèles par style. Cela suggère que les séquences jouées selon un style donné s'y conforment totalement, et offrent peu de possibilités de variation – la distribution des N-grammes estimée sur la séquence semble donc déterminée par le style. Ces résultats montrent enfin l'intérêt des N-grammes généralisés : les modèles les plus informatifs ne sont ni des modèles de N-grammes classiques, ni des modèles de N-grammes périodiques, mais tiennent compte à la fois des dépendances à court et long terme. Soulignons pour terminer que ces résultats ne mesurent qu'un critère d'information issu du modèle appris, et non le gain de performances réel qu'il apporte pour la tâche de transcription musicale. Le gain de performances dépend, en outre, de la qualité de l'estimation des probabilités, ainsi que de la véracité des informations fournies par le modèle acoustique (*Garbage In, Garbage Out*).

4.5.3 Une approche non supervisée : Correction d'erreur par minimisation de la complexité

L'approche que nous venons de décrire souffre de deux inconvénients. D'une part, elle nécessite l'apprentissage d'un modèle – tâche pour laquelle, comme nous venons de le voir, un compromis entre généricité, et pouvoir prédictif doit être trouvé. D'autre part, la prise en compte de dépendances à des échelles variées (temps, mesure, section du morceau), si elle est rendue possible par l'emploi de N-grammes généralisés, n'en est pas moins limitée : le choix du support présuppose la connaissance de la durée d'une mesure, et d'une section, et lorsque N est grand, la qualité des estimations des probabilités diminue. Ces défauts suggèrent une approche entièrement non-supervisée, ne reposant pas sur un modèle statistique des séquences. Les observations la motivant sont les suivantes :

- Les séquences que nous voulons transcrire sont régulières, répétitives, et peuvent être décrites efficacement par des représentations hiérarchiques comme celles illustrées dans la figure 4.5.
- Le produit brut de la transcription ne possède qu'approximativement de telles régularités – deux répétitions d'un même segment pourront être transcrites avec des erreurs différentes, les rendant non semblables.

Il apparaît dès lors qu'un moyen d'améliorer les scores de reconnaissance consisterait à corriger la transcription de manière à rétablir sa "symétrie" – sa capacité à être décrite en termes de structures répétitives simples. Nous nous intéresserons ainsi, tout d'abord, à la définition d'un critère de complexité mesurant le caractère "asymétrique" (au sens qui vient d'être défini) d'une séquence. Nous détaillerons ensuite une procédure de correction cherchant à le minimiser, tout en restant compatible avec les indications fournies par le modèle acoustique.

4.5.3.1 Un critère de complexité pour les séquences rythmiques

La complexité de Kolmogorov d'une séquence S est définie comme la longueur du plus court programme, représenté avec un alphabet binaire, dans un modèle de calcul abstrait donné (machine de Turing par exemple), générant S. Ce plus court programme fournit ainsi une description minimale de S, et sa longueur fournit donc une mesure absolue de la quantité d'information contenue dans S. Cette grandeur n'est pas calculable, elle peut cependant être approximée à l'aide d'un algorithme de compression – dans ce cas, le plus court programme générant S est la version compressée de S, suivie du programme la décompressant.

De telles mesures de complexité ont été utilisées pour des applications musicales, par exemple dans [CVW04] ou [LS05] pour mesurer la similarité entre mélodies ; ou dans [MW06] pour discriminer la mélodie principale (considérée de complexité maximale) d'une oeuvre polyphonique, par rapport à l'accompagnement. Toutes font appel à des variantes des algorithmes de compression LZ77 ou LZ78 [ZL78] pour approximer la complexité de Kolmogorov.

Nous nous proposons ici d'utiliser un autre algorithme de compression pour mesurer la complexité des séquences : l'algorithme SEQUITUR [NMW97]. Trois raisons motivent ce choix. Tout d'abord, SEQUITUR s'est montré plus efficace que l'agorithme LZ78 pour diverses tâches de compression de texte [NMWM94] – et fournit ainsi une meilleure approximation de la description minimale d'une séquence. Ensuite, cet algorithme infère, à partir de la séquence à compresser non pas un dictionnaire de préfixes fréquents (comme c'est le cas avec l'algorithme LZ78), mais une grammaire hors-contexte. Il est ainsi possible de prendre en compte des structures récursives et hiérarchiques comme celles présentées dans la figure 4.5. Enfin, l'algorithme est susceptible d'être modifié pour inclure, dans les grammaires inférées, des opérateurs spécifiques au type de données à traiter – par exemple des opérateurs d'inversion ou de transposition (pour la musique), ou de complémentation des bases (pour les séquences d'ADN) [EL03].

Nous rappelons ici brièvement le principe de l'algorithme SEQUITUR.

Inférence en ligne d'une grammaire hors-contexte à partir d'une séquence. L'algorithme SEQUITUR traite séquentiellement (c'est à dire en ligne, symbole par symbole) la séquence

à compresser et met à jour sa représentation sous forme de grammaire G de manière à vérifier les deux propriétés suivantes :

Unicité des bigrammes. Un bigramme ne doit pas apparaître plus d'une fois dans les membres de droite des productions de G. Deux cas peuvent se présenter :

Création d'une nouvelle production. Dans le cas où la grammaire contient les productions $A \to XabY$ et $B \to ZabT$, une nouvelle production $C \to ab$ est créée, et les productions originales sont modifiées en $A \to XCY$ et $B \to ZCT$.

Réutilisation d'une production existante. Dans le cas où la grammaire contient les productions $A \to XabY$ et $B \to ab$, la première production est modifiée en $A \to XBY$.

Utilité des règles de production. Chaque production doit être utilisée au moins deux fois. Ainsi, si la grammaire contient $A \to XBY$ et $B \to ZT$, et que le non-terminal B apparaît uniquement dans la première production, la deuxième production est supprimée et la première devient $A \to XZTY$.

Un exemple sur la séquence $abcbcabcbc$ est donné dans la table 4.7.

Une structure de données efficace permet de représenter les règles de production et l'index recensant l'utilisation des bigrammes dans chacune des règles [NMW97]. Cela assure à l'algorithme SEQUITUR une complexité linéaire en la longueur de la séquence à traiter.

Dans le cas où l'on veut permettre l'inférence de règles de production de type $A \to \varphi(B)C$, où φ désigne une transformation bijective de B (transposition, substitution de symboles) préservant sa longueur, l'algorithme SEQUITUR peut toujours être utilisé pour inférer la grammaire. La modification consiste à remplacer la règle d'unicité des bigrammes, par une règle d'unicité des bigrammes sous l'action de φ : pour tous bigrammes ab et cd apparaissant dans les membres de droite des productions de G, on doit avoir $\varphi(ab) \neq cd$. Dans le cas où cette contrainte est violée, la grammaire est modifiée comme suit :

$$\begin{array}{lcl} A & \to & XabY \\ B & \to & ZcdT \end{array} \implies \begin{array}{lcl} A & \to & XCY \\ B & \to & Z\varphi(C)T \\ C & \to & ab \end{array} \tag{4.36}$$

Il n'existe pas, dans le cas général, d'implémentation efficace de cet algorithme. Cependant, dans les cas simples où $\varphi(xy) = \varphi(x)\varphi(y)$ (resp. $\varphi(xy) = \varphi(y)\varphi(x)$), l'implémentation efficace de [NMW97] est toujours valide. Dans ce cas, à chaque fois qu'un bigramme xy entre dans l'index, on stocke également dans l'index le bigramme $\varphi(x)\varphi(y)$ (resp. $\varphi(y)\varphi(x)$). Dans un cadre plus général où plusieurs transformations $(\varphi_i)_{i\in\{1,\ldots,N\}}$ sont considérées, chacune pouvant être itérée, on stocke dans l'index les bigrammes correspondant à toutes les transformations dans le groupe engendré par les $(\varphi_i)_{i\in\{1,\ldots,N\}}$. Quelques exemples sont donnés dans la table 4.8.

Dans le cadre des applications musicales traitant des séquences mélodiques monophoniques, les opérateurs intéressants à considérer pourraient être la transposition ou le renversement de séquence. Un autre exemple intéressant d'utilisation de tels opérateurs pour l'inférence de grammaire concerne le Tabla [GR03], où les frappes peuvent être sourdes (jouées avec la paume de la main) ou résonnantes (jouées avec le doigt), et où les compositions peuvent présenter des répétitions d'une même séquence où toutes les frappes sourdes sont remplacées par des frappes résonnantes (et vice-versa). Dans le cadre des rythmes de batterie, il est possible de formuler un opérateur déplaçant le jeu des cymbales – c'est à dire substituant une frappe sur une cymbale par une frappe sur une autre cymbale, et laissant les autres frappes inchangées. Puisque nous nous restreignons dans cette section aux trois instruments grosse caisse, caisse claire et cymbale hi-hat, l'algorithme SEQUITUR original est utilisé, mais nous soulignons que pour d'autres applications, l'utilisation de tels opérateurs améliore le pouvoir descriptif des grammaires inférées.

Représentation d'une grammaire sous forme binaire. Nous rappelons que nous visons à calculer une approximation de la complexité d'une séquence s en la compressant. À l'issue de l'étape d'inférence de grammaire précédente, nous disposons d'une représentation efficace de s sous forme de grammaire. De manière à représenter cette grammaire sous forme de flux binaire, les membres de

Étape	Règle appliquée	Grammaire inférée
a bcbcabcbc		$S \rightarrow a$
ab cbcabcbc		$S \rightarrow ab$
abc bcabcbc		$S \rightarrow abc$
abcb cabcbc		$S \rightarrow abcb$
abcbc abcbc		$S \rightarrow a\underline{bcbc}$
abcbc abcbc	Unicité des bigrammes, création	$S \rightarrow aAA$ $A \rightarrow bc$
abcbca bcbc		$S \rightarrow aAAa$ $A \rightarrow bc$
abcbcab cbc		$S \rightarrow aAAab$ $A \rightarrow bc$
abcbcabc bc		$S \rightarrow aAAa\underline{bc}$ $A \rightarrow \underline{bc}$
abcbcabc bc	Unicité des bigrammes, réutilisation	$S \rightarrow \underline{aA}A\underline{aA}$ $A \rightarrow bc$
abcbcabc bc	Unicité des bigrammes, création	$S \rightarrow BAB$ $A \rightarrow bc$ $B \rightarrow aA$
abcbcabcb c		$S \rightarrow BABb$ $A \rightarrow bc$ $B \rightarrow aA$
abcbcabcbc		$S \rightarrow BAB\underline{bc}$ $A \rightarrow \underline{bc}$ $B \rightarrow aA$
abcbcabcbc	Unicité des bigrammes, réutilisation	$S \rightarrow \underline{BABA}$ $A \rightarrow bc$ $B \rightarrow aA$
abcbcabcbc	Unicité des bigrammes, création	$S \rightarrow CC$ $A \rightarrow bc$ $B \rightarrow aA$ $C \rightarrow BA$
abcbcabcbc	Utilité	$S \rightarrow CC$ $A \rightarrow bc$ $C \rightarrow aAA$

TAB. 4.7 – Exemple d'inférence de grammaire par l'algorithme SEQUITUR pour la séquence $abcbcabcbc$

Séquence	Transformations autorisées	Grammaire produite
cde.cde.gab.bag.	Transposition	$S \rightarrow AAt_7(A)bag.$ $A \rightarrow cde$
cde.cde.gab.bag.	Retournement	$S \rightarrow AAr(B).B.$ $A \rightarrow cde$ $B \rightarrow bag$
cde.cde.gab.bag.	Retournement et transposition	$S \rightarrow AAt_7(A)r(B).$ $A \rightarrow t_5(B).$ $B \rightarrow gab$

TAB. 4.8 – Exemple d'inférence de grammaire avec transformations

droite des productions sont concaténés, séparés par un symbole spécial # délimitant les productions. Ainsi, la grammaire :

$$\begin{array}{lcl} S & \rightarrow & AABA \\ A & \rightarrow & aCB \\ B & \rightarrow & Cd \\ C & \rightarrow & bc \end{array} \tag{4.37}$$

sera représentée par la séquence $AABA\#aCB\#Cd\#bc$. Si l'on désigne par Ω l'alphabet contenant les symboles terminaux, non-terminaux et le délimiteur #, dans le cas où un code entropique (code de Huffman, code arithmétique) est utilisé pour coder cette séquence, une approximation de la longueur du message binaire correspondant est donnée par :

$$l(G) \approx -\sum_{a \in \Omega} C(a) \log_2 \frac{C(a)}{N} \tag{4.38}$$

Où $\frac{C(a)}{N}$ est la fréquence du symbole a dans la séquence, N la longueur de la séquence.

Nous résumons ainsi la procédure retenue pour l'approximation de la complexité d'une séquence rythmique :

1. Inférence d'une grammaire hors-contexte $G(s)$ décrivant la séquence s, à l'aide de l'algorithme SEQUITUR.
2. "Mise à plat" de la grammaire $G(s)$ sous forme de séquence de symboles.
3. Codage de cette séquence de symboles par un code entropique, et calcul de la longueur de la séquence binaire résultante. Dans le cas où un code entropique optimal est utilisé, on peut directement calculer la longueur de la séquence binaire à partir de la fréquence d'apparition de chacun des symboles, sans effectuer le codage.

Notons que le critère de complexité obtenu satisfait bien notre objectif : la complexité moyenne des séquences de notre base de test est de $K + 984$ bits ; la complexité moyenne de leurs transcriptions obtenues par le seul critère acoustique est de $K + 1179$ bits, où K est une constante, omise par la suite, représentant la longueur, en bits, d'un décodeur de Huffman, suivi d'un programme reconstruisant S à partir de la grammaire.

4.5.3.2 Correction de séquence rythmique par minimisation de la complexité

On se propose maintenant d'utiliser ce critère de complexité pour améliorer la transcription des séquences. Le système de classification utilisant les paramètres acoustiques fournit les probabilités $P(S_n = s_n|t, \pi)$. Si $s = (s_n)$ est une séquence candidate, on lui affecte le score suivant :

$$F(s) = \sum_{n=1}^{L} \log P(S_n = s_n | t, \pi) - \alpha l(G(s)) \tag{4.39}$$

Le premier terme pénalise les séquences incompatibles avec les indications fournies par les paramètres acoustiques, le second terme pénalise les séquences complexes. On notera la ressemblance entre ce critère et les critères d'information utilisés dans la sélection d'ordre de modèles (de type Akaike), ou les critères de vraisemblance pénalisées – dans tous les cas, il s'agit de trouver une description compacte des données (peu complexe) compatible avec des observations. Ces méthodes s'inspirent du principe du rasoir d'Occam – parmi les séquences compatibles avec les observations (ou les probabilités fournies par le modèle acoustique), il est raisonnable de penser que le musicien a joué la séquence plus simple – c'est à dire la plus régulière et symétrique.

La séquence optimale s^* est ainsi obtenue en maximisant ce critère. Il n'existe malheureusement pas d'algorithme déterministe efficace permettant d'effectuer cette maximisation. En particulier, si on écrit s comme la concaténation de sous-séquences s_1 et s_2, on n'a pas de relation simple entre $l(G(s))$, $l(G(s_1))$ et $l(G(s_2))$. Cela interdit l'emploi de méthodes de programmation dynamique (comme l'algorithme de Viterbi utilisé dans le cas des modèles à N-grammes), dont le principe exige qu'une solution optimale au problème considéré puisse être construite à partir de solutions optimales à ses sous-problèmes.

Une recherche exhaustive dans l'espace de toutes les séquences possibles A^L est bien entendu impossible. Nous proposons alors l'emploi d'algorithmes évolutionnaires[7] [Mit98] pour produire la séquence optimale. Le choix de cette méthode est motivé par le fait que les séquences se représentent trivialement sous forme de "chromosomes" pour lesquels l'opérateur de recombinaison a du sens : on espère produire une bonne transcription en combinant des fragments de transcriptions valides. Autrement dit, le choix d'un codage de la structure à optimiser sous forme de chromosomes, une des étapes clés dans la mise en oeuvre des méthodes évolutionnaires, est ici triviale.

La mise en oeuvre de cette méthode d'optimisation est détaillée ici :

1. Initialisation d'une population de $N_{pop} = 200$ séquences $(s^i)_{i \in \{1,\ldots,N_{pop}\}}$. Cette population est initialisée avec la séquence optimale selon un critère purement acoustique,

$$\operatorname*{argmax}_s \sum_{n=1}^{L} \log P(S_n = s_n | t, \pi) \tag{4.40}$$

   à laquelle on fait subir des mutations aléatoires.

2. Reproduction. On forme $N_{exp} = 4N_{pop}$ séquences filles par la procédure suivante :
   a Choix aléatoire de deux parents s^1 et s^2 parmi la population courante.
   b Recombinaison. Un point de recombinaison $p \in \{1, \ldots, L\}$ est choisi aléatoirement. La séquence fille est alors déterminée par $s^f(n) = s^1(n), \forall n \in \{1, \ldots, p\}$ et $s^f(n) = s^2(n), \forall n \in \{p+1, \ldots, L\}$.
   c Mutation. Une position de mutation $p \in \{1, \ldots, L\}$ est choisie aléatoirement. La probabilité que le symbole en position p mute en a est alors donnée par $P(S_p = a | t, \pi)$.

3. Sélection. Une population de N_{pop} séquences survit. Cette population est constituée :
   – Des $0.9N_{pop}$ individus pour lesquels le critère F est maximal. Le calcul du critère F étant coûteux, un cache utilisant une politique LRU (*least recently used*) est utilisé pour éviter de calculer deux fois le critère F sur une même séquence.
   – De $0.1N_{pop}$ individus tirés aléatoirement parmi les individus restants.

4. Répétition des phases de reproduction, mutation, sélection sur $N = 50$ générations.

Une des particularités de cette implémentation réside dans le contrôle des probabilités de mutation. Cela limite, dans la pratique, l'exploration de l'espace des solutions aux séquences pour

[7] Nos premiers essais utilisant le recuit simulé se sont avérés infructueux, car demandant un refroidissement très lent pour ne pas tomber dans des minima locaux.

Exemple 1	Exemple 2
Séquence incomplète et complétion proposée	
$abcab?$	$abbaabbacddcdd.abababababcddcdd.abb????ac???dd.$
$abcab\mathbf{c}$	$abbaabbacddcdd.abababababcddcdd.abb\mathbf{aabb}ac\mathbf{ddc}dd.$
Grammaire minimale	
$S \rightarrow AA$ $A \rightarrow abc$	$S \rightarrow ABBCA$ $A \rightarrow DDC$ $B \rightarrow EE$ $C \rightarrow FF.$ $D \rightarrow Eba$ $E \rightarrow ab$ $F \rightarrow cdd$

TAB. 4.9 – Exemples de complétion automatique de séquence par minimisation de la complexité

lesquelles le premier terme de vraisemblance $\sum_{n=1}^{L} \log P(S_n = s_n|t, \pi)$ est élevé. Nous observons en fait que même lorsque le terme de vraisemblance est dominé par le terme de régularisation (c'est à dire quand $\alpha >> 1$) le contrôle des probabilités de mutation permet de produire des solutions conciliant le critère de complexité et les indices acoustiques.

En dehors de son application à la correction de séquences rythmiques, évaluée au prochain chapitre, cette approche peut aussi être utilisée pour la tâche de complétion de séquences comme illustré dans la table 4.9.

4.6 Résultats expérimentaux

Nous détaillons maintenant les performances obtenues par le système de transcription de la batterie présenté dans ce chapitre, en mettant l'accent sur l'apport de nos contributions.

4.6.1 Protocole

4.6.1.1 Base de données

S'il existe de nombreuses bases de données de sons isolés contenant des frappes de batterie [JW89; BBHL99; Fri], l'offre est beaucoup plus limitée en matière de séquences rythmiques annotées. La base de données RWC [GHNO02] contient des morceaux de musique populaire dont l'annotation est fournie sous forme de fichiers MIDI. Malheureusement, nombre de ces morceaux emploient des batteries synthétiques et/ou séquencées qui ne reproduisent ni la diversité des timbres d'une batterie acoustique, ni ses subtilités de jeu. Plus récemment, dans le cadre du projet MAMI de l'université de Ghent, l'annotation de 50 extraits musicaux longs de 30 secondes a été réalisée [TLD+05]. Cependant, les extraits musicaux étant protégés par copyright, cette base n'a pu être rendue publique, seules les annotations réalisées le sont.

Dans le cadre de cette thèse, une base intitulée ENST-drums a été enregistrée et annotée pour dépasser ces contraintes et permettre de nouveaux types d'expériences. En particulier, la disponibilité de pistes séparées pour chaque élément de la batterie et pour l'accompagnement permet de tester la robustesse des algorithmes sous diverses conditions de mixage, et d'évaluer des méthodes de séparation de sources, ce qui n'était jusqu'ici pas possible. Le contenu de la base ainsi que les

FIG. 4.9 – Batteries et batteurs dans la base ENST-drums

processus d'enregistrement et d'annotation sont documentés dans le second article donné dans l'annexe C. Une partie de la base a été rendue publique et a été distribuée à ce jour à une dizaine de laboratoires.

Nous avons utilisé pour les expériences menées dans ce chapitre les séquences *minus one* de cette base. Ces séquences sont constituées de 17 oeuvres musicales instrumentales mixées sans batterie, d'une durée moyenne de 71 secondes, sur lesquelles 3 batteurs différents ont improvisé la partie rythmique, chacun sur une batterie différente (petite batterie jazz/latin portable, batterie country/pop de taille moyenne, batterie complète rock, voir figure 4.9). Une caractéristique intéressante de ce type d'enregistrements est qu'il permet d'ajuster le mixage de la batterie et de l'accompagnement, de manière à tester la robustesse du système de transcription en présence d'autres instruments. Les expériences ont ainsi pu être répétées pour 4 mixages différents, dans lesquels l'accompagnement instrumental est successivement supprimé (batterie seule), atténué de 6 dB, équilibré avec la batterie, et amplifié de 6 dB.

Cette base de données peut être considérée comme diverse et difficile en termes de style et de jeu : certaines séquences sont jouées aux balais, aux fagots ou aux mailloches ; d'autres mettent l'accent sur un style de jeu riche et naturel. De plus, l'annotation est exhaustive et inclut en particulier les *ghost notes*, des frappes peu accentuées utilisées pour donner un effet de "groove" à un rythme autrement trop simple. De telles frappes sont particulièrement difficiles à détecter. L'accompagnement instrumental est lui même riche et de styles variés (musette, blues, funk, swing...), utilisant des instruments acoustiques (contrebasse, vibraphone, piano, accordéon), électro-acoustiques (guitare électrique, guitare electro-acoustique, orgue Hammond, piano Fender Rhodes) ou des synthétiseurs.

Tous les signaux sont échantillonnés à 44.1 kHz et traités à cette fréquence.

4.6.1.2 Protocole d'apprentissage et de test

Nous avons voulu éviter à tout prix deux erreurs communément rencontrées dans la littérature :

- Dans certaines études, les paramètres des algorithmes d'apprentissages sont choisis par validation croisée, et les résultats publiés sont ces mêmes résultats de validation croisée. De tels résultats ne permettent pas de juger du pouvoir de généralisation des algorithmes utilisés. Nous veillerons à ce que les paramètres des algorithmes de classification soient choisis par validation croisée sur un sous-ensemble de la base, et testés sur un autre.
- L'implémentation classique de la validation croisée dans des outils logiciels comme WEKA [oW03] ou Spider [WEBS] peut placer des frappes issues d'une même séquence dans les ensembles de test et d'apprentissage. Nous veillerons à ce que les ensembles de test et d'apprentissage n'aient non seulement aucun exemple en commun, mais également qu'ils contiennent des frappes venant de séquences différentes, jouées par des batteurs différents, avec des accompagnements différents.

Ces deux contraintes suggèrent le protocole dit de validation emboîtée suivant. Tout d'abord, les 17 séquences d'accompagnement de la base de données sont divisées en 3 groupes (un groupe

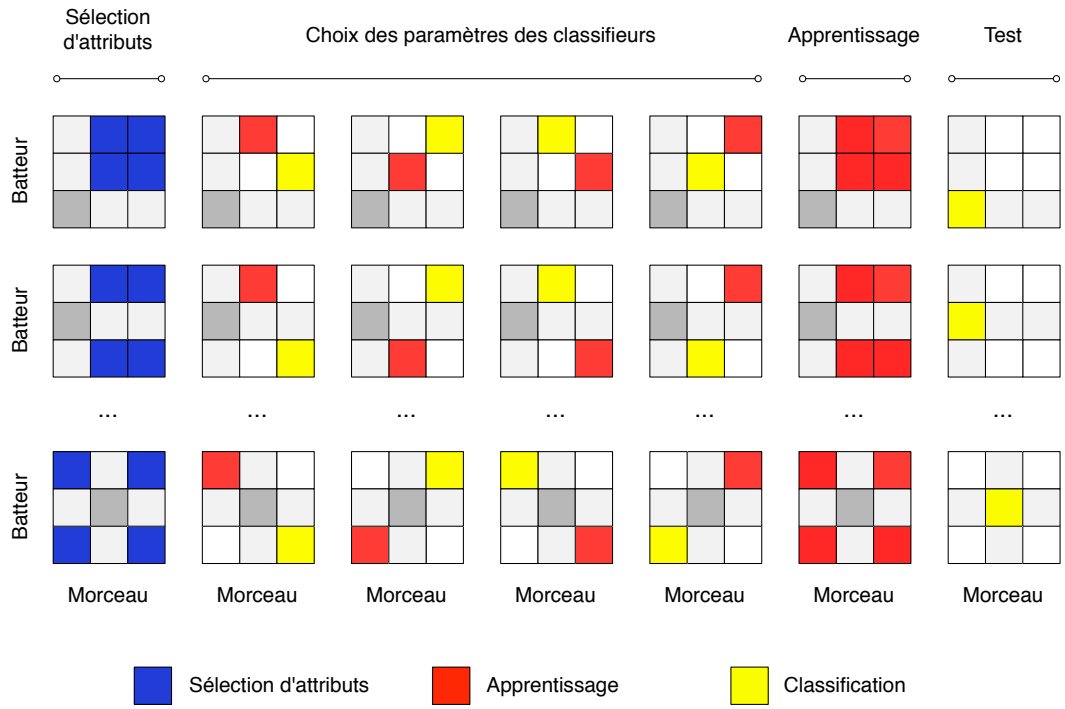


FIG. 4.10 – Protocole de validation emboîtée utilisé

contient les 5 morceaux les plus longs, les deux autres groupes 6 morceaux). Soit S_{ij} le sous-ensemble de la base de données contenant les morceaux du i-ème groupe, joué par le j-ème batteur. L'évaluation est ensuite conduite selon le protocole décrit dans l'algorithme 4 et illustré dans la figure 4.10.

Ce protocole assure que les paramètres choisis pour C, σ, le nombre d'attributs d et l'algorithme de sélection d'attributs fournissent des bonnes propriétés de généralisation, puisque dans la boucle intérieure de notre protocole, les ensembles de test et d'apprentissage correspondent à la fois à des morceaux et à des batteries différentes. Le surapprentissage est évité en s'assurant que les données sur lesquelles les classifieurs seront utilisés en fin de chaîne n'ont aucun point commun avec les données sur lesquels les attributs et les paramètres des classifieurs ont été choisis.

4.6.1.3 Métriques

La qualité de la transcription est évaluée avec des mesures classiques de précision et de rappel, calculées pour chaque classe d'instrument. Soient N_k^d le nombre de frappes de l'instrument k détectées par le système, N_k^c le nombre de frappes correctes détectées par le système (un écart de 50 ms au plus est toléré entre l'onset actuel et l'onset détecté) ; et N_k le nombre de frappes de l'instrument k qu'il aurait fallu détecter. La précision et le rappel sont alors définis par :

$$P_k = \frac{N_k^c}{N_k^d} \tag{4.41}$$

$$R_k = \frac{N_k^c}{N_k} \tag{4.42}$$

Algorithme 4 : Protocole d'évaluation

entrées : Base de données divisée en 9 groupes S_{ij}, $t_i, \mathbf{x}_i$ pour chaque séquence
pour chaque $(i_0, j_0) \in \{1,2,3\} \times \{1,2,3\}$ **faire**
  pour chaque *Instrument considéré* **faire**
    $A \leftarrow \bigcup_{i \neq i_0, j \neq j_0} S_{ij}$
    Sélection des attributs dans le sous-ensemble A par RFE-SVM
    Sélection des attributs dans le sous-ensemble A par IRMFSP
    pour chaque $(C, \sigma, d, alg) \in \mathcal{D}(C) \times \mathcal{D}(\sigma) \times \mathcal{D}(d) \times \{\textit{RFE-SVM}, \textit{IRMFSP}\}$ **faire**
      erreur généralisation $\leftarrow 0$
      pour chaque $i_1 \neq i_0, j_1 \neq j_0$ **faire**
        $(\boldsymbol{\alpha}, b) \leftarrow$ Entraîner C-SVM (C, σ) sur $S_{i_1 j_1}$ avec les d meilleurs attributs produits par alg
        erreur $\leftarrow$ Tester SVM $(\boldsymbol{\alpha}, b, \sigma)$ sur $S_{i_2 j_2}$, avec $i_2 \notin \{i_0, i_1\}, j_2 \notin \{j_0, j_1\}$
        erreur généralisation $\leftarrow$ erreur généralisation + erreur
      fin
    fin
    Entraîner C-SVM (C^*, σ^*) sur A avec les d^* meilleurs attributs produits par alg^*, où C^*, σ^*, d^*, alg^* minimisent l'erreur de généralisation
  fin
  Utiliser les classifieurs entraînés pour transcrire les séquences dans $S_{i_0 j_0}$
fin
sorties : Une transcription automatique de chaque séquence de la base

Ces mesures dépendent du seuil de décision, par exemple, un seuil de décision très haut assurera une bonne précision mais un mauvais rappel. La F-mesure tente de résumer ce compromis entre rappel et précision, et est définie comme suit :

$$F_k = \frac{2P_k R_k}{P_k + R_k} \tag{4.43}$$

4.6.2 Résultats

4.6.2.1 Performances en transcription

Un résumé des performances est donné dans la table 4.10. Nous commentons ces résultats, ainsi que d'autres analyses détaillées supplémentaires si besoin, dans les paragraphes qui suivent. Précisons avant tout que les résultats sont tronqués avant la première décimale non significative - les résultats donnés avec une décimale après la virgule ont ainsi un intervalle de confiance à 95% d'amplitude inférieure à 0.1.

Apport du pré-traitement d'accentuation de la piste de batterie Commençons tout d'abord par comparer les deux systèmes n'utilisant aucune fusion : le système effectuant la détection sur le signal original, et le système effectuant la détection sur le signal pré-traité par les méthodes décrites au chapitre précédent. Les résultats sont donnés dans les deux premiers groupes de colonnes de la table 4.10. Notre première observation est que globalement, le pré-traitement n'améliore que légèrement les performances en détection de caisse claire et de hi-hat. Les gains les plus importants sont observés dans les situations où l'accompagnement est le plus fort – situation où le pré-traitement prend tout son intérêt. Les performances en détection de grosse caisse sont, elles, légèrement dégradées.

Des résultats plus détaillés, présentés par batterie, sont donnés dans la table 4.11. Nous observons tout d'abord que pour les séquences jouées sur la batterie 1, les meilleurs résultats sont presque toujours obtenus en utilisant le pré-traitement. Comment cela s'explique-t-il ? La batterie 1 a un timbre très différent des autres, en particulier à cause de sa grosse caisse sonnant comme un tom grave,

	Signal original			**Signal pré-traité**			**Fusion précoce**			**Fusion tardive**		
Instrument	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$
Accompagnement $-\infty$ dB												
BD	66.4	67.8	67.1	60.4	75.2	67.0	62.8	62.7	62.8	**65.6**	**80.5**	**72.3**
SD	52.4	80.1	63.3	57.0	70.1	62.9	51.1	78.3	61.8	**58.5**	**75.7**	**66.0**
HH	81.3	76.8	79.0	82.5	78.6	80.5	86.5	76.6	81.3	**85.2**	**79.2**	**82.1**
Accompagnement -6 dB												
BD	65.7	72.1	68.7	54.3	69.3	60.9	63.7	61.5	62.6	**64.6**	**79.2**	**71.1**
SD	54.7	72.4	62.3	57.3	69.0	62.6	**56.6**	**75.1**	**64.5**	**57.7**	**73.2**	**64.5**
HH	81.2	75.8	78.4	79.5	78.4	79.0	80.5	77.3	78.9	**82.4**	**78.2**	**80.3**
Accompagnement $+0$ dB												
BD	61.7	58.4	60.0	54.1	65.8	59.4	61.1	61.0	61.1	**62.0**	**70.2**	**65.8**
SD	46.4	66.7	54.7	50.6	66.1	57.4	**52.0**	**69.5**	**59.5**	50.6	70.7	59.0
HH	80.8	70.6	75.4	79.5	73.3	76.3	78.9	74.9	76.8	**83.1**	**73.0**	**77.7**
Accompagnement $+6$ dB												
BD	60.0	54.3	57.0	55.1	58.5	56.8	55.5	54.9	55.2	**60.9**	**62.6**	**61.7**
SD	37.6	54.7	44.6	41.3	56.5	47.7	**48.0**	**58.7**	**52.8**	42.8	60.4	50.1
HH	76.7	65.6	70.6	74.7	68.4	71.4	74.7	67.7	71.1	**78.0**	**68.0**	**72.6**

TAB. 4.10 – Rappel R, Précision P et F-mesure F pour la transcription de la batterie avec accompagnement

	Batterie 1		Batterie 2		Batterie 3	
Signal pré-traité ?	○	●	○	●	○	●
Accompagnement $-\infty$ dB						
BD	21.5	**50.8**	**94.3**	83.8	**84.0**	75.2
SD	58.4	**66.7**	66.8	**77.9**	**63.6**	60.5
HH	65.9	**66.4**	**83.2**	**83.2**	76.3	**81.8**
Accompagnement -6 dB						
BD	20.7	**60.3**	**87.6**	71.1	**85.7**	66.0
SD	**63.8**	60.3	**68.5**	66.2	57.4	**58.0**
HH	63.8	**64.8**	79.2	**82.3**	**80.2**	76.7
Accompagnement $+0$ dB						
BD	16.5	**56.6**	**82.9**	67.5	**81.6**	64.3
SD	51.3	**53.9**	62.6	**62.8**	50.6	**53.9**
HH	61.8	**63.8**	76.3	**79.7**	**77.0**	75.0
Accompagnement $+6$ dB						
BD	27.9	**54.6**	**76.8**	66.5	**77.6**	64.1
SD	45.7	**47.6**	48.8	**49.0**	41.8	**46.7**
HH	60.5	**61.0**	71.2	**72.6**	**70.1**	69.9

TAB. 4.11 – Performances (F-mesure en %) par batterie, avec et sans pré-traitement, pour divers mixages

et de sa petite caisse claire sonnant très aigüe. De manière à permettre la meilleure généralisation possible d'un classifieur entraîné sur les batterie 2 et 3 à la batterie 1, les attributs utilisés doivent être robustes à ces différences de timbre, en faisant abstraction de la hauteur des composantes tonales. Les attributs calculés sur le signal pré-traité ne dépendent pas des composantes tonales présentes dans le signal, et permettent une généralisation acceptable des batteries 2 et 3 à la batterie 1.

Nous constatons ensuite que pour les batteries 2 et 3, les performances en détection de grosse caisse sont beaucoup plus faibles sur le signal pré-traité. Cela s'explique par le fait que pour ces batteries, la grosse caisse produit une composante harmonique de fréquence très basse. La seule composante harmonique dans les régions les plus basses du spectre vient de la grosse caisse, et est ainsi éliminée lors de la projection sur l'espace bruit durant le pré-traitement.

Enfin, nous observons que dans la majorité des cas, la détection de la hi-hat est plus aisée sur le signal pré-traité. Une des explications possibles est que la projection sur l'espace bruit supprime les parties harmoniques entretenues, à décroissance lente, du signal de batterie (frappes sur les toms par exemple). La détection de frappes courtes et impulsives (frappe de hi-hat fermée) jouées après une frappe à long temps de décroissance est alors plus facile. Cela explique pourquoi, même sur les signaux où la batterie joue seule, le pré-traitement peut avoir un intérêt.

Influence du mixage Sans surprise, les performances se détériorent lorsque le niveau de l'accompagnement instrumental augmente. Nous observons cependant quelques cas où les performances en classification sont meilleures avec un accompagnement instrumental de niveau faible (-6 dB) que sans accompagnement. Une justification possible est la suivante : la présence d'une musique d'accompagnement augmente la diversité de l'ensemble d'apprentissage, et permet ainsi de meilleures capacités de généralisation. Cette observation suggère que le meilleur moyen d'entraîner un système de transcription de solo de batterie est d'utiliser non pas des soli, mais des enregistrements de batterie avec un accompagnement faible pour diversifier les données.

Apport des méthodes de fusion Nous avons vu que le pré-traitement d'accentuation de la piste de batterie ne conduit pas toujours à de meilleures performances. Cela souligne l'intérêt des méthodes de fusion qui vont tirer au mieux partie des attributs calculés sur les deux signaux dispo-

Instrument	$R\%$	$P\%$	$F\%$
Maximum			
BD	**66.9**	**65.0**	**65.8**
SD	59.2	58.6	58.9
HH	88.4	67.7	76.7
Minimum			
BD	47.3	66.1	55.2
SD	38.2	83.3	52.3
HH	72.2	78.1	75.0
Somme pondérée, $\alpha = 0.5$			
BD	**62.0**	**70.2**	**65.8**
SD	**50.6**	**70.7**	**59.0**
HH	**83.1**	**73.0**	**77.7**
Plus confiant			
BD	56.0	66.5	60.8
SD	44.1	78.9	56.6
HH	77.9	76.5	77.2
Produit			
BD	60.2	63.8	62.0
SD	53.2	58.4	55.7
HH	82.7	67.8	74.6

TAB. 4.12 – Performances (Rappel R, Précision P, F-mesure F) pour un mixage équilibré, avec diverses méthodes de fusion tardive

nibles. Pour les signaux où l'accompagnement instrumental est présent, même à faible volume, les méthodes de fusion produisent les meilleurs résultats. Cela est particulièrement vérifié pour la fusion tardive – les meilleurs résultats sont dans ce cas obtenus avec l'opérateur somme pondérée avec un poids égal pour les deux sources d'information. Nous présentons dans la table 4.12 les résultats obtenus pour chacun des opérateurs de fusion considérés, sur les enregistrements avec batterie et musique d'accompagnement équilibrés.

Nous livrons dans les sous-sections qui suivent des résultats relatifs non pas aux performances des classifieurs, mais aux attributs et paramètres des classifieurs choisis automatiquement lors de l'étape d'apprentissage.

4.6.2.2 Résultats de la sélection d'attributs

Vue d'ensemble des attributs sélectionnés La table 4.13 liste les 4 premiers attributs sélectionnés par la méthode IRMFSP (nous verrons plus tard que cette méthode est la plus apte à extraire des jeux d'attributs de petite taille) sur les ensembles d'attributs extraits du signal original, du signal pré-traité, ou des deux signaux, par instrument à reconnaître et par type de mixage.

Nous soulignons tout d'abord la pertinence des attributs mesurant la distribution de l'énergie en sortie de filtres spécifiques à la batterie – qu'il s'agisse de ceux utilisant les filtres de Tanghe et al. [TDB05] ou ceux que nous avons proposés dans [GR04]. Pour chaque instrument et chaque type de mixage, au moins un de ces attributs est presque toujours sélectionné. Notons cependant qu'ils sont parfois utilisés de façon surprenante. Par exemple, parmi les attributs $lRMS_{gband,k}$, le plus caractéristique de la caisse claire est $lRMS_{gband,3}$, puisqu'il mesure l'énergie dans la bande où est concentrée 95% de l'énergie d'une frappe de caisse claire. Or, en présence d'accompagnement à un volume équilibré ou fort, l'attribut de cette catégorie utilisé pour la détection de la caisse claire est $lRMS_{gband,8}$, mesurant l'énergie dans la bande $[10000, 15000]$ Hz. Nous expliquons cela par le

Instrument	Attributs signal original	Attributs signal pré-traité	Attributs joints
Accompagnement $-\infty$ dB			
BD	$lRMS_{bd}$ Ldr_{15} $lRMS_{gband,2}$ $lRMSrel_{hh,bd}$	$lRMSrel^*_{bd}$ Ldr^*_{14} $lRMS^*_{bd}$ $\mu MFCC^*_0$	$lRMS_{bd}$ $lRMSrel^*_{bd,sd}$ $\sigma MFCC_{12}$ $\sigma MFCC^*_{12}$
SD	Ldr_{10} $lRMSrel_{sd}$ $lRMS_{gband,6}$ $\mu MFCC_0$	Ldr^*_{14} Ldr^*_{12} Ldr^*_{13} $lRMS^*_{gband,3}$	Ldr^*_{13} $lRMSrel_{sd}$ Ldr^*_{14} Ldr^*_{12}
HH	S_{kurt} $lRMS_{hh}$ Ldr_{24} $\mu MFCC_0$	$\sigma MFCC^*_0$ Ldr^*_{24} $lRMS^*_{hh}$ $OBSIR^*_7$	S_{kurt} $lRMS^*_{hh}$ Ldr^*_{24} $\sigma MFCC_0$
Accompagnement -6 dB			
BD	$lRMS_{bd}$ $\mu MFCC_0$ $\sigma MFCC_{12}$ $lRMSrel_{sd}$	$lRMS^*_{bd}$ T^*_A Et^* $lRMS^*_{gband,1}$	$lRMS_{bd}$ $\sigma MFCC_{12}$ $\sigma MFCC^*_{11}$ T^*_A
SD	$\sigma MFCC_0$ Ldr_{11} Ldr_{10} $\mu MFCC_0$	$\mu MFCC^*_0$ Ldr^*_{12} $\sigma MFCC^*_0$ Ldr^*_{13}	Ldr^*_{12} $\sigma MFCC^*_{12}$ $\mu MFCC^*_0$ $lRMSrel_{sd}$
HH	S_{kurt} Ldr_{24} $lRMS_{gband,4}$ Et	S^*_{kurt} Ldr^*_{24} S^*_{flat} $lRMS^*_{gband,8}$	S_{kurt} Ldr_{24} $\sigma MFCC^*_{12}$ S^*_{kurt}
Accompagnement $+0$ dB			
BD	$lRMS_{bd}$ $lRMS_{gband,1}$ $\sigma MFCC_{12}$ $lRMSrel_{bd,sd}$	T^*_A $lRMS^*_{bd}$ $lRMS^*_{gband,1}$ $OBSIR^*_3$	$lRMS_{bd}$ T^*_A $\sigma MFCC_{12}$ $lRMS_{gband,1}$
SD	$lRMS_{gband,8}$ $lRMSrel_{sd}$ Ldr_{11} $OBSIR_2$	Ldr^*_{10} $\mu MFCC^*_0$ $\sigma MFCC^*_0$ $lRMS^*_{gband,8}$	$lRMS^*_{gband,8}$ Ldr^*_{10} $\mu MFCC^*_0$ Ldr^*_{12}
HH	S_{kurt} $lRMS_{gband,6}$ S_{flat} Et	S^*_{kurt} $lRMS^*_{gband,8}$ Ldr^*_{24} S^*_{flat}	S^*_{kurt} $lRMS^*_{hh}$ $lRMS^*_{gband,8}$ $\sigma MFCC^*_{12}$
Accompagnement $+6$ dB			
BD	$lRMS_{bd}$ $lRMS_{gband,1}$ $Crest$ $\sigma MFCC_{12}$	$lRMS^*_{gband,1}$ T^*_A $lRMS^*_{bd}$ $OBSIR^*_3$	$lRMS_{bd}$ $lRMS_{gband,1}$ $\sigma MFCC_{12}$ $\sigma MFCC^*_{12}$
SD	$lRMS_{gband,8}$ $Crest$ $\mu MFCC_0$ Ldr_{23}	Ldr^*_{10} $lRMS^*_{gband,8}$ Ldr^*_{11} $lRMS^*_{hh}$	$lRMS^*_{gband,8}$ Ldr^*_{10} Ldr^*_{12} $lRMSrel^*_{sd}$
HH	S_{kurt} $lRMS_{gband,8}$ $Crest$ $lRMS_{gband,7}$	S^*_{kurt} $lRMS^*_{gband,8}$ S^*_{flat} $\sigma MFCC^*_1$	S^*_{kurt} $lRMS^*_{gband,8}$ S^*_{flat} $\sigma MFCC^*_{12}$

TAB. 4.13 – 4 premiers attributs sélectionnés sur les ensembles d'attributs extraits du signal original, du signal pré-traité (*), ou des deux signaux, par instrument à reconnaître et par type de mixage

fait que la bande de fréquence associée à $lRMS_{gband,3}$ contiendra de nombreux partiels associés à d'autres instruments dans l'accompagnement. En conséquence, $lRMS_{gband,3}$ est peu robuste à l'ajout de bruit. À l'inverse $lRMS_{gband,8}$ ne contiendra pas de partiels issus des instruments harmoniques et restera robuste. Reste à expliquer quelle information $lRMS_{gband,8}$ livre quant à la présence de la caisse claire. Nous suggérons que dans le cas d'une frappe de caisse claire avec timbre, le bruit produit par le timbre occupe une partie de la bande de fréquence associée à $lRMS_{gband,8}$.

Les paramètres spectraux semblent surtout intéressants pour la détection des frappes de hi-hat, en particulier le kurtosis ou la platitude spectrale. Tous deux caractérisent le même phénomène : en présence d'une hi-hat, qui peut être grossièrement modélisée par un bruit coloré, le contraste du spectre diminue. Leur équivalent perceptuel, l'étendue Et est également sélectionné.

Les paramètres cepstraux semblent d'intérêt limité : ceux sélectionnés sont les moyennes et variances du premier coefficient, donc une mesure d'énergie et de variabilité de l'énergie dans la fenêtre d'observation.

Le seul paramètre temporel sélectionné est le paramètre T_A du modèle d'enveloppe, sélectionné pour la détection de grosse caisse. Ce paramètre fournit une mesure de l'amplitude du signal au début de la fenêtre d'observation. Nous supposons que le caractère impulsionnel de la frappe de grosse caisse, pour laquelle l'énergie est concentrée en début de fenêtre, explique le choix de ce paramètre.

Les attributs psychoacoustiques sont relativement peu utilisés, en dehors de la sonie relative pour des valeurs de 10 à 14 Barks (de 1250 à 2250 Hz environ). Le rôle joué par ces attributs semble difficile à justifier – pourquoi sont-ils préférés à un seul attribut qui mesurerait l'énergie dans une telle bande de fréquence ($lRMS_{gband,6}$ par exemple) ? Il s'agit là peut être d'une limite rencontrée par l'algorithme IRMFSP : de tels attributs ne sont pas sélectionnés par l'algorithme RFE-SVM.

Complémentarité des attributs extraits sur les signaux originaux et pré-traités Dans le cas où la sélection d'attributs est réalisée sur les attributs joints (fusion précoce), il est intéressant d'évaluer la part d'attributs extraits du signal original et du signal pré-traité. À cet effet, nous avons sélectionné avec l'algorithme RFE-SVM les 10 meilleurs attributs parmi ceux extraits du signal original et du signal pré-traité. Nous les présentons groupés par catégorie dans la table 4.14

Nous observons que le nombre d'attributs extraits à partir du signal pré-traité augmente avec le niveau de l'accompagnement instrumental. La hi-hat et la caisse claire bénéficient le mieux des attributs extraits du signal pré-traité. Pourtant, au moins 2 attributs sont à chaque fois sélectionnés parmi les attributs extraits du signal original. Cela justifie notre intuition initiale selon laquelle les informations contenues dans les deux signaux seraient complémentaires, expliquant ainsi les bons résultats obtenus par les méthodes de fusion.

4.6.2.3 À propos de l'apprentissage

Paramètres optimaux par problèmes de classification Nous nous intéressons maintenant aux paramètres optimaux sélectionnés à chaque tour du protocole de validation emboîtée. Les paramètres optimaux les plus fréquemment choisis pour chaque problème de classification sont donnés dans le tableau 4.15.

La détection de la grosse caisse est la tâche pouvant être effectuée efficacement avec le moins de paramètres : nous avons vu en effet que les paramètres spécifiques (puissance dans des bandes de fréquences très basses) sont très pertinents pour cette tâche. La détection de la caisse claire est la tâche exigeant le plus d'attributs – sans doute parce que la caisse claire est, parmi les instruments considérés, celui dont l'énergie est concentrée dans la bande de fréquences la plus susceptible de contenir des partiels des autres instruments harmoniques. Le nombre d'attributs extraits est rarement élevé. Nous nous attendions en fait à voir le nombre d'attributs sélectionné croître à mesure que le niveau de l'accompagnement musical augmente, pour mieux appréhender la variabilité croissante des signaux. Ce n'est pas le cas. Une première explication serait que les classifieurs utilisés sont inefficaces en grandes dimensions, et que d doit ainsi rester faible ; mais les SVM sont connues pour

	Attributs signal original						**Attributs signal pré-traité**					
Instr.	T	D	S	C	P	Total	T	D	S	C	P	Total
Accompagnement $-\infty$ dB												
BD	1	5	0	1	1	**8**	2	0	0	0	0	2
SD	1	1	1	1	1	**5**	0	2	1	1	1	**5**
HH	0	2	0	0	1	3	1	1	3	1	1	**7**
Accompagnement -6 dB												
BD	1	3	0	1	1	**6**	1	1	0	2	0	4
SD	2	1	0	1	0	4	0	3	0	3	0	**6**
HH	2	0	0	0	2	4	1	0	3	1	1	**6**
Accompagnement $+0$ dB												
BD	0	2	0	0	0	2	1	4	0	3	0	**8**
SD	2	2	0	0	0	4	2	1	0	3	0	**6**
HH	1	0	0	0	0	1	1	1	5	1	1	**9**
Accompagnement $+6$ dB												
BD	0	4	0	0	0	4	1	4	0	1	0	**6**
SD	2	1	0	0	0	3	2	3	0	2	0	**7**
HH	2	0	0	0	0	2	1	0	4	0	3	**8**

TAB. 4.14 – Nombre d'attributs temporels (T), de distribution d'énergie (D), spectraux (S), cepstraux (C) et psychoacoustiques (P) extraits par la méthode RFE-SVM

	Signal original				**Signal pré-traité**				**Fusion précoce**			
Accompagnement $-\infty$ dB												
Instr.	d^*	C^*	σ^*	alg^*	d^*	C^*	σ^*	alg^*	d^*	C^*	σ^*	alg^*
BD	4	2	2	I	16	2	1	I	16	2	2	R
SD	16	2	$\frac{1}{4}$	R	16	2	$\frac{1}{2}$	I	8	128	2	I
HH	32	16	1	R	16	16	2	I	8	16	2	I
Accompagnement -6 dB												
BD	4	2	2	I	8	2	1	I	4	2	2	I
SD	32	2	$\frac{1}{4}$	R	32	2	$\frac{1}{2}$	R	32	2	$\frac{1}{2}$	R
HH	16	128	1	R	8	128	2	I	32	128	2	R
Accompagnement $+0$ dB												
BD	4	16	2	R	16	128	2	I	4	16	2	I
SD	16	128	2	R	32	2	$\frac{1}{4}$	R	32	128	2	I
HH	32	128	1	R	8	128	1	I	16	128	1	R
Accompagnement $+6$ dB												
BD	16	16	2	I	16	128	2	I	32	16	2	R
SD	64	2	$\frac{1}{8}$	R	32	2	$\frac{1}{2}$	R	16	128	2	I
HH	32	128	1	R	16	16	1	I	32	128	1	I

TAB. 4.15 – Paramètres optimaux choisis pour chaque problème de classification : nombre d'attributs choisis, paramètre de régularisation des C-SVM, taille du noyau σ, et algorithme de sélection d'attributs (I pour IRMFSP, R pour RFE-SVM)

être résistantes à la "malédiction de la dimensionnalité". Nous suggérons plutôt que pour chacun des problèmes de classification considérés, seul un petit nombre d'attributs est suffisamment robuste.

Nous remarquons que C^* prend souvent une valeur élevée quand σ^* prend une valeur élevée, et inversement, les petites valeurs de σ^* sont presque toujours associées à des valeurs faibles de C^*. En fait, C^* et σ^* correspondent à deux stratégies différentes pour contrôler la généralisation : maximiser la marge tout en permettant à la surface de décision de prendre des formes arbitrairement complexes (σ^* et C^* faibles) ; ou garder une surface de décision simple, tout en s'assurant qu'elle discrimine au mieux les exemples (σ^* et C^* élevés).

Concluons enfin quant aux performances relatives des algorithmes de sélection d'attributs IRMFSP et RFE-SVM. Les résultats laissent supposer que RFE-SVM est le plus souvent choisi pour les grands ensembles d'attributs qu'il sélectionne ($d \geq 16$), et IRMFSP sur les petits ensembles d'attributs ($d < 16$). Nous confirmons cette observation dans le paragraphe qui suit.

Performances comparées de RFE-SVM et IRMFSP Dans cette expérience, nous utilisons une variante du protocole de validation emboîtée (algorithme 4), dans laquelle d et alg sont fixés à l'avance. Nous calculons alors, pour l'ensemble des 9 itérations (itération sur les trois batteurs, itération sur les trois sous-ensembles de morceaux), la moyenne de l'erreur de généralisation obtenue par le classifieur de paramètres optimaux, sur l'ensemble des 3 problèmes de détection de grosse caisse, caisse claire et hi-hat. Les résultats sont donnés dans la table 4.16, et confirment notre observation que l'algorithme IRMFSP est plus efficace sur les petits ensembles d'attributs ($d \in \{4, 8\}$), tandis que RFE-SVM donne de meilleures performances dans les autres situations.

d	**Err. gén. (%), RFE-SVM**	**Err. gén. (%), IRMFSP**
4	25.4	**24.0**
8	23.4	**22.3**
16	**21.7**	21.9
32	**21.7**	22.4
64	**22.0**	22.6
96	**22.1**	22.7

TAB. 4.16 – Performances (% d'erreur de généralisation) comparées des classifieurs utilisant les attributs sélectionnés par les méthodes RFE-SVM et IRMFSP

Séparabilité Nous donnons dans la figure 4.11 les surfaces de décision projetées sur les 3 premières composantes principales. Les attributs utilisés pour l'apprentissage des SVM dont nous avons tracé les surfaces de décision sont extraits à la fois du signal original et du signal pré-traité, et ont été sélectionnés par l'algorithme RFE-SVM. Nous observons qu'en dépit de l'étape de sélection d'attributs, les ensembles d'apprentissage pour le détecteur de caisse claire et le détecteur de hi-hat en présence d'un accompagnement instrumental sont peu séparables (la séparation est plus facile en l'absence d'accompagnement). Ces résultats montrent que les limites de notre méthode ne sont pas inhérentes au classifieur retenu, mais aux attributs extraits : il sera nécessaire d'utiliser des attributs plus discriminants ou plus robustes, de manière à améliorer les performances.

Analyse des vecteurs de support Intuitivement, les vecteurs de support correspondent aux exemples les plus difficiles à classer. Nous avons, dans l'expérience qui suit, analysé la composition de l'ensemble des vecteurs de support pour deux problèmes de classification. Dans les deux cas, de manière à limiter le nombre de vecteurs de supports à analyser et annoter, nous avons tiré aléatoirement 100 d'entre eux.

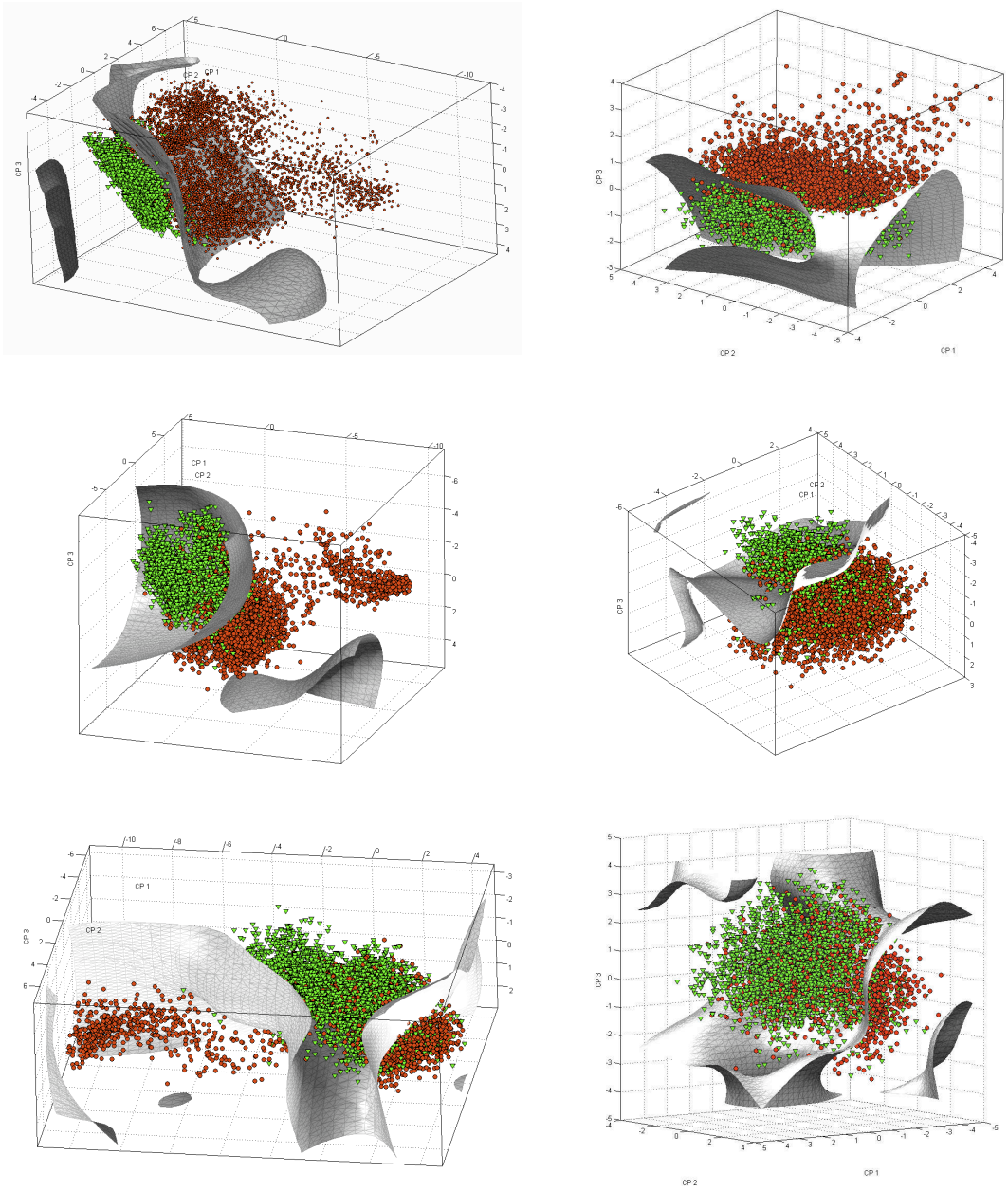


FIG. 4.11 – Surfaces de décisions projetées sur les 3 premières composantes principales. À gauche : accompagnement $-\infty$ dB. À droite : accompagnement 0 dB. De haut en bas : grosse caisse, caisse claire, hi-hat. La classe positive est représentée en vert

Classe	Nombre	Description des exemples d'apprentissage associés
−	32	Frappes diverses d'autres instruments de la batterie
−	18	Combinaisons de frappes sonnant similairement à une caisse claire
−	14	Chevauchement entre caisse claire et autre frappe
−	3	Segments courts
+	17	Frappes de caisse claire sans particularité
+	10	Frappes de caisse claire légères (*ghost notes*)
+	3	Combinaison de caisse claire avec tom ou cymbale
+	3	Segments courts

TAB. 4.17 – Composition des vecteurs de support pour la détection de caisse claire sur des signaux de batterie sans accompagnement

Classe	Nombre	Description des exemples d'apprentissage associés
−	17	Basse prédominante
−	13	Accord au piano ou à la guitare très percussif prédominant
−	12	Caisse claire prédominante
−	7	Exemples suivant immédiatement une frappe de grosse caisse
+	18	Frappes de grosse caisse simultanée à une note de basse
+	9	Frappes simultanées à un accord joué à la guitare ou au piano
+	8	Combinaisons grosse caisse + cymbale crash ou caisse claire
+	7	Frappes de grosse caisse simultanée à un instrument mélodique
+	5	Onsets mal alignés résultant en une troncature de la frappe
+	4	Grosse caisse en solo

TAB. 4.18 – Composition des vecteurs de support pour la détection de grosse caisse sur des signaux de musique avec accompagnement instrumental mixé au même niveau que la batterie

Le premier problème étudié est celui de la classification de la caisse claire en l'absence d'accompagnement. L'ensemble d'apprentissage contient 8038 exemples, dont 465 sont des vecteurs de support. Le second problème étudié est celui de la classification de la grosse caisse en présence d'accompagnement, à un niveau équilibré. Parmi les 8578 exemples de l'ensemble d'apprentissage, 1290 sont des vecteurs de support. La composition des 100 vecteurs de support tirés aléatoirement est donnée respectivement dans les tables 4.17 et 4.18 qui recensent, pour chaque problème, à quel type d'exemples d'apprentissage correspondent les vecteurs de support positifs et négatifs.

Ces résultats soulignent l'approche discriminative employée par les SVM : nous voyons que les vecteurs de support correspondent à toutes les situations difficiles rencontrées en transcription de signaux percussifs. Dans le second cas, le nombre de vecteurs de support associés à des exemples où la basse est prédominante montre qu'il s'agit là d'un des problèmes les plus difficiles à résoudre dans la détection de frappes de grosse caisse.

4.6.2.4 Comparaison avec d'autres méthodes

Nous donnons dans la table 4.19 les résultats obtenus avec notre système (fusion tardive), avec une variante de notre système dans laquelle le pré-traitement d'accentuation de la piste de batterie est remplacé par la méthode d'extraction décrite par Helén et Virtanen dans [HV05], et avec le système

	Méthode proposée			**Pré-séparation**			**Tanghe et al.**		
Instr.	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$
Accompagnement $-\infty$ dB									
BD	**65.6**	**80.5**	**72.3**	**68.5**	**76.5**	**72.3**	58.5	87.2	70.0
SD	**58.5**	**75.7**	**66.0**	55.1	77.1	64.2	44.4	71.9	54.9
HH	**85.2**	**79.2**	**82.1**	80.6	76.1	78.3	82.9	65.0	72.9
Accompagnement -6 dB									
BD	**64.6**	**79.2**	**71.1**	64.7	74.8	69.4	45.8	69.5	55.2
SD	**57.7**	**73.2**	**64.5**	43.4	68.8	53.1	19.1	71.3	30.2
HH	**82.4**	**78.2**	**80.3**	77.9	69.8	73.7	82.7	53.9	65.3
Accompagnement $+0$ dB									
BD	**62.0**	**70.2**	**65.8**	54.4	59.4	56.8	33.9	67.7	45.2
SD	**50.6**	**70.7**	**59.0**	33.6	51.1	40.5	12.7	63.2	21.1
HH	**83.1**	**73.0**	**77.7**	71.2	65.1	68.3	81.1	51.2	62.8
Accompagnement $+6$ dB									
BD	**60.9**	**62.6**	**61.7**	36.9	53.8	43.7	18.7	53.7	27.7
SD	**42.8**	**60.4**	**50.1**	22.5	43.4	29.7	8.7	54.8	15.0
HH	**78.0**	**68.0**	**72.6**	60.2	62.2	61.2	77.2	48.9	59.9

TAB. 4.19 – Performances comparées du système de transcription proposé (avec fusion tardive), d'un système de transcription utilisant l'algorithme de Helén et Virtanen comme pré-traitement, et du système de transcription développé par Tanghe et al

de transcription développé par Tanghe et al. [TDB05], dont une implémentation est distribuée publiquement [Tan05]. En absence d'accompagnement, les performances de ces systèmes sont similaires à celles que nous obtenons, mais en présence d'accompagnement, leurs performances se dégradent rapidement.

Nous espérons que suite à la diffusion publique de la base ENST-drums, d'autres équipes testeront leurs algorithmes sur cette base et publieront leurs résultats.

4.6.2.5 Apport des modèles de séquence

Nous terminons enfin en étudiant l'apport des deux techniques employant des connaissances musicales présentées dans la section 4.5. Ces études sont menées en utilisant les séquences avec mixage équilibré (qui sont les plus proches des conditions d'utilisation réelles en indexation), jouées par les batteurs 2 et 3 – les performances en transcription pour le batteur 1 n'ont pas été jugées suffisantes.

Nous donnons dans le tableau 4.20 les F-mesures pour la séquence originale, et diverses méthodes de correction d'erreur. Nous observons tout d'abord que sans surprise, les gains les plus grands sont obtenus avec les modèles oracle, c'est à dire les modèles ayant été appris sur la séquence à reconnaître. Cependant, les performances ne sont pas uniformes en fonction du contexte : pour la hi-hat, les meilleurs performances sont obtenues avec un contexte long (modèle de pentagrammes), tandis que pour la caisse claire et la grosse caisse, des contextes plus courts doivent être utilisés. Le gain de performances offert par le modèle local est plus modeste. Les meilleurs résultats sont obtenus en considérant les supports $-4, -2, -1$ ou $-8, -2, -1$. Soulignons que ce modèle a l'avantage d'être non-supervisé.

Les performances offertes par les modèles par style et les modèles par style avec oracle sont très proches. Cela peut s'expliquer par deux phénomènes :

Paramètres	BD	SD	HH
Référence			
	79.4	59.6	76.7
Modèle oracle			
-1	**82.6**	63.3	79.2
-2,-1	82.0	**67.0**	80.6
-4,-1	81.7	64.6	80.9
-8,-1	82.3	63.8	80.3
-16,-1	81.0	63.2	80.2
-3,-2,-1	80.9	66.7	81.5
-4,-2,-1	82.2	65.7	82.5
-8,-2,-1	81.2	65.4	81.2
-16,-2,-1	82.4	66.0	82.1
-4,-3,-2,-1	78.7	66.0	**82.9**
-16,-8,-2,-1	81.4	64.7	81.3
Modèle local			
-1	80.8	60.2	77.9
-2,-1	81.3	60.6	**78.2**
-4,-1	81.2	**61.2**	77.6
-8,-1	81.0	60.9	77.8
-16,-1	81.2	60.1	77.7
-3,-2,-1	81.3	60.8	77.2
-4,-2,-1	**81.6**	61.1	77.6
-8,-2,-1	81.5	61.1	77.7
-16,-2,-1	**81.6**	60.8	77.2
-4,-3,-2,-1	81.1	61.0	77.5
-16,-8,-2,-1	81.5	60.1	76.5
Modèle par style			
-1	79.4	60.4	78.0
-2,-1	80.2	60.8	79.6
-4,-1	80.9	60.9	78.7
-8,-1	**81.2**	61.4	78.8
-16,-1	80.1	60.1	78.6
-3,-2,-1	78.1	**61.8**	**80.3**
-4,-2,-1	77.4	59.8	78.8
-8,-2,-1	77.2	59.9	78.6
-16,-2,-1	78.5	59.1	78.2
-4,-3,-2,-1	75.4	61.6	80.0
-16,-8,-2,-1	79.3	59.0	79.1
Modèle par style avec oracle			
-1	79.4	60.4	78.0
-2,-1	80.2	60.9	79.7
-4,-1	80.5	61.2	79.5
-8,-1	**81.2**	**61.9**	79.1
-16,-1	80.2	60.2	78.8
-3,-2,-1	78.1	61.8	**80.3**
-4,-2,-1	78.5	60.1	79.3
-8,-2,-1	78.7	59.8	79.3
-16,-2,-1	78.8	59.2	78.6
-4,-3,-2,-1	76.4	61.5	**80.3**
-16,-8,-2,-1	79.3	59.0	79.4
Minimisation de la complexité			
	81.3	61.7	80.4

TAB. 4.20 – Performances des méthodes de correction d'erreur supervisées (modèles de séquence) et non-supervisées

– L'identification du style réalisée par le modèle sans oracle est souvent correcte (dans 61% des cas).
– Même si le modèle par style utilisé est incorrect suite à une erreur de classification, ce modèle intègre tout de même des propriétés générales du jeu de la batterie pouvant s'appliquer à tous les styles, et donc suffisantes pour corriger les erreurs.

La méthode non-supervisée de minimisation de la complexité offre des performances similaires aux modèles par style. Ses performances pourraient sans doute être améliorées par une recherche plus exhaustive (plus de générations, et population plus grande, lors de la simulation de l'évolution), mais son intérêt est alors limité par son coût excessif en calculs.

4.7 Conclusion

Nous avons présenté dans ce chapitre un système complet de transcription de la piste de batterie d'un enregistrement musical multi-instrumental. L'originalité de ce système est qu'il traite en parallèle le signal à transcrire, et ce même signal pré-traité par la méthode d'accentuation de la piste de batterie présentée au chapitre précédent. Après avoir segmenté les signaux en en détectant les onsets, de nombreux paramètres acoustiques en sont extraits. La classification est effectuée à l'aide de machines à vecteurs de support, assurant un excellent compromis entre apprentissage et généralisation. Puisque certains des attributs extraits du signal original ne sont plus robustes en présence d'un accompagnement musical superposé à la batterie ; et que d'autres attributs ne sont pas robustes aux artefacts introduits par la méthode d'accentuation de la piste de batterie, nous avons eu recours à des techniques de sélection d'attributs pour éliminer les attributs trop peu robustes, et à deux approches de fusion (précoce et tardive) pour tirer au mieux partie de l'information complémentaire présente dans les deux signaux. Nos résultats montrent ainsi que les systèmes de classification effectuant une fusion des informations présentes dans les deux signaux sont plus performants que ceux exploitant ou le signal original, ou le signal dont la piste de batterie a été accentuée. Une propriété intéressante du pré-traitement d'accentuation de la piste de batterie mise en lumière dans nos expériences est également qu'il peut faciliter la généralisation, en faisant abstraction des différences de taille des fûts entre batteries. Nous avons également observé expérimentalement qu'un système de transcription robuste de soli de batterie ne doit pas nécessairement être entraîné sur des soli de batterie, mais sur des séquences avec un faible accompagnement instrumental, de manière à gagner en diversité dans la base d'apprentissage, et donc en pouvoir de généralisation. Les résultats de la sélection d'attributs ont révélé quels attributs étaient pertinents (et robustes) pour la détection de frappes de grosse caisse, caisse claire, et hi-hat. En particulier, ils ont montré la supériorité d'attributs ad-hoc (énergie dans des bancs de filtres adaptés) par rapport à des attributs classiques comme les MFCC. Nous avons également étudié comment des modèles de séquence, ou des techniques non-supervisées de minimisation de complexité de séquences peuvent contribuer à améliorer les résultats de la transcription, de façon certes modérée.

Nos résultats ont également montré quelques limites de notre approche. Tout d'abord les attributs sélectionnés ne permettent pas la séparation des classes dans certains sous-problèmes de classification rencontrés – quelques pistes quant aux situations mettant en difficulté notre système ont été dévoilées par l'analyse de la composition des vecteurs de support. Ensuite, les améliorations offertes par les modèles de séquence se sont avérées modérées, alors que l'analyse du corpus d'apprentissage laissait apparaître de fortes relations entre un symbole et son contexte. Nous suggérons que ce résultat s'explique non pas par l'impuissance des modèles de séquence en question, mais par la procédure visant à obtenir une représentation symbolique de la séquence, dans laquelle une partie de l'information est perdue par quantification et regroupement des onsets, et par le manque de fiabilité des probabilités a posteriori fournies en sortie des classifieurs. Notre intuition initiale selon laquelle les classifieurs produiraient des probabilités a posteriori proches du seuil de décision, mais du mauvais côté, en cas d'exemples difficiles est fausse : nos observations suggèrent plutôt que lorsqu'un classifieur commet une erreur, il ne "doute" pas. Le seul moyen d'améliorer la qualité des scores acoustiques est, comme nous l'avons vu plus haut, d'extraire de meilleurs attributs du signal.

En dépit de ces limites, les performances obtenues par notre système sont cependant acceptables pour des applications d'indexation et de transcription rythmique, et sont supérieures à celles d'autres systèmes, pourtant conçus pour le cas polyphonique, dont les performances se dégradent dès lors qu'un accompagnement instrumental est ajouté.

Nous allons désormais nous intéresser à un problème connexe à celui de la transcription : comment séparer au mieux la piste de batterie d'un enregistrement de musique. Nous avons déjà fourni une réponse simple avec le système d'accentuation de la piste de batterie du chapitre précédent. Nous apportons dans le chapitre qui suit plusieurs améliorations à ce système, en particulier en exploitant la transcription pour améliorer la qualité de la séparation. Nous introduirons également d'autres méthodes de séparation, et discuterons le problème suivant : faut-il d'abord séparer un signal pour mieux le transcrire, ou faut-il d'abord le transcrire pour mieux le séparer ?

Publications liées à ce chapitre

Les versions successives du système de transcription présenté dans ce chapitre ont été décrites dans différents articles.
Nos premiers travaux en transcription de signaux percussifs [GR03] traitaient le cas du Tabla (et non de la batterie) et soulignaient particulièrement l'intérêt des modèles de séquences pour améliorer les performances de la transcription. Leur extension et application à la batterie est décrite pour la première fois dans [GR04]. Plusieurs améliorations du système développé (notamment une évaluation plus approfondie des SVM), ainsi que son intégration à un système de requête par le contenu sont introduites dans [GR05e] et [GR05b]. L'extension au cas polyphonique est considérée dans [GR05c]. Notons que dans ce dernier article, ne sont utilisés que des attributs calculés sur le signal pré-traité, et qu'aucune sélection des attributs n'est effectuée. Les développements les plus récents, tels qu'ils sont décrits dans ce chapitre, sont présentés dans [GR07].
La base ENST-drums utilisée pour les évaluations est décrite dans [GR06b].

CHAPITRE 5

Extraction de la piste de batterie dans un signal de musique

Dans ce chapitre est étudié le problème de l'extraction de la piste de batterie à partir d'un signal de musique. Ce chapitre peut être vu comme une extension ou un approfondissement des méthodes présentées dans le chapitre 3. Cependant, notre objectif est différent : au chapitre 3, nous cherchions à accentuer la piste de batterie en n'utilisant aucune information a priori quant à la partition rythmique jouée par la batterie, puisque notre objectif était précisément d'obtenir cette partition. Cette tâche est peu contraignante quant à la qualité du signal extrait – la seule contrainte étant que le signal séparé permette l'extraction d'attributs apportant une information complémentaire aux attributs extraits du signal original. Nous avons vu au chapitre précédent que cette contrainte était satisfaite. Dans ce chapitre, notre objectif est d'extraire un signal le plus fidèle possible à la piste de batterie du signal de musique considéré – cette problématique étant exactement celle de la séparation de sources. Les applications envisagées sont essentiellement celles de remixage de la batterie dans des signaux de musique, mais ce problème n'en est pas pour autant déconnecté de celui de la transcription. Tout d'abord, si de telles méthodes de séparation de sources peuvent être développées, elles fourniront ainsi un pré-traitement efficace pour la transcription. Par ailleurs, comme nous allons le voir, des méthodes de séparation particulièrement efficaces peuvent être conçues si l'on connaît, a priori, une partition de ce qui est joué par le batteur. Nous soulignerons ainsi, dans ce chapitre, les relations entre les problèmes de transcription et séparation. Une brève vue d'ensemble des méthodes de séparation de sources génériques est donnée dans la section 5.1. Nous en explicitons les limites dans le cas du problème d'extraction de la piste de batterie, et présenterons quelques méthodes conçues spécifiquement pour la batterie. Dans la section 5.2, nous introduisons une méthode utilisant des masques temps/fréquence/sous-espace, qui peut être vue comme une extension de la séparation harmonique/bruit présentée au chapitre 3. Nous présentons dans la section 5.3 une autre méthode de séparation de sources proposée par Benaroya [Ben03], et voyons comment elle peut être mise en oeuvre et modifiée pour la séparation de la piste de batterie. Plusieurs des méthodes discutées dans ce chapitre font l'objet d'une évaluation objective dans la section 5.4.

5.1 Bref état de l'art

Nous donnons ici un bref état de l'art des méthodes de séparation de sources, principalement destiné à montrer la spécificité du problème de la séparation de la piste de batterie : nous montrons d'abord les limites des méthodes classiques, et nous présenterons ensuite quelques solutions qui y ont été apportées.

5.1.1 Séparation de sources

Dans le cas où l'enregistrement utilisé est multicanal et contient autant de canaux qu'il existe de sources sonores, la séparation peut être effectuée par des algorithmes classiques d'analyse en composantes indépendantes – *Independent Component Analysis* (ICA). Cette situation idéale ne correspond pas à celle à laquelle nous sommes confrontés, où les enregistrements sont au mieux stéréophoniques, et contiennent plus de deux sources. Quelques hypothèses quant à la procédure de mixage et au non-recouvrement des représentations temps/fréquence des sources nous ont permis, au chapitre 3 de mettre en oeuvre une méthode de séparation opérant sur des signaux stéréophoniques. Ces hypothèses n'étant pas toujours vérifiées, les performances obtenues sont insuffisantes, et cette méthode ne peut donc être vue que comme un pré-traitement.

Parmi les solutions proposées au problème de la séparation de sources avec un seul capteur, on distinguera plusieurs méthodes.

Méthodes supervisées : Modèle de source et refiltrage De telles méthodes nécessitent la formulation d'un modèle des sources à extraire, dont les paramètres doivent être appris sur des signaux isolés de chacune des sources. Il est ainsi possible de formuler un modèle du mélange des sources, dont l'estimation des paramètres à partir du mélange observé permet de déduire la contribution de chacune des sources. Les modèles mis en oeuvre sont divers : modèles statistiques comme les HMM dans [Row01], ou des réseaux bayésiens dans [VR04b], l'estimation des paramètres se faisant au maximum de vraisemblance ; ou représentation d'une source comme un "sac de trames" typiques, obtenues par quantification vectorielle [EW06]. La séparation d'une source se fait dans tous les cas par filtrage ou masquage. Dans l'application d'extraction de la piste de batterie, nous souhaitons séparer deux sources : la batterie, et les autres instruments non percussifs. La diversité des sources à séparer est problématique : il semble difficile de disposer d'un modèle capable, à lui seul, de décrire tous les sons percussifs et tous les sons non-percussifs.

Méthodes non-supervisées basées sur des critères psychoacoustiques Ellis présente dans [Ell96] un système d'analyse de signaux utilisant des règles de groupement issues de la psychoacoustique (par exemple des partiels évoluant simultanément seront perçus comme appartenant à la même source) pour grouper les trajectoires de partiels dans le plan temps/fréquence, et ainsi former des objets sonores. Une reformulation de cette méthode comme un problème de clustering des points temps-fréquence est donnée par Bach et Jordan dans [BJ06]. De telles méthodes sont particulièrement adaptées aux signaux harmoniques, mais ne permettent pas la séparation de sources bruitées, comme cela est requis pour la séparation de signaux percussifs. Même pour les instruments à percussion contenant une forte proportion de composantes harmoniques (toms), la décroissance des partiels est trop rapide pour assurer le suivi de leur trajectoire.

Méthodes non-supervisées d'élimination de la redondance Elles visent à obtenir une décomposition du spectrogramme comme une somme de quelques sources sonores. La seule hypothèse formulée quant à ces sources est que leur spectrogramme puisse être écrit comme le produit externe d'un profil spectral et d'une enveloppe temporelle – autrement dit que les sources peuvent être vues comme des processus aléatoires gaussiens stationnaires, modulés lentement en amplitude. La décomposition est obtenue soit par PCA puis par ICA – la méthode porte alors le nom d'analyse en sous espaces indépendants [CW00] ; par NMF [LS01] ; ou par des techniques de codage parcimonieux [Vir03]. Cependant, l'hypothèse formulée quant à la forme des spectres de ces sources n'est pas toujours valide pour les sources percussives : modulation de fréquence pour les toms et la grosse caisse, et transitoires au voisinage de la frappe rendent ce modèle inadéquat. En conséquence, l'application directe de ces méthodes peut se traduire par de la sur-séparation : le choc de la mailloche sur la grosse caisse et la composante périodique qui suit ce choc, ou la section où la fréquence fondamentale d'un tom est modulée, et la section où elle se stabilise, sont extraits comme des sources distinctes.

De plus, un inconvénient commun à toutes les méthodes non-supervisées est la nécessité de fixer a priori le nombre de sources à extraire, et de reconnaître a posteriori, parmi les sources séparées, celles qui correspondent à des instruments percussifs. Une mauvaise estimation a priori du nombre de sources peut conduire à une sur-séparation – le même instrument est séparé en deux composantes, et devient donc difficile à identifier, ou à une sous-séparation – un instrument harmonique et un instrument percussif jouant souvent simultanément sont séparés en une seule et même source. Deux solutions sont possibles : utiliser des connaissances a priori sur les sources à extraire (dans ce cas, il s'agit de séparation supervisée), ou utiliser des méthodes d'apprentissage statistique pour classer et regrouper les sources extraites.

Applications de ces méthodes à la piste de batterie L'application directe d'une des méthodes que nous venons de présenter a été effectuée par Virtanen et Helén dans [HV05] : des SVM sont utilisés pour reconnaître et sélectionner les sources percussives, parmi celles extraites par NMF. La thèse de FitzGerald [Fit04] contient également quelques exemples de séparation des pistes de grosse caisse, caisse claire et hi-hat[1] produites par ISA, même si elles ne sont données qu'à titre illustratif (l'application de séparation et de remixage n'est pas envisagée).

5.1.2 Méthodes de séparation spécifiques à la batterie

Observons tout d'abord que les systèmes de transcription suivant l'approche MatAda produisent, en plus de la transcription, des modèles temporels ou temps/fréquence de chacun des instruments de la batterie détectée. De tels modèles permettent ainsi, en combinaison avec la partition, de re-synthétiser une piste de batterie : pour chaque instrument, un train d'impulsions indiquant à quels instants ont été détectées des frappes de cet instrument est convolué par le modèle temporel de cet instrument (ou par le signal reconstitué à partir du modèle temps-fréquence). Cette solution a été proposée par Zils et al. dans [ZPDG02] pour des modèles temporels, et par Yoshii et al. dans [YGO05] pour des modèles temps/fréquence. Notons que dans les deux cas, la piste de batterie reconstruite perd les variations de dynamique et de timbre contenues dans le signal original, puisque chaque frappe de la batterie sera toujours synthétisée de la même façon. Le signal obtenu ne peut dès lors être ajouté ou soustrait au signal original pour réaliser un remixage de la piste de batterie.

En dehors de ces systèmes, deux méthodes de séparation exploitant des propriétés typiques des signaux de batterie ont été proposées.

Barry et al. observent dans [BFCL05] que les variations brusques du flux spectral dans les signaux de musique sont principalement dues aux instruments percussifs. Ils proposent donc de moduler le spectrogramme par une mesure d'impulsivité déduite du SEF. Cette méthode, extrêmement peu coûteuse en calculs, n'extrait cependant que la composante transitoire de chaque instrument percussif.

Nous avons introduit dans [GR05d] une technique de séparation spécifique à la batterie qui est décrite et étendue dans la section suivante. Elle possède plusieurs avantages : tout d'abord, elle ne nécessite pas de connaître a priori le nombre de sources à extraire, puisqu'elle modélise la piste de batterie comme une seule et même source - de fait, elle ne requiert pas non plus l'identification des sources extraites. Ensuite, elle est "conservative", au sens où aucune information (de phase, par exemple), n'est perdue lors de l'opération d'analyse et de synthèse, permettant l'extraction d'un signal pouvant être ajouté ou soustrait au signal original pour les applications de remixage. Enfin, elle est non-supervisée, et ne demande que l'apprentissage de paramètres génériques pouvant décrire une large gamme de signaux.

[1] Notons que nous ne nous intéressons pas ici à l'extraction des pistes individuelles de grosse caisse, caisse claire, et hi-hat. Nous nous intéressons seulement à la reconstruction de la piste de batterie du signal original, en tant qu'une seule et unique source.

5.2 Filtrage temps/fréquence/sous-espace (TFS)

5.2.1 Principe

Comme nous l'avons vu dans les sections 3.2 et 3.4, un signal de musique peut être analysé de manière à obtenir une décomposition harmonique/bruit dans chacune des sous-bandes d'un banc de filtres. Soit x_{hk} (resp. x_{rk}) la composante déterministe (resp. stochastique) extraite dans le signal de sous-bande issu de la k-ième voie du banc de filtres. Le banc de filtres que nous utilisons étant multi-résolution, ces signaux n'ont pas tous la même fréquence d'échantillonnage. Soient $\hat{x}_{hk}$ (resp. $\hat{x}_{rk}$) leur version pleine bande, obtenue par expansion et application du filtre de synthèse. Au chapitre 3, nous avons simplement utilisé les composantes stochastiques dans chacune des bandes pour produire un signal $\sum_{k=1}^{8} \hat{x}_{rk}$ où le contenu percussif est accentué. Nous avons montré au chapitre précédent l'intérêt offert par ce signal pour les applications de transcription. Cependant, la qualité de ce signal est insuffisante pour les applications de séparation. En effet, ce signal contient, en plus des composantes stochastiques issues des percussions, les composantes stochastiques issues des autres instruments (choc des marteaux sur les cordes de piano par exemple). Par ailleurs, la grosse caisse et la caisse claire contiennent quelques composantes déterministes qui doivent être présentes dans le signal reconstruit.

Nous proposons alors de reconstruire la piste de batterie en appliquant des gains variables dans le temps à chacune des composantes déterministes et stochastiques de sous-bande :

$$s(n) = \sum_{k=1}^{8} \alpha_{hk}(n)\hat{x}_{hk}(n) + \alpha_{rk}(n)\hat{x}_{rk}(n) \tag{5.1}$$

Les gains (α_{hk}) et (α_{rk}) permettent de ne sélectionner dans le signal reconstruit que les composantes qui peuvent être associées à des instruments de la batterie. Nous présentons dans les sections qui suivent comment ces gains sont obtenus.

5.2.2 Masques temps/fréquence/sous-espace (TFS)

Tout d'abord, dans une phase d'apprentissage, la décomposition décrite dans les sections 3.2 et 3.4 est appliquée à une frappe isolée de chacun des instruments considérés (grosse caisse, caisse claire, et hi-hat). Soit i un indice identifiant l'instrument considéré, et soit N la longueur (en nombre d'échantillons) des signaux considérés. À l'issue de cette analyse, sont produites les composantes déterministes et stochastiques des signaux de sous-bande, notées respectivement $\hat{x}^i_{hk}$ et $\hat{x}^i_{rk}$. L'enveloppe d'amplitude de chacun de ces signaux est approximée par une exponentielle décroissante en suivant la procédure décrite en A.4, produisant les enveloppes e^i_{hk} et e^i_{rk}. Ces deux enveloppes modélisent ainsi l'évolution temporelle de l'amplitude des composantes déterministes et stochastiques du signal dans chacune des sous-bandes. Notons que cette opération peut être répétée sur plusieurs instances de frappes isolées de chacun des instruments, auquel cas les enveloppes d'amplitude extraites de chaque frappe sont moyennées avant l'estimation de l'exponentielle approximant cette enveloppe moyenne. La figure 5.1 illustre les enveloppes d'amplitude extraites de la base, pour les trois instruments considérés.

Nous soulignons qu'en raison du nombre limité de bandes utilisées pour la décomposition, et du lissage des enveloppes d'amplitude réalisées lors de leur approximation par une exponentielle, les modèles appris dépendent peu de la batterie utilisée – le modèle que nous utilisons ici est suffisament peu expressif pour ne pas sur-apprendre le timbre ou l'"accordage" spécifique d'une batterie.

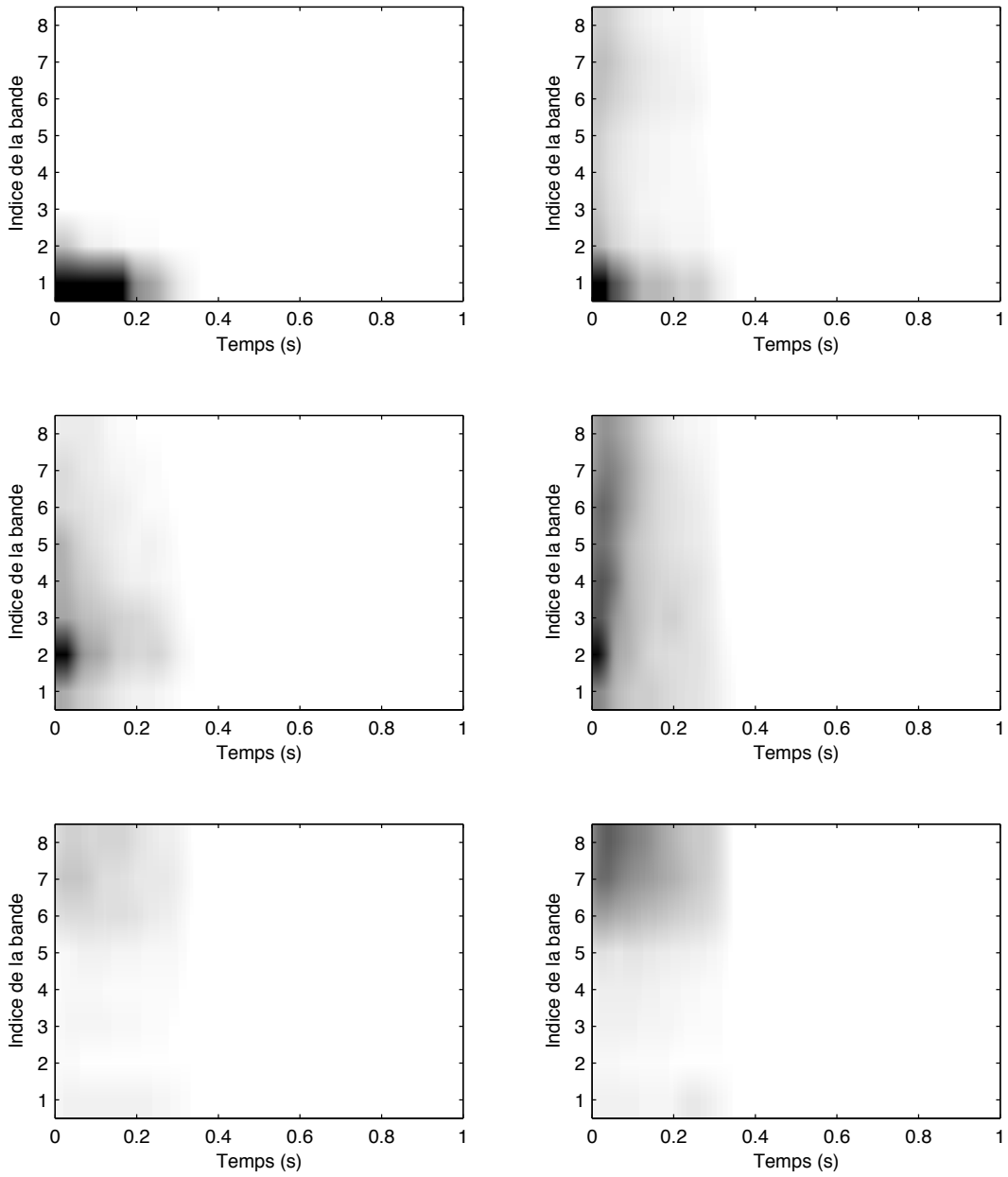


FIG. 5.1 – Enveloppes d'amplitude pour chacun des signaux de sous-bande. À gauche : partie déterministe ; À droite : partie stochastique. De haut en bas : grosse caisse, caisse claire, hi-hat

5.2.3 Détection des frappes de batterie

L'étape suivante consiste à détecter les occurrences des frappes de grosse caisse, de caisse claire et de hi-hat (ou des autres instruments pour lesquels on a estimé des masques) à partir du signal de musique dont on cherche à extraire la piste de batterie. N'importe quel détecteur ou système produisant une transcription peut être utilisé à cet effet, par exemple le système de transcription décrit au chapitre précédent, ou une annotation de référence si elle est disponible. Nous présentons ici une méthode de détection simplifiée, semblable à la procédure de mise en correspondance utilisée dans [YGO04a], qui consiste à détecter une frappe sur l'instrument i à l'onset n_0 lorsque la fonction de détection $D^i(n_0)$ définie ci-dessous dépasse un seuil τ_i fixé à l'avance[2] :

$$D^i(n_0) = \sum_{k=1}^{8} \sum_{n=0}^{N-1} \left[e^i_{hk}(n)\hat{x}_{hk}(n_0+n) + e^i_{rk}(n)\hat{x}_{rk}(n_0+n) \right]^2 \tag{5.2}$$

Cette fonction de détection est une mesure d'énergie pondérée pour ne prendre en compte que les sous-bandes, et les composantes harmoniques/bruit caractéristiques de chaque instrument à détecter.

5.2.4 Remasquage

Si K_i frappes de l'instrument i ont été détectées aux instants $\{t^i_1, \ldots, t^i_{K_i}\}$ (t_k est exprimé en échantillons), on définit la fonction $\mathbb{I}^i(n)$ selon :

$$\mathbb{I}^i(n) = \sum_{k=1}^{K_i} \delta(t^i_k - n) \tag{5.3}$$

Si le signal n'avait contenu que les événements percussifs décrits par $\mathbb{I}^i(n)$, son enveloppe d'amplitude dans chacun des signaux de sous-bandes aurait pu être approximée par :

$$\hat{e}^i_{hk}(n) = (\mathbb{I}^i * e^i_{hk})(n) \tag{5.4}$$
$$\hat{e}^i_{rk}(n) = (\mathbb{I}^i * e^i_{rk})(n) \tag{5.5}$$

Les gains variables sont alors calculés selon :

$$\alpha_{hk}(n) = \max_i \hat{e}^i_{hk}(n) \tag{5.6}$$
$$\alpha_{rk}(n) = \max_i \hat{e}^i_{rk}(n) \tag{5.7}$$

Intuitivement, ces gains recréent dans chaque sous-bande et pour chaque composante harmonique/bruit l'enveloppe temporelle que le signal aurait eu s'il n'avait contenu que les événements percussifs décrits par tous les $\mathbb{I}^i(n)$. L'utilisation du maximum pour estimer l'enveloppe temporelle ou le spectre d'un mélange à partir du spectre ou des enveloppes des sources individuelles est discuté dans [Row01].

Notons que l'algorithme que nous avons présenté en [GR05d] peut être décrit par le même formalisme – dans ce cas, les masques e^i_{rk} sont binaires et empiriquement définis pour chaque instrument, et les e^i_{hk} sont nuls.

[2] Le même post-traitement de normalisation de la fonction de détection qu'en 4.2.2 peut être appliqué, de façon à utiliser un même seuil τ pour tous les instruments

5.3 Filtrage pseudo-Wiener et modèles spectraux

Nous présentons maintenant une méthode supervisée développée par Benaroya dans [Ben03]. Après en avoir résumé le principe dans la section 5.3.1, nous en discutons la mise en oeuvre dans la section 5.3.2, où nous en proposons diverses modifications pour améliorer ses performances en séparation de la piste de batterie.

5.3.1 Principe

5.3.1.1 Modèle de signal, filtrage de Wiener

Considérons deux processus gaussiens stationnaires s_1 et s_2, de d.s.p $\sigma_1^2(f)$ et $\sigma_2^2(f)$. Le filtre de Wiener, dont la réponse $H_i(f)$ est donnée ci-dessous, permet alors d'obtenir la meilleure estimée de s_i à partir du mélange $s_1 + s_2$:

$$H_i(f) = \frac{\sigma_i^2(f)}{\sigma_1^2(f) + \sigma_2^2(f)} \tag{5.8}$$

Les sources que nous souhaitons séparer ne peuvent être considérées que comme localement stationnaires, et ne peuvent pas être décrites par une seule d.s.p. De manière à prendre en compte ces deux phénomènes, les sources peuvent alors être considérées comme un mélange de processus gaussiens stationnaires dans des proportions variant lentement dans le temps :

$$s_i(n) = \sum_{l \in L_i} a_l(n) b_l(n) \tag{5.9}$$

Où $a_i(n) \geq 0$ est un gain lentement variable et $b_l(n)$ est un processus gaussien stationnaire de d.s.p σ_l^2, et L_i un ensemble d'indices. Les d.s.p σ_l^2 seront par la suite appelées modèles spectraux.

Dans ce cas, la source s_i peut être estimée à partir du procédé suivant, décrit dans [BDBG03] :

1. Une représentation temps-fréquence $\hat{X}(m,k)$ de x est obtenue, par exemple à l'aide d'un banc de filtres ou d'une TFCT. m est l'indice de la trame, $k \in \{0, \dots K-1\}$ est l'indice de la bande ou du canal.
2. Pour chaque trame m, la densité spectrale de puissance observée est décomposée comme une somme des modèles spectraux : $|\hat{X}(m,k)|^2 \approx \sum_{l \in L_1 \cup L_2} \hat{a}_l(m) \sigma_l^2(k)$. Nous verrons dans la section suivante comment cette décomposition peut être effectuée.
3. La représentation temps-fréquence de la source s_i est estimée par :

$$|\hat{S}_i(m,k)|^2 = \frac{\sum_{l \in L_i} \hat{a}_l(m) \sigma_l^2(k)}{\sum_{l \in L_1 \cup L_2} \hat{a}_l(m) \sigma_l^2(k)} |\hat{X}(m,k)|^2 \tag{5.10}$$

   Cette opération correspond à un filtrage de Wiener pour des processus dont la d.s.p peut être considérée comme localement stationnaire, et porte le nom de filtrage pseudo-Wiener.

5.3.1.2 Décomposition non-négative d'un spectre sur une base de modèles spectraux

L'étape 2 de la méthode présentée précédemment requiert l'approximation d'un vecteur positif $|\hat{X}(m,k)|^2$ comme une somme pondérée, par des coefficients $\hat{a}_l(m) \geq 0$, de vecteurs positifs $\left(\sigma_l^2(k)\right)_{l \in L_1 \cup L_2}$. Définissons :

$$\mathbf{V} = \left[\ |\hat{X}(m,0)|^2 \quad \dots \quad |\hat{X}(m,K-1)|^2 \ \right]^T \tag{5.11}$$

$$\mathbf{H} = \left[\ \hat{a}_0(m) \quad \dots \quad \hat{a}_{L-1}(m) \ \right]^T \tag{5.12}$$

$$\mathbf{W} = \begin{bmatrix} \sigma_0^2(0) & \dots & \sigma_{L-1}^2(0) \\ \vdots & \ddots & \vdots \\ \sigma_0^2(K-1) & \dots & \sigma_{L-1}^2(K-1) \end{bmatrix} \tag{5.13}$$

Avec ces notations, il s'agit de factoriser $\mathbf{V}$ sous la forme $\mathbf{V} \approx \mathbf{WH}$. Notons qu'à la différence des problèmes classiques de NMF, $\mathbf{W}$ est ici entièrement connue et n'a pas à être déterminée. Une règle multiplicative minimisant itérativement la divergence de Kullback-Leibler entre $\mathbf{V}$ et $\mathbf{WH}$ est donnée dans [LS01] :

$$H_l^{n+1} = H_l^n \frac{\sum_{k=0}^{K-1} W_{lk} V_k / (WH^n)_k}{\sum_{k=0}^{K-1} W_{lk}} \tag{5.14}$$

Ou, reprenant nos notations[3] :

$$\hat{a}_l^{n+1}(m) = \hat{a}_l^n(m) \frac{\sum_{k=0}^{K-1} \sigma_l^2(k) \frac{|\hat{X}(m,k)|^2}{E^n(m,k)}}{\sum_{k=0}^{K-1} \sigma_l^2(k)} \tag{5.15}$$

$$E^n(m,k) = \sum_{l=0}^{L-1} \sigma_l^2(k) \hat{a}_l^n(m) \tag{5.16}$$

Notons que des contraintes de parcimonie peuvent être utilisées pour imposer la non-nullité d'un nombre réduit de coefficients $a_l(m)$, donnant lieu à de nouvelles règles de mise à jour [BDBG03]. De telles contraintes sont par exemple utilisées par Cont dans [Con06] pour décomposer la d.s.p observée sur une base de d.s.p correspondant à différentes notes d'un même instrument, à des fins de suivi de partition en contexte polyphonique.

5.3.1.3 Extraction d'une base de modèles spectraux

L'approche que nous venons de présenter est supervisée au sens où elle nécessite l'apprentissage de modèles spectraux pour les deux sources à séparer (ici, batterie et accompagnement instrumental). Benaroya et al. proposent dans [BDBG03] plusieurs méthodes afin d'obtenir une famille de d.s.p décrivant chacune des sources. La méthode la plus efficace est une méthode de clustering. Pour chacune des sources, est considéré un enregistrement (ou plusieurs enregistrements concaténés) de cette source isolée. Une représentation temps-fréquence en est extraite. Les trames de cette représentation sont regroupées à l'aide d'un algorithme de clustering (les k-moyennes [DHS01] par exemple), utilisant un critère de corrélation. Les centroïdes de chaque cluster définissent les $\left(\sigma_l^2(k)\right)_{l \in L_i}$.

5.3.2 Mise en oeuvre et améliorations pour la séparation de la piste de batterie

Nous proposons maintenant différentes améliorations de cette méthode, pour sa mise en oeuvre sur des signaux de musique dont on souhaite séparer la piste de batterie.

[3] Cette règle de mise à jour permet une convergence plus rapide que celle dérivée par Benaroya dans [Ben03] et utilisée par exemple dans [BBG06].

5.3.2.1 Apprentissage des modèles spectraux

Dans cette étude, nous utilisons $\#K_1 = 16$ modèles spectraux pour la batterie, et $\#K_2 = 128$ modèles spectraux pour la musique d'accompagnement.

Nous avons observé qu'en utilisant l'algorithme de clustering avec critère de corrélation décrit dans [BDBG03], les d.s.p extraites des signaux de batterie seule contiennent des mélanges, dans des proportions diverses, de la caisse claire, de la hi-hat et de la grosse caisse. De tels mélanges sont redondants, puisqu'ils peuvent être obtenus par combinaison linéaire non-négative des d.s.p des frappes isolées. En conséquence, nous avons suivi une autre approche pour extraire les 16 d.s.p à partir des enregistrements de batterie seule : ces d.s.p ont été extraites par NMF. Quelques exemples tirés du dictionnaire extrait par clustering et par NMF sont donnés dans la figure 5.2 (première et deuxième colonne). On observe que les éléments extraits par NMF sont moins redondants.

Cette approche n'est pas applicable à l'extraction d'un dictionnaire pour la musique d'accompagnement. En effet, l'application d'une NMF avec un nombre aussi élevé de composantes fournit comme dictionnaire de d.s.p, un ensemble de raies couvrant toutes les fréquences les plus basses du spectre. Cette représentation, si elle permet effectivement de bien décrire les d.s.p des signaux de musique d'accompagnement, n'est pas assez spécifique. Nous avons jugé satisfaisant le dictionnaire de d.s.p appris par clustering (dernière colonne sur la figure 5.2) – ses éléments sont principalement des peignes harmoniques de fréquence fondamentale et de distributions des amplitudes des partiels variées.

5.3.2.2 Adaptation par enrichissement du dictionnaire

La deuxième amélioration que nous proposons consiste en la procédure d'adaptation suivante : durant l'étape de décomposition, le dictionnaire de d.s.p utilisé pour la batterie $\left(\sigma_l^2(k)\right)_{l\in L_1}$ est enrichi par la d.s.p de la composante stochastique du signal x observée à la trame m. En effet, ce modèle spectral additionnel fournit une bonne estimée de la d.s.p de la contribution de la batterie dans le signal observé. En particulier, il permet une bonne représentation de la composante stochastique du signal de batterie, qui n'est pas prise en compte par les 16 modèles spectraux $\left(\sigma_l^2(k)\right)_{l\in L_1}$.

5.3.2.3 Utilisation d'une représentation temps/fréquence à résolution variable

La troisième amélioration que nous proposons est relative au choix de la taille de fenêtre utilisée pour la décomposition temps/fréquence (TFCT). Un compromis doit être trouvé entre les fenêtres courtes et fenêtres longues. Les premières sont adaptées aux segments contenant des frappes de batterie ou des événements très localisés dans le temps, mais disposent d'une mauvaise résolution fréquentielle et produisent des artefacts désagréables lorsque les coefficients $a_k(m)$ varient rapidement entre fenêtres courtes adjacentes. Les fenêtres longues, efficaces pour les segments contenant les parties entretenues des instruments non-percussifs, peuvent créer des phénomènes de pré-echo, ou peuvent adoucir les transitoires dans le signal reconstruit.

De manière à éviter ce choix difficile, nous utilisons une taille de fenêtre variable dans la décomposition temps/fréquence. Cette pratique est courante en codage audio, pour éviter les problèmes de pré-écho [BG02]. Deux tailles de fenêtres sont utilisées, $L_1 = 2048$ et $L_2 = 256$. Les dictionnaires de modèles spectraux sont appris pour ces deux tailles de fenêtres. Le signal, échantillonné à 44.1kHz, est traité par blocs de 2048 échantillons avec un chevauchement entre blocs de 50%. Si le bloc examiné contient un onset (comme détecté dans la section 4.2), il est traité comme huit fenêtres de 256 échantillons, dans le cas contraire comme une seule fenêtre. De manière à assurer une reconstruction parfaite, des fenêtres de transition sont définies en cas de changement de taille. Les fenêtres, appliquées durant les phases d'analyse et synthèse, sont construites à partir d'arches de sinusoïdes, comme cela est fait par exemple pour le codeur audio MPEG-2 AAC (Figure 5.3).

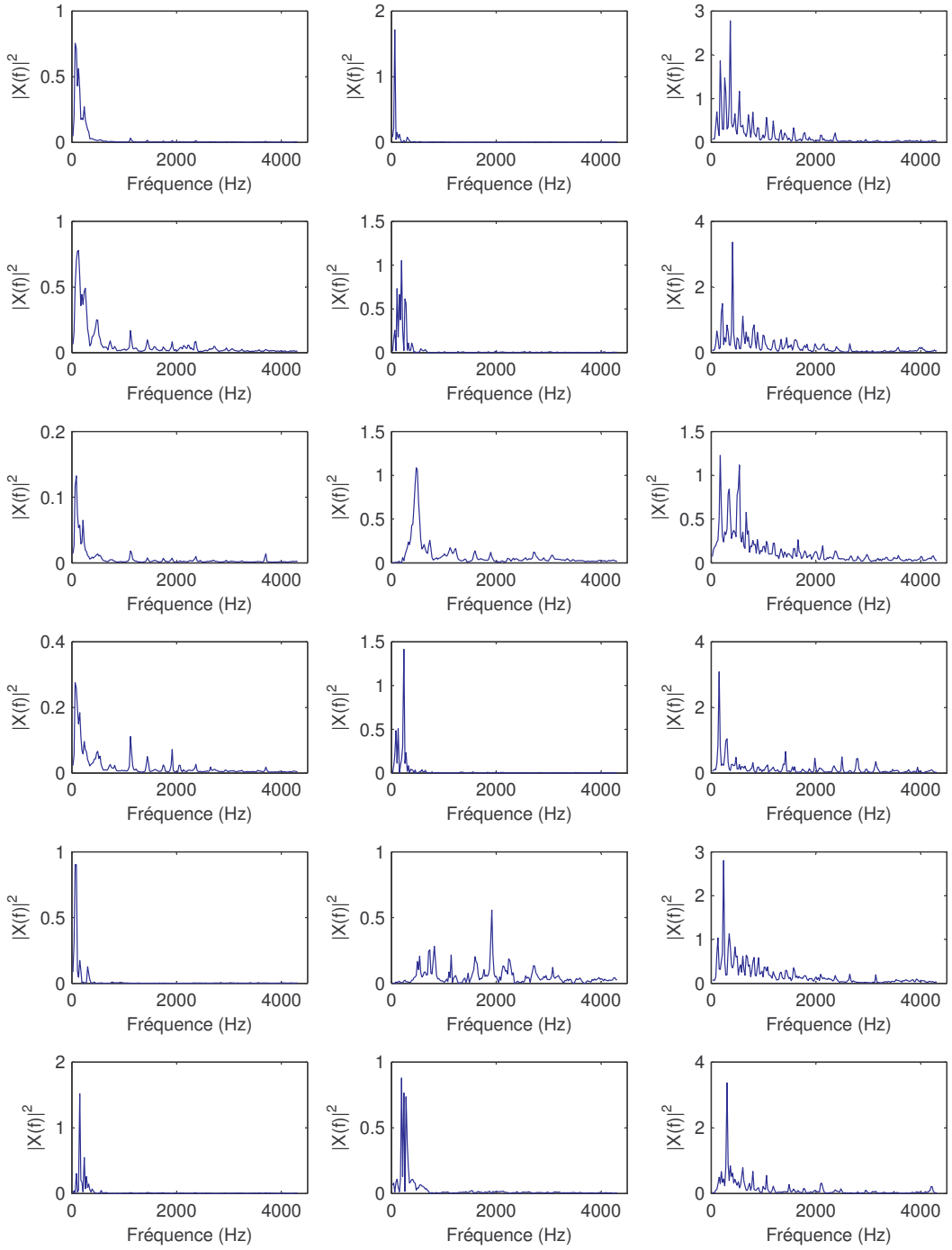


FIG. 5.2 – Quelques exemples de d.s.p tirées des dictionnaires appris sur des signaux des deux classes à séparer. De gauche à droite : dictionnaire pour la batterie, appris par clustering ; pour la batterie, appris par NMF ; pour l'accompagnement, appris par clustering

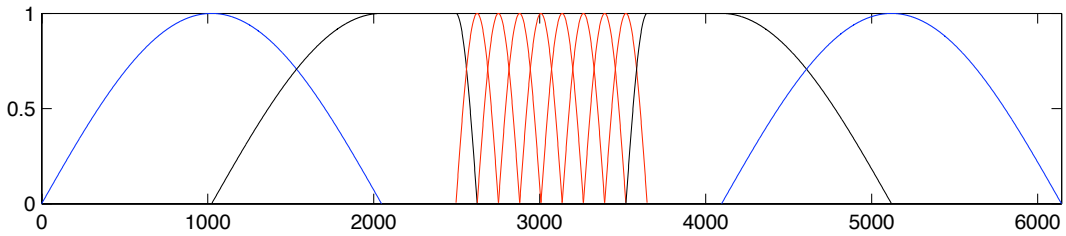


FIG. 5.3 – Fenêtres longues, courtes et de transition utilisées pour l'analyse et la synthèse

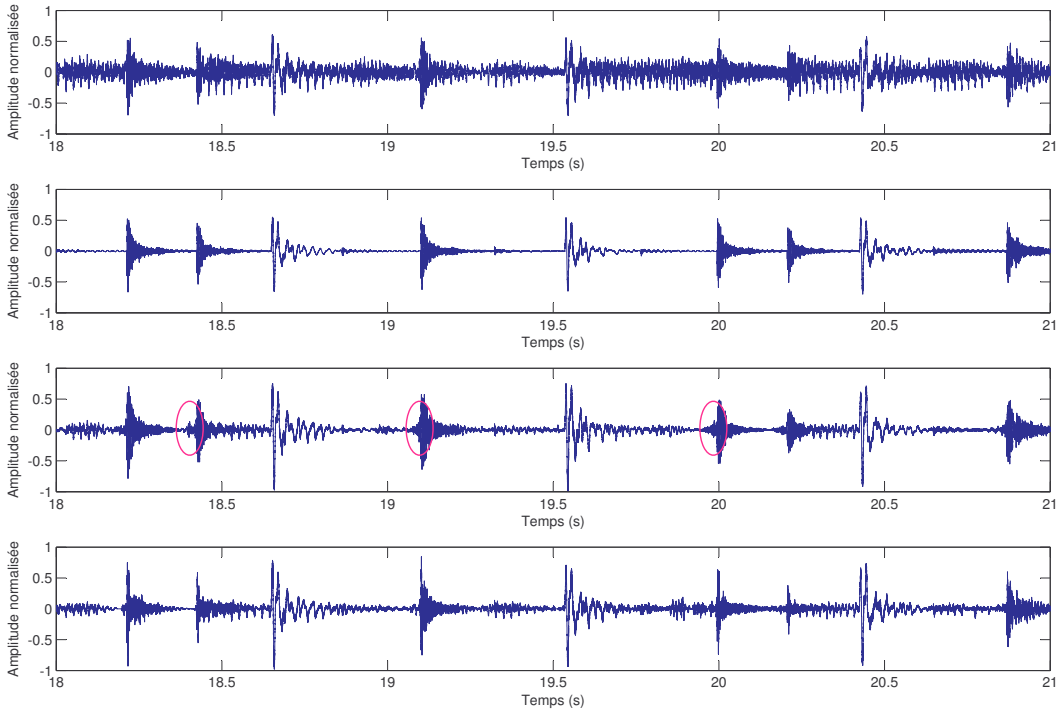


FIG. 5.4 – Signal de musique original, piste de batterie originale, piste de batterie séparée par filtrage pseudo-Wiener, piste de batterie séparée par filtrage pseudo-Wiener avec adaptation de la taille de fenêtre

La figure 5.4 illustre l'apport de cette méthode. Le pré-écho observé pour les frappes de caisse claire dans le troisième signal (filtrage pseudo-Wiener avec taille de fenêtre fixe) disparaît lors de l'utilisation d'une taille de fenêtre adaptative.

5.4 Résultats expérimentaux

5.4.1 Évaluation objective

5.4.1.1 Corpus et protocole

L'évaluation objective est réalisée comme au chapitre précédent sur les séquences *minus one* du corpus ENST-drums (voir 4.6.1.1). Nous évaluons la séparation de la piste de batterie dans trois situations : lorsque l'accompagnement est atténué par rapport à la batterie (de -6 dB), lorsque l'accompagnement est équilibré par rapport à la batterie (0 dB), et lorsque l'accompagnement est amplifié par rapport à la batterie (de 6 dB).

Contrairement à d'autres travaux en séparation de sources utilisant à des fins d'évaluation des mélanges synthétiques de signaux sans relation les uns avec les autres, ou des signaux synthétiques produits par exemple par des échantillonneurs ou des synthétiseurs à table d'ondes, les signaux que nous utilisons ici sont à la fois naturels et conçus pour être mélangés – seule la procédure de mixage des signaux est artificielle. Une telle variété et quantité de signaux d'évaluation contrastent avec la taille modeste, se limitant parfois à quelques secondes seulement, des signaux utilisés dans les évaluations des méthodes de séparation de sources proposées dans la littérature.

Soulignons que certaines des méthodes que nous évaluons demandent un apprentissage, par exemple pour l'estimation des masques TFS ou des modèles spectraux. Pour cet apprentissage, les signaux originaux de batterie et d'accompagnement ont été utilisés. Une telle procédure peut favoriser les approches supervisées, puisqu'on évalue ici leur capacité à séparer les signaux sur lesquelles elles ont été entraînées. Cependant, le peu de degrés de liberté permis par les paramètres des masques TFS, la généricité des modèles appris dans le cadre du filtrage pseudo-Wiener, et la diversité de la base d'apprentissage laissent suggérer que les masques et modèles appris sont suffisamment génériques pour éviter le surapprentissage. Cela explique peut-être pourquoi cette pratique, moins rigoureuse qu'une validation croisée, est courante dans la littérature relative à la séparation de sources.

5.4.1.2 Métriques

Les mesures de performance utilisées sont celles définies dans [GBVF03]. Soit s_d (resp. s_a) le signal original de batterie (resp. d'accompagnement). L'estimée $\hat{s}_d$ de la piste de batterie fournie par l'algorithme à évaluer peut être projetée sur les signaux originaux de batterie et d'accompagnement :

$$\hat{s}_d = \langle \hat{s}_d, s_d \rangle s_d + \langle \hat{s}_d, s_a \rangle s_a + \epsilon_{artif} \tag{5.17}$$

où ϵ_{artif} est le résiduel à l'issue de ces deux projections. Le rapport signal à distorsion – *Signal to Distortion Ratio* (SDR) est une mesure globale de la qualité de séparation, tandis que le SIR et le rapport signal à artefacts – *Signal to Artefact Ratio* (SAR) mesurent respectivement la quantité d'accompagnement, et d'artefacts de reconstruction ou de séparation présents dans le signal séparé. Ils sont définis comme suit :

$$SDR = 10\log_{10}\frac{||\langle\hat{s}_d, s_d\rangle s_d||^2}{||\langle\hat{s}_d, s_a\rangle s_a + \epsilon_{artif}||^2} \quad (5.18)$$

$$SIR = 10\log_{10}\frac{||\langle\hat{s}_d, s_d\rangle s_d||^2}{||\langle\hat{s}_d, s_a\rangle s_a||^2} \quad (5.19)$$

$$SAR = 10\log_{10}\frac{||\langle\hat{s}_d, s_d\rangle s_d + \langle\hat{s}_d, s_a\rangle s_a||^2}{||\epsilon_{artif}||^2} \quad (5.20)$$

5.4.1.3 Résultats

Les résultats sont donnés dans la table 5.1, pour différents algorithmes[4] :

Modulation d'amplitude consiste à utiliser le système de transcription décrit au chapitre précédent pour détecter les onsets correspondant à des frappes de batterie, et à moduler le signal en amplitude par une enveloppe exponentielle décroissante, de constante de temps égale à 100 ms, après chacun de ces onsets.

NMF+SVM est une réimplémentation de la méthode décrite par Helén dans [HV05], en utilisant des classifieurs entraînés sur l'ensemble de la base de données.

Modulation spectrale est la méthode présentée dans [BFCL05], utilisant l'implémentation de référence de FitzGerald avec les paramètres optimaux $\Psi = 1$, $T = 6$ dB, et $N = 8192$.

ICA par sous-bande est la méthode décrite en 3.3.1.2, extrayant des sources percussives à partir d'enregistrements stéréophoniques (toutes les autres méthodes sont évaluées sur des signaux monophoniques).

Projection espace bruit est la projection sur l'espace bruit, dans chacune des sous-bandes, comme décrit en 3.4.

Accentuation batterie est la combinaison des deux traitements précédents, comme utilisé au chapitre précédent pour la transcription.

Filtrage TFS, détecteur simple est le filtrage TFS décrit dans la section 5.2, utilisant un détecteur de frappes de batterie simple donné dans l'équation 5.2.

Filtrage TFS, bon détecteur est le filtrage TFS décrit dans la section 5.2, utilisant le détecteur de frappes de batterie décrit au chapitre précédent.

Filtrage TFS, oracle est le filtrage TFS décrit dans la section 5.2, utilisant comme partition l'annotation exacte du signal à séparer.

Pseudo-Wiener est l'approche décrite dans la section 5.3, sous sa formulation originale par Benaroya et al.

Pseudo-Wiener, amélioré utilise les améliorations que nous avons proposées dans la section 5.3.2.

Pour les signaux où la batterie est prédominante ou équilibrée avec l'accompagnement, les meilleurs résultats sont obtenus avec le filtrage pseudo-Wiener. Dans tous les cas, les améliorations que nous proposons conduisent à de meilleures performances. Cette méthode produit également de bons résultats quand la musique d'accompagnement est prédominante. Des résultats comparables sont obtenus avec le filtrage TFS, utilisant le module de transcription de la piste de batterie du chapitre précédent. Sans surprise, les performances sont encore meilleures lorsque la partition exacte est connue (filtrage TFS avec oracle).

Les améliorations offertes par le filtrage TFS par rapport à une simple projection sur l'espace bruit se traduisent par une augmentation du SDR et du SIR. Cependant, la projection sur l'espace bruit peut être considérée comme une méthode plus conservatrice, au sens où elle introduit moins d'artefacts dans le signal séparé.

[4] Le lecteur pourra se faire une impression de la qualité des signaux produits en écoutant les exemples à l'adresse suivante : `http://www.tsi.enst.fr/~gillet/ENST-drums/separation/`.

	Acc. −6 dB			Acc. +0 dB			Acc. +6 dB		
Méthode	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
Modulation d'amplitude	3.9	11.2	6.1	1.2	5.2	4.9	-3.5	-1.2	3.7
NMF+SVM	5.2	14.4	6.2	2.2	**10.7**	3.5	-1.4	**6.9**	0.2
Modulation spectrale	0.7	13.8	1.3	-0.8	8.0	0.9	-3.9	2.1	0.0
ICA par sous-bande	5.7	10.0	9.7	0.1	4.9	5.9	-6.3	-2.2	2.6
Projection espace bruit	8.3	10.2	14.5	3.0	4.3	**11.5**	-2.7	-1.6	8.9
Accentuation batterie	8.7	10.0	13.2	3.4	5.2	11.4	-2.2	-1.5	**9.0**
TFS, détecteur simple	7.6	14.0	9.6	3.4	6.8	7.7	-2.4	-0.6	6.3
TFS, bon détecteur	7.5	**15.9**	8.7	4.6	10.0	7.1	0.4	4.1	4.7
TFS, oracle	8.8	15.8	8.9	4.8	**10.7**	7.5	**0.6**	4.8	5.0
Pseudo-Wiener	8.6	10.4	**14.8**	3.1	9.4	5.1	-0.4	4.8	2.9
Pseudo-Wiener, amélioré	**10.1**	15.7	12.2	**5.5**	**10.7**	8.0	0.2	5.1	3.9

TAB. 5.1 – Rapports signal à distorsion, signal à interférences, et signal à artefacts pour diverses méthodes de séparation de la piste de batterie, sur les séquences *minus one* du corpus ENST-drums

Nous insistons également sur le fait que la méthode proposée dans [HV05] obtient des SIR élevés – illustrant ainsi sa capacité à discriminer fortement la batterie des autres instruments. Cependant, elle se caractérise, tout comme la modulation spectrale, par des SAR particulièrement bas. Cela souligne les difficultés rencontrées par les méthodes qui tentent de reconstruire le signal à partir d'une représentation temps-fréquence synthétique, plutôt que de filtrer le signal original. En particulier, ces méthodes sont confrontées au problème de la reconstruction de la phase à partir de la TFCT, et les métriques que nous utilisons sont sensibles aux erreurs de phase.

5.4.2 Vers de nouvelles métriques ?

Nos résultats se heurtent aux limites des métriques objectives utilisées : comme nous venons de le voir, les erreurs de reconstruction de la phase handicapent certaines méthodes. Ces erreurs de reconstruction sont effectivement gênantes pour les applications de remixage, ou le signal de batterie séparé va être ajouté ou soustrait au signal original, et demande donc d'avoir une phase synchrone avec celle du signal original – il s'agissait là d'une de nos motivations à utiliser la représentation banc de filtres + séparation harmonique/bruit, qui permet une reconstruction parfaite. Cependant, dans les applications où le signal séparé n'a pas à être combiné au signal original, la perte de l'information de phase n'est plus gênante. Dans ce cas, des métriques robustes aux erreurs de phase doivent être envisagées – par exemple, on pourrait mesurer la norme de la différence entre les spectrogrammes des deux sources, ou même envisager une représentation temps/fréquence perceptuelle (banc de filtres en bandes critiques par exemple).

Les mesures objectives utilisées sont par ailleurs incapables d'évaluer si les différences entre le signal original et le signal séparé sont audibles ou non – celles-ci pourraient en effet se trouver en dessous du seuil de masquage, et donc inaudible. Des rapports distorsion/interférences/artefacts à masque pourraient par exemple être considérés.

Enfin, ces métriques pénalisent tout autant les erreurs de séparation sur la partie entretenue de la frappe de batterie que sur son attaque. Il serait intéressant de considérer une métrique mesurant la capacité de la méthode de séparation employée à correctement reproduire le caractère percussif et les transitoires de la source considérée, propriétés essentielles pour la batterie. Cela peut être obtenu en comparant des mesures globales d'impulsivité ou de percussivité, comme celles définies en 3.3.2, ou bien en disposant d'un modèle génératif des signaux de batterie, dont les paramètres seraient appris

sur le signal original, nous permettant de calculer sa vraisemblance à partir du signal séparé. Un tel modèle semble cependant difficile à formuler[5].

Soulignons cependant, que toutes les mesures envisagées ici sont non-linéaires, et ne permettent donc pas l'explication de l'erreur entre le signal séparé et le signal original en termes d'interférences d'une part, et d'artefacts d'autre part. De telles mesures seraient également incapables de tolérer des invariances dans les signaux séparés (reconstruction à un gain près, à un gain lentement variable dans le temps près, à un délai près, etc.) – situation qui est prise en compte par les SDR, SIR et SAR en adaptant l'opération de projection.

5.5 Conclusion

Après avoir donné un aperçu des méthodes de séparation de sources mono-capteur proposées dans la littérature, et des difficultés posées par leur application à la séparation de la piste de batterie, nous avons présenté deux méthodes de séparation de la piste de batterie. La première méthode s'appuie sur une modélisation des enveloppes d'amplitude de chacune des composantes harmonique/bruit des signaux de sous-bande. Combinée à un module de transcription de la piste de batterie, il est possible de reconstituer l'enveloppe d'amplitude des composantes harmoniques/bruit de sous-bande du signal de batterie à extraire, permettant ainsi une séparation par masquage/filtrage. La deuxième méthode étend les travaux de Benaroya, en en proposant plusieurs améliorations spécifiques à la batterie : enrichissement du dictionnaire de d.s.p avec la d.s.p de la composante stochastique du signal observé, utilisation de tailles de fenêtres variables, et méthode alternative d'apprentissage du dictionnaire de d.s.p pour la batterie.

L'évaluation, conduite sur un sous-ensemble varié du corpus ENST-drums, souligne l'intérêt de nos contributions. Les méthodes les plus puissantes sont des méthodes supervisées, utilisant une étape d'apprentissage pour estimer par exemple des modèles spectraux ou des masques TFS. Cette étape d'apprentissage peut cependant mettre en difficulté de telles méthodes. Pour certaines applications, la séparation doit être efficace sur une large gamme de signaux, y compris par exemple des signaux de batteries électroniques. Les méthodes supervisées peuvent être mises en défaut dans de telles situations. Une direction de recherche intéressante peut alors consister en l'utilisation des techniques d'adaptation (comme proposé par Ozerov et al. pour la séparation de voix chantée [OPGB05]).

Nous avons également souligné différentes limites des mesures de performance utilisées. En particulier, pour le problème de la séparation de la piste de batterie, le caractère percussif et les transitoires du signal original doivent être restitués. Il apparaît ainsi essentiel de développer de nouvelles métriques mesurant la qualité de la séparation sur les parties stables et transitoires du signal à extraire. Faute de mieux, les tests d'écoute subjectifs sont le seul moyen d'évaluer la qualité de la séparation pour des applications à large échelle, comme par exemple l'inclusion d'un contrôle du volume de la batterie dans les lecteurs de musique.

Publications liées à ce chapitre

Nos premiers travaux utilisant le remasquage des signaux stochastiques de sous-bande pour la reconstruction d'une piste de batterie sont détaillés dans [GR05d]. Cet article inclus en particulier une évaluation subjective mesurant la qualité des signaux séparés pour une application de remixage. Les autres méthodes discutées et évaluées dans ce chapitre sont présentées dans [GR07].
La base ENST-drums utilisée pour les évaluations est décrite dans [GR06b].

[5] Les masques TFS tout comme les modèles spectraux ne fournissent pas un modèle des signaux de batterie, mais plutôt un modèle des observations ou des paramètres qu'on en extrait.

Conclusion de la partie I

Un certain nombre de problèmes rencontrés en indexation audio consistent à extraire, à partir d'un signal de musique polyphonique complexe, une description de haut niveau d'une de ses parties. De tels problèmes incluent par exemple la détection de la mélodie, la reconnaissance de l'instrument jouant un solo, ou, dans le contexte de cette thèse, la transcription de la piste de batterie. De tels problèmes doivent-ils être résolus par une étape préliminaire de séparation de sources, de manière à isoler la partie qu'on cherche à analyser, ou doit-on traiter le signal globalement ? Nous avons montré tout au long de cette première partie que les deux approches peuvent être suivies en parallèle. Les expériences que nous avons réalisées en transcription de la piste de batterie suggèrent en effet que les artefacts introduits par la méthode de séparation de sources employée peuvent dégrader la robustesse de certains attributs, tandis que d'autres attributs gagnent en pouvoir discriminant à l'issue de cette étape de séparation. Il apparaît dès lors intéressant de combiner l'information présente dans le signal original et le signal séparé, et plusieurs stratégies de fusion peuvent alors être mises en oeuvre.

L'absence de modèle génératif pouvant décrire les signaux de batterie nous a conduit à utiliser, pour la transcription, une approche discriminative utilisant des méthodes d'apprentissage statistique et une vaste palette de paramètres acoustiques. De nombreuses questions restent ouvertes quant à l'interprétation de ces attributs lorsqu'ils sont extraits sur des signaux polyphoniques, ou quant à leur robustesse à l'ajout d'un accompagnement instrumental. Nous suggérons que des méthodes supervisées de sélection d'attributs peuvent fournir des réponses à ces questions, et permettre le développement de systèmes de transcription efficaces.

Les indices acoustiques ne sont pas les seuls à permettre la transcription : des modèles de séquence peuvent guider la transcription en incorporant des règles musicales ou stylistiques simples, tandis que des méthodes de minimisation de mesures de complexité peuvent rétablir le caractère symétrique et répétitif des séquences rythmiques. Cependant, de telles méthodes ne sont réellement efficaces que lorsque les scores fournis par les modèles acoustiques sont fiables, et elles opèrent dans le domaine symbolique – une part d'information peut donc être perdue lors de la quantification de la séquence à transcrire.

Les performances satisfaisantes obtenues par les algorithmes de séparation présentés au chapitre 5 suggèrent la question suivante : Pourquoi ne pas utiliser ces méthodes de séparation, plutôt que les méthodes plus simples utilisées au chapitre 3, comme pré-traitement avant la transcription ? Simplement parce qu'une des méthodes présentées requiert une transcription de la piste de batterie, et parce que les performances de l'autre dépendent d'une étape d'apprentissage. Une telle situation est similaire aux problèmes d'estimation avec variables cachées, dans lesquels l'ensemble des variables à estimer (dans notre cas, la séparation), et l'ensemble des variables latentes (dans notre cas, la transcription, ou un modèle de chaque instrument percussif utilisé) sont difficiles à estimer de façon jointe, mais faciles à estimer l'une par rapport à l'autre. Cette observation suggère des approches itératives, où les étapes de transcription et de séparation sont effectuées séquentiellement, l'une étant donné l'autre, jusqu'à convergence, le processus de séparation étant informé par la partition obtenue à l'étape précédente, et le processus de transcription utilisant des attributs extraits à la fois du signal séparé et du signal original pour plus de robustesse.

De façon concurrente, il serait intéressant de disposer de représentations permettant l'estimation jointe de la partition et du signal séparé. Cette approche est en quelque sorte suivie par les méthodes employant des décompositions comme la NMF ou l'ISA, dans lesquelles les profils spectraux et les enveloppes temporelles peuvent être estimés conjointement, et où ils jouent le rôle d'une représentation intermédiaire permettant à la fois la transcription et la resynthèse. Cependant,

différents traitements sont nécessaires pour effectivement déduire une transcription, ou effectivement reconstruire un signal séparé, à partir de cette représentation intermédiaire. Une direction de recherche intéressante consisterait alors à découvrir une représentation intermédiaire de haut-niveau, à la fois proche de la source et de la transcription, pour laquelle il existe une procédure efficace d'estimation jointe de tous les paramètres.

Deuxième partie

Transcription audiovisuelle du jeu de la batterie

CHAPITRE 6

Transcription musicale et multimodalité : état de l'art et problématique

Nous nous proposons dans cette seconde partie d'étendre le système de transcription de signaux de batterie présenté dans la partie précédente pour qu'il intègre une information visuelle fournie par une ou plusieurs caméras filmant le batteur. L'objectif est double : améliorer les performances de la transcription, et extraire des informations de jeu complémentaires difficiles à obtenir à partir de la modalité audio seule. Les applications envisagées sont celles de l'interaction musicien/machine – capture précise du jeu d'un soliste ou aide à l'apprentissage. L'utilisation du système dans des circonstances où les conditions de prise de vue sont moins contrôlées (annotation automatique de vidéos de concerts) est également discutée.

À notre connaissance, ce problème n'a jamais été traité dans la littérature. Il existe cependant différents problèmes ayant des points communs avec le nôtre, dont nous pourrons nous inspirer. Nous en proposerons une typologie dans la section 6.1. Un état de l'art de chacun de ces problèmes est donné dans la section 6.2. Cet état de l'art n'a pas l'ambition d'être exhaustif, mais cherche plutôt à introduire les principaux modèles statistiques et techniques de traitement d'image mis en oeuvre. Enfin, nous présenterons l'approche que nous avons décidé de suivre dans la section 6.3, en guise d'introduction aux chapitres qui suivent.

6.1 Spécificité du problème à résoudre et typologie des tâches connexes

Le problème que nous nous proposons de résoudre possède les spécificités suivantes :

- Les gestes ou mouvements à analyser seront produits dans un contexte musical : les gestes seront courts et rapides (plusieurs d'entre eux effectués par seconde), répétés pour former des séquences, et chacun d'entre eux appartiendra à un ensemble fini de catégories.
- Les mouvements à reconnaître seront essentiellement ceux des membres supérieurs du corps humain.
- L'acquisition de l'information de jeu sera effectuée de façon non-intrusive par des capteurs vidéos, et non pas, par exemple, par des accéléromètres ou des capteurs d'efforts fixés sur les articulations du musicien. Par ailleurs, nous étendons la contrainte de non-instrusivité pour exclure l'usage par les musiciens de gants ou baguettes colorés ou le jeu sur un fond coloré. Nous exigeons ainsi que notre système soit capable de traiter *a posteriori* des séquences vidéo prises dans des conditions d'éclairage normales, et idéalement sans mouvements de caméra.
- L'information extraite devra pouvoir être fusionnée ou corrélée avec une information extraite d'un signal audio – il devra également être possible d'effectuer directement une reconnais-

Contexte musical ?	Mouvements des membres ?	Vidéo/non-intrusif ?	Fusion avec l'audio ?	Tâche	Références
●	○	●	●	Transcription audiovisuelle de piano	[SKT97; SC03]
●	●	○	●	Analyse de danse	[SNI04; KPS03]
●	●	●	○	Suivi vidéo d'activités musicales	[CMR+03; Dah00; Dah04; Mur03; MAJ04]
●	●	○	○	Contrôle gestuel d'instruments	[WD01; WD04]
○	●	●	○	Reconnaissance des gestes et postures	[DB97; KKVB+05; YOI92; Bra97; Min05; PSH97; WH00]
○	○	●	●	Reconnaissance de parole audiovisuelle	[PNLM04]
○	○	●	●	Localisation de sources sonores	[FDFV00; FD01; HM00]
○	○	●	●	Séparation de sources audiovisuelle	[HC02; SSG+02; SGJS04; WCH+05]

TAB. 6.1 – Quelques problèmes connexes traités dans la littérature

sance multimodale audiovisuelle.

Aucun problème ne combinant ces quatre aspects n'a été traité dans la littérature. Il existe cependant différentes familles de problèmes connexes retenant certains de ces aspects, dont un résumé est donné dans la table 6.1. Nous dressons maintenant un état de l'art de chacun de ces problèmes.

6.2 État de l'art

6.2.1 Transcription audiovisuelle de piano

Un problème similaire au nôtre – transcrire le jeu d'un instrument à partir de signaux audio et vidéo – a été abordé pour le piano, dans deux études.

Dans [SKT97], Saitoh et al. décrivent un système de transcription guidé par la vidéo, acquise par une caméra située à la verticale du clavier. Trois modules de traitement d'image sont décrits : un module de segmentation de l'image du clavier en régions correspondants à chacune des touches, par détection de segments de droite (transformée de Hough) ; un module de détection de la position de la main utilisant un critère de couleur ; et un détecteur de touches enfoncées utilisant un critère de luminosité – ce critère ne permettant de détecter que l'enfoncement des touches blanches. L'analyse audio consiste en un système très rudimentaire recherchant les maxima d'énergie en sortie d'un banc de filtres à Q constant. La transcription musicale est effectuée par une approche hiérarchique : si l'enfoncement d'une touche blanche est détecté, cette information vidéo est directement utilisée pour la transcription. Sinon, la position de la main est utilisée pour proposer un ensemble de notes candidates, qui seront départagées en utilisant le détecteur audio. L'évaluation est effectuée sur une séquence monophonique de 29 notes. 4 erreurs sont commises par le système vidéo seul, 1 erreur par le système multimodal. Soulignons que cette étude est limitée par la simplicité du module d'analyse audio, et par son application au simple cas monophonique. En particulier, dès lors que le nombre

de notes jouées simultanément sera inconnu, la méthode hiérarchique proposée ne sera plus valide – même si l'enfoncement d'une touche blanche est détecté, l'analyse audio doit tout de même être effectuée pour tester l'enfoncement éventuel d'une ou plusieurs autres touches noires.

Dans [SC03], Smaragdis et Casey considèrent une représentation d'une séquence vidéo sous forme d'une suite de vecteurs $\mathbf{x}(m)$ à $160 \times 120 + 128$ composantes, où chaque trame est représentée par un vecteur. Les 160×120 premières composantes contiennent les valeurs de luminosité de chaque pixel de la trame et les 128 autres composantes contiennent le module du spectre du signal observé sur la durée d'une trame. Une analyse en sous-espaces indépendants – *Independent Subspace Analysis* (ISA) – présentée dans la section 2.2.3.1 – est effectuée à partir de ce vecteur, produisant une décomposition de la séquence sous la forme $\mathbf{X} = \mathbf{F}\mathbf{T}^T$, où $\mathbf{F}$ contient des composantes audiovisuelles caractérisées par un profil spectral et un masque vidéo, et $\mathbf{T}$ contient des enveloppes représentant, en fonction du temps, l'activation de ces composantes. Cette approche est appliquée à des signaux synthétiques (points clignotants sur une image associés à des sinusoïdes de diverses fréquences), et à une courte séquence de jeu de piano. Pour ce dernier exemple, chaque composante indépendante audiovisuelle extraite correspond au spectre d'une note associé au contour de la touche correspondante. L'application à la transcription musicale est envisageable à condition de connaître le nombre de composantes, et de disposer d'un détecteur de fréquence fondamentale pour associer chaque composante à une note (comme fait dans [BBR07]).

6.2.2 Analyse de danse

Shiratori et al. présentent dans [SNI04] un système multimodal d'analyse des mouvements de danse destiné à extraire, à partir d'une chorégraphie, des gestes et postures élémentaires. Les mouvements sont capturés à l'aide d'un système d'analyse vidéo intrusif (exigeant la pose de marqueurs colorés sur le corps du danseur, et la prise de vue multi-caméra sur fond uniforme), afin d'extraire les positions du centre de gravité du corps du danseur, de ses mains et de ses pieds. Un système de détection du tempo [GM95] est utilisé pour extraire une pulsation rythmique. La segmentation en postures s'effectue sur un critère de minimum de vitesse, éventuellement aligné avec la grille rythmique. Les approches précédentes, telles le système de Kim et al. décrit dans [KPS03], n'offrent pas une telle précision dans la segmentation. Parmi les applications envisagées, figurent la transcription – l'extraction des postures et leur reconnaissance ultérieure permettant ainsi une forme de transcription supervisée ; ainsi que la synthèse de mouvements de danse à partir d'enregistrements musicaux comme évoqué dans [KPS03].

6.2.3 Suivi vidéo d'activités musicales

Plus proche de notre application se trouvent divers systèmes d'analyse vidéo d'activités musicales (principalement, le jeu d'un instrument). Camurri et al. décrivent dans [CMR+03] un système d'analyse des mouvements d'un pianiste, à des fins d'analyse de l'expressivité. L'acquisition des paramètres de mouvement est faite de façon non-intrusive, à l'aide de quatre caméras. Les paramètres extraits sont les positions de la tête de l'instrumentiste sur deux axes gauche/droite et avant/arrière. Ces paramètres sont corrélés à des paramètres de vélocité générés par l'instrument (il s'agit d'un piano MIDIfié) pour vérifier diverses hypothèses sur les modes d'expression corporelle des pianistes. Des analyses similaires ont été effectuées pour le jeu de la batterie par Dahl dans [Dah00] et [Dah04]. Le système d'analyse vidéo comprend deux caméras. Il est moyennement intrusif, puisqu'il n'exige que des marqueurs lumineux aux extrémités des baguettes et sur les bras du musicien. Ce dispositif permet l'étude des différentes stratégies développées par les musiciens pour contrôler la force de frappe sur l'instrument, selon l'accent et les nuances de jeu. L'application envisagée n'est donc pas la transcription musicale, mais plutôt l'acquisition de paramètres musicologiques – ces systèmes n'intégrant d'ailleurs pas la modalité audio.

Une autre activité musicale ayant donné lieu au développement d'un système de suivi vidéo est la direction d'orchestre. Murphy décrit dans [Mur03] une méthode pour suivre les mouvements

de la baguette d'un chef d'orchestre. Deux sous-systèmes sont introduits. Tout d'abord, un sous-système est chargé de localiser la position initiale de la baguette au sein d'une trame (par exemple, dans la première trame d'une séquence) : les contours de l'image sont extraits par l'algorithme de Canny [Can86], et la baguette est identifiée en cherchant deux segments de droite parallèles dans l'image. En supposant que la section de la baguette est constante et de l'ordre de quelques pixels, cette recherche peut être effectuée par deux automates finis déterministes opérant en parallèle sur les lignes et colonnes de l'image. Le second sous-système permet de mettre à jour la position de la baguette, connaissant sa position précédente. Du calcul du flot optique et de la recherche de vecteurs vitesses alignés, sont déduits les vecteurs vitesses de la base et du sommet de la baguette, permettant d'obtenir une estimée de la nouvelle position de la baguette dans la trame courante. Cette position est alors utilisée pour lancer une procédure de recherche semblable à celle effectuée par le premier sous-système – mais cette fois restreinte au voisinage de la position supposée de la baguette. Le suivi est effectué par deux caméras, et permet l'extraction de la vitesse et de la position de la baguette. À un niveau supérieur, les trajectoires extraites sont segmentées en mouvements élémentaires (tels ceux utilisés pour battre la mesure), permettant l'extraction d'une pulsation rythmique. La reproduction d'un signal de musique dont le tempo est connu peut alors être alignée sur les mouvements de la baguette, comme décrit dans [MAJ04].

6.2.4 Contrôle gestuel de la synthèse sonore

Si jusqu'ici nous avons présenté des systèmes capturant les mouvements de musiciens jouant d'instruments réels, il est également possible d'analyser les gestes de musiciens jouant d'instruments fictifs (que leur conception s'inspire ou non d'instruments acoustiques), et d'utiliser les paramètres extraits pour contrôler un synthétiseur. Or, si l'analyse du jeu d'un instrument réel demande des techniques non-intrusives d'acquisition, pour préserver le timbre et l'ergonomie de jeu de l'instrument, et donc un suivi vidéo, l'analyse du jeu sur un instrument fictif ou "contrôleur" peut se faire plus aisément à l'aide de capteurs (de force ou d'accélération), sur le corps de l'instrument lui-même[1]. Différents types de capteurs, stratégies d'acquisition des paramètres de mouvement, et d'association des paramètres gestuels aux paramètres de synthèse sonore sont discutés par Wanderley et Depalle dans [WD04].

6.2.5 Reconnaissance des gestes et postures

En dehors de ce contexte musical, différentes applications (surveillance, indexation vidéo) requièrent la segmentation et la reconnaissance de gestes effectués par des humains au sein de séquences vidéos. Les approches les plus simples [DB97] éliminent la dimension temporelle du mouvement : à partir d'une séquence vidéo, est produite une image unique formée de la somme de masques binaires représentant, pour chaque trame, les régions en mouvement. Cette "enveloppe" de la trajectoire peut être utilisée pour discriminer différentes actions. De façon similaire, des paramètres de trajectoire (position, vitesse et accélération d'un marqueur) peuvent être extraits pour chaque trame de la séquence d'images. La séquence formée par ces vecteurs de paramètres peut être représentée par un unique vecteur d'attributs, contenant par exemple les premiers moments des distributions de chacun de ces paramètres. Une telle approche est décrite dans [KKVB+05] pour la reconnaissance d'émotions véhiculées dans les gestes.

Certaines applications requièrent cependant de prendre en compte la dimension temporelle des gestes, soit parce que la séquence à traiter comporte plusieurs actions successives, soit parce qu'il est nécessaire de segmenter l'action reconnue en ses mouvements élémentaires. Dans ce cas, la séquence de vecteurs d'attributs extraits d'un marqueur de l'image est modélisée par des HMM, chaque état

[1] Pour quelques applications exigeant le suivi précis de la position d'objets, l'utilisation de capteurs vidéos est plus pertinente. Cependant, pour de telles applications, des marqueurs spécifiques peuvent être employés. Par exemple, les systèmes D-Touch [CSR03] ou Reactivision [BKJ05] exigent de localiser plusieurs objets sur une surface plane. L'analyse peut alors se faire aisément à l'aide de la modalité vidéo, en repérant chaque objet à l'aide de marques fiduciaires.

correspondant à une étape du mouvement [YOI92]. Soulignons que ce type de modèle n'impose aucune contrainte quant à la méthode d'extraction des paramètres de trajectoire – par suivi de marqueurs colorés ou par analyse du flot optique [Min05]. Dans le cas où K points sont suivis sur la séquence vidéo (par exemple, un marqueur pour chaque membre), l'usage de HMM couplés [Bra97] ou factoriels [GJ97] est utile pour modéliser des situations intermédiaires entre l'indépendance totale des mouvements de chaque membre (produit de K HMM modélisant les vecteurs d'attributs de taille D extraits pour chaque point à suivre), et leur dépendance totale (un seul HMM modélisant un vecteur de taille $D \times K$). L'application de tels modèles à des activités aussi diverses que le Tai-Chi [Bra97] ou le Ping-Pong [BOP97] a été effectuée avec succès. Cependant, dans toutes ces applications, les modèles ont été entraînés sur des séquences filmées avec le même angle de vue – les attributs extraits (positions ou vitesse), et donc les modèles appris, sont peu robustes aux changements d'orientation ou d'angle de prise de vue. Une des applications de la reconnaissance de gestes exigeant la plus grande robustesse face à de tels changements est la reconnaissance du langage des signes (voir [PSH97] pour une revue détaillée). Pour ce problème, une paramétrisation spécifique [PSH97] utilisant un modèle 3D de la main, ou des techniques semi-supervisées de sélection d'attributs robustes à l'orientation peuvent être envisagées [WH00].

6.2.6 Traitement audiovisuel de la parole

Différents systèmes de traitement de la parole cherchent à exploiter le fait que la perception de la parole est bimodale – des expériences comme celles réalisées par McGurk montrent en effet que le cerveau intègre les modalités auditives et visuelles. Nous présentons ici quelques solutions proposées à différents problèmes couramment rencontrés en traitement audiovisuel de la parole.

6.2.6.1 Reconnaissance de la parole audiovisuelle

Les systèmes de reconnaissance de la parole audiovisuelle cherchent à exploiter la complémentarité de l'information contenue dans le signal audio, et dans une image des lèvres ou du visage du locuteur pour améliorer la reconnaissance. S'il existe un consensus sur la paramétrisation à utiliser en reconnaissance automatique de la parole à partir de la modalité audio (coefficients de prédiction linéaire ou MFCC), une large gamme de méthodes a été proposée dans la littérature pour segmenter et paramétriser l'image des lèvres : contours actifs ou modèles d'apparence pour la segmentation, modèles paramétriques de la forme des lèvres tel les *facial animation parameters* définis dans la norme MPEG-4 [AWWK02], ou simples attributs géométriques pour la paramétrisation. Nous ne présenterons ici ni ces paramètres ni leur procédé d'extraction.

Une problématique moins spécifique est la fusion des modalités audio et vidéo pour la reconnaissance de la parole. Potamianos et al. [PNLM04] recensent les architectures suivantes :

Fusion des attributs par concaténation Dans cette architecture, les attributs audio et vidéo sont concaténés. Le vecteur d'attributs ainsi formé peut être utilisé de façon identique aux vecteurs d'attributs audio seuls utilisés classiquement en reconnaissance de la parole, par exemple en utilisant des HMM [RJ93]. Cette solution simple est illustrée dans la figure 6.1. Elle permet l'intégration à moindre coût de l'information vidéo dans les systèmes de reconnaissance de la parole existants.

Fusion des attributs par sélection et concaténation Le critère de Fisher est utilisé pour identifier, parmi les attributs audio et vidéo, les attributs les plus discriminants. Ces attributs sont concaténés pour former un vecteur d'attributs utilisé comme précédemment.

Débruitage des attributs audio par projection Ici, des attributs audio "débruités" sont obtenus par une projection du vecteur d'attributs audiovisuels concatenés. Le choix de la projection peut être vu comme un problème de régression linéaire – il s'agit de déterminer la projection permettant la meilleure approximation, au sens des moindres carrés, des paramètres acoustiques qui auraient été calculés sur le signal de parole sans bruit ; à partir des paramètres acoustiques extraits du signal bruité augmenté des observations vidéo.

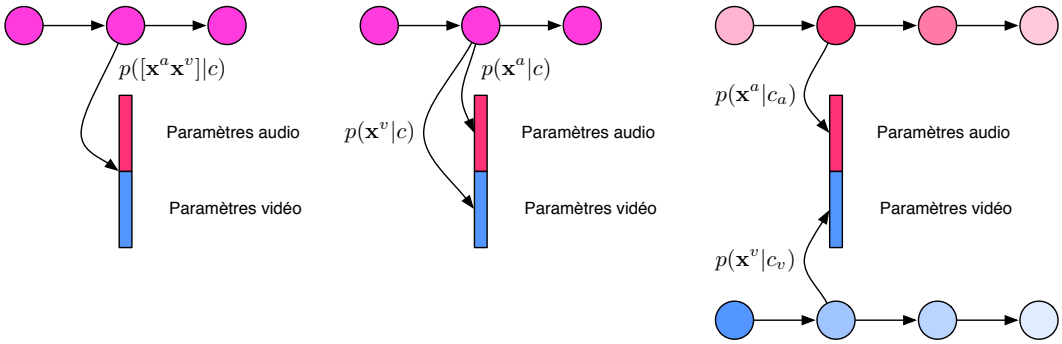


FIG. 6.1 – Utilisation de HMM pour la reconnaissance de parole audiovisuelle : vecteurs d'attributs concaténés, HMM bimodal à états synchrones, HMM produit

Fusion des vraisemblances d'un HMM bimodal à états synchrones Plutôt que d'associer à chaque état c du HMM la distribution des attributs audiovisuels concaténés $[\mathbf{x}^a\mathbf{x}^v]$ (modélisée par exemple comme un mélange de gaussiennes), on modélise indépendamment les distributions $p(\mathbf{x}^a|c)$ et $p(\mathbf{x}^v|c)$. La fonction de vraisemblance $p([\mathbf{x}^a\mathbf{x}^v]|c)$ est remplacée lors de l'apprentissage ou de la reconnaissance par $p(\mathbf{x}^a|c)^{\alpha_a} \times p(\mathbf{x}^v|c)^{\alpha_v}$ où α_a et α_v sont des constantes pondérant l'influence des modalités audio et vidéo. Soulignons que dans cette méthode, n'est modifiée que la couche d'observations du HMM. Les observations audio et vidéo sont à tout moment expliquées par le même état sous-jacent du HMM (voir figure 6.1).

Fusion par HMM produit Les observations audio et vidéo sont expliquées par deux HMM évoluant de façon découplée : le score associé à une observation de paramètres audiovisuels est calculé selon $p(\mathbf{x}^a|c_a)^{\alpha_a} \times p(\mathbf{x}^v|c_v)^{\alpha_v}$ où c_a et c_v sont respectivement les états dans lequel se trouvent les HMM utilisés pour la modalité audio, et pour la modalité vidéo (voir figure 6.1). Ce découplage permet de prendre en compte le décalage temporel entre les mouvements des lèvres et la parole (par exemple un mouvement des lèvres anticipant la prononciation d'un phonème). Son coût en calcul est cependant élevé, puisque la reconnaissance, s'effectuant par l'algorithme de Viterbi, doit explorer à chaque pas $N_a \times N_v$ états, où N_a (resp. N_v) est le nombre d'états accessibles du HMM modélisant les paramètres audio (resp. vidéo).

Une comparaison de ces méthodes de fusion sur un même corpus est effectuée dans [PNLM04].

6.2.6.2 Localisation du locuteur dans une scène

Avant d'appliquer de telles méthodes de reconnaissance de la parole, il peut être nécessaire de localiser le locuteur dans une scène complexe pouvant contenir d'autres objets ou êtres humains en mouvement, ou même d'autres locuteurs. Le problème de la localisation du locuteur dans une scène vidéo est traité dans différents travaux.

Hershey et Movellan proposent dans [HM00] de modéliser par des gaussiennes multivariées la distribution des paramètres audio, vidéo, et du vecteur contenant les paramètres audio et vidéo joints. Il est alors possible de calculer analytiquement l'information mutuelle entre chacun des attributs audio et chacun des attributs vidéo, qui s'exprime comme une fonction simple du coefficient de corrélation de Pearson entre les attributs considérés. Hershey considère ensuite, pour chacun des pixels de l'image, son information mutuelle avec un paramètre d'énergie du signal audio. Le centroïde de l'ensemble des pixels où l'information mutuelle dépasse un seuil donné indique alors la position du locuteur.

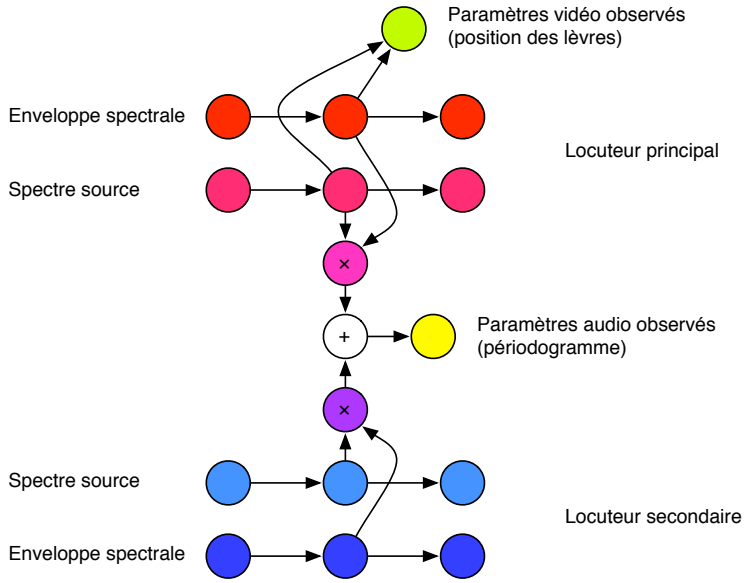


FIG. 6.2 – Modèle factoriel pour le débruitage audiovisuel de la parole

L'hypothèse formulée par Hershey et Movellan, selon laquelle la distribution jointe des attributs audio $\mathbf{x}^a$ et $\mathbf{x}^v$ est gaussienne est contestée par Fisher et al. dans [FDFV00]. L'alternative proposée consiste à trouver des projections $\phi^a_{\boldsymbol{\alpha}^a}$ et $\phi^v_{\boldsymbol{\alpha}^v}$ des attributs audio et vidéo maximisant l'information mutuelle entre $\phi^a_{\boldsymbol{\alpha}^a}(\mathbf{x}^a)$ et $\phi^v_{\boldsymbol{\alpha}^v}(\mathbf{x}^v)$. La classe de fonctions ϕ^a et ϕ^v considérées correspond au perceptron à une couche $\phi^a_{\mathbf{w}}(\mathbf{x}^a) = f(\mathbf{w} \cdot \mathbf{x}^a)$ où f est une fonction non-linéaire continue, par exemple une sigmoïde. Les coefficients $\boldsymbol{\alpha}^a$ et $\boldsymbol{\alpha}^v$ s'interprètent alors comme des poids indiquant la contribution de chacun des attributs audio (resp. vidéo) à la formation d'un attribut maximalement corrélé à la vidéo (resp.à l'audio). Dans le cas où, pour une trame donnée, les attributs vidéo sont les luminosités des pixels, et les attributs audio le périodogramme du signal de parole sur la durée de la trame, il est montré dans [FDFV00] que les poids vidéos $\boldsymbol{\alpha}^v$ sont élevés pour les intensités de pixels de la bouche du locuteur, tandis que les poids audio $\boldsymbol{\alpha}^a$ sont élevés pour les régions du spectre occupées par la parole du locuteur. Dans [FD01], les projections considérées sont linéaires : Dans ce cas, la recherche de la projection optimale peut être effectuée par une méthode efficace de descente de gradient.

6.2.6.3 Séparation audiovisuelle de la parole

Une dernière famille de systèmes de traitement de la parole utilisant la modalité vidéo sont les systèmes de séparation de sources (ou de débruitage). Dans [HC02], Hershey et Casey proposent un modèle factoriel original des signaux de parole, dans lequel le périodogramme d'une trame de signal observé est décrit comme le produit d'une enveloppe spectrale, et du spectre de source. La séquence des enveloppes spectrales, et la séquence des spectres de sources sont modélisées par deux HMM découplés. Un modèle similaire est construit pour des signaux de bruit d'ambiance, ou pour des signaux de parole venant d'un autre locuteur. Un modèle factoriel double des signaux de parole bruités (ou perturbés par un deuxième locuteur) est formé en considérant les signaux observés comme la somme de signaux produits par ces deux modèles. L'estimation de la séquence d'états la plus probable à partir d'une séquence de parole permet d'associer chaque point temps/fréquence à une des deux sources (locuteur principal ou bruit/locuteur secondaire). Hershey et Casey introduisent ensuite dans le modèle factoriel associé au locuteur principal une couche d'observations vidéos (le

modèle complet est illustré dans la figure 6.2). L'estimation de la séquence d'états la plus probable peut ainsi être effectuée en utilisant à la fois l'information audio et vidéo. Les résultats donnés dans [HC02] montrent que le taux de reconnaissance de mots isolés est toujours amélioré lorsque les observations vidéo sont prises en compte – le gain de performances pouvant s'élever jusqu'à 60% lorsque le signal est modérément bruité (rapport signal à bruit de 12 dB). Notons que ce modèle s'applique aux situations où l'on ne dispose que d'une seule source d'observations audio (séparation à un seul capteur).

Le cas de la séparation à plusieurs capteurs est traité par Sodoyer et al. dans [SSG+02]. Si on considère le mélange comme instantané (les signaux observés sont des combinaisons linéaires des sources à séparer), le problème de séparation consiste alors à chercher une matrice de démixage maximisant un critère donné. Une ICA classique maximise par exemple une mesure d'indépendance des sources extraites. Sodoyer et al. propose de déterminer la matrice de démixage maximisant la cohérence audiovisuelle entre la première source, et les observations vidéo. Dans le cas où on suppose la matrice de mixage constante au cours du temps, cette cohérence peut être mesurée comme le produit des probabilités jointes $\prod_{i=1}^{N} p(\mathbf{x}_i^a, \mathbf{x}_i^v)$ où N est la longueur de la séquence à traiter, les $\mathbf{x}_i^a$ représentent les paramètres audio extraits (coefficients de prédiction linéaire), et les $\mathbf{x}_i^v$ représentent les paramètres vidéo extraits (deux paramètres de position des lèvres). La loi jointe $p(\mathbf{x}^a, \mathbf{x}^v)$ est un mélange de gaussiennes, dont les paramètres sont appris sur un corpus de signaux non bruités. Notons que dans le cas où la matrice de mixage varie au cours du temps, il suffit de calculer pour chaque trame la matrice de démixage instantanée maximisant $p(\mathbf{x}_i^a, \mathbf{x}_i^v)$ – sans intégration temporelle. L'optimisation de ce critère étant dans tous les cas coûteuse, une amélioration proposée par les mêmes auteurs dans [SGJS04] consiste à utiliser une méthode de séparation de sources classique (JADE), et d'identifier, parmi les sources extraites, celle dont la cohérence avec les observations vidéos est la plus forte, à l'aide du critère de probabilité jointe.

Une approche similaire est utilisée par Wang et al. dans [WCH+05] – elle est cette fois étendue aux mélanges convolutifs. Pour de tels mélanges, il est nécessaire d'utiliser un critère assurant à la fois l'indépendance des sources extraites, et la cohérence de la source principale avec les observations vidéo.

6.3 Discussion

6.3.1 Que retenir de l'état de l'art ?

Que pouvons nous retenir des solutions proposées à ces problèmes semblables au nôtre ? Des systèmes spécifiques au piano présentés en 6.2.1, nous pouvons retenir quelques pistes quant aux méthodes de segmentation d'image à utiliser (critère géométrique sur la forme de l'instrument).

L'application de l'ICA audiovisuelle aux séquences vidéos de batterie en situation polyphonique semble difficile : elle pose les mêmes problèmes que son homologue unimodale présentée en 2.2.3.1. Nous avons évoqué en 2.2.3.1 la possibilité d'utiliser une information a priori pour éviter le problème de la sur/sous-séparation et de l'identification des sources. Cette approche n'est malheureusement pas possible dans le cas multimodal, car s'il est possible d'apprendre a priori un modèle générique des timbres de la caisse claire, de la grosse caisse et des hi-hats, il n'est pas possible d'apprendre un modèle générique a priori de l'image de la scène (la position des éléments de la batterie change d'une scène à une autre). L'ISA audiovisuelle ne peut ainsi être utilisée que de façon non-supervisée, forme sous laquelle se pose le problème de l'identification des sources. Cet échec souligne une spécificité de notre problème : s'il est possible de formuler un modèle générique de la distribution des attributs audio extraits pour différents instruments de la batterie, il n'est pas possible de formuler un tel modèle générique pour les attributs vidéos, qui dépendent de la position des instruments dans l'espace.

Dans les systèmes d'analyse de la danse présentés en 6.2.2, l'intégration des modalités audio et vidéo ne peut se faire que pour la tâche de segmentation. Au delà, il n'existe pas de corrélation entre les postures et, par exemple, le contenu spectral ou mélodique de chacun des segments. Ce n'est

pas le cas pour la batterie ; s'il sera possible de fusionner les informations extraites des flux audio et vidéo pour segmenter chacune des frappes, on souhaitera également combiner ces deux sources d'information pour la détection des postures et des instruments joués.

Parce que notre but est de transcrire des séquences de batterie audiovisuelles *a posteriori* (document déjà enregistré, dont il n'est pas possible de contrôler les conditions de prise de vue), nous avons exclu l'utilisation de systèmes intrusifs, demandant la pose de marqueurs sur les baguettes ou le port de gants colorés. Les systèmes de suivi discutés en 6.2.4 sont difficilement exploitables.

Les méthodes de traitement d'image développées par [Mur03] pour le suivi de la baguette du chef d'orchestre semblent pouvoir être directement réutilisées pour extraire la position et l'orientation des baguettes du batteur. Cependant, son implémentation, réalisée en collaboration avec Kevin McGuinness au *Centre for Digital Video Processing* (Dublin City University) s'est avérée infructueuse : dans le cas de la batterie, le mouvement des baguettes est très rapide, rendant le suivi impossible, et imposant un fort flou de bougé. Nous avons également rencontré quelques difficultés pour le suivi des balais et des fagots (l'attache sombre qui lit les fagots crée des discontinuités et arêtes supplémentaires dans le contour de la baguette extrait par l'algorithme de Canny). Notons également que nous avons réalisé l'annotation manuelle de la position et de l'orientation des baguettes sur trois séquences vidéos. À partir de ces données de suivi idéales, nous avons tenté de déterminer les attributs les plus discriminants pour la transcription du jeu. Le critère le plus informatif est la position de la tête de la baguette relativement à la surface des fûts et des cymbales : un tel critère peut être extrait par des méthodes plus simples, ne demandant pas l'estimation des paramètres de position et d'orientation.

Nous avons présenté en 6.2.5 différentes méthodes de classification et de reconnaissance des postures. Nous avons insisté sur le caractère spécifique des modèles appris, qui ne sont pas robustes à des changements d'angles de prise de vue, ou d'orientation. Il semble ainsi difficile d'appliquer de telles méthodes à la transcription audiovisuelle du jeu de la batterie. Tout d'abord, la contrainte de robustesse aux changements d'orientation et d'angle de prise de vue exige d'utiliser des paramètres relatifs (comme cela est fait en reconnaissance de la langue des signes) – correspondant par exemple à un modèle 3D du batteur. Cependant, la reconnaissance des gestes à partir de ces seuls paramètres serait insuffisante pour transcrire le jeu du batteur, puisque la signification musicale de chaque geste dépend également de la position des instruments (un même geste peut être une frappe sur un tom ou la caisse claire selon la disposition de la batterie). Nous distinguons ainsi deux sources de variabilité dans les séquences de batterie : la variabilité dans les angles de prise de vue, qui empêchent l'apprentissage de modèles universels des trajectoires – problème pouvant être résolu par l'utilisation de paramètres relatifs à un modèle 3D du musicien ; et la variabilité dans les positions des instruments – donnant à deux gestes strictement identiques des sens différents. Cette deuxième source de variabilité rend difficile l'apprentissage de modèles de jeu universels, pouvant être appliqués à une vidéo d'un batteur/d'une batterie inconnus : Quels attributs permettraient de décrire par un même modèle une "frappe de caisse claire" dans chacune des scènes présentées dans la figure 4.9 ? Il semble donc raisonnable de croire que les approches décrites en 6.2.5 ne permettent qu'un niveau de description peu fin – la simple reconnaissance de l'action "jeu de la batterie" dans des séquences vidéos.

Pour ces mêmes raisons, nous excluons les méthodes supervisées à base de HMM telles celles utilisées en reconnaissance audiovisuelle de la parole (section 6.2.6). En fait, nous avons évalué un système s'inspirant de telles méthodes dans une étude préliminaire publiée dans [GR05a][2]. Les attributs vidéo utilisés correspondent à une estimation de la quantité de mouvement dans des régions d'intérêt définies par l'utilisateur, et les attributs audio sont ceux du système de transcription de soli introduit dans [GR04]. Deux approches sont discutées pour la fusion : fusion précoce par concaténation des attributs et utilisation de la PCA pour former des attributs audiovisuels décorrélés ; et fusion tardive par multiplication des scores de vraisemblance issus de classifieurs audio et vidéo entraînés indépendamment les uns les autres, ou par utilisation d'une règle de décision "au plus confiant" (donnée dans le tableau 4.4). Si les résultats se sont montrés satisfaisants – augmentation de 5.2 points du taux de reconnaissance des frappes –, la méthode utilisée ne permet pas la formulation d'un modèle universel du jeu de la batterie pouvant être appris et testé sur des séquences utilisant des batteries ou angles de prise de vue différents. Les attributs vidéos eux-mêmes dépendent

[2] article reproduit dans l'annexe C

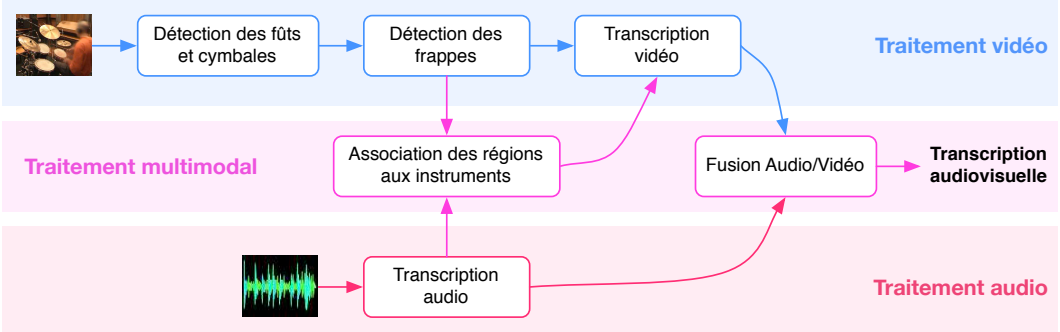


FIG. 6.3 – Architecture du système proposé pour la transcription audiovisuelle du jeu de la batterie

de l'orientation de la scène, et leur robustesse dépend du processus de calibration (définition des régions d'intérêt).

6.3.2 Approche proposée

Il apparaît à l'issue des discussions précédentes que la reconnaissance vidéo et la reconnaissance audio du jeu de la batterie diffèrent en un point : s'il existe un modèle universel, indépendant de la batterie et du batteur, du timbre d'un tom ou d'une grosse caisse, il n'existe pas de modèle universel, indépendant de la scène, des gestes du musicien jouant ces instruments. Ainsi, toute modélisation de paramètres vidéo, ou de paramètres joints audiovisuels ne peut se faire que localement, de façon spécifique à la séquence à traiter – autrement dit, un système utilisant la fusion précoce ne serait pas capable de généralisation. Nous proposons ainsi d'utiliser la fusion tardive, dans laquelle seront fusionnées les décisions produites par un système de classification audio universel (tel celui présenté au chapitre 4), et un système de classification vidéo local.

Quel système de classification vidéo utiliser, pour quels attributs ? Nous avons vu que la simple reconnaissance des gestes du batteur est insuffisante pour permettre la transcription d'une séquence rythmique, puisque le sens de chacun de ces gestes dépend de la disposition des éléments de la batterie. Il apparaît alors nécessaire d'analyser la scène vidéo pour déterminer la position de chacun des éléments de la batterie. Nous considérons ainsi deux groupes de paramètres vidéo, calculés pour chacun des éléments :

1. La quantité de mouvement à l'intérieur de chaque région d'intérêt. En effet, chaque élément de la batterie est mis en mouvement immédiatement après avoir été frappé – si les cymbales (crash et ride) sont les plus mobiles, le mouvement d'un tom mal fixé, ou même de la caisse claire, est également décelable.
2. La position d'une baguette relativement à chacune des régions d'intérêt, mesurée comme la proportion de pixels de la baguette présents à l'intérieur de la région. Cet attribut ne requiert pas la détermination de paramètres de position ou de vitesse, mais simplement la segmentation de la baguette dans la séquence.

Avec une telle paramétrisation, la détection d'une frappe de batterie dans une des régions d'intérêt est aisée, puisqu'une frappe se manifeste par l'intersection de la baguette et de la région considérée, suivie d'une augmentation et d'une décroissance rapide de la quantité de mouvement au sein de la région. Cette paramétrisation permet ainsi de s'affranchir de l'utilisation de classifieurs, puisque la reconnaissance des frappes pour chaque instrument peut se formuler comme un problème de détection.

Un dernier problème reste à résoudre : lors de l'analyse de la scène, comment associer chacune des régions d'intérêt extraites à l'instrument de la batterie qui lui correspond ? Si l'on dispose d'une transcription audio suffisamment fiable, ce problème peut être résolu par des méthodes semblables à celles décrites dans la section 6.2.6.2. Cette association entre régions d'intérêt et classes d'instruments pourra également être effectuée en utilisant des connaissances sur les propriétés de couleur des instruments (par exemple, les cymbales sont métalliques), et en cherchant l'association maximisant la cohérence entre les transcriptions effectuées par le module de détection vidéo, et le module de transcription audio.

L'architecture retenue est présentée dans la figure 6.3. Nous présentons dans le chapitre 7 les méthodes de traitement d'image utilisées pour l'analyse de la scène. Le chapitre 8 est consacré à la détection de frappes à partir de la modalité vidéo, et à la fusion des décisions – qui nécessitera l'association des régions d'intérêt extraites de la vidéo à des classes d'instruments. Notons qu'au long de ce chapitre, nous proposerons également plusieurs variantes de l'architecture retenue, pour offrir plus de robustesse aux changements d'angle de prise de vue ou d'éclairage au cours du temps, ainsi que pour tirer avantage de l'intervention d'un opérateur humain (transcription semi-automatique).

CHAPITRE 7

Segmentation de scènes de jeu de batterie

Sont présentées dans ce chapitre différentes techniques de traitement d'image pour segmenter automatiquement une séquence vidéo de jeu de batterie, afin d'extraire des éléments d'intérêt.

Une première tâche consiste à identifier la position des différents instruments de la batterie. Plus particulièrement, nous souhaitons localiser le sommet de chaque élément, c'est à dire la région susceptible d'être frappée par la baguette – surface de la cymbale et peau tendue sur le fût[1]. La section 7.1 présente des méthodes capables de produire une telle segmentation à partir d'images fixes. Ces méthodes sont étendues dans la section 7.2 pour traiter des séquences d'images. Nous présenterons également une méthode de segmentation basée sur un critère de mouvement visant à identifier les régions mises en mouvement simultanément, ainsi qu'une méthode de segmentation supervisée exploitant une transcription idéale produite par un système de transcription audio, ou une partition de référence.

La section 7.3 traite de la segmentation des baguettes et des avant-bras du batteur, à l'aide d'une méthode de soustraction adaptative de l'arrière-plan.

7.1 Segmentation des éléments de la batterie dans une scène : cas des images fixes

Nous présentons ici plusieurs critères complémentaires pour la segmentation des éléments de la batterie dans une image fixe : un critère de couleur, un critère morphologique et un critère géométrique.

7.1.1 Pré-traitement

Avant toute segmentation, l'image est pré-traitée par l'application d'un filtre bilatéral gaussien. Ce filtre non-linéaire, introduit par Tomasi et Manduchi dans [TM98], permet le débruitage de l'image et l'élimination des détails tout en préservant la netteté des contours. Si $\mathbf{I}$ est l'image à filtrer et $\mathbf{I}'$ l'image traitée, alors :

$$\mathbf{I}'(x, y) = (\mathbf{I} * k(x, y))(x, y) \tag{7.1}$$

$k(x, y)$ est un noyau gaussien pondéré, différent pour chaque point de l'image, défini par :

[1] Nous ne traitons pas le cas de la grosse caisse qui est hors champ dans les séquences que nous avons utilisées.

FIG. 7.1 – Filtrage bilatéral gaussien

$$\mathbf{k}(x_0, y_0)(x, y) = \underbrace{\exp\left(-\frac{1}{2}\frac{x^2+y^2}{\sigma_d^2}\right)}_{\text{Noyau gaussien classique}} \underbrace{\exp\left(-\frac{1}{2}\frac{||\mathbf{I}(x_0+x, y_0+y) - \mathbf{I}(x_0, y_0)||^2}{\sigma_r^2}\right)}_{\text{Pondération par un critère de similarité photométrique}} \tag{7.2}$$

Intuitivement, le second terme élimine dans un lissage par un noyau gaussien la contribution des pixels trop différents du pixel central. Nous avons utilisé les paramètres $\sigma_d = \sigma_r = 4$, et avons appliqué successivement 5 filtrages à l'image. Un exemple de résultat est donné dans la figure 7.1.

7.1.2 Critère de couleur

Qu'il s'agisse des cymbales ou des fûts, les éléments de la batterie ont une couleur qui leur est propre. Si l'on associe à chaque pixel de l'image un vecteur d'attributs, correspondant à des descripteurs de couleur, il est possible d'entraîner un classifieur discriminant les pixels selon les deux classes *élément de la batterie* (notée par la suite E) et *autre élément* (notée par la suite $\bar{E}$).

7.1.2.1 Attributs pour la segmentation

Les attributs suivants sont ainsi extraits pour chaque pixel de l'image :

- Composantes rouges, vertes et bleues (r, g, b) du pixel, normalisées dans l'intervalle $[0, 1]$. Ces composantes s'obtiennent directement à partir de la représentation de l'image.
- Rapports entre les composantes r, g, b définis comme suit :

$$r_{rg} = \frac{r}{g} \qquad r_{rb} = \frac{r}{b} \qquad r_{gb} = \frac{g}{b} \tag{7.3}$$

- Composantes de teinte, saturation et valeur (h, s et v) du pixel. Ces composantes s'obtiennent à partir des composantes r, g, b par les relations suivantes :

$$\begin{aligned} m &= \min\{r,g,b\} \\ v &= \max\{r,g,b\} \\ s &= \begin{cases} 0 & \text{si } v = 0 \\ 1 - \frac{m}{v} & \text{sinon} \end{cases} \end{aligned} \qquad h = \begin{cases} -1 & \text{si } v = m \\ 60\frac{g-b}{v-m} & \text{si } v = r \text{ et } g \geq b \\ 60\frac{g-b}{v-m} + 360 & \text{si } v = r \text{ et } g < b \\ 60\frac{b-r}{v-m} + 120 & \text{si } v = g \\ 60\frac{r-g}{v-m} + 240 & \text{sinon} \end{cases}$$

– Composantes de couleur dans l'espace CIE $L*u*v*$, définies par les relations suivantes :

$$\begin{aligned} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \\ L* &= \begin{cases} 903.3Y & \text{si } Y < 0.008856 \\ 116\sqrt[3]{Y} - 16 & \text{sinon} \end{cases} \\ u* &= 13L* \left(\frac{4X}{X + 15Y + 3Z} - 0.197839 \right) \\ v* &= 13L* \left(\frac{9Y}{X + 15Y + 3Z} - 0.463842 \right) \end{aligned}$$

7.1.2.2 Classification des pixels

Si l'on note $\mathbf{x}_i$ le vecteur d'attributs de couleur associé au i-ème pixel d'une image, et y_i la classe correspondante ($y_i = +1$ si y_i appartient à un élément de la batterie, $y_i = -1$ sinon), on se ramène à la formulation classique d'un problème de classification supervisée. Cependant, contrairement à ce que nous avons pu faire dans la section 4.4, la quantité de données à traiter ici est bien plus importante, puisque le nombre de pixels à classifier pour segmenter une image de taille 720×576 est de l'ordre de 4×10^5. Il est donc nécessaire de choisir un classifieur dont l'évaluation de la fonction de décision est peu coûteuse en termes de temps de calcul. Cela exclut des méthodes comme les SVM, les k plus proches voisins, ou même les approches bayésiennes utilisant des mélanges de gaussiennes pour représenter les densités associées à chaque classe. Notre choix de méthode de classification des pixels pour la segmentation se porte donc vers les arbres de décision, dont la fonction de décision associée se limite à une hiérarchie de comparaisons sur les attributs. La complexité de cette fonction de décision peut être aisément contrôlée au moment de l'apprentissage en limitant la profondeur de l'arbre appris. Nous avons plus particulièrement utilisé l'algorithme d'apprentissage **C4.5** [Qui93], tel qu'il est implémenté dans WEKA [WE05].

7.1.2.3 Évaluation sur les scènes de la base ENST-drums

De manière à évaluer les performances de ce classifieur, 6 images moyennes (voir section 7.2.1.2) de séquences ont été extraites de la base ENST-drums (voir annexe C) – une image pour chacun des trois batteurs et par angle de prise de vue. Chacune de ces images a été annotée manuellement en marquant les zones correspondant aux éléments de la batterie (cymbales et sommet des fûts). Le protocole de validation choisi est celui du *leave one out* - pour chaque sous-ensemble possible de 5 images, un classifieur est entraîné sur ces 5 images et testé sur l'image restante. L'ensemble d'apprentissage est ainsi constitué, pour chaque itération, de l'ordre de 2×10^6 pixels. De manière à limiter la profondeur de l'arbre de décision construit, la valeur 4×10^5 a été donnée au critère d'arrêt de l'algorithme **C4.5**. Ainsi, les feuilles de l'arbre de décision appris ne décrivent pas moins de 2.5% des pixels de l'ensemble de l'apprentissage.

Les résultats de classification sont donnés pour différents jeux d'attributs dans le tableau 7.1. Nous pouvons constater que le passage dans les espaces de couleur transformés HSV ou $L*U*V*$ est d'intérêt limité : les performances de classification à partir des simples composantes RGB et de

Attributs utilisés	Éléments (E)			Autres ($\bar{E}$)		
	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$
HSV	69.2	78.1	73.4	96.4	94.5	95.4
L*U*V*	71.2	77.0	74.0	96.1	94.8	95.4
RGB, Rapports RGB	71.2	77.4	74.1	**96.2**	**94.8**	**95.5**
Tous	**74.6**	**74.7**	**74.6**	95.4	95.3	95.4

TAB. 7.1 – Évaluation des attributs de couleur pour la segmentation des éléments de la batterie : Rappel R, Précision P, F-mesure F

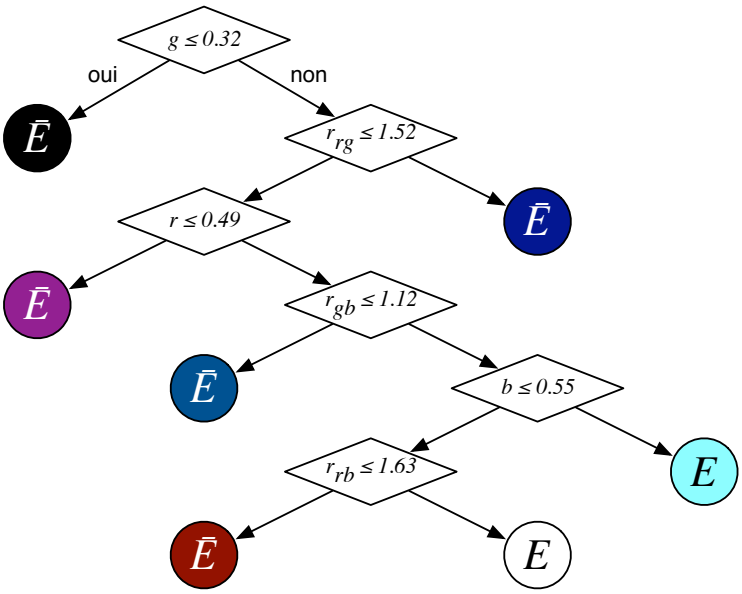


FIG. 7.2 – Critère de couleur appris

leurs rapports étant similaires à celles obtenues avec tous les attributs. Par la suite, ces seuls attributs seront utilisés, puisqu'ils correspondent à l'espace de couleur original des images que nous traitons. Un exemple d'arbre de décision appris est donné dans la figure 7.2. La segmentation d'une image de test est donnée dans la figure 7.3. Les couleurs des régions sont celles du noeud correspondant de l'arbre de décision, les régions claires sont celles d'intérêt.

Si les résultats de cette segmentation sont satisfaisants, cette méthode n'en souffre pas moins de trois défauts importants. Tout d'abord, le critère de couleur appris (région de teinte jaune, ou très lumineuse) n'est pas robuste aux variations d'éclairage ou à un mauvais calibrage des couleurs de la caméra. Ensuite, certains éléments de la scène à l'arrière-plan peuvent avoir des couleurs similaires aux éléments de la batterie – dans l'exemple donné, une partie du meuble à l'arrière-plan et le crâne du batteur sont reconnus comme régions d'intérêt. Enfin, cette méthode n'extrait pas les régions individuelles associées à chaque élément de la batterie.

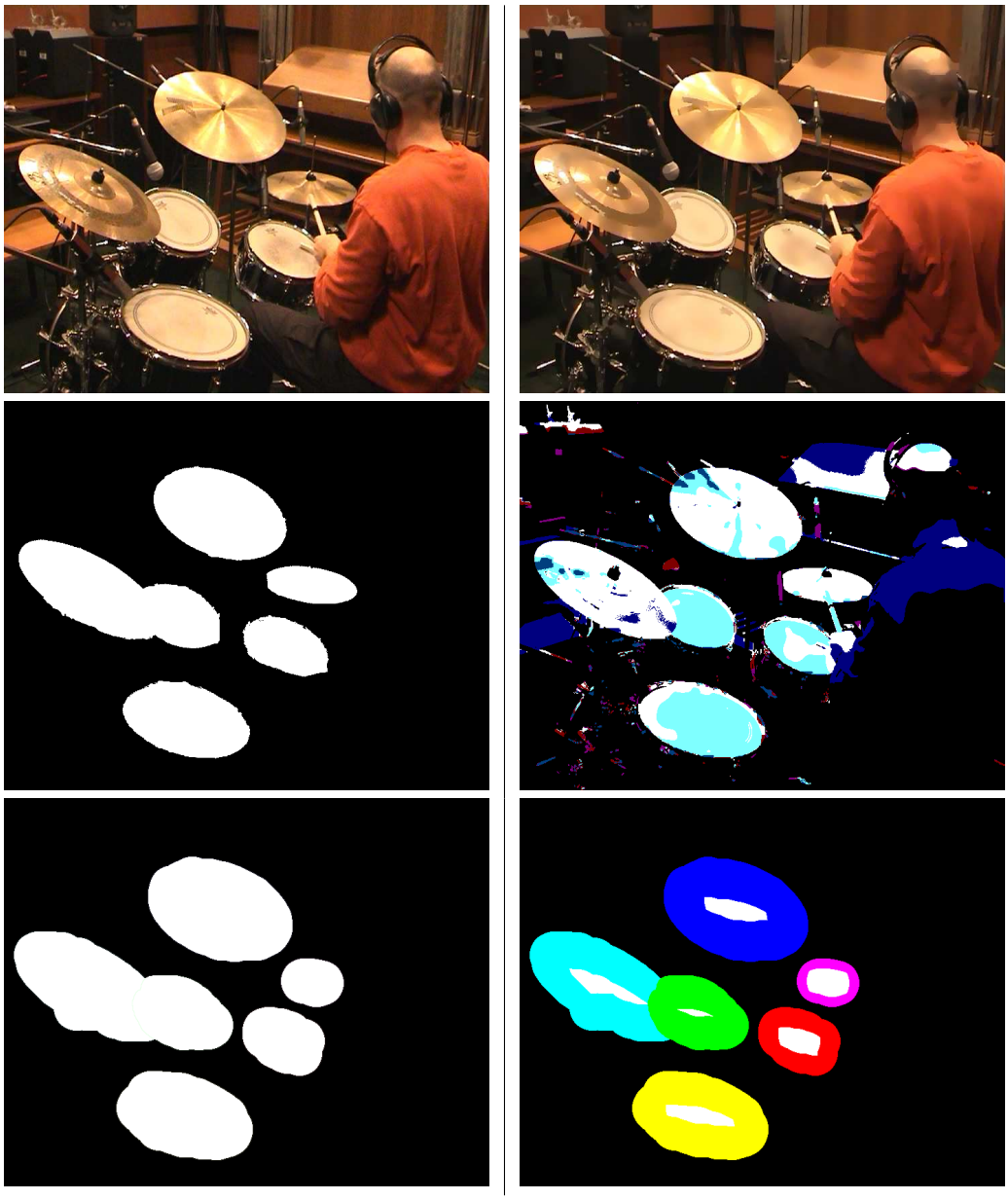


FIG. 7.3 – Segmentation des éléments de la batterie par critère de couleur : image originale, pré-traitée ; régions d'intérêt manuellement annotées et régions extraites par le critère de couleur ; post-traitements morphologiques

7.1.3 Critère morphologique

Une solution possible à ces deux derniers problèmes consiste à appliquer au résultat de la segmentation par critère de couleur une série d'opérations morphologiques choisies pour modéliser certaines connaissances a priori sur les dimensions et la forme des régions à extraire. Une première dilatation, avec pour élément structurant un disque de rayon égal à 10 pixels, permet d'inclure dans la région extraite d'éventuelles zones d'ombre sur la surface du fût ou de la cymbale, et les dômes des cymbales (leur couleur sombre les exclut de la segmentation par la couleur).

Ensuite, une ouverture par un disque de rayon de rayon égal à 30 pixels permet de ne retenir que les régions aux bords arrondis. Le résultat est donné en bas à gauche de la figure 7.3. La dernière étape consiste en l'extraction de régions individuelles. Des érosions successives par des disques de rayon égal à 3 pixels sont appliquées. À chaque itération k, si une composante connexe d'aire inférieure à 1500 pixels (correspondant alors à une version "effondrée" d'une région elliptique d'intérêt) est présente dans l'image, elle est soustraite de l'image et forme une région. Une dilatation par un disque de rayon égal à $3k$ est ensuite appliquée à chaque région extraite pour restaurer sa taille originale. Le résultat est donné en bas à droite de la figure 7.3, les régions effondrées associées à chaque composante étant représentées en blanc. Cette approche morphologique échoue cependant lorsque l'angle de vue est tel que les régions d'intérêt apparaissent comme très oblongues – c'est le cas par exemple de la hi-hat dans la figure 7.3. Il faudrait dans ce cas utiliser plusieurs éléments structurants correspondant à des ellipses allongées, sous diverses orientations. Le coût en calcul résultant de cette approche est tel que nous avons décidé de ne pas poursuivre dans cette voie.

7.1.4 Critère géométrique

Toutes les régions à extraire ayant une apparence ellipsoïdale (éventuellement occultée), ce critère géométrique peut être utilisé pour la segmentation. Les différentes étapes du système de détection d'ellipse développé[2] sont détaillées ici, et sont illustrées en 7.5 :

7.1.4.1 Extraction des contours

Les contours de l'image sont extraits par une variante de l'algorithme de Canny. Le pré-traitement par un filtre gaussien suggéré par Canny est remplacé par le filtrage bilatéral gaussien décrit en 7.1.1, de manière à préserver la netteté des contours. Le calcul du gradient est effectué sur l'image en couleurs (dans l'espace $L * u * v$) plutôt qu'en niveaux de gris. Le gradient utilisé est ainsi obtenu en pondérant les gradients calculés à l'aide d'opérateurs de Sobel de taille 3×3 sur les 3 composantes L, $u*$ et $v*$. Les étapes suivantes – éliminations des non-maxima de gradient et seuillage à hysteresis des contours – sont inchangées. Est ainsi obtenue une image en niveaux de gris $\mathcal{C}(x, y)$, telle que $\mathcal{C}(x, y)$ est nulle si (x, y) n'est pas sur un contour, et est égale à la norme du gradient en ce point sinon. $\mathcal{C}(x, y)$ est seuillée avec deux seuils ; un seuil bas (20) pour obtenir une représentation détaillée des contours $\mathcal{C}_d(x, y)$, et un seuil haut (80) pour obtenir une représentation grossière $\mathcal{C}_g(x, y)$ des contours.

7.1.4.2 Regroupement des pixels de contour

Les points de $\mathcal{C}_g(x, y)$ sont ensuite regroupés pour former des groupes $(C_i)_{i\in\{1,\ldots,N\}}$ de points susceptibles d'appartenir au contour d'un même objet. Le regroupement s'effectue selon deux critères (un exemple est donné figure 7.4) :

[2]Ce travail a été réalisé en collaboration avec Kevin McGuinness, du *Centre for Digital Video Processing*, Dublin City University.

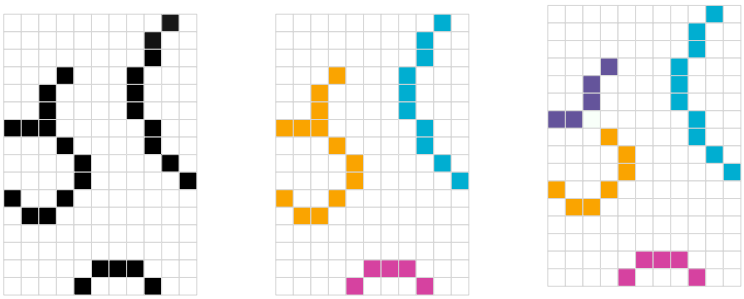


FIG. 7.4 – Regroupement des contours : critère de proximité, prise en compte de la courbure

1. Regroupement par proximité : des pixels voisins (au sens de la connexité 8) seront associés au même groupe. Ce critère seul est cependant susceptible de regrouper les contours de deux objets distincts, l'un occultant l'autre. Le deuxième critère évite cette situation.
2. Non-regroupement par critère de courbure : La courbure locale est calculée en chaque point de C_i. Si une valeur forte de courbure est détectée en (x, y), les voisins de (x, y) sont associés à des groupes différents.

La règle de regroupement est ainsi la suivante : Si $(x_0, y_0) \in C_i$, si (x, y) est dans le voisinage en connexité 8 de (x_0, y_0), si $\mathcal{C}_g(x, y) = 1$, et si (x, y) n'est pas un point de courbure élevée alors $(x, y) \in C_i$. La courbure en un point (x_0, y_0) est mesurée comme l'inverse du rayon du cercle osculateur en ce point. Le rayon du cercle osculateur est approximé de la façon suivante : les points de contour dans un voisinage circulaire de rayon r de (x_0, y_0), c'est à dire vérifiant $\mathcal{C}_g(x, y) = 1$ et $(x - x_0)^2 + (y - y_0)^2 < r$ sont considérés. Le rayon du meilleur cercle passant par ces points est estimé, à l'aide de la méthode décrite dans [Tau91].

7.1.4.3 Recherche d'ellipses

Les ensembles de points C_i, $1 \leq i \leq N$, et $C_i \cup C_j$, $1 \leq i < j \leq N$ sont successivement considérés. Pour chaque ensemble de points, les paramètres d'une ellipse optimale passant par ces points sont déterminés, et différents critères sont utilisés pour déterminer sa pertinence. Si l'ellipse est acceptée, les groupes contenant les points considérés sont éliminés. L'ajustement des paramètres est réalisé par la méthode des moindres carrés décrite par Fitzgibbon et al. dans [FPF99]. Soient $S = \{(x_i, y_i), 1 \leq i \leq n\}$ un ensemble des points considéré, $\mathbf{x}_i = \begin{bmatrix} x_i^2 & x_i y_i & y_i^2 & x_i & y_i & 1 \end{bmatrix}^T$, et $\mathbf{\Theta} = \begin{bmatrix} a & b & c & d & e & f \end{bmatrix}^T$ les paramètres de l'ellipse. L'ellipse optimale de paramètres $\mathbf{\Theta}^*$ vérifie :

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\operatorname{argmin}} \sum_{i=1}^{n} (\mathbf{\Theta}^T \mathbf{x}_i)^2 \tag{7.4}$$

$$b^2 < 4ac \tag{7.5}$$

Supposant une mise à l'échelle des coefficients, la deuxième contrainte peut s'écrire : $4ac - b^2 = 1$, soit $\mathbf{\Theta}^T \mathbf{C} \mathbf{\Theta} = 1$ avec :

$$C_{ij} = \begin{cases} 2 & (i, j) \in \{(1, 3), (3, 1)\} \\ -1 & i = j = 2 \\ 0 & \text{sinon} \end{cases} \tag{7.6}$$

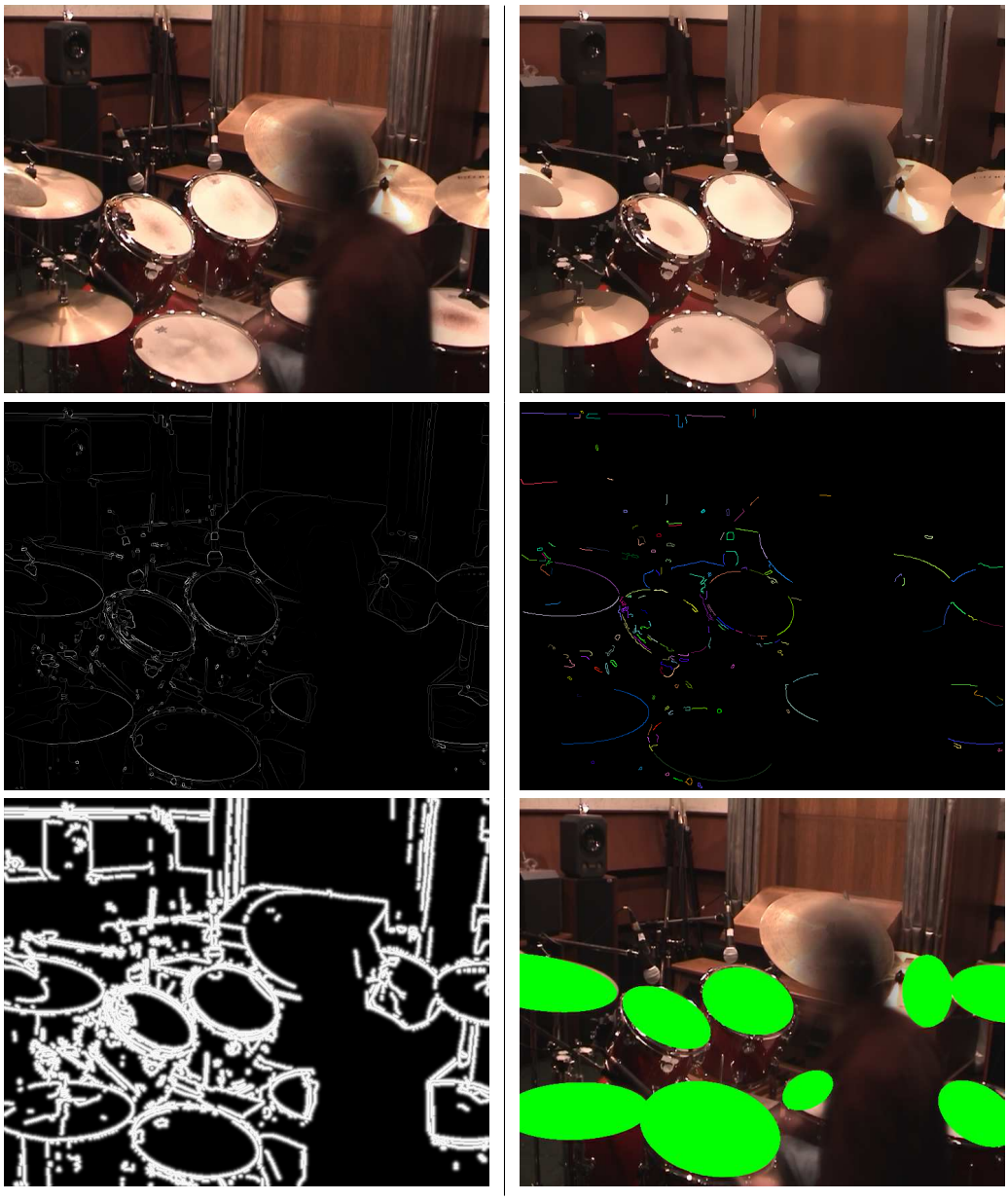


FIG. 7.5 – Détection d'ellipses : image originale, image pré-traitée, contours, groupes de contours, distance de chaque point aux contours détaillés, ellipses détectées

Il est montré dans [FPF99] que le problème de minimisation peut être reformulé sous forme d'un problème de valeurs propres généralisées :

$$\mathbf{D}^T\mathbf{D}\boldsymbol{\Theta} = \lambda\mathbf{C}\boldsymbol{\Theta} \tag{7.7}$$

Où $\mathbf{D} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n \end{bmatrix}^T$. Les paramètres optimaux correspondent alors au seul vecteur propre dont la valeur propre associée est positive. La pertinence de l'ellipse paramétrée par $\boldsymbol{\Theta}^*$, notée $\mathcal{E}_{\boldsymbol{\Theta}^*}$, est ensuite mesurée par les critères suivants :

		Sans modèle de couleur			Avec modèle de couleur		
Batteur	Angle	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$
1	1	83	33	47	**83**	**100**	**91**
1	2	**100**	**100**	**100**	**100**	**100**	**100**
2	1	67	38	48	**67**	**73**	**70**
2	2	56	56	56	**56**	**83**	**67**
3	1	37	23	28	**37**	**100**	**54**
3	2	90	70	79	**90**	**100**	**95**

TAB. 7.2 – Évaluation de la détection d'ellipses pour la segmentation des éléments de la batterie : Rappel R, Précision P, F-mesure F

Dimensions Les dimensions des grand et petit axes, ainsi que l'aire de l'ellipse, sont restreintes à un intervalle fixé – $[20, 200]$ pixels pour les dimensions, $[1500, 10000]$ pixels pour l'aire.

Mesure d'ajustement des points à l'ellipse Pour chaque point de l'ensemble S pour lequel l'ellipse optimale a été estimée, la mesure d'ajustement suivante est calculée :

$$C_1 = \frac{1}{|S|} \sum_{\mathbf{x} \in S} \exp\left(\frac{-d(\mathbf{x}, \mathcal{E}_{\mathbf{\Theta}^*})^2}{2\sigma^2}\right) \tag{7.8}$$

Où $d(\mathbf{x}, \mathcal{E}_{\mathbf{\Theta}^*}) = \min_{\mathbf{e} \in \mathcal{E}_{\mathbf{\Theta}^*}} d(\mathbf{x}, \mathbf{e})$ est la distance d'un point à l'ellipse. Cette mesure prend une valeur dans l'intervalle $]0, 1]$. Une ellipse doit vérifier $C_1 > 0.8$ pour être sélectionnée, avec la tolérance $\sigma = 4$ pixels.

Mesure d'ajustement de l'ellipse aux contours Soit $E_{\mathbf{\Theta}^*}$ l'ensemble des pixels constituant $\mathcal{E}_{\mathbf{\Theta}^*}$ après *rasterisation*, réalisée selon [Bon02]. La mesure d'ajustement suivante est calculée :

$$C_2 = \frac{1}{|E|} \sum_{\mathbf{e} \in E_{\mathbf{\Theta}^*}} \exp\left(\frac{-d(\mathbf{e}, \mathcal{C}_d)^2}{2\sigma^2}\right) \tag{7.9}$$

Les calculs de la distance de chaque point du contour de l'ellipse candidate aux contours détaillés détectés $d(\mathbf{e}, \mathcal{C}_d)$ sont réalisés en calculant une fois pour toute la transformée de distance euclidienne de l'image $\mathcal{C}_d$, à l'aide de l'algorithme de programmation dynamique décrit dans [DH04]. Une ellipse est rejetée si $C_2 < 0.5$.

Occlusion Est calculée la proportion de pixels à l'intérieur de l'ellipse considérée occultant les ellipses précédemment détectées. Une ellipse occultant plus de 40% d'une ellipse précédemment détectée est rejetée.

Pour réduire le coût en calculs, ces critères sont vérifiés dans cet ordre. Il est de plus possible, dans le cas où l'éclairage de la scène est bien contrôlé, de prendre en compte les contraintes de couleur suivantes :

Homogénéité de couleur La somme des variances des composantes L, $u*$ et $v*$ des pixels à l'intérieur de la partie non occultée de l'ellipse considérée est calculée. Une ellipse est rejetée si la variance totale excède 21.

Compatibilité des couleurs avec un modèle Une ellipse est rejetée si elle contient plus de 20% de pixels considérés comme n'appartenant pas à un élément de la batterie selon le modèle de couleur présenté en 7.1.2.

7.1.4.4 Évaluation sur les scènes de la base ENST-drums

Les 6 images moyennes utilisées précédemment ont été utilisées pour l'évaluation, avec les valeurs des paramètres données dans la section précédente. Seul le critère de couleur utilisé pour

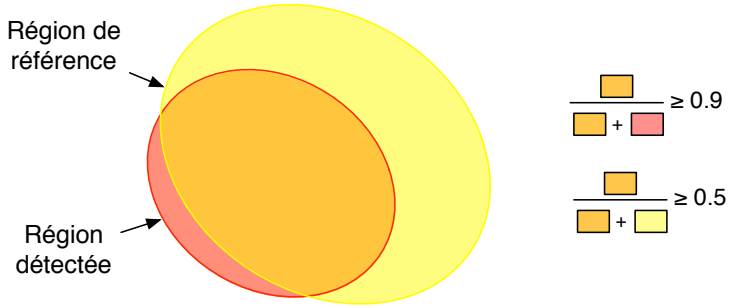


FIG. 7.6 – Critère de validité des régions obtenues par segmentation

sélectionner les ellipses pertinentes demande un apprentissage – ce critère a été appris sur toutes les images autres que l'image évaluée. Les ellipses obtenues et donc la segmentation produite ont été évaluées par comparaison avec une segmentation de référence produite par un opérateur humain : une ellipse est considérée valide s'il existe une région R dans la segmentation de référence telle qu'au moins 50% des pixels de R soient à l'intérieur de l'ellipse, et qu'au moins 90% des pixels à l'intérieur de l'ellipse soient aussi dans R (voir figure 7.6).

Les taux de rappel et de précision sont donnés dans la table 7.2. Les résultats suggèrent que le critère de couleur doit nécessairement être pris en compte pour que la segmentation ne produise pas de régions incorrectes. Nous ferons donc par la suite la supposition que les conditions d'éclairage permettent l'utilisation d'un tel critère.

7.2 Segmentation des éléments dans une séquence d'images

Les méthodes de segmentation présentées jusqu'ici ne traitent qu'une trame individuelle d'une séquence vidéo. Nous étendons maintenant ces méthodes (dans la section 7.2.1), ou en introduisons de nouvelles (dans les sections 7.2.2 et 7.2.3) pour prendre en compte la dimension temporelle d'une séquence vidéo, et produire une unique segmentation à partir de l'ensemble des trames de la séquence.

7.2.1 De la segmentation d'images fixes à la segmentation de séquences d'images

7.2.1.1 Fusion des segmentations

Une première approche consiste à appliquer l'algorithme de segmentation présenté en 7.1.4 à chaque trame de la séquence vidéo. Soient $(\mathcal{R}_i(m))_{i\in\{1,\ldots,n(m)\}}$ les $n(m)$ régions produites pour chaque trame m de la séquence, $1 \leq m \leq M$.

La fusion des segmentations est aisée quand $n(m) = 1, \forall m$, et qu'une seule région est à extraire : on peut par exemple utiliser une procédure de vote et former la région $\mathcal{R}$ contenant les points présents, dans une large proportion τ, dans les régions individuelles $\mathcal{R}_1(m)$:

$$(x,y) \in \mathcal{R} \Leftrightarrow \left(\frac{1}{M} \sum_{m=1}^{M} \mathbb{I}_{\mathcal{R}_1(m)}((x,y)) \right) \geq \tau \tag{7.10}$$

Cette procédure n'est plus valable quand le nombre de régions à extraire est plus grand que 1. Par exemple, lorsque la scène comporte deux régions d'intérêt, que les régions $\mathcal{R}_i(m)$ correspondent au premier objet pour la moitié des trames de la séquence, et au deuxième objet pour les autres trames, une procédure de vote avec $\tau < 0.5$ n'extrairait aucune région, et avec $\tau > 0.5$ n'extrairait qu'une seule région constituée de l'union des deux régions d'intérêt. D'autres situations difficiles peuvent être rencontrées : des régions invalides peuvent temporairement être extraites sur certaines trames, les frontières d'une région peuvent varier d'une trame à l'autre selon l'occlusion, et une région peut temporairement n'être que partiellement extraite en cas d'occlusion (C'est le cas d'une cymbale ou du tom basse, occultés par le batteur dans la figure 7.5). La solution retenue consiste à former des groupes de régions similaires, parmi toutes les régions extraites sur l'ensemble des trames, et à sélectionner le représentant de chacun des groupes les plus représentés.

Soit $\mathcal{R} = \bigcup_{m=1}^{M} \bigcup_{i=1}^{n(m)} \{\mathcal{R}_i(m)\}$ l'ensemble des régions extraites. Des groupes de régions peuvent être formés à l'aide d'un algorithme de regroupement agglomératif glouton ([DHS01], pp 552–553), qui regroupe à chaque étape les régions les moins dissimilaires. L'usage de mesures de dissimilarités classiques entre régions, comme le nombre de pixels présents dans la différence symétrique des deux régions, n'est pas envisageable ici car trop coûteux en calculs (plusieurs milliers de régions sont extraites sur les séquences considérées). Nous utilisons ici un critère plus simple, tirant parti du fait que les régions à comparer sont des ellipses. Les ellipses extraites peuvent être en effet paramétrisées sous la forme$(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, où $\boldsymbol{\mu}$ est leur centre et $\boldsymbol{\Sigma}$ est une matrice dont les valeurs propres positives sont les dimensions des grand et petit axes, et dont les vecteurs propres définissent les directions de ces axes. La dissimilarité entre deux ellipses peut alors être mesurée par les mêmes critères que ceux utilisés traditionnellement pour comparer des distributions gaussiennes bivariées (dont les supports sont des ellipses), comme par exemple la distance de Bhattacharrya :

$$d_B(\mathcal{R}_1, \mathcal{R}_2) = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left[\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \right]^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \log \frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}} \tag{7.11}$$

Le résultat de la procédure de regroupement agglomératif est un dendogramme, dont une coupe à un seuil de distance donné (ici, $\delta = 0.15$) fournit des groupes de régions. Les groupes contenant plus de $0.4M$ (c'est à dire, correspondant à des régions identifiées dans plus de 40% des trames de la séquence) sont retenus.

7.2.1.2 Fusion des images puis segmentation

La procédure de segmentation étant très coûteuse en calculs, une méthode plus efficace consiste à fusionner d'abord les images de la séquence pour obtenir une image unique, sur laquelle la segmentation sera appliquée une seule fois. L'intérêt de cette fusion est qu'elle peut permettre d'éliminer l'occlusion temporaire d'un élément de la batterie par le corps du batteur.

La méthode la plus simple consiste à moyenner les images de la séquence à traiter. Cependant, elle produit un flou autour des instruments de la batterie souvent mis en mouvement (hi-hat par exemple), et crée des différences d'intensité visibles dans les régions temporairement occultées (voir figure 7.7).

Une autre solution que nous avons développée utilise une variante non-adaptative de l'algorithme de segmentation de l'image en arrière-plan/avant-plan détaillé dans la section 7.3.

Soit $P(x,y) = \{\mathbf{I}(x,y,m), 1 \leq m \leq M\}$ l'ensemble des vecteurs contenant les composantes RGB que prend le pixel (x,y) au long de la séquence. Ces vecteurs sont considérés comme des observations indépendantes, identiquement distribuées, dont la densité est modélisée par un mélange de $K = 3$ gaussiennes multivariées de moyenne $\boldsymbol{\mu}_k^{(x,y)}$, de matrice de covariance diagonale $\boldsymbol{\Sigma}_k^{(x,y)}$, et de poids $\pi_k^{(x,y)}$, $k \in \{1,2,3\}$. Les paramètres $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)^{(x,y)}$ peuvent être estimés au maximum

FIG. 7.7 – Fusion des images pour l'élimination de l'occlusion avant segmentation : modélisation de l'arrière-plan et moyennage

de vraisemblance par l'algorithme EM. Chaque composante du mélange peut s'interpréter comme la contribution d'un objet susceptible d'être trouvé en (x, y) : le poids $\pi_k^{(x,y)}$ indique la proportion de trames dans lesquelles cet objet est présent en (x, y), $\boldsymbol{\mu}_k^{(x,y)}$ représente sa couleur moyenne, et $\boldsymbol{\Sigma}_k^{(x,y)}$ la variabilité de sa couleur. Nous pouvons alors déterminer parmi ces K composantes celle expliquant la couleur des pixels d'arrière-plan. En effet, lorsque les conditions d'éclairage sont fixes et que l'arrière-plan est statique, la couleur d'un objet d'arrière-plan est fixe, et est donc issue d'une composante de mélange dotée d'une faible variance. Par ailleurs, si l'on suppose que l'occlusion par les membres du batteur est temporaire, la composante du mélange dotée du poids le plus fort est celle qui explique la couleur de l'arrière-plan. Un compromis entre ces deux règles permet ainsi de construire une image d'arrière-plan $B(x, y)$ à partir des modèles appris :

$$B(x,y) = \boldsymbol{\mu}_{k^*(x,y)} \tag{7.12}$$

$$\text{avec } k^*(x,y) = \underset{k}{\operatorname{argmax}} \frac{\pi_k^{(x,y)}}{\sqrt{|\boldsymbol{\Sigma}_k^{(x,y)}|}} \tag{7.13}$$

Un exemple d'image d'arrière-plan extraite est donné dans la figure 7.7. On constate que l'occlusion causée par les membres du batteur a été éliminée avec succès. Par contre, les éléments de la batterie fréquemment mis en mouvement (hi-hat) apparaissent rétrécis, et leurs bords sont crénelés. En effet, lorsqu'ils sont mis en mouvement, ils révèlent une partie de l'arrière-plan derrière eux, qui sera extraite.

Soulignons que dans le cas où les conditions d'éclairage varient au cours du temps, cette méthode peut être mise en difficulté : si l'on considère par exemple que les conditions d'éclairage ont été modifiées au milieu de la séquence, une des composantes expliquera l'arrière-plan avant le changement, une autre composante expliquera l'arrière-plan après le changement. Ces deux composantes auront des poids voisins – ce sera donc le critère de variance qui déterminera laquelle des composantes, en chaque pixel, formera l'arrière-plan. Le risque est grand que l'image d'arrière-plan formée $B(x, y)$ alterne les pixels sous les deux conditions d'éclairage, créant du bruit, ou des arêtes superflues. Dans ce cas, le simple moyennage des trames permet une estimation plus robuste de l'arrière-plan.

7.2.2 Segmentation par factorisation du mouvement

Supposons que l'on dispose d'une fonction $\mathcal{A}(x, y, m)$ mesurant la quantité de mouvement (par exemple la norme du vecteur vitesse) au point (x, y) à la trame m. Les objets à segmenter étant rigides, tous les points qui les composent sont donc mis en mouvement simultanément. Par ailleurs, le déplacement des objets à segmenter autour de leur position au repos est limitée à quelques pixels pour les fûts, quelques dizaines de pixels pour les cymbales. On peut alors approximer $\mathcal{A}(x, y, m)$ sous la forme :

$$\mathcal{A}(x, y, m) \approx \sum_{k=1}^{K} a_k(m)\mathcal{A}_k(x, y) \tag{7.14}$$

Où $a_k(m) \geq 0$ représente l'activation de l'objet k à la trame m, et $\mathcal{A}_k(x, y) \geq 0$ est un masque nul pour (x, y) hors de la région associée à l'objet k. Une telle approximation peut être obtenue par factorisation non négative (NMF) de la matrice $\mathbf{A}$ définie par $A_{i+jW,k} = \mathcal{A}(i, j, k)$, où W est la largeur de l'image. Notons que nous avons jusqu'ici négligé la contribution des mouvements du batteur dans $\mathcal{A}(x, y, m)$. Nous pouvons soit :

- Considérer que cette contribution peut également s'écrire sous la forme $\sum_{k=1}^{K} a_k(m)\mathcal{A}_k(x, y)$. Dans ce cas, les composantes obtenues par factorisation non-négative expliqueront à la fois les mouvements du batteur et des éléments de la batterie.
- Utiliser le critère de couleur défini en 7.1.2 pour déterminer si le pixel en $\mathbf{I}(x, y, m)$, correspond ou non à un élément de la batterie. C'est cette solution que nous avons retenue.

Soit $C(\mathbf{I}(x, y, m))$ la fonction prenant la valeur 1 si le pixel $\mathbf{I}(x, y, m)$ a la couleur d'un élément de la batterie, 0 sinon. Un estimateur simple d'intensité de mouvement peut être obtenu en considérant la différence entre deux trames successives $\Delta(x, y, m) = ||\mathbf{I}(x, y, m) - \mathbf{I}(x, y, m - 1)||$. Nous utilisons ainsi :

$$\mathcal{A}(x, y, m) = \begin{cases} 0 & \text{si } C(\mathbf{I}(x, y, m)) = 0 \\ 0 & \text{si } C(\mathbf{I}(x, y, m)) = 1 \text{ et } \Delta(x, y, m) < \tau \\ \Delta(x, y, m) & \text{sinon} \end{cases} \tag{7.15}$$

La matrice $\mathbf{A}$ est formée et une factorisation non-négative en est obtenue, définissant des masques $\mathcal{A}_k(x, y)$. Cependant, ces masques ne fournissent pas immédiatement les régions d'intérêt. Tout d'abord, le critère de couleur ne discrimine pas toujours correctement les éléments de la batterie, et $\mathcal{A}(x, y, m)$ peut ainsi parfois inclure une contribution correspondant au mouvement des baguettes ou de la tête du batteur. Ensuite, un même élément de la batterie peut être représenté par plusieurs composantes – dans nos expériences, c'est par exemple le cas de la cymbale crash qui peut être frappée à des positions différentes. Enfin, deux éléments distincts, en particulier s'ils sont fréquemment joués simultanément, peuvent occuper la même composante. Pour remédier à ces situations, le nombre de composantes à extraire est volontairement fixé à une valeur élevée ($K = 25$), et les composantes extraites sont classées et regroupées : L'algorithme de détection d'ellipses présenté en 7.5 est appliqué sur chacun des masques $\mathcal{A}_k(x, y)$, et les ellipses éventuellement produites sont groupées selon la méthode décrite en 7.2.1.

Des exemples de masques extraits pour trois instruments (hi-hat, tom medium et cymbale crash) sont donnés dans la figure 7.8. Soulignons que l'intérêt de cette approche est limité par la difficulté des post-traitements visant à classer et regrouper les masques extraits. En particulier l'élimination des composantes dues au mouvement du batteur requiert un critère de couleur et une détection d'ellipses dans les masques $\mathcal{A}_k(x, y)$ – deux sous-systèmes pouvant à eux seuls fournir une segmentation satisfaisante.

FIG. 7.8 – Masques obtenus par factorisation non-négative d'une mesure de la quantité de mouvement des éléments de la batterie

7.2.3 Segmentation supervisée : calibration à partir d'une transcription de référence

Dans les applications d'interaction musicien/machine, il serait envisageable de demander au musicien de jouer, à des fins de calibration, une courte séquence de référence utilisant tous les instruments de la batterie ; ou bien de jouer individuellement chaque instrument de façon suffisamment lente et détachée pour qu'on puisse considérer la transcription audio qui en résulterait comme parfaite.

Soit i un instrument de la batterie et $\mathbb{I}_i(m)$ une fonction obtenue à partir d'une transcription audio idéale, ou de la partition de référence, valant 1 si l'instrument i est joué à la trame m, et 0 sinon. On cherche à former, à partir de $\mathbb{I}_i(m)$, une fonction $a_i(m)$ exprimant l'intensité de mouvement de l'instrument i à la trame m – mesurée par exemple comme la moyenne des normes des vecteurs vitesse sur la surface de l'instrument. Deux comportements sont à distinguer : les fûts et la hi-hat fermée reviennent très rapidement à leur position au repos, tandis que les cymbales sont libres de se déplacer par rapport à leur position au repos. Pour chaque instrument, est ainsi définie une enveloppe temporelle $e_i(m)$. Pour les fûts et la hi-hat, $e_i(m)$ est une exponentielle décroissante de constante de temps égale à 3 trames, pour les autres cymbales, $e_i(m)$ est une exponentielle décroissante de constante de temps égale à 15 trames. La quantité de mouvement prédite $\hat{a}_i(m)$ pour l'instrument i est ainsi :

$$\hat{a}_i(m) = (\mathbb{I}_i * e_i)(m) \tag{7.16}$$

Soit $\mathcal{A}(x, y, m)$ la mesure de quantité de mouvement décrite dans la section précédente. En suivant l'approche présentée par Hershey et Movellan dans [HM00], nous pouvons associer à l'instrument i la région constituée des pixels (x, y) tels que l'information mutuelle entre l'intensité de mouvement $\mathcal{A}(x, y, m)$ observée et l'intensité de mouvement prédite $\hat{a}_i(m)$ dépasse un seuil τ :

$$-\frac{1}{2}\log(1 - \hat{\rho}^2_{x,y,i}) > \tau \tag{7.17}$$

Où $\hat{\rho}_{x,y,i}$ est l'estimée du coefficient de corrélation de Pearson entre $\mathcal{A}(x, y, m)$ et $\hat{a}_i(m)$:

$$\hat{\rho}_{x,y,i} = \frac{\sum_{m=1}^{M} \underline{\hat{a}_i}(m)\underline{\mathcal{A}}(x, y, m)}{\sqrt{\left(\sum_{m=1}^{M} \underline{\hat{a}}_i^2(m)\right)\left(\sum_{m=1}^{M} \underline{\mathcal{A}}^2(x, y, m)\right)}} \tag{7.18}$$

$\underline{\hat{a}_i}$ et $\underline{\mathcal{A}}$ désignent respectivement les versions centrées de $\hat{a}_i$ et $\mathcal{A}$. Si l'on suppose que les éléments sont en mouvement une fraction négligeable du temps, on a $\underline{\hat{a}_i} \approx \hat{a}_i$ et $\underline{\mathcal{A}} \approx \mathcal{A}$, et l'on retrouve la méthode utilisée dans [GR05a] pour la calibration automatique.

Un exemple est donné dans la figure 7.9.

FIG. 7.9 – Régions extraites par corrélation de l'intensité de mouvement dans l'image avec l'intensité de mouvement prédite par la transcription de référence : cymbales crash et hi-hat

7.3 Segmentation des baguettes

Nous nous intéressons maintenant à la segmentation du batteur et des baguettes dans une séquence vidéo. Si l'on néglige le mouvement des éléments de la batterie, ce problème peut être formulé comme un problème de segmentation d'objets en mouvement par rapport à l'arrière-plan. Cette formulation a l'avantage de n'exiger aucun a priori sur la forme et la couleur des baguettes. Elle est donc robuste à la fois au flou de bougé (à cause duquel une baguette peut apparaître comme un secteur circulaire), et au jeu avec balais, mailloches ou fagots.

La segmentation arrière-plan fixe/avant-plan animé est classiquement effectuée par des méthodes adaptatives d'estimation et de soustraction de l'arrière-plan. Ces méthodes consistent à classer chaque pixel de l'image en les catégories avant/arrière plan, selon leur différence avec l'image d'arrière-plan, puis à mettre à jour l'arrière-plan à partir des pixels classés comme y appartenant (voir par exemple [RMK95]).

Nous avons ici utilisé une variante de la méthode proposée par Stauffer et Grimson dans [SG99]. Nous rappelons que $P(x, y) = \{\mathbf{I}(x, y, m), 1 \leq m \leq M\}$ est l'ensemble des vecteurs contenant les composantes RGB que prend le pixel (x, y) au long de la séquence. Dans la section 7.2.1.2, nous avons fait l'hypothèse que les vecteurs $P(x, y)$ pouvaient être considérés comme des observations indépendantes, identiquement distribuées selon un mélange de K gaussiennes multivariées. Stauffer et Grimson proposent un modèle dans lequel les paramètres du mélange – poids $\pi_k^{(x,y)}$, moyennes $\boldsymbol{\mu}_k^{(x,y)}$ et matrices de covariance $\boldsymbol{\Sigma}_k^{(x,y)}$ varient au cours du temps. Cela offre deux avantages pratiques. Tout d'abord, l'apprentissage d'un tel modèle se fait en ligne, et est donc à la fois causal et peu coûteux en termes de calculs. Ensuite, cela autorise le modèle de l'arrière plan à varier lentement au cours du temps. Dans les applications de suivi de traffic, cela permet par exemple de prendre en compte les variations d'éclairage au long de la journée. Dans notre application, cela permet d'inclure dans le modèle de l'arrière plan le buste et la tête du batteur, dont les mouvements se limitent à des changements lents de posture – ne sont ainsi suivis que les mouvements des mains, des bras et des baguettes.

La mise à jour du modèle, pour un pixel (x, y) à la trame m se fait de la façon suivante. Tout d'abord les probabilités que le pixel observé $\mathbf{I}(x, y, m)$ soit issu de chacune des K composantes du mélange sont calculées. Deux cas se présentent :

– Si ces probabilités sont très faibles, la composante $k^\dagger$ de poids le plus faible est remplacée par une composante de poids faible, de moyenne $\mathbf{I}(x, y, m)$, et de variance élevée :

$$\pi_{k^\dagger}^{(x,y)}(m) = 0.1 \tag{7.19}$$

$$\boldsymbol{\mu}_{k^\dagger}^{(x,y)}(m) = \mathbf{I}(x,y,m) \tag{7.20}$$

$$\boldsymbol{\Sigma}_{k^\dagger}^{(x,y)}(m) = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 30 \end{bmatrix} \tag{7.21}$$

– Sinon, soit $M_k^{(x,y)}(m)$ une fonction de k valant 1 si k est la composante dont est le plus vraisemblablement issu $\mathbf{I}(x,y,m)$, 0 sinon. Les paramètres du modèle sont mis à jour selon :

$$\pi_k^{(x,y)}(m) = (1-\alpha)\pi_k^{(x,y)}(m-1) + \alpha M_k^{(x,y)}(m) \tag{7.22}$$

$$\boldsymbol{\mu}_k^{(x,y)}(m) = (1-\rho)\boldsymbol{\mu}_k^{(x,y)}(m-1) + \rho\mathbf{I}(x,y,m) \tag{7.23}$$

$$\left(\boldsymbol{\Sigma}_k^{(x,y)}(m)\right)^2 = (1-\rho)\left(\boldsymbol{\Sigma}_k^{(x,y)}(m-1)\right)^2 + \rho\mathbf{C}(x,y,m) \tag{7.24}$$

Avec :

$$\mathbf{C}(x,y,m) = (\mathbf{I}(x,y,m) - \boldsymbol{\mu}_k^{(x,y)}(m))^T(\mathbf{I}(x,y,m) - \boldsymbol{\mu}_k^{(x,y)}(m)) \tag{7.25}$$

$$\rho = \alpha\frac{p(\mathbf{I}(x,y,m)|\boldsymbol{\mu}_k^{(x,y)}(m), \boldsymbol{\Sigma}_k^{(x,y)}(m))}{\sum_{k=1}^K p(\mathbf{I}(x,y,m)|\boldsymbol{\mu}_k^{(x,y)}(m), \boldsymbol{\Sigma}_k^{(x,y)}(m))} \tag{7.26}$$

Pour chaque pixel de l'image, on considère que la composante de poids le plus fort et de variance la plus faible explique le fond de l'image, qu'on peut reconstruire selon :

$$B(x,y,m) = \boldsymbol{\mu}_{k^*(x,y,m)} \tag{7.27}$$

$$\text{avec } k^*(x,y,m) = \underset{k}{\operatorname{argmax}} \frac{\pi_k^{(x,y)}(m)}{\sqrt{|\boldsymbol{\Sigma}_k^{(x,y)}(m)|}} \tag{7.28}$$

Un pixel est considéré comme appartenant à l'avant-plan (dans notre cas, aux baguettes) si la composante dont il est le plus vraisemblablement issu n'est pas la composante expliquant le fond :

$$F(x,y,m) = \begin{cases} 0 & \text{si } k^*(x,y,m) = \operatorname{argmax}_k p(\mathbf{I}(x,y,m)|\boldsymbol{\mu}_k^{(x,y)}(m), \boldsymbol{\Sigma}_k^{(x,y)}(m)) \\ 1 & \text{sinon} \end{cases} \tag{7.29}$$

Un exemple est donné dans la figure 7.10, pour deux trames tirées de la même séquence. Le modèle de l'arrière-plan s'est adapté pour prendre en compte le changement de posture du batteur.

Il est également possible de définir une mesure souple d'appartenance à l'avant-plan, correspondant à la probabilité (normalisée) que le pixel observé est issu d'une autre composante que celle expliquant l'arrière-plan :

$$p_F(x,y,m) = \frac{\sum_{k\neq k^*(x,y,m)} p(\mathbf{I}(x,y,m)|\boldsymbol{\mu}_k^{(x,y)}(m), \boldsymbol{\Sigma}_k^{(x,y)}(m))}{\sum_{k=1}^K p(\mathbf{I}(x,y,m)|\boldsymbol{\mu}_k^{(x,y)}(m), \boldsymbol{\Sigma}_k^{(x,y)}(m))} \tag{7.30}$$

7.4 Conclusion

Nous avons introduit dans ce chapitre différentes méthodes de segmentation d'images pouvant être utilisées pour l'analyse visuelle de scènes de jeu de batterie. Un modèle de couleur des éléments

FIG. 7.10 – Segmentation des baguettes par segmentation de l'avant-plan en mouvement. Modèle de l'arrière-plan, et trame originale avec marquage coloré de l'avant-plan détecté

de la batterie a été proposé. En dépit de sa précision, il ne permet pas, utilisé seul, d'obtenir une segmentation individuelle de chaque élément de la batterie. Un post-traitement de la segmentation obtenue par des opérateurs morphologiques modélisant des connaissances sur la forme et la dimension des instruments est possible, mais coûteux, car l'apparence d'un élément dépend de l'angle de prise de vue. Une voie plus prometteuse consiste en l'utilisation d'un critère géométrique : les éléments de la batterie peuvent être efficacement segmentés en extrayant des ellipses dans la scène. La méthode proposée consiste à extraire les contours de l'image, à former des groupes de pixels de contour connexes, à ajuster les paramètres d'une ellipse passant par les pixels de chaque groupe ou couple de groupes, et à sélectionner les ellipses sur des critères de taille, d'ajustement aux contours de l'image, d'occlusion, et éventuellement de couleur.

Deux approches ont été discutées pour appliquer ces méthodes de segmentation à une séquence d'images : la fusion des segmentations, par clustering des régions extraites ; ou la fusion des images avant segmentation, par extraction d'une image d'arrière-plan éliminant l'occlusion des instruments par les baguettes ou les membres du batteur. Les résultats les plus satisfaisants ont été obtenus à l'aide d'un modèle d'arrière plan utilisant un mélange de gaussiennes. Nous avons également introduit deux méthodes basées sur le mouvement : une méthode non-supervisée basée sur la NMF, et extrayant des régions mises en mouvement simultanément ; ainsi qu'une méthode supervisée extrayant, pour chaque instrument, les régions de l'image dont l'intensité de mouvement est très corrélée avec le jeu de cet instrument. La première méthode est effectivement capable d'extraire des régions

correspondant aux différents instruments de la batterie, mais demande différents post-traitements pour reconnaître et regrouper ces composantes d'intérêt. La seconde méthode, déjà évaluée dans des travaux préliminaires [GR05a] produit des résultats satisfaisants.

Nous avons enfin proposé l'usage d'un algorithme classique d'estimation adaptative de l'arrière-plan pour effectuer la segmentation des baguettes et des mains du batteur. Nous n'avons pas réalisé d'évaluation de la segmentation produite. Cependant, les attributs utilisés au chapitre suivant pour la transcription audiovisuelle du jeu de l'instrument exploitent cette segmentation.

Nous concluons en soulignant quelques limites des méthodes utilisées dans ce chapitre. Tout d'abord, les méthodes proposées pour le traitement de séquences vidéo ne sont pas robustes aux changements d'angle de prise de vue (zoom, travelling), puisqu'elles exploitent la redondance ou la similarité entre trames successives de la séquence. Quelques pistes seront données en conclusion de ce manuscrit – faute de mieux, seul un traitement image par image avec les méthodes détaillées en 7.1 et un appariement des régions extraites trame à trame peut être envisagé. La robustesse des méthodes présentées dépend également de la stabilité de l'éclairage, puisque nous avons vu que toutes les méthodes présentées gagnent à utiliser un modèle de couleur des éléments de la batterie. Seul le critère géométrique introduit en 7.1.4 peut être utilisé dans n'importe quelles conditions d'éclairage mais ses performances se dégradent sans moyen simple de reconnaître les ellipses pertinentes.

Nous soulignons également qu'à l'exception de la méthode supervisée présentée en 7.2.3, les méthodes de segmentation introduites dans ce chapitre reconnaissent, mais n'identifient pas, les éléments de la batterie dans une scène. D'autres méthodes devront donc être utilisées pour étiqueter chaque région extraite par le nom de l'instrument qui lui correspond – étape nécessaire pour la transcription audiovisuelle du jeu de l'instrument.

Publications liées à ce chapitre

Le module de détection d'ellipses présenté dans ce chapitre est décrit dans [MGOR07].

CHAPITRE 8

Transcription audiovisuelle de séquences de batterie

Nous présentons dans ce chapitre un système de transcription de séquences audiovisuelles de jeu de batterie. La première section est consacrée à la détection des frappes dans des régions d'intérêt, à partir de la modalité vidéo seule, sous les aspects suivants : extraction de paramètres à partir d'une segmentation de l'image, et détection des frappes à partir des paramètres. Nous abordons ensuite dans la section 8.2 le problème de la fusion du résultat de cette détection avec le produit d'un système de transcription audio. Cette fusion exige d'abord la résolution du problème suivant : Si le système d'analyse vidéo est capable de détecter les frappes dans chaque région d'intérêt, il est incapable d'identifier à quel instrument est associée chacune des régions. Nous proposons un critère de couleur et un critère de compatibilité audiovisuelle permettant l'identification des instruments dans la scène. Les performances du système résultant sont évaluées sur des séquences de la base ENST-drums. Avant de conclure, nous présentons dans la section 8.3 des variantes de notre système de transcription audiovisuelle adaptées à divers scénarios d'usage : ces variantes tirent avantage d'un opérateur humain ou d'une transcription de référence et/ou s'adaptent à des conditions de prise de vue mal contrôlées.

8.1 Détection des frappes dans une séquence vidéo

La détection des frappes est effectuée en calculant différents paramètres (présentés en 8.1.1) à partir de la segmentation de l'image, telle qu'elle a été réalisée au chapitre précédent ; puis en recherchant des pics, dont la forme se rapproche d'un modèle donné, dans les fonctions décrivant l'évolution de ces paramètres au cours du temps (section 8.1.2).

8.1.1 Calcul des paramètres

Nous supposons ici que la séquence à traiter a été au préalable segmentée, produisant :
- Un ensemble de régions $\mathcal{R}_i$, chaque région correspondant à un instrument de la batterie.
- Pour chaque pixel de chaque trame, une mesure de l'appartenance de ce pixel à l'avant-plan, $p_F(x, y, m) \in [0, 1]$.

Deux groupes d'attributs sont calculés :

Mouvement des instruments Lorsqu'un instrument de la batterie est frappé, il est mis en mouvement. En conséquence, une frappe sur un instrument de la batterie se traduit toujours par une variation de la quantité de mouvement dans la région de l'image lui correspondant. Nous utilisons à cet effet une variante de la mesure de quantité de mouvement décrite dans la section 7.2.2.

Tout d'abord pour chaque pixel (x, y), la suite $L(x, y, m)$ des luminosités des pixels $\mathbf{I}(x, y, m)$ est filtrée par un filtre dérivateur de longueur égale à 5, produisant la suite $\Delta(x, y, m)$. Une mesure d'intensité de mouvement seuillée est fournie comme précédemment par :

$$\mathcal{A}(x, y, m) = \begin{cases} 0 & \text{si } |\Delta(x, y, m)| < \tau \\ |\Delta(x, y, m)| & \text{sinon} \end{cases} \tag{8.1}$$

Enfin, $\mathcal{A}(x, y, m)$ est lissée spatialement par convolution par un noyau gaussien de paramètre $\sigma = 3$, produisant une mesure d'intensité de mouvement $\mathcal{A}'(x, y, m)$ qu'on peut intégrer sur chacune des régions d'intérêt :

$$M_i(m) = \sum_{(x,y)\in\mathcal{R}_i} \mathcal{A}'(x, y, m) \tag{8.2}$$

Intersection des baguettes et des régions d'intérêt Le jeu d'un instrument de la batterie se traduit toujours par l'intersection de la région associée à une baguette et de la région associée à l'instrument. Il est important de noter cependant que la réciproque peut être fausse - en cas d'occlusion, les deux régions peuvent s'intersecter sans que cela corresponde nécessairement à une frappe. Nous pouvons ainsi mesurer, pour chaque région d'intérêt, la fraction de pixels appartenant à la baguette qu'elle contient :

$$B_i(m) = \sum_{(x,y)\in\mathcal{R}_i} p_F(x, y, m) \tag{8.3}$$

8.1.2 Détection

Les fonctions $M_i(m)$ et $B_i(m)$ définies précédemment possèdent des pics aux instants où l'instrument associé à la région d'intérêt i est frappé (voir les exemples dans la figure 8.1). Dans le cas de $M_i(m)$, ce pic est dû à la contribution (très localisée dans le temps) du mouvement de la baguette dans la région d'intérêt, suivi d'une composante décroissante correspondant au mouvement de l'instrument autour de sa position centrale. Cette composante peut être modélisée par une exponentielle décroissante de constante de temps élevée pour les cymbales, qui disposent d'une plus grande liberté de mouvement ; et de constante de temps courte pour les autres éléments. Dans le cas de $B_i(m)$, le pic est de forme triangulaire, et est très localisé dans le temps – il correspond à l'entrée puis à la sortie de la baguette dans la région. Nous suggérons les modèles suivants pour ces pics : $r_B(m)$ pour les pics dans $B_i(m)$, $r_{MC}(m)$ pour les pics dans $M_i(m)$ quand la région $\mathcal{R}_i$ est associée à une cymbale, et $r_{MF}(m)$ quand la région $\mathcal{R}_i$ est associée à un autre instrument.

$$r_B(m) = \begin{cases} 0 & |m| \geq 3 \\ 1 - \frac{|m|}{3} & |m| < 3 \end{cases} \tag{8.4}$$

$$r_{MF}(m) = \begin{cases} 1 - \frac{|m|}{3} & -3 < m < 0 \\ e^{-\frac{m}{3}} & m \geq 0 \end{cases} \tag{8.5}$$

$$r_{MC}(m) = \begin{cases} 1 - \frac{|m|}{3} & -3 < m < 0 \\ e^{-\frac{m}{15}} & m \geq 0 \end{cases} \tag{8.6}$$

Ces modèles sont illustrés dans la figure 8.2. Nous proposons de modéliser les fonctions $B_i(m)$ sous la forme suivante :

$$B_i(m) = w(m) + \sum_{k=1}^{K} a_k r_B(m - t_k) \tag{8.7}$$

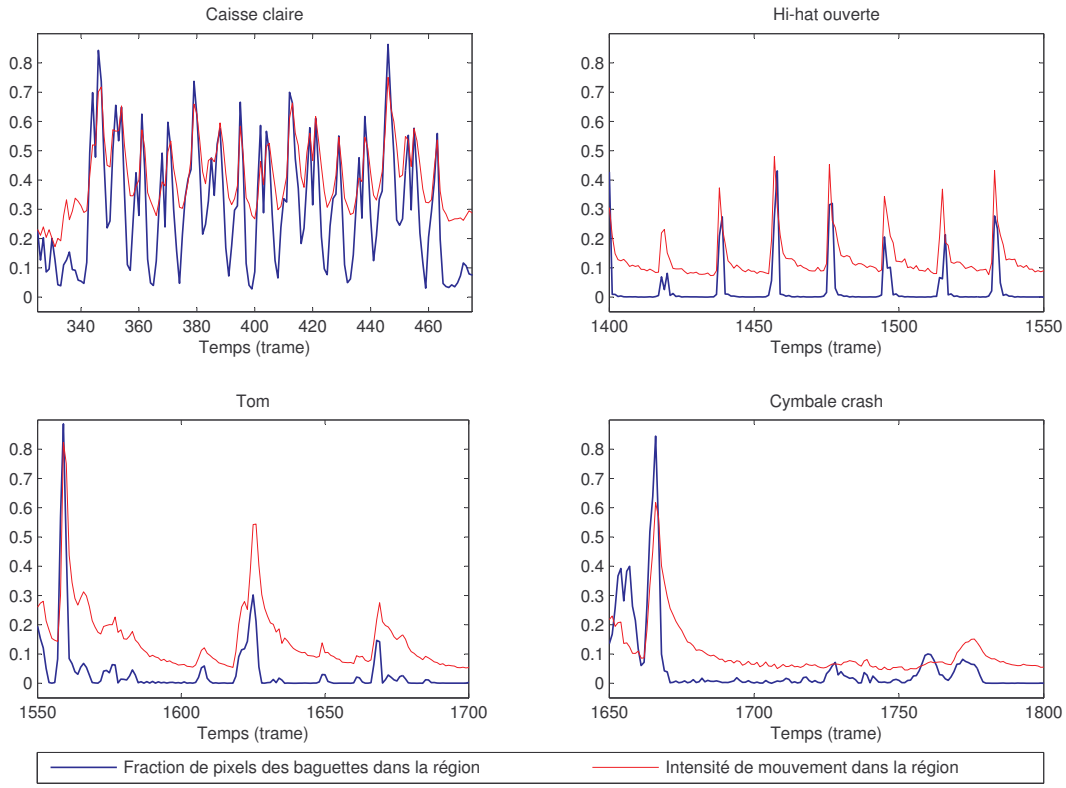


FIG. 8.1 – Exemples de paramètres extraits

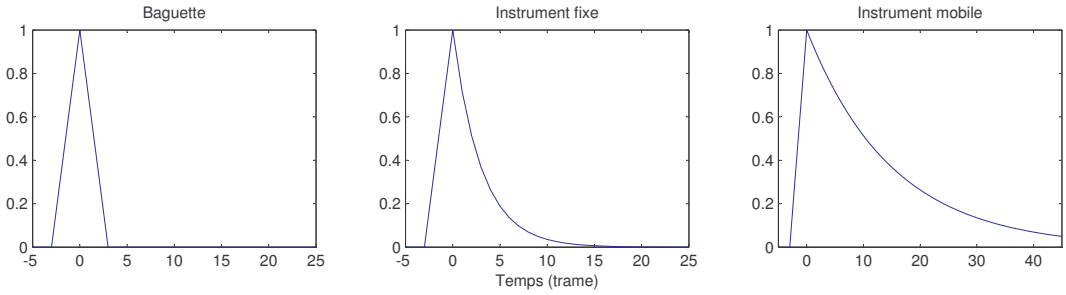


FIG. 8.2 – Modèles de pics $r_B(m)$, $r_{MF}(m)$ et $r_{MC}(m)$

Où K est le nombre de frappes de l'instrument i considéré au cours de la séquence, t_k les instants (exprimés en trames) où elles sont jouées, et a_k un facteur d'intensité. $w(m)$ représente le bruit dans la fonction $B_i(m)$ qui peut être dû aux mouvements des baguettes passant dans la région d'intérêt sans la heurter, ou aux erreurs de segmentation – régions non attribuées aux baguettes et apparaissant à l'intérieur d'une des régions d'intérêt, comme par exemple le buste du batteur. Ce bruit est modélisé comme la réalisation d'un bruit blanc gaussien dont la moyenne $\mu_{B_i}(m)$ et la variance $\sigma^2_{B_i}(m)$ varient lentement dans le temps. On suppose que la contribution de ces artefacts de segmentation est moindre par rapport aux mouvements qu'on souhaite réellement détecter. Ainsi, on peut supposer que $a_k \gg \sigma_{B_i}(m)$. La détection des pics $r_B(m)$ dans ce signal se fait alors en deux étapes :

Estimation des paramètres du bruit Nous considérons à cet effet une fenêtre longue de 251 trames, centrée en m, $W(m) = [B_i(m-125)\dots B_i(m)\dots B_i(m+125)]$. Les estimateurs classiques de la moyenne et de la variance ne peuvent pas être utilisés ici, puisque $W(m)$ peut contenir des valeurs extrêmes dues à la présence de pics. Des estimations plus robustes de $\mu_{B_i}(m)$ et de $\sigma^2_{B_i}(m)$ peuvent être obtenues en utilisant respectivement la médiane de $W(m)$; et la variance tronquée (estimation classique de la variance après rejet du premier et du dernier décile). On considère alors la fonction :

$$B'_i(m) = \frac{B_i(m) - \hat{\mu}_{B_i}(m)}{\hat{\sigma}_{B_i}(m)} \tag{8.8}$$

Selon l'hypothèse de variation lente de $\mu_{B_i}(m)$ et $\sigma_{B_i}(m)$, $B'_i(m)$ peut également s'écrire sous la forme :

$$B'_i(m) = w'(m) + \sum_{k=1}^{K} a'_k r_B(m - t_k) \tag{8.9}$$

Où $w'(m)$ est cette fois ci une réalisation d'un bruit blanc gaussien centré de variance unitaire.

Détection des pics La détection des pics peut être effectuée en filtrant $B'_i(m)$ par un filtre adapté (corrélateur) de réponse impulsionnelle $\alpha r_B(-m)$, où α est une constante de normalisation d'énergie $\alpha = \left(\sum_{m=-\infty}^{\infty} r_B^2(m)\right)^{-1}$. Soit $B''_i(m)$ le résultat de ce filtrage. Nous pouvons alors calculer la probabilité qu'un échantillon observé puisse être attribué au bruit :

$$\bar{p}_{B_i}(m) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-B''_i(m)^2}{2}\right) \tag{8.10}$$

Des développements similaires fournissent une fonction de détection $\bar{p}_{M_i}(m)$ mesurant la probabilité qu'à l'instant m la valeur observée de $M_i(m)$ puisse être expliquée par le bruit. Dans ce cas, le filtre adapté utilise r_{MC} si à la région $\mathcal{R}_i$ correspond une cymbale, r_{MF} sinon.

Une frappe sur un instrument est caractérisée à la fois par un mouvement de la baguette dans $\mathcal{R}_i$, et l'augmentation de la quantité de mouvement dans cette même région. Cette règle conjonctive peut être exprimée par le produit des probabilités que les paramètres observés à la trame m ne puissent pas être expliqués par le bruit :

$$p_i(m) = (1 - \bar{p}_{B_i}(m))(1 - \bar{p}_{M_i}(m)) \tag{8.11}$$

Cette probabilité peut être comparée à un seuil de décision pour produire un ensemble d'instants (ou d'indices de trames) H_i^{video} auxquels l'instrument associé à la région i est joué.

Classifieur	Angle de vue	Erreurs (%)
SVM	1	**30**
SVM	2	**15**
AdaBoost+C4.5	1	35
AdaBoost+C4.5	2	25

TAB. 8.1 – Classification cymbales/fûts par critère de couleur

8.2 Transcription audiovisuelle par fusion tardive

8.2.1 Prélude à la fusion : Association automatique des régions aux classes d'instruments

Avant de combiner les résultats de la détection d'événements effectuée sur le flux vidéo, avec ceux d'une transcription audio, il est nécessaire d'identifier à quel instrument de la batterie (cymbale, caisse claire, tom) correspond chaque région $\mathcal{R}_i$. En effet, à l'exception de la méthode de segmentation supervisée décrite en 7.2.3, les algorithmes décrits au chapitre précédent segmentent les régions contenant des instruments de la batterie, mais sont incapables d'identifier l'instrument qu'elles contiennent.

Soit $\{\mathcal{I}_1 \dots \mathcal{I}_{N_I}\}$ l'ensemble des N_I instruments de la batterie utilisés. Cet ensemble utilise une nomenclature détaillée des éléments de la batterie : en particulier, il inclut les différentes tailles de toms (tom alto, medium, basse, basse 2), et les différents types de cymbales (ride, splash, crash, chinoise). La tâche d'identification des instruments consiste à trouver une injection φ de l'ensemble des régions $\mathcal{R}$ vers l'ensemble des instruments $\mathcal{I}$.

Cette identification est rendue difficile par différentes situations rencontrées dans notre base d'évaluation : l'existence de séquences jouées par un batteur gaucher empêche l'utilisation d'heuristiques basées sur la position des éléments dans la scène par rapport au batteur, tandis que la présence de rythmes afro ou salsa joués essentiellement sur les toms plutôt que la caisse claire met en difficulté les heuristiques utilisant la fréquence des frappes. Nous utilisons donc deux critères plus robustes pour l'identification des instruments associés aux régions.

8.2.1.1 Critère de couleur pour la discrimination cymbales / fûts

Les instruments $\mathcal{I}_j$ peuvent être classés en deux grandes catégories : les fûts (toms et caisse claire), et les cymbales (hi-hat, ride, crash, splash). Soit $\mathcal{C}_\mathcal{I}(\mathcal{I}_j)$ la catégorie associée à l'instrument $\mathcal{I}_j$.

Les cymbales, qui sont principalement faites d'un alliage de cuivre, peuvent être identifiées par leur couleur. À cet effet, nous extrayons de chaque région segmentée $\mathcal{R}_i$ un histogramme à 16 classes des valeurs de teinte, saturation, et luminosité, produisant un vecteur d'attributs $\mathbf{x}_i$ de taille 48. Deux méthodes de classification ont été comparées pour la discrimination des deux classes considérées : agrégation de dix arbres de décisions (l'apprentissage des poids et des arbres est dirigé par l'algorithme AdaBoost), et C-SVM avec noyau gaussien (voir annexe B), avec pour paramètres $C = 5$ et $\sigma = 1$. Les résultats obtenus par le protocole *leave one out* sont donnés dans la table 8.1 – nous utiliserons par la suite les SVM donnant de meilleures performances.

Soit $\mathcal{C}_\mathcal{R}(\mathcal{R}_i)$ la catégorie associée à la région $\mathcal{R}_i$ par la classification automatique. Une matrice de compatibilité peut alors être définie entre les régions et les instruments selon :

$$C_{i,j}^{coul} = \delta_{\mathcal{C}_\mathcal{I}(\mathcal{I}_j)}^{\mathcal{C}_\mathcal{R}(\mathcal{R}_i)} \tag{8.12}$$

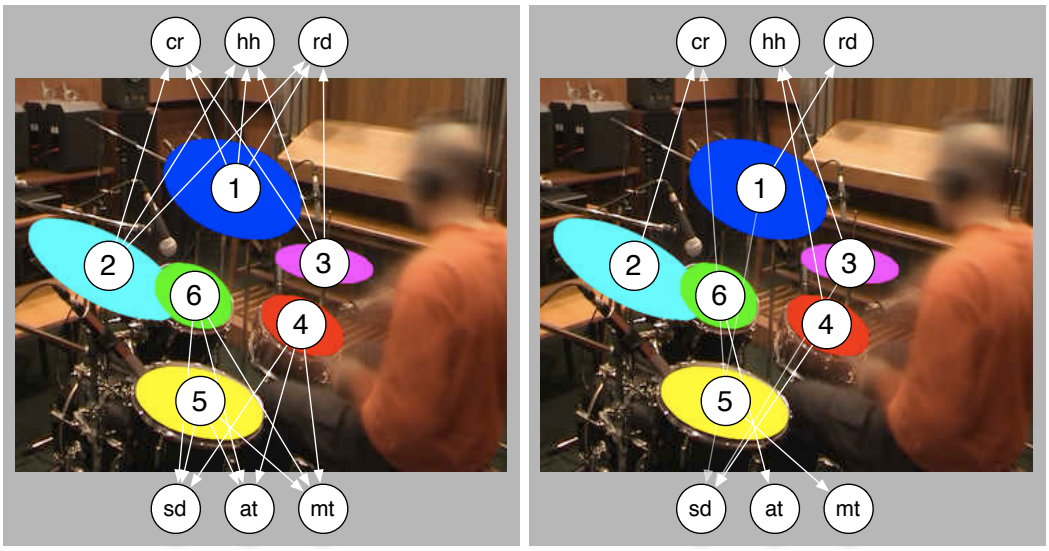


FIG. 8.3 – Compatibilité région/instrument selon des critères de couleur et de consistance avec la transcription audio

Un exemple est donné dans la figure 8.3 (à gauche). La matrice de compatibilité correspondante est :

Instrument $\mathcal{I}_j$	Region $\mathcal{R}_i$ / $\mathcal{C}_\mathcal{R}(\mathcal{R}_i)$ / $\mathcal{C}_\mathcal{I}(\mathcal{I}_j)$	$\mathcal{R}_1$ Cym.	$\mathcal{R}_2$ Cym.	$\mathcal{R}_3$ Cym.	$\mathcal{R}_4$ Fût	$\mathcal{R}_5$ Fût	$\mathcal{R}_6$ Fût
Caisse claire	Fût	0	0	0	1	1	1
Hi-hat	Cym.	1	1	1	0	0	0
Cymbale crash	Cym.	1	1	1	0	0	0
Cymbale ride	Cym.	1	1	1	0	0	0
Tom alto	Fût	0	0	0	1	1	1
Tom medium	Fût	0	0	0	1	1	1

Dans la pratique, ce critère de couleur est calculé avant la procédure de détection des frappes de la batterie présentée dans la section 8.1.2. Ce critère permet ainsi de choisir les modèles de pics r_{MF} (pour les fûts) ou r_{MC} (pour les cymbales) les plus pertinents pour la détection.

8.2.1.2 Critère de compatibilité des transcriptions extraites

Le système de transcription audio présenté au chapitre 4 produit pour chaque instrument $\mathcal{I}_j$ un ensemble H_j^{audio}, contenant les instants auxquels une frappe sur cet instrument a été détectée. Si la région $\mathcal{R}_i$ est associée à l'instrument $\mathcal{I}_j$, les transcriptions audio H_j^{audio} et vidéo H_i^{video} doivent être consistantes, et contenir des éléments communs l'un à l'autre (elles ne sont cependant pas identiques, car des frappes détectées à partir du signal audio ne le sont pas toujours sur la vidéo, et inversement – justifiant l'intérêt de la fusion). De manière à mesurer cette consistance, nous proposons le critère suivant :

$$C_{i,j}^{compat} = \frac{|H_i^{video} \cap H_j^{audio}|}{\sqrt{|H_i^{video}|}\sqrt{|H_j^{audio}|}} \tag{8.13}$$

Ce critère peut être soit vu comme le nombre de co-occurences, normalisé par la moyenne géométrique du nombre d'événements détectés à partir de chaque modalité considérée, ou comme une approximation du coefficient de corrélation de Pearson calculé sur des versions seuillées des fonctions de détection – valant 1 si une frappe est détectée à la trame m, et 0 sinon.

Soulignons que les classifieurs audio et vidéo ont des résolutions temporelles différentes, et qu'un événement peut être détecté avec un léger décalage entre une modalité et l'autre. Pour permettre la mise en correspondance des événements, les durées dans H_i^{video} et H_j^{audio} sont quantifiées uniformément avec un pas de 100 ms.

8.2.1.3 Association région/instrument optimale

Soit $C_{i,j}$ un critère de compatibilité entre régions et instruments, construit par exemple à partir des critères $C_{i,j}^{compat}$ et $C_{i,j}^{coul}$. L'association région/instrument optimale φ^* est celle maximisant le score de compatibilité totale, c'est à dire :

$$\varphi^* = \underset{\varphi}{\operatorname{argmax}} \sum_i C_{i,\varphi(i)} \tag{8.14}$$

Résolution par couplage de graphe Ce problème peut être reformulé comme la recherche d'un couplage de poids maximal dans un graphe biparti. Soit $G(V, E)$ un graphe biparti dont les sommets sont les régions et instruments $V = (\bigcup_i \mathcal{R}_i) \cup (\bigcup_j \mathcal{I}_j)$, dont les arêtes connectent tous les sommets $E = (\bigcup_i \mathcal{R}_i) \times (\bigcup_j \mathcal{I}_j)$, avec un poids $w(e) = C_{i,j}$ si $e = \{\mathcal{R}_i, \mathcal{I}_j\}$. L'algorithme de Kuhn-Munkres [Kuh55] permet de résoudre ce problème avec une complexité en $\mathcal{O}(N^3)$, où $N = \max\{N_I, N_R\}$.

On pose $A = \mathcal{I}, B = \mathcal{R}$, et on suppose, quitte à inverser le rôle de A et B, que $|A| \leq |B|$. Soit $l : (A \cup B) \mapsto \mathbb{R}$, vérifiant $\forall a \in A, \forall b \in B, l(a) + l(b) \geq w((a, b))$. Pour l donné, G_l est le sous-graphe de G contenant tous les sommets V et les arrêtes $E_l = \{(a, b) \in A \times B, l(a) + l(b) = w((a, b))\}$. Soit $adjacents_G(X)$ l'ensemble des sommets adjacents aux sommets de X, dans G. L'algorithme de Kuhn-Munkres est donné dans l'algorithme 5, sans détailler la recherche du chemin alternant (que nous effectuons par une recherche en profondeur).

Critère à maximiser Nos premières expériences utilisaient le critère $C_{i,j} = C_{i,j}^{compat} C_{i,j}^{coul}$ pour la recherche de l'association optimale. Ce critère correspond à une procédure d'association hiérarchique dans laquelle sont construites en parallèle une association optimale entre le groupe de régions classées comme cymbales selon le critère de couleur, et les cymbales ; et une association entre les régions classées comme fûts et les fûts. Cependant, le critère de couleur seul n'étant pas toujours fiable (voir résultats dans la table 8.1), cette méthode de combinaison n'est pas optimale. Le critère choisi doit en effet être capable d'associer une région et un instrument très compatibles selon le critère C^{compat}, même s'ils sont incompatibles selon C^{coul} – comme ce peut être le cas lors d'une erreur de classification de la région par le critère de couleur. Notre choix s'est tourné vers le critère $C = \frac{1}{2}C^{compat} + \frac{1}{2}C^{coul}$.

8.2.2 Fusion des transcriptions audio et vidéo

Notons tout d'abord qu'à l'issue de cette étape d'association des instruments aux régions, nous pouvons recalculer les fonctions $p_i(m)$ en choisissant pour chaque région le modèle de pics adapté à l'instrument qui lui est associé. Nous disposons ainsi pour chaque instrument $\mathcal{I}_j$ d'une fonction $p_j^1(m) = p_{\varphi^{-1}(j)}(m)$ indiquant la probabilité que cet instrument soit joué à la trame m. Cette opération peut être répétée pour chaque capteur vidéo.

Si N_s flux sont considérés (incluant flux vidéos et flux audio), une transcription peut finalement être obtenue en agrégeant les probabilités $p_j^1(m) \ldots p_j^{N_s}(m)$. En supposant l'information apportée par chaque flux fiable et complémentaire, une règle de disjonction peut être utilisée pour la fusion :

Algorithme 5 : Algorithme de Kuhn-Munkres

entrées : A, B, w
$l(a) \leftarrow \max_{b \in B} w((a,b)), \forall a \in A$
$l(b) \leftarrow 0, \forall b \in B$
$M \leftarrow \emptyset$
// Tant qu'il existe des paires non formées
tant que $\exists a \in A, \forall b \in B, (a,b) \notin M$ **faire**
  $S \leftarrow \{a\}$
  $T \leftarrow \emptyset$
  étendre ← Vrai
  // Cherche un chemin augmentant et l'étend tant que possible
  tant que *étendre* **faire**
    tant que $adjacents_{G_l}(S) = T$ **faire**
      $\alpha \leftarrow \min_{a \in S, b \in B \setminus T} l(a) + l(b) - w((a,b))$
      $l(a) \leftarrow l(a) - \alpha, \forall a \in S$
      $l(b) \leftarrow l(b) + \alpha, \forall b \in T$
    fin
    $z \leftarrow$ élément choisi dans adjacents$_{G_l}(S) \setminus T$
    si $\exists y \in A, (y,z) \in M$ **alors**
      $S \leftarrow S \cup \{y\}$
      $T \leftarrow T \cup \{z\}$
    sinon
      étendre ← Faux
    fin
  fin
  // Il existe un chemin entre a et z alternant entre $M \cap E_l$ et M
  $C \leftarrow recherche_chemin_alternant(a, z, M, E_l)$
  $M \leftarrow \{(a,b), (x,y) \in C$ et $x \in A\}$
fin
sorties : M

$$p_j(m) = 1 - \prod_{s=1}^{N_s} (1 - p_j^s(m)) \tag{8.15}$$

8.2.3 Résultats expérimentaux

Les expériences ont été menées sur 51 séquences vidéo de la base ENST-drums – 17 séquences jouées par 3 batteurs, déjà utilisées pour l'évaluation de la transcription audio en 4.6.1.1. Les signaux audio incluent un accompagnement instrumental, mixé de façon équilibrée avec les signaux de batterie.

Chaque séquence est enregistrée par deux caméras, sous deux angles de vue : derrière le batteur, à sa gauche ; et en hauteur, face à la batterie. Contrairement aux expériences effectuées en transcription audio, la taxonomie utilisée ici est plus détaillée, puisqu'elle inclut les toms et les cymbales, qui sont annotés différemment selon leur taille et type (tom alto, medium, basse ; cymbale crash, ride, chinoise, splash). Cette information de taille et type est particulièrement difficile à extraire par les classifieurs audio.

Le protocole retenu est le suivant, pour chaque séquence à transcrire :

1. Le signal audio est transcrit par le système de transcription décrit au chapitre 4 (parmi les variantes proposées, nous utilisons celle basée sur la fusion tardive[1]). Une méthode semblable

[1]Les variantes utilisant des modèles de séquence ne sont pas considérées. En effet, dans le cas d'un système de trans-

	Batteur 1		Batteur 2		Batteur 3	
Angle de vue	1	2	1	2	1	2
% Régions correctement identifiées	64.3	76.5	60.0	73.3	72.5	64.3

TAB. 8.2 – Identification des instruments à partir des régions : performances

à celle utilisée au chapitre 4 pour la détection des frappes de caisse claire, de grosse caisse et de hi-hat est utilisée pour chaque catégorie de toms et de cymbales : sélection d'attributs, et classification par SVM sur le signal original et le signal dont la batterie a été accentuée.

2. Pour chaque angle de vue, une image d'arrière plan de la séquence est calculée. Cette image est utilisée pour la segmentation des instruments, en utilisant la méthode de détection d'ellipses présentée en 7.1.4. Les classifieurs intervenant dans la définition du critère de couleur ont été appris sur des images n'incluant pas la batterie et l'angle de vue de la séquence considérée. L'extraction de l'avant-plan de chaque trame est également réalisé.
3. Chaque région extraite est classée selon les catégories fûts/cymbales (comme précédemment, les classifieurs utilisés ont été entraînés sur des données étrangères).
4. Pour chaque angle de vue, les paramètres visuels sont extraits, permettant la détection des frappes à l'aide du modèle de pic adapté à la catégorie fût/cymbale décidée précédemment. L'association optimale entre régions et instruments peut alors être effectuée, en utilisant le critère de couleur et de compatibilité avec la transcription audio. Soulignons que nous excluons de l'ensemble $\mathcal{I}$ la grosse caisse, qui n'est que partiellement visible dans les séquences utilisant l'angle 1, et hors champ dans les séquences utilisant l'angle 2 – les résultats de la transcription vidéo sont donc toujours nuls pour cet instrument.
5. La détection des frappes est effectuée une seconde fois, en utilisant le modèle de pic adapté à l'instrument finalement associé à la région.
6. Les sorties des classifieurs audio et vidéo sont agrégées, par règle disjonctive.
7. La transcription obtenue est comparée avec la transcription de référence. Les mesures utilisées sont le rappel, la précision et la F-mesure, comme définis en 4.6.1.3. Une tolérance de deux trames (80 ms) entre la position réelle et la position détectée est autorisée[2]. Pour les toms et cymbales, les résultats donnés sont la moyenne des différentes sous-catégories (tom alto, medium, basse...).

Nous donnons d'abord dans la table 8.2 les performances pour la tâche d'identification des instruments de la batterie à partir des régions. Les résultats de transcription sont donnés dans la moitié supérieure de la table 8.3.

Nous observons tout d'abord que pour la caisse claire et la hi-hat, les performances de la transcription vidéo (sans fusion avec l'audio) sont en deçà de celles obtenues par l'analyse du signal audio seul. La rapidité du jeu sur ces instruments peut justifier ces mauvaises performances. Par contre, pour la détection des toms et cymbales, les meilleurs résultats sont obtenus par les détecteurs vidéo : l'information détaillée sur le type de cymbale ou la hauteur du tom est plus facilement extractible à partir de la vidéo.

Pour tous les instruments, la combinaison des deux capteurs vidéo fournit des résultats supérieurs à ceux obtenus en utilisant un seul capteur : la prise de vue multi-caméra est donc un moyen efficace de lutter contre l'occlusion. Par contre, à l'exception de la caisse claire dont la transcription la plus précise est obtenue en combinant audio et vidéo capturée par l'angle 1, les autres instruments sont mieux transcris par des systèmes de transcription unimodaux (hi-hat et grosse-caisse par

cription audiovisuel, nous suggérons l'application du modèle de séquence en fin de chaîne, c'est à dire après la fusion des transcriptions audio et vidéo.

[2]Nos expériences avec la modalité audio seule utilisaient une tolérance de 50 ms. Cela explique les scores légérement supérieurs du système de transcription audio par rapport aux résultats donnés dans la table 4.10.

Segmentation automatique																	
Modalité			Grosse caisse			Caisse claire			Toms			Hi-hat			Cymbales		
Audio	Vidéo 1	Vidéo 2	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$
Transcription unimodale																	
●	○	○	**70.5**	**68.1**	**69.3**	64.1	61.8	62.9	5.3	11.7	7.3	**89.5**	**69.8**	**78.4**	15.8	17.8	16.8
○	●	○	0.0	0.0	0.0	71.0	37.4	49.0	73.7	6.4	11.8	49.1	47.4	48.3	96.0	17.3	29.4
○	○	●	0.0	0.0	0.0	49.3	28.9	36.5	92.3	11.7	20.8	66.7	59.5	62.9	85.4	17.5	29.1
○	●	●	0.0	0.0	0.0	66.9	40.9	50.8	**92.1**	**12.5**	**22.0**	71.1	61.8	66.1	**87.5**	**18.3**	**30.3**
Transcription multimodale																	
●	●	○	**70.5**	**68.1**	**69.3**	**69.4**	**68.0**	**68.7**	84.2	8.7	15.7	83.4	71.3	76.9	45.6	19.3	27.1
●	○	●	**70.5**	**68.1**	**69.3**	76.2	58.4	66.1	93.7	9.9	17.9	84.4	70.8	77.0	45.9	19.1	26.9
●	●	●	**70.5**	**68.1**	**69.3**	77.1	61.0	68.1	95.3	9.8	17.8	83.3	72.1	77.3	48.6	20.7	29.0

Segmentation manuelle																	
Modalité			Grosse caisse			Caisse claire			Toms			Hi-hat			Cymbales		
Audio	Vidéo 1	Vidéo 2	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$	$R\%$	$P\%$	$F\%$
Transcription unimodale																	
●	○	○	**70.5**	**68.1**	**69.3**	64.1	61.8	62.9	5.3	11.7	7.3	89.5	69.8	78.4	15.8	17.8	16.8
○	●	○	40.6	39.2	39.9	68.6	42.4	52.4	63.3	6.2	11.3	61.1	64.6	62.8	89.1	16.1	27.3
○	○	●	0.0	0.0	0.0	57.1	37.5	45.3	67.3	7.7	13.8	69.0	65.8	67.4	**86.5**	**16.8**	**28.2**
○	●	●	40.6	39.2	39.9	76.7	40.8	53.2	76.1	7.8	14.1	74.2	66.4	70.1	93.7	16.5	28.0
Transcription multimodale																	
●	●	○	68.1	64.7	66.3	**82.1**	**63.5**	**71.6**	67.0	5.9	10.9	81.3	70.3	75.4	39.4	16.4	23.2
●	○	●	**70.5**	**68.1**	**69.3**	69.7	67.7	68.7	68.2	10.3	17.9	**85.8**	**73.7**	**79.3**	44.5	18.8	26.4
●	●	●	68.1	64.7	66.3	77.3	65.3	70.8	**64.0**	**12.0**	**20.3**	95.6	63.7	76.5	69.5	17.6	28.1

TAB. 8.3 – Rappel R, Précision P et F-mesure F pour la transcription audiovisuelle de la batterie avec accompagnement

système audio, toms et cymbales par système vidéo à deux capteurs). Cela suggère donc une fusion de type “meilleur expert”, dans lequel chaque instrument est transcrit à partir de la modalité la mieux adaptée.

8.3 Autres stratégies pour la transcription musicale audiovisuelle

Nous introduisons et comparons à présent diverses variantes du système présenté et évalué dans ce chapitre.

8.3.1 Variations sur la segmentation

8.3.1.1 Intervention d’un opérateur humain

Nous avons privilégié jusqu’ici les approches entièrement automatiques, envisageant les applications d’indexation. Pour les applications d’interaction musicien/machine ou d’aide à l’apprentissage, il est possible de requérir l’intervention de l’utilisateur pour la calibration du système. Trois niveaux d’implication peuvent être définis :

Validation de la segmentation, et association des régions aux instruments L’utilisateur désigne sur une image les ellipses correctes parmi celles extraites automatiquement. Dans de telles approches de segmentation supervisée par un utilisateur humain, les coûts associés à une fausse acceptation et un faux rejet sont asymétriques : dans le premier cas, l’utilisateur doit juste désigner une ellipse incorrecte, tandis que dans le second cas, il doit dessiner l’ellipse manquante. Il peut donc s’avérer plus efficace d’assouplir les critères définis en 7.1.4 pour le filtrage des ellipses. L’utilisateur désigne ensuite l’instrument associé à chaque région.

Segmentation manuelle L’utilisateur désigne successivement, pour chaque instrument, la région de l’image associée à l’instrument considéré – en la peignant sur une image de référence. L’intérêt d’une segmentation entièrement manuelle est qu’elle ne contraint pas la forme de la région d’intérêt. L’utilisateur peut par exemple inclure non pas seulement le sommet, mais aussi le corps (le fût) de l’instrument, de manière à disposer d’un critère de mouvement plus robuste : une frappe est détectée quand le corps de l’instrument est mis en mouvement.

Nous avons évalué cette approche de segmentation manuelle, avec le système de détection présenté dans ce chapitre[3]. Les résultats sont donnés dans la deuxième partie de la table 8.3. Notons tout d’abord que la grosse caisse est partiellement visible sur les séquences filmées depuis l’angle 1 – cela permet donc sa transcription à partir de la modalité vidéo seule. Cependant, la transcription est bien moins robuste qu’à partir de la modalité audio – causant également une dégradation des performances en fusion audio/vidéo. En dehors du cas de la grosse caisse, les très bonnes performances offertes par la fusion par règle disjonctive suggèrent la complémentarité des informations extraites par les détecteurs audio et vidéo. L’amélioration de la qualité de la segmentation bénéficie le plus à la caisse claire et la hi-hat, dont les scores de transcription vidéo sont meilleurs. En conséquence, pour ces instruments, les meilleures performances sont obtenues par fusion (et non plus à partir de l’audio seul). Le cas des toms et des cymbales est surprenant : les performances obtenues avec le procédé de segmentation automatique sont meilleures qu’avec une segmentation manuelle. Cette situation peut s’expliquer par le fait que des ellipses invalides, ou mal ajustées aux bords de l’instrument peuvent améliorer la détection du mouvement de l’instrument ou la présence d’une baguette sur l’instrument.

[3]Notre but initial étant principalement d’évaluer comment les erreurs introduites aux étapes de segmentation et d’association des instruments aux régions se cumulent et contribuent à dégrader les performances.

FIG. 8.4 – Segmentation manuelle détaillée

Cela suggère l'utilisation de régions étendues incluant la baguette ou les avant-bras du batteur lors de la frappe.

Segmentation détaillée L'utilisateur désigne, à travers une interface similaire à celle utilisée précédemment deux régions par instrument : une région correspondant à la surface de l'instrument, et une région autour de l'instrument où est susceptible d'être détecté un mouvement lors du jeu de l'instrument. Par exemple, pour une cymbale, cette région inclut le voisinage de la cymbale par où arrive la baguette, et le voisinage du poignet du batteur dans la posture qu'il adopte pour frapper la cymbale. Un exemple de segmentation (montré dans l'interface utilisée pour la réaliser) est donné dans la figure 8.4. Un attribut supplémentaire mesurant la quantité de mouvement dans cette région est ainsi disponible, et peut être utilisé de la même façon que les deux autres attributs définis en 8.1.1. La détection d'une frappe exige alors les trois conditions suivantes : intersection de la baguette dans la région, mouvement dans la région, et mouvement dans la région périphérique peu avant la frappe.

Segmentation par le jeu d'une séquence de référence Dans ce cas, l'utilisateur doit jouer une séquence de référence permettant la calibration. Il peut soit s'agir d'une phrase dont la partition est connue, ou d'une séquence où chaque instrument est isolément joué – phrase pour laquelle on peut supposer que la classification audio est parfaite. La segmentation s'effectue alors par la méthode décrite en 7.2.3.

8.3.2 Variations sur le procédé de reconnaissance

Classifieurs supervisés locaux Nous avons justifié en 6.3 notre motivation à former des attributs simples permettant une transcription par détection des pics : L'emploi d'un classifieur supervisé n'est pas possible, puisque les attributs extraits dépendent de la séquence considérée et de la configuration de la batterie utilisée – il est donc impossible d'apprendre un modèle "universel" du jeu de la batterie.

Cependant, si la séquence à traiter est suffisamment longue, et que nous disposons d'une transcription de référence d'une de ses parties, nous pouvons apprendre un classifieur local, entraîné sur, et pour, la batterie utilisée dans la séquence.

Une telle approche a déjà été utilisée dans le cas de la transcription audio par Sandvold et al. dans [SGH04], ou dans [GR05c], afin de disposer d'un système de transcription spécialisé pour la batterie à transcrire.

Dans le cadre de la transcription vidéo, l'intérêt d'un tel classifieur est multiple. Tout d'abord, il rend inutile la tâche d'association des régions aux instruments – lors de la phase d'apprentissage le classifieur associé à chacun des instruments identifiera le poids optimal des attributs calculés sur chacune des régions ; cette étape peut en outre être facilitée par des méthodes de sélection d'attributs, telles celles présentées en 4.4.2. Par ailleurs, nous nous sommes restreints jusqu'ici à des choix d'attributs facilitant la détection de frappes par recherche de pics. Les classifieurs pouvant implémenter des règles de décision plus complexes qu'un simple seuil (ou conjonction de seuils), d'autres attributs peuvent être extraits et considérés, par exemple les moyennes, variances, et moments d'ordre supérieur des coordonnées des points considérés comme formant l'avant-plan.

Deux démarches sont possibles pour utiliser des classifieurs supervisés :

- Une pré-segmentation temporelle de la séquence par détection de pics dans les attributs de mouvement (par analogie avec la détection d'onsets), suivie du calcul d'un unique vecteur d'attributs par segment. Dans ce cas, les attributs calculés peuvent être intégrés sur différentes plages temporelles comme réalisé dans [GR05a] : les attributs liés au mouvement de l'instrument sont intégrés sur toute la longueur du segment, tandis que ceux liés au mouvement des baguettes sont intégrés sur un voisinage du début du segment. La reconnaissance s'effectue alors par classification supervisée de ces vecteurs d'attributs. Les développements relatifs au choix d'une taxonomie, d'une approche discriminative vs explicative, des attributs et des paramètres des classifieurs présentés au chapitre 4 s'appliquent sans modification à ce problème. Notons que lorsque cette approche est suivie, l'intégration des informations audio et vidéo peut s'effectuer de façon précoce, en entraînant le classifieur local sur des vecteurs d'attributs incluant à la fois des paramètres audio et vidéo. Dans les expériences réalisées en [GR05a] (reproduit dans l'annexe C), c'est cette méthode de fusion qui a donné les résultats les plus satisfaisants, par rapport à la fusion d'un classifieur local vidéo et d'un classifieur audio universel (entraîné sur une base diverse).
- Une segmentation/reconnaissance simultanée par l'emploi de modèles temporels (HMM par exemple). Un nouvel avantage des classifieurs locaux apparaît alors : ils permettent l'apprentissage d'un modèle de l'évolution temporelle des attributs propre à la batterie considérée. Par contraste, le système de détection présenté en 8.1 utilise des modèles temporels définis a priori.

Soulignons toutefois quelques unes des limites de cette approche :

- La séquence de référence utilisée pour l'apprentissage du classifieur local doit être suffisamment longue pour permettre l'apprentissage. Des modèles explicatifs comme les GMM ou les HMM possèdent de nombreux paramètres, et requièrent donc un volume de données d'apprentissage considérable. Par opposition, les approches discriminatives (en particulier les méthodes à noyaux) sont plus robustes lorsque les observations sont peu nombreuses[4].
- Ces méthodes exigent que la distribution d'un attribut (conditionnellement au jeu/non-jeu d'un instrument) soit constante au cours du temps. C'est le cas uniquement lorsque les conditions de prise de vue sont stables, ce qui exclut l'usage de telles méthodes sur des séquences dont l'éclairage ou l'angle de prise de vue varient.

Reconnaissance itérative Disposer d'une transcription de référence facilite à la fois la segmentation et l'association des régions aux instruments, et permet l'apprentissage et l'utilisation de classifieurs locaux comme vu précédemment. Comment faire lorsqu'une telle transcription n'est pas disponible ? Nous suggérons l'emploi d'un processus de reconnaissance itérative, dans lequel une première transcription est obtenue entièrement automatiquement, soit en utilisant un classifieur audio

[4]Ce problème d'apprentissage à partir d'un ensemble de données très réduit est à nouveau rencontré au chapitre 10.

seul (si le signal audio est de bonne qualité, et si l'accompagnement musical n'est pas prédominant), ou un classifieur audiovisuel utilisant une segmentation et une calibration automatique. Cette transcription peut alors être considérée comme référence pour la segmentation, la calibration, ou l'apprentissage d'un classifieur local. Ce procédé peut être itéré, en utilisant la transcription produite à l'étape précédente comme référence pour l'identification des régions. Ce processus est similaire aux approches utilisées en transcription audio, convergeant itérativement vers une transcription et un modèle d'instruments, l'un optimisé par rapport à l'autre.

8.3.3 Quelles solutions choisir ?

Nous résumons dans la table 8.4 et dans cette section nos discussions relatives aux conditions d'utilisation des méthodes décrites dans ce chapitre et au chapitre précédent.

Les contraintes relatives à l'utilisation des classifieurs locaux ont déjà été présentées : une transcription de référence doit être disponible, et les attributs calculés doivent avoir une interprétation constante au long de la séquence.

Dans les situations où l'angle de vue ne varie pas au cours du temps, l'utilisation d'une segmentation par recherche des régions maximisant l'information mutuelle avec la référence est souhaitable dès qu'une référence est disponible. Par ailleurs, la présence d'un opérateur humain ou la disponibilité d'une transcription de référence permet de simplifier la tâche d'association des régions aux instruments. Cette association peut également être faite implicitement par le classifieur local, ou par l'étape de sélection d'attributs qui a précédé son apprentissage.

Considérons maintenant le cas des séquences où l'angle de prise de vue varie continûment au cours du temps. Dans le cas où une transcription entièrement automatique est souhaitée, la segmentation doit être effectuée trame à trame par une méthode automatique (détection d'ellipses), et les régions extraites doivent être appariées. Une approche concurrente consisterait à utiliser des contours actifs (*snakes*) suivant la région. Dans le cas où un opérateur humain est présent, une telle segmentation peut être manuellement initialisée, et suivie trame à trame. La segmentation des baguettes ne peut plus se faire par segmentation arrière-plan fixe/avant-plan en mouvement, puisqu'ici l'arrière-plan apparaît en mouvement. Si les mouvements de caméra sont lents, on peut envisager une compensation du mouvement par mise en correspondance des images successives.

Quoi qu'il en soit, l'interprétation différente qu'auront les attributs au cours du temps exclut l'usage de classifieurs locaux – la détection des frappes devra se faire par recherche des pics. Reste à définir quelle stratégie adopter pour l'association des régions aux instruments. En absence d'une séquence de référence, la recherche du couplage maximal sur critères de compatibilité avec l'audio (et la couleur, si l'éclairage est stable) doit être envisagée. Si une transcription de référence est disponible, la compatibilité avec cette référence, plutôt qu'avec la transcription audio peut être considérée. Notons que même dans le cas où l'angle de prise de vue change, une interface adéquate peut permettre à un opérateur humain d'annoter les régions avec l'instrument qui leur est associé.

8.4 Conclusion

Nous avons présenté dans ce chapitre une méthode de détection des frappes de batterie à partir d'une segmentation de la séquence en régions (chaque région est associée à un instrument), et en arrière-plan/avant-plan : Des paramètres mesurant l'intensité de mouvement dans chaque région, et le degré d'intersection de la baguette et de la région sont formés, la détection est ensuite effectuée en recherchant des pics dans les fonctions qu'ils définissent. Nous avons par la suite traité le problème de la fusion de cette analyse vidéo avec le produit d'une transcription audio. Avant toute fusion, il est nécessaire d'identifier quel instrument de la batterie est associé à chacune des régions. Nous avons à cet effet proposé deux critères : un critère de couleur, utilisant une SVM pour discriminer les instruments selon leur apparence, et un critère de compatibilité avec la transcription audio. Ces deux critères sont combinés, et définissent un graphe dont un couplage maximal fournit une association optimale des instruments aux régions. La fusion entre l'audio et la vidéo est alors possible, par

Référence ?	Éclairage variable ?	Intervention humaine ?	Mouvements de caméra ?	Segmentation des instruments	Analyse vidéo	Association régions/instruments
○	○	○	○	Ellipses + couleur sur modèle du fond	Détection	Couplage maximal, compatibilité avec l'audio et la couleur
●	○	○	○	Inf. mutuelle avec la référence	Détection	Selon référence
				Inf. mutuelle avec la référence	Classifieur local	Sélection d'attributs
○	●	○	○	Ellipses, sur moyenne des trames	Détection	Couplage maximal, compatibilité avec l'audio
●	●	○	○	Inf. mutuelle avec la référence	Détection	Selon référence
○	○	●	○	Manuelle	Détection	Par opérateur humain
●	○	●	○	Par référence ou manuelle	Détection	Selon référence ou opérateur humain
				Par référence ou manuelle	Classifieur local	Sélection d'attributs
○	●	●	○	Manuelle	Détection	Selon opérateur humain
●	●	●	○	Par référence ou manuelle	Détection	Selon référence ou opérateur humain
○	○	○	●	Ellipses + couleur, suivi de région	Détection	Couplage maximal, compatibilité avec l'audio et la couleur
●	○	○	●	Ellipses + couleur, suivi de région	Détection	Couplage maximal, compatibilité avec la référence et la couleur
○	●	○	●	Ellipses, suivi de région	Détection	Couplage maximal, compatibilité avec l'audio
●	●	○	●	Ellipses, suivi de région	Détection	Couplage maximal, compatibilité avec la référence
○	○	●	●	Manuelle, suivi de région	Détection	Opérateur humain
●	○	●	●	Manuelle, suivi de région	Détection	Couplage maximal, compatibilité avec la référence et la couleur ; ou opérateur humain
○	●	●	●	Manuelle, suivi de région	Détection	Opérateur humain
●	●	●	●	Manuelle, suivi de région	Détection	Couplage maximal, compatibilité avec la référence ; ou opérateur humain

TAB. 8.4 – Choix recommandé de méthodes de segmentation, de détection de frappes et d'association régions/instruments, selon le scénario d'utilisation

l'application d'une règle disjonctive – qui suppose que chaque modalité fournit une information fiable et complémentaire.

L'évaluation est effectuée sur un ensemble de séquences tirées de la base ENST-drums, pour différentes combinaisons de modalités. Pour la plupart des instruments, les meilleures performances sont obtenues avec des classifieurs unimodaux. En particulier, le jeu des toms et cymbales, pour lesquels une taxonomie détaillée a été utilisée, est plus efficacement transcrit à partir de la modalité vidéo. L'apport de la fusion n'est significatif que pour la caisse claire. Outre la difficulté inhérente à la tâche de détection vidéo des frappes, une partie des erreurs commises par le système s'explique par les erreurs de segmentation, et les erreurs d'association régions/instruments. De manière à évaluer la contribution de ces erreurs, les expériences ont été répétées en utilisant une segmentation manuelle des régions. Dans ce cas, les performances optimales sont obtenues par fusion. Une découverte surprenante est que pour certains instruments (toms et cymbales), une segmentation automatique imparfaite conduit à de meilleurs résultats qu'une segmentation manuelle.

Nous avons enfin discuté quelques variantes possibles de notre système, utilisant d'autres méthodes de segmentation ou de classification. En particulier, la disponibilité d'une séquence de référence ou l'intervention d'un opérateur humain facilitent les tâches de segmentation et d'association instruments/régions. Dans le cas où une séquence de référence est disponible, l'apprentissage de classifieurs locaux peut être envisagée, permettant l'emploi de méthodes d'apprentissage statistiques éprouvées. Nous avons également présenté une méthode itérative de transcription, dans laquelle une première transcription (audio ou audiovisuelle) est utilisée comme référence pour la segmentation ou l'apprentissage. Si nous n'avons pu, faute de temps, évaluer cette méthode sur la base ENST-drums, nous avons évalué sa pertinence dans une étude préliminaire publiée dans [GR05a]. Pour résumer nos discussions sur la robustesse de chacune des méthodes évoquées à différentes situations d'usage, nous avons suggéré un choix de méthodes adaptées à chaque scénario d'utilisation, qui peut servir de cadre à des développements et évaluations expérimentales futures.

Publications liées à ce chapitre

Les méthodes de détection et fusion introduites dans ce chapitre, ainsi que les résultats des expériences réalisées, ont été publiés dans [MGOR07]. Notre étude préliminaire du problème de la transcription audiovisuelle de séquences vidéo de jeu de batterie, utilisant une approche basée sur l'apprentissage supervisé de modèles locaux, a également fait l'objet d'un article [GR05a].

Conclusion de la partie II

Le problème de la transcription automatique de scènes musicales audiovisuelles est atypique, et peu traité dans la littérature. Nos propositions de solutions, pour une application concrète, constituent donc l'une des contributions originales de cette thèse. Malgré les similarités apparentes entre la transcription musicale audiovisuelle et les problèmes de la reconnaissance des gestes et postures ou le traitement audiovisuel de la parole, les solutions proposées à ces problèmes ne s'appliquent que peu ou mal à la transcription musicale audiovisuelle. Parmi les raisons expliquant cet échec, nous avons souligné en particulier l'impossibilité de formuler des modèles universels des gestes et des paramètres extraits de la séquence vidéo – ces paramètres et modèles dépendant de l'angle de prise de vue et de la configuration de l'instrument. Cette asymétrie entre le problème de transcription audio – pour lequel un modèle universel du timbre de chaque instrument peut être construit – et du problème d'analyse vidéo – dépendant de la scène – suggère l'emploi de la fusion tardive, la seule à même de combiner des classifieurs de nature et portée différentes.

Nous avons ainsi retenu le système de classification audio supervisé/universel présenté au chapitre 4, et choisi de fusionner ses sorties avec un système de détection non-supervisé/local utilisant la modalité vidéo.

L'impossibilité d'utiliser des méthodes d'apprentissage statistique nous a conduit à construire des attributs véhiculant une information de haut-niveau, modélisant deux connaissances a priori sur le jeu de l'instrument : un instrument est mis en mouvement lorsqu'il est joué, et la baguette le heurte au moment du jeu. À cet effet, des méthodes de segmentation des instruments ont dû être développées. Nous avons retenu deux critères complémentaires pour la segmentation : un critère de couleur, et un critère géométrique, utilisant une méthode originale de détection d'ellipses dans une image. Le critère de couleur n'est pas robuste aux changements de conditions d'éclairage, mais il rend plus robuste la détection d'ellipses en permettant de rejeter des régions incorrectes. Différentes approches ont été proposées pour la fusion d'image en vue de la segmentation, ou la fusion des segmentations (solution rejetée car trop coûteuse). Des méthodes supervisées et non-supervisées utilisant des attributs d'intensité de mouvement ont également été proposées, bien que leur évaluation objective n'ait pas été réalisée. La segmentation des baguettes et des mains du batteur a été effectuée par une méthode simple, utilisant une segmentation adaptative avant-plan en mouvement/arrière-plan. Notons que cette méthode est peu robuste dans les situations où d'autres musiciens sont en mouvement sur la scène, et ne permet pas la segmentation de scènes où la caméra est en mouvement. Dans une telle situation, un réel suivi de la position des baguettes doit être effectué, opération qui apparaît comme très difficile.

Le processus de détection des frappes est grandement facilité par le fait que les attributs extraits sont de haut niveau : il consiste en une détection des pics par filtrage adapté, utilisant des modèles de pics propres à chaque catégorie d'instrument. La difficulté principale rencontrée dans la mise en oeuvre d'une approche entièrement automatique et non-supervisée est l'identification des instruments associés aux régions. La solution originale proposée consiste à formuler ce problème comme un problème de couplage maximal dans un graphe, sur divers critères de compatibilité. La fusion réalisée est ainsi celle maximisant la compatibilité entre les informations présentes dans les flux.

Les résultats expérimentaux suggèrent que pour certaines tâches, la transcription vidéo ou multimodale est plus robuste que la transcription audio, même si les gains de performances restent modestes.

Dans cette partie, l'accent a été mis sur le traitement non-supervisé, entièrement automatique, de scènes musicales. En conséquence, les méthodes proposées n'ont pas toujours été les plus ro-

bustes à des situations adverses, comme le changement de conditions d'éclairage ou de prise de vue. Différentes variantes du système évalué ont été proposées pour s'adapter à ces conditions adverses et/ou pour tirer parti d'informations supplémentaires, fournies par un opérateur humain ou une transcription de référence d'un fragment de la séquence. Ces variantes n'ont cependant pas été évaluées, et fournissent juste un plan de travail pour une série d'évaluations futures.

Ces considérations sur la robustesse du système et son usage automatique/semi-automatique nous conduisent à la situation paradoxale suivante : Les applications où les conditions de prise de vue sont les plus contrôlées (usage en interaction musicien/machine) sont celles où l'intervention d'un opérateur humain est possible ; tandis que les applications qui requièrent un traitement entièrement automatique (indexation de vidéos de concert par exemple) sont celles pour lesquelles les conditions de prise de vue sont les plus variables. La méthode proposée dans cette partie est ainsi presque trop générique pour les applications d'interaction musicien/machine (pour lesquelles une approche semi-automatique peut suffire), et pas encore assez robuste pour traiter des documents audiovisuels musicaux commerciaux. Cet échec relatif motive la dernière partie de cette thèse.

Troisième partie

Vers l'analyse des documents audiovisuels musicaux

CHAPITRE 9

Problématique

Nous avons présenté dans la partie précédente un système d'analyse audiovisuel du jeu de la batterie. Si un tel système peut être utilisé dans des applications d'interaction musicien/machine, ou d'apprentissage assisté par ordinateur, les différentes contraintes que nous avons formulées quant aux conditions de prise de vue ne permettent pas son utilisation sur n'importe quel document audiovisuel musical – une retransmission télévisée d'un concert, ou un clip vidéo par exemple. Faut-il alors en conclure que les quelques applications évoquées ci-dessus sont le seul domaine où analyse de scènes vidéo et transcription musicale peuvent se rejoindre ? Nous allons proposer, dans cette partie, d'autres applications se trouvant à l'intersection de ces deux domaines, et présenter un système capable de traiter une large gamme de documents audiovisuels musicaux.

La problématique sera cependant différente : dans la partie précédente, nous utilisons à la fois l'information vidéo et audio pour effectuer une transcription précise de la partie audio. Dans cette section, nous cherchons à combiner les modalités audio et vidéo, pour extraire un nouveau type de description du document audiovisuel (relatif, par exemple, à son genre), ou pour permettre de nouvelles applications (recherche d'une séquence vidéo accompagnant une oeuvre musicale). Autrement dit, nous nous intéressons dans cette partie à des méthodes pouvant traiter des documents audiovisuels musicaux bien plus diversifiés qu'au chapitre précédent, mais qui en extraient une information de plus haut niveau, moins détaillée (bien que d'intérêt).

Ainsi, les problèmes traités dans cette partie et la partie précédente ne s'excluent pas. En particulier, les méthodes d'analyse du contenu présentées dans cette partie peuvent permettre de découvrir, dans une base de données de documents audiovisuels musicaux, quels documents se prêtent particulièrement bien aux méthodes de transcription audiovisuelle décrites précédemment.

Un bref état de l'art des systèmes d'analyse automatique du contenu des documents audiovisuels musicaux est donné dans la section 9.1 – nous ne présentons ici que les sytèmes prenant réellement en compte leur dimension musicale. Dans la section 9.2, nous décrivons en détail la problématique de cette troisième partie, en montrant sa spécificité par rapport aux autres approches proposées dans la littérature.

9.1 État de l'art

Soulignons tout d'abord que de nombreux systèmes d'indexation de documents audiovisuels combinant les modalités audio et vidéo, ou découvrant des associations entre ces modalités ont été développés pour des tâches aussi diverses que l'identification des scènes d'interviews dans les journaux télévisés [ATD02], la découverte d'association entre mots-clés et concepts audiovisuels [XKC+04] ou la recherche de célébrités dans des documents audiovisuels [IVWF06]. Un nombre encore plus important de systèmes d'indexation de documents audiovisuels se concentrent sur la modalité la plus pertinente pour la tâche à accomplir : par exemple vidéo ou audio pour la segmentation en programmes et la classification du contenu, texte pour la reconnaissance de mots-clés.

Si de tels systèmes peuvent être utilisés pour indexer des documents audiovisuels musicaux, ils n'en exploitent pas les spécificités et n'en extraient pas une description adaptée à leur nature. Nous ne dresserons pas ici d'état de l'art de ce domaine trop large, mais nous nous restreindrons plutôt à ses applications spécifiques aux documents audiovisuels musicaux.

9.1.1 Analyse automatique de clips vidéos

Différents systèmes ont été proposés pour l'analyse des clips vidéos, plus spécifiquement pour en extraire des résumés. Les méthodes classiques de génération de résumés exploitent principalement la modalité vidéo, par exemple en effectuant un découpage de la séquence en plans, et en extrayant les plans les moins redondants entre eux selon une mesure de similarité visuelle (voir par exemple [HYM02] pour l'évaluation de telles mesures). Cependant, pour résumer un clip vidéo, les modalités audio et textuelles (transcription des paroles) doivent aussi être prises en compte.

Ainsi, Agnihotri et al. décrivent dans [ADKZ03; ADK04] un système de résumé de clips vidéos guidé par l'analyse des informations textuelles affichées à l'écran. Une segmentation du document en plans est d'abord effectuée, utilisant comme attributs la sortie d'un détecteur de visage, un histogramme de couleurs, et un détecteur de texte. Elle permet non seulement le découpage d'un long document audiovisuel en les différents clips vidéos ou programmes qui le composent, mais elle autorise aussi, à un niveau de structuration plus fin, la sélection d'images clés montrant l'artiste (si disponible), et d'une image où apparaît le titre et le nom de l'artiste – comme affiché au début et à la fin de la vidéo par la plupart des chaînes. La transcription automatique des paroles affichées à l'écran, et le clustering des phrases obtenues permettent la détection du refrain (correspondant aux paroles les plus souvent répétées) et l'extraction du segment audio correspondant. Les sorties du détecteur de refrain, de visage et de paroles répétées sont intégrées dans un réseau Bayésien permettant de calculer la probabilité qu'un segment de vidéo donné présente un intérêt. Les segments les plus intéressants sont sélectionnés pour constituer un résumé vidéo. Les autres informations extraites (refrain audio, images clés, titre et artiste) peuvent être présentées dans une interface facilitant la navigation dans une base de données de clips vidéos. Notons que cette approche est inapplicable aux clips vidéos de musique instrumentale (techno par exemple), pour lesquels aucune parole n'est disponible.

Shao et al. présentent dans [SXK03] un système de résumé n'exploitant que des informations audiovisuelles (et non une transcription des paroles). Le contenu audio est segmenté par un algorithme de clustering permettant l'extraction du refrain et des couplets. La séquence vidéo est segmentée en plans, dont sont extraits des images clés. Le clustering des images clés permet d'extraire un ensemble E de plans non-redondants. Le résumé final est obtenu en jouant une séquence de 7 extraits audio, accompagnés de séquences vidéos tirées de E. Un effort particulier est fait pour s'assurer que pour chaque extrait audio, la séquence vidéo choisie sera similaire à celle accompagnant originellement l'extrait audio. Notons que ce traitement distinct de l'audio et de la vidéo suppose que le contenu vidéo est indépendant de la musique – une propriété vraie uniquement pour une classe limitée de clips vidéos. Ce système est étendu dans [SXK04] pour inclure une analyse des paroles affichées à l'écran aidant à l'identification du refrain, comme proposé par Agnihotri et al.

9.1.2 Illustration sonore ou visuelle automatique

Une autre tâche liant l'analyse musicale et l'analyse de séquences vidéos fréquemment étudiée dans la littérature est la requête ou synthèse d'extraits musicaux par la vidéo à des fins d'illustration sonore, ou le montage de séquences vidéo guidé par une séquence musicale.

Dans [FCG02], Foote et al. décrivent un système de montage de vidéos familiales guidé par la musique. La structure d'une oeuvre musicale est extraite par analyse de sa matrice d'auto-similarité. La séquence vidéo est segmentée, et ses plans sont choisis selon un critère de qualité (mesure de l'exposition et des mouvements intempestifs de caméra), pour ensuite être associés à chaque segment audio. Soulignons que les plans sont choisis dans l'ordre chronologique sur le seul critère de qualité, et que rien n'est fait pour s'assurer de leur synchronie ou de leur compatibilité avec la musique.

Dans [MKYH03], Mulhem et al. proposent un système d'aide à l'illustration musicale de séquences vidéo se basant cette fois-ci sur des règles d'associations entre propriétés visuelles et caractéristiques musicales. Ces règles sont tirées du traité d'esthétique audiovisuelle de Zettl [Zet98], et relient par exemple les changements de tonalité aux changements de plans, la quantité de mouvement à l'énergie sonore ou la tonalité à la teinte de l'éclairage. Elles permettent de définir un espace dit pivot dont chaque dimension représente le concept audiovisuel intervenant dans chacune de ces règles d'association. Des vecteurs d'attributs audio et vidéo peuvent être projetés sur cet espace. Une mesure de similarité entre contenu vidéo et audio est alors définie par la distance des projetés des attributs extraits dans l'espace pivot. Cette méthode est utilisée pour sélectionner un extrait sonore accompagnant une séquence vidéo donnée. La relation entre tempo et intensité de mouvement a également été utilisée pour la même application par Yang et Brown dans [YB04]. Notons que les approches évoquées ici correspondent à une problématique commune : mesurer par un score la compatibilité entre flux vidéo et audio. Dans un tout autre contexte – celui de la détection d'attaques dans les systèmes d'identification biométrique audiovisuelle, des mesures de synchronie entre signal de parole et vidéo des lèvres sont données par Bredin et Chollet dans [BC07].

Une approche différente décrite par Nayak et al. dans [NSK03], consiste à utiliser les règles d'esthétique audiovisuelles mentionnées précédemment pour directement synthétiser une musique à partir des attributs extraits de la vidéo (plutôt que de rechercher cette musique au sein d'une base de données de signaux existants).

Précisons, pour conclure cette section, que les effets vidéo synchrones à la musique étant très prisés, quelques logiciels commerciaux existent déjà en dehors du domaine de la recherche. Abaltat Beat [Aba07] facilite la composition de musique à l'image en générant une grille rythmique dont les pulsations coïncident le mieux avec les changements de plan d'une séquence vidéo. Aucune analyse vidéo n'est effectuée, les documents vidéos utilisés en entrée devant être accompagnés d'une *edit decision list* spécifiant leur structure. En ce qui concerne l'illustration d'une séquence musicale, le logiciel de compositage et d'effets spéciaux Apple Motion 3 [App07] est capable de lier une propriété d'un objet graphique à la valeur d'un descripteur extrait du signal audio (énergie dans des bandes de fréquences, fréquence fondamentale, centroïde spectral ou fonction de détection d'onsets).

9.2 Approche proposée

9.2.1 Principe

Notons tout d'abord que les systèmes de résumé de clips vidéo, s'ils peuvent faciliter l'exploration d'une collection de documents par un utilisateur humain, ne permettent pas l'extraction de descriptions supplémentaires permettant la classification ou le regroupement des documents. Quelle pourrait être cette information supplémentaire ? Nous suggérons qu'une information pertinente à extraire d'un clip vidéo pourrait être son *genre visuel*. En effet, parmi les différents types de documents audiovisuels musicaux (retransmission de concerts, opéras ou spectacles musicaux, clips vidéos, émissions de variétés, danse), les clips vidéos sont les plus variés, en raison de la vaste palette de relations sémantiques associant la vidéo et la musique l'accompagnant. Si la plupart des clips vidéo commerciaux montrent des danseurs et des musiciens, d'autres ont une trame narrative (construite par exemple à partir des paroles de la chanson ou de propriétés de haut-niveau de la musique – ambiance, structure), tandis que les réalisateurs les plus créatifs comme Spike Jonze ou Michel Gondry [Jon03; Gon03; Div02] ont inventé de nouvelles formes de métaphores audiovisuelles. Cette caractéristique de *genre visuel*, qui offre un axe de description complémentaire au genre musical, dépend de la relation liant la séquence vidéo à son accompagnement musical : illustre-t-elle une activité synchrone à la musique (danse, jeu des musiciens) ? Accompagne-t-elle la structure de la musique (narration) ? Répondre à de telles questions nécessite la définition de mesures de synchronie entre divers niveaux de descriptions du contenu audio (notes, sections), et du contenu vidéo (plans, séquences, mouvements).

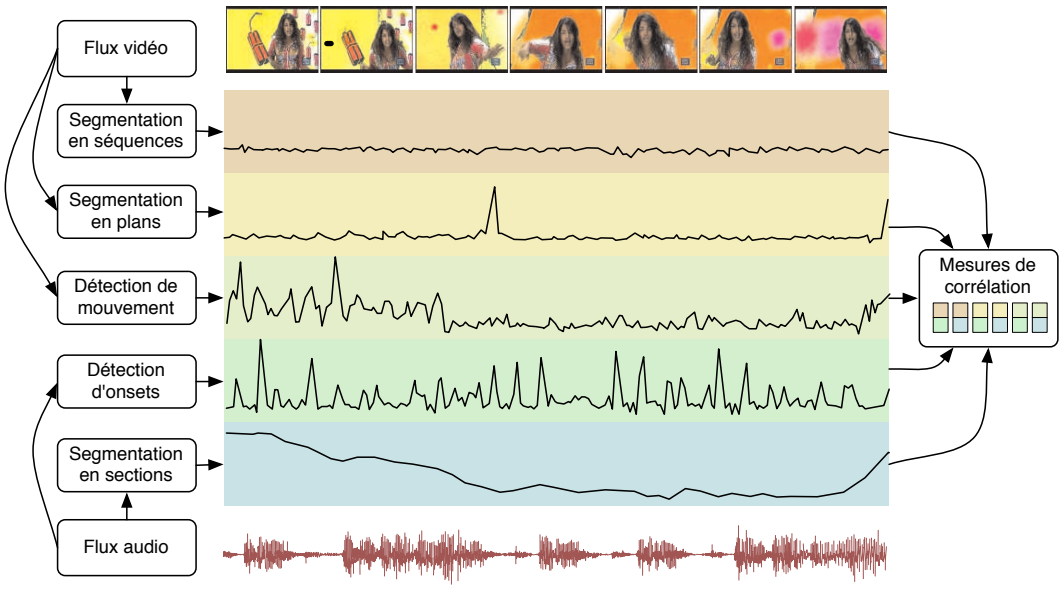


FIG. 9.1 – Structuration et analyse de synchronie dans les documents audiovisuels musicaux

En dehors de leur application à la caractérisation du *genre visuel* des clips vidéos, de telles mesures peuvent également être utilisées pour permettre des requêtes de modalités croisées (par exemple, recherche d'accompagnement musical illustrant une vidéo donnée). Nous précisons cependant qu'à l'inverse de certaines des méthodes présentées dans la section 9.1.2, les mesures de synchronie ne requièrent aucune connaissance a priori quant aux règles d'esthétique audiovisuelle liant des descripteurs vidéo à des descripteurs audio. En fait, une condition nécessaire pour que soit perçue une relation d'association entre un attribut vidéo (par exemple la luminosité), et un attribut audio (par exemple, la sonie), est que les changements brusques d'un attribut coïncident avec des changements brusques de l'autre [Lip05]. Cette condition de synchronie n'est certe pas suffisante, mais elle apparaît plus robuste et générale que l'utilisation de critères esthétiques. Elle permet en outre de révéler des associations à plusieurs niveaux sémantiques – changements de séquence, de scène, ou mouvements.

Nous nous proposons ainsi de définir, dans cette partie, des mesures de synchronie des changements observés dans les documents audiovisuels musicaux.

9.2.2 Architecture du système

L'architecture du système qui sera étudié dans cette partie est donnée dans la figure 9.1. Les contenus audio et vidéo sont tout d'abord analysés afin d'en extraire leur structure, à des degrés divers :

- Les événements les plus saillants dans les signaux de musique sont les changements de notes ou d'accords. Une segmentation de bas niveau d'une oeuvre musicale peut ainsi être obtenue par détection des onsets. À un niveau immédiatement supérieur, il est également possible d'extraire les pulsations rythmiques définissant le tempo.
- De façon similaire, à la granularité la plus fine, les événements les plus saillants dans une vidéo sont les changements brusques de mouvement (pas de danse, mouvements des musiciens pour jouer une note, mouvements dans une séquence d'action).

– À un plus haut niveau, une oeuvre musicale peut être segmentée en sections, caractérisées par des propriétés de dynamique, de tonalité ou de timbre différentes. De telles sections correspondent à la structure musicale de l'oeuvre, en termes de refrain, couplet, intro ou ponts.
– De façon similaire, à un haut niveau, une séquence vidéo peut être segmentée en plans, et ces plans peuvent être groupés en séquences.

Ces différents modules de segmentation seront présentés au chapitre 10, dans lequel sont en particulier introduites et évaluées différentes méthodes originales pour la segmentation en sections d'enregistrements musicaux. Tous les modules de segmentation produisent une fonction de détection, dont les pics matérialisent les changements à l'échelle considérée.

Des mesures de corrélation (ou plus précisément de synchronie des changements) peuvent alors être définies entre les flux audio et vidéo, pour chaque paire de niveaux de structuration, par exemple, synchronie entre les changements de plan et de section dans la musique, ou entre les mouvements et la pulsation rythmique. Ces mesures de corrélation seront présentées au chapitre 11. Nous démontrerons dans ce même chapitre leur intérêt pour diverses applications.

CHAPITRE **10**

Détection des changements dans les documents audiovisuels musicaux

Nous présentons dans ce chapitre les différents modules de segmentation utilisés dans notre système – nous nous intéressons à la fois à la segmentation du contenu audio et vidéo d'un document audiovisuel musical, et ce à plusieurs échelles. Le problème de la segmentation en sections d'un enregistrement musical est traité dans la section 10.1 – nous en présentons différentes solutions basées sur des méthodes à noyaux. À un plus bas niveau, la segmentation d'un signal de musique en notes peut être réalisée par un détecteur d'onsets classique. Nous avons déjà traité ce problème dans la section 4.2 et n'y reviendrons pas. Les approches retenues pour la structuration du flux vidéo en plans et séquences sont présentées dans la section 10.2. Enfin, la méthode choisie pour réaliser la segmentation à bas niveau d'une séquence vidéo est décrite dans la section 10.3. Elle consiste à détecter les variations d'une mesure de quantité de mouvement.

10.1 Détection des changements de section dans les signaux de musique

Nous nous intéressons dans cette section à la segmentation temporelle d'une oeuvre musicale en sections (refrain, couplet, intro, pont), chacune d'entre elle se distinguant des autres ou bien par sa tonalité, sa dynamique ou par des changements de timbre et d'instrumentation.

L'étape commune à toutes les méthodes de segmentation présentées dans la littérature consiste en l'extraction d'une suite de vecteurs de paramètres acoustiques à partir du signal à segmenter. Les attributs sont typiquement extraits sur des fenêtres longues de plusieurs centaines de millisecondes. Notons qu'il n'existe aucun consensus sur les attributs à extraire. Les traitements qui suivent sont également variés. Une approche courante dans la littérature, introduite par Foote [Foo99; CF02] consiste à construire à partir de la suite d'attributs une matrice d'auto-similarité. Les sections répétées se matérialisent alors par des blocs apparaissant au dessus de la diagonale. Différents critères pour grouper ou fusionner les sections détectées peuvent éventuellement être utilisés en post-traitement [PK06]. Une approche concurrente consiste à utiliser un algorithme de clustering incluant une contrainte temporelle (deux trames proches sont très susceptibles d'appartenir au même groupe), ou de façon équivalente, un HMM utilisé de façon non-supervisée [PBR02]. Chaque trame est ainsi associée à un groupe (ou à un état du HMM), définissant le segment auquel elle appartient.

Le point commun de ces deux approches est qu'elles cherchent à obtenir une segmentation en regroupant des trames ou des ensembles de trames similaires. Les méthodes que nous présentons dans cette section cherchent plutôt à directement identifier les frontières des sections, qui peuvent se caractériser de la façon suivante : les vecteurs d'attributs extraits du signal suivant le changement de section sont "nouveaux" relativement aux vecteurs d'attributs extraits du signal précédent le chan-

Catégorie	Notation	Dim.	Description
D	E_k^t	12	Énergie en sortie d'un b.d.f en demi-tons
D	$OBSIR_i$	7	Rapports d'énergie dans un b.d.f. en bandes d'octaves [ERD06b]
C	$\mu MFCC_k$	13	Moyenne des MFCC
S	S_{cntr}, S_{sprd}, S_{skew}, S_{kurt}	4	Moments spectraux [GR04]
T	ZCR	1	Taux de passage par zéro classique
T	T_{var}, T_{skew}, T_{kurt}	3	Moments de la forme d'onde
T	E_{mean}, E_{var}, E_{skew}, E_{kurt}	4	Moments de l'enveloppe d'amplitude
P	Ldr_i	24	Sonie spécifique relative [Pee04]
P	Acu	1	Acuité [Pee04; Zwi77]
P	Et	1	Étendue [Pee04]

TAB. 10.1 – Récapitualtif des 70 attributs utilisés pour la segmentation audio. Leur calcul est détaillé dans l'annexe A

gement de section. Avant d'exposer les outils statistiques permettant une telle mesure de nouveauté, nous présentons d'abord la paramétrisation du signal utilisée dans nos travaux.

10.1.1 Paramétrisation du signal

Nous considérons pour la segmentation un ensemble de 70 attributs candidats, parmi lesquels seront sélectionnés les attributs les plus efficaces. Ces attributs sont répertoriés dans la table 10.1. L'annexe A offre une définition détaillée de chacun de ces attributs.

Cet ensemble d'attributs candidats regroupe les attributs les plus utilisés dans la littérature relative à la segmentation d'oeuvres musicales (MFCC, banc de filtres en demi-tons), des attributs génériques (moments spectraux, rapports d'énergie entre octaves adjacentes, et leurs équivalents perceptuels, taux de passage par zéro), et des moments calculés dans le domaine temporel pour mesurer des propriétés rythmiques (impulsivité).

Les attributs sont extraits sur des fenêtres longues de 2 secondes. Cette taille, particulièrement longue, permet d'une part de compenser ou lisser les variations rapides et périodiques de certains attributs de timbre, et de disposer d'un horizon d'observation suffisamment long pour extraire des paramètres mesurant les propriétés rythmiques. De manière cependant à disposer de suffisamment d'observations, le taux de chevauchement entre fenêtres successives est de $\frac{1}{16}$; 8 vecteurs de paramètres sont ainsi extraits chaque seconde. On notera par la suite $\mathbf{x}(m)$ le vecteur d'attributs extrait pour la m-ième trame.

10.1.2 Sélection d'attributs pour la segmentation

Nous avons déjà introduit en 4.4.2 la problématique de la sélection d'attributs pour la classification, et les grandes familles de solutions proposées dans la littérature. Nous nous intéressons maintenant à l'utilisation de ces méthodes pour sélectionner les meilleurs attributs pour la tâche de segmentation :

Premièrement, les méthodes en boucle fermée (*wrapper*) peuvent être utilisées de la même façon, en utilisant comme mesure de performance d'un ensemble d'attributs non plus le taux de reconnaissance en sortie d'un classifieur, mais une des mesures de performance typiques utilisées en segmentation (précision, rappel, F-mesure). Notons que les risques de surapprentissage sont tout aussi grands.

Deuxièmement, les méthodes embarquées ou les filtres requièrent intrinsèquement d'être appliqués à des problèmes de classification, puisqu'elles exploitent la structure d'un classifieur, ou des mesures de pouvoir discriminant (critère de Fisher par exemple). La seule exception sont les méthodes de type filtres n'utilisant aucun critère de pouvoir discriminant, mais simplement un critère de non-redondance. Cependant, même si le problème de la détection de changements brusques dans les signaux de musique n'est pas en soi un problème de classification, il est possible de définir un critère de nature discriminative pour le choix des attributs : les attributs à extraire sont ceux qui permettront le mieux de discriminer les trames de deux sections distinctes, mais qui ne discrimineront pas des trames tirées d'une même section. En d'autres termes, nous pouvons considérer deux paires de segments adjacents dans une oeuvre musicale comme définissant deux classes à discriminer et choisir les attributs les plus discriminants pour ces deux classes.

Algorithme 6 : Sélection d'attributs localement discriminants et vote

entrées : $\mathbf{x}^i(m)$, $y^i(m)$, L_i, N
pour $n \in \{1, \ldots, F\}$ **faire**
  $v_n \leftarrow 0$
fin
pour $i \in \{1, \ldots, N\}$ **faire**
  pour $j \in \{2, \ldots, L_i\}$ **faire**
    $T \leftarrow \{(\mathbf{x}^i(m), -1), y^i(m) = j - 1\} \cup \{(\mathbf{x}^i(m), +1), y^i(m) = j\}$
    $S \leftarrow$ sélection d'attributs(T)
    pour tous les $n \in S$ **faire**
      $v_n \leftarrow v_n + 1$
    fin
  fin
fin
sorties : v

Cette formulation fait cependant apparaître une différence par rapport au problème classique de la sélection d'attributs : dans notre problème, les attributs à sélectionner seront les attributs les plus efficaces sur l'ensemble des paires de sections adjacentes – et chacun des problèmes de discrimination associés – tandis qu'en sélection d'attributs pour la classification, les attributs à sélectionner seront les plus efficaces sur un unique problème de classification. Nous proposons de résoudre cette difficulté par une procédure de vote. Le protocole utilisé pour la sélection d'attributs est ainsi décrit dans l'algorithme 6. Nous noterons $\mathbf{x}^i(m)$ les vecteurs d'attributs extraits du i-ème morceau de la base d'apprentissage (contenant au total N morceaux), $y^i(m)$ l'indice du numéro de section dans laquelle se situe la trame m au sein de ce morceau, L_i le nombre total de sections, et *sélection d'attributs* une procédure de sélection d'attributs pour les problèmes de classification supervisée, renvoyant les indices des attributs les plus efficaces (les attributs sont indicés de 1 à F).

N'importe quelle méthode de sélection d'attributs conçue pour la classification supervisée peut être utilisée dans cette procédure. Dans les expériences qui suivent, nous avons utilisé comme critère de sélection d'attributs le critère de Fisher donné dans l'équation 4.14 : les attributs sélectionnés sont ceux qui maximisent ce critère. Le très bon rapport performances / coût en calculs de ce critère a été souligné dans [ERD06b].

Le nombre d'attributs sélectionnés a été fixé à 32 par validation croisée dans les expériences de segmentation décrites par la suite. À des fins de validation, nous avons également effectué l'expérience suivante : la base de données de signaux de musique utilisée (décrite dans l'annexe D.2) a été divisée en deux sous-groupes contenant chacun la moitié des signaux. Pour chacun des deux groupes, la procédure de sélection d'attributs décrite dans l'algorithme 6 a été appliquée et les 32 attributs recevant le plus de votes ont été sélectionnés. Les attributs sélectionnés dans les deux groupes sont les mêmes, bien que leur ordre diffère. Cela suggère que cet ensemble d'attributs pertinents pour la segmentation est stable, et que la phase de sélection d'attributs peut être effectuée une fois pour toutes, et non de façon adaptative pour chacun des signaux à traiter.

Les attributs sélectionnés sont donnés dans la table 10.2. Un des critères les plus importants

Groupe d'attributs	Sélectionnés	Sélection
Filtres en demi-tons	0 / 12	
OBSIR	7 / 7	$OBSIR_5$, $OBSIR_4$, $OBSIR_7$, $OBSIR_6$, $OBSIR_3$, $OBSIR_2$, $OBSIR_1$
Moments spectraux	4 / 4	S_{sprd}, S_{cntr}, S_{kurt}, S_{skew}
MFCC	3 / 13	$\mu MFCC_0$, $\mu MFCC_1$, $\mu MFCC_2$
Taux de passage par zéro	1 / 1	ZCR
Moments de la forme d'onde	2 / 3	T_{var}, T_{kurt}
Moments de l'enveloppe	2 / 4	E_{mean}, E_{var}
Psychoacoustiques	13 / 26	Et, Ldr_1, Ldr_2, Acu, Ldr_{22}, Ldr_{24}, Ldr_{23}, Ldr_{21}, Ldr_3, Ldr_{20}, Ldr_{17}, Ldr_{19}, Ldr_{18}

TAB. 10.2 – Attributs sélectionnés pour la segmentation en sections de signaux de musique

pour la segmentation semble être la puissance du signal, mesurée de diverses façons par les attributs $\mu MFCC_0$, T_{var} et E_{mean}, tous sélectionnés. La pertinence des attributs $OBSIR$ et des moments spectraux suggère également l'importance des critères de timbre pour la segmentation. De façon surprenante, aucun attribut extrait de la sortie d'un banc de filtres en demi-tons n'est sélectionné. De tels attributs sont pourtant couramment utilisés dans la littérature. Une première explication possible est que notre base de données est plus diverse que celles utilisées dans la littérature (de taille souvent limitée), et qu'elle inclut en particulier des signaux des genres électroniques ou hip-hop dans lesquels la distinction des sections se fait avant tout par des changements d'instrumentation, plutôt que par des modulations de tonalité. Par ailleurs, parmi les attributs sélectionnés, des changements de tonalité pourraient être perçus par des modification du taux de passage par zéro, si nous le considérons comme une estimation très grossière de fréquence fondamentale, ou par les moments spectraux. Il semble également que les autres études ont sous-estimé l'importance du timbre, de la texture et du rythme, pris en compte dans les autres attributs choisis.

10.1.3 Segmentation par détection de nouveauté

Comme nous l'avons vu, nous cherchons à obtenir une segmentation en détectant les frontières de segments. La détection de ces points de changement peut être formulée comme un problème de détection de nouveauté, qui consiste à déterminer, étant donné un ensemble d'exemples de référence (des vecteurs de paramètres acoustiques par exemple), si un ensemble d'observations sont générées par le même processus que celui par lequel ont été générés les exemples de référence.

Ainsi, détecter si un changement de section s'est produit à la trame m_0 correspond à décider si les observations pour les trames d'indices $m > m_0$ (l'ensemble de ces trames forme les *données futures*) sont nouvelles par rapport aux trames d'indices $m < m_0$ (*données passées*). En pratique, seulement un nombre limité d'observations sont considérées pour les *données passées* et les *données futures*.

Toutes les méthodes que nous allons décrire par la suite reposent alors sur la même formulation (illustrée dans la figure 10.1). Une fenêtre glissante centrée en m_0, de longueur $2L + 1$ est considérée. m_0 est considéré comme la frontière entre deux sections si les données futures $S_2(m_0) = \{\mathbf{x}(m), m_0 + 1 \leq m \leq m_0 + L\}$ sont nouvelles par rapport aux données passées $S_1(m_0) = \{\mathbf{x}(m), m_0 - L \leq m \leq m_0 - 1\}$. De manière à simplifier les notations, pour une valeur de m_0 donnée, les données futures et passées seront notées S_1 et S_2 et nous noterons $W = S_1 \cup S_2$. Nous ferons par la suite l'hypothèse que les vecteurs de S_i sont des vecteurs aléatoires indépendants, identiquement distribués selon P_i.

Les différentes solutions proposées au problème de la détection de nouveauté diffèrent par la classe de modèles utilisés pour P_1 et P_2, et par le critère utilisé pour les comparer. Les trois familles

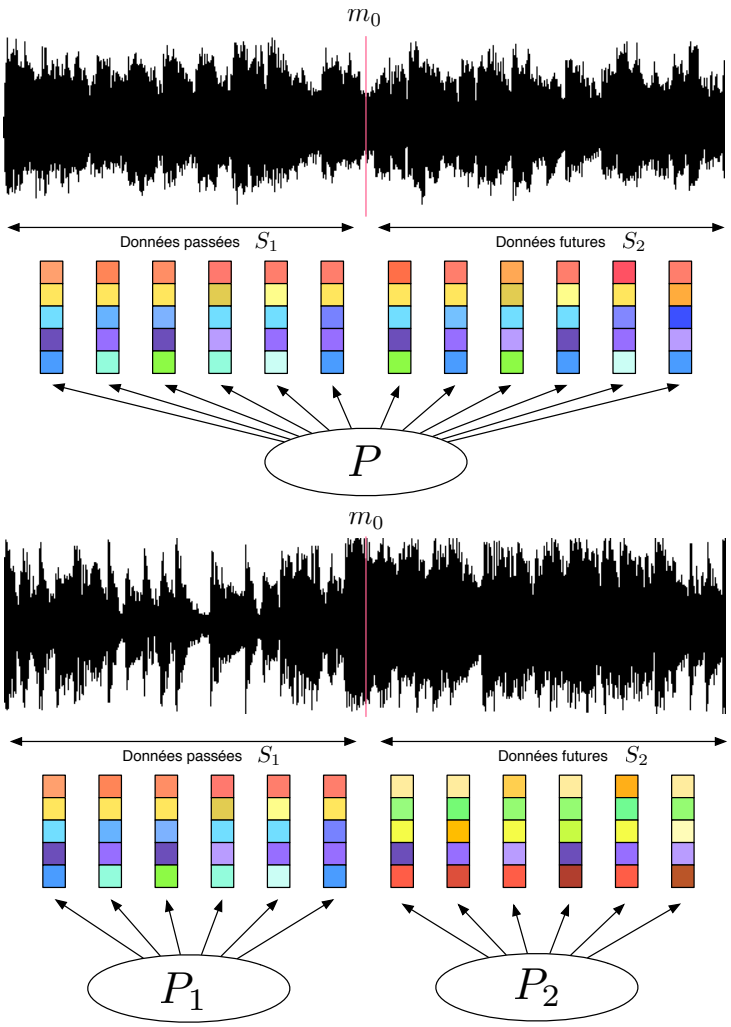


FIG. 10.1 – Principe de la segmentation par détection de nouveauté. En haut, m_0 n'est pas une frontière de section. En bas, m_0 est une frontière de section.

de méthodes que nous présentons illustrent cette diversité.

10.1.3.1 Critère d'information Bayésien

Le critère d'information bayésien – *Bayesian Information Criterion* (BIC) est un critère de vraisemblance pénalisée classique utilisé en sélection de modèle. Il a été utilisé avec succès pour des tâches de segmentation parole/musique ou pour la segmentation en locuteurs [CG98; ZH00]. Pour un modèle M paramétré par N paramètres θ_j, décrivant un ensemble de L réalisations d'une variable aléatoire $\mathbf{x}$ le BIC est défini par :

$$BIC(M) = -\frac{1}{2}N \log L + \log l(\mathbf{x}, \boldsymbol{\theta}^*) \tag{10.1}$$

Où $\boldsymbol{\theta}^*$ sont les paramètres de M estimés au maximum de vraisemblance, et $\log l(\mathbf{x}, \boldsymbol{\theta}^*)$ la valeur maximale de la log-vraisemblance. Par exemple, dans le cas où nous observons L réalisations d'un vecteur aléatoire gaussien de $\mathbb{R}^d$, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, avec $\boldsymbol{\Sigma}$ complète :

$$\begin{aligned} N &= \underbrace{d}_{\text{paramètres libres pour } \mu} + \underbrace{\frac{1}{2}d(d+1)}_{\text{paramètres libres pour } \boldsymbol{\Sigma}} \\ \log l(\mathbf{x}, \boldsymbol{\theta}^*) &= \sum_{i=1}^{L} -\frac{1}{2}\log|\boldsymbol{\Sigma}^*| - \frac{1}{2}(\mathbf{x_i} - \boldsymbol{\mu^*})^T \boldsymbol{\Sigma}^{*-1}(\mathbf{x_i} - \boldsymbol{\mu^*}) = -\frac{1}{2}L\log|\boldsymbol{\Sigma}^*| - \frac{1}{2}Ld \end{aligned}$$

Dans le problème de segmentation, nous souhaitons comparer les deux modèles suivants :

M_1 **:** Les données dans S_1 et S_2 sont toutes distribuées selon $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

M_2 **:** Les données dans S_i sont distribuées selon $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

Les BIC de ces modèles sont :

$$BIC(M_1) = -\frac{1}{2}N\log 2L - \frac{1}{2}(2L)\log|\boldsymbol{\Sigma}^*| - \frac{1}{2}(2L)d \tag{10.2}$$

$$BIC(M_2) = -\frac{1}{2}2N\log 2L - \frac{1}{2}L\log|\boldsymbol{\Sigma}_1^*| - \frac{1}{2}Ld - \frac{1}{2}L\log|\boldsymbol{\Sigma}_2^*| - \frac{1}{2}Ld \tag{10.3}$$

De manière à choisir le meilleur de ces modèles, nous nous intéressons à la différence $\Delta BIC = BIC(M_2) - BIC(M_1)$:

$$\Delta BIC = \frac{1}{2}\left(2L\log|\boldsymbol{\Sigma}^*| - L\log|\boldsymbol{\Sigma}_1^*| - L\log|\boldsymbol{\Sigma}_2^*| - (d + \frac{1}{2}d(d+1))\log 2L\right) \tag{10.4}$$

Cette expression ne dépend que des matrices de covariance estimées sur S_1, S_2 et W, qui sont faciles à calculer. Cependant, dans notre application, $d = 32$ et $L = 64$ (fenêtres futures et passées longues de 8 secondes), il n'est donc pas raisonnable d'estimer des matrices de covariance pleines aussi grandes à partir de si peu de données. Nous imposerons alors à $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_1$ et $\boldsymbol{\Sigma}_2$ d'êtres diagonales. L'expression de ΔBIC ne change pas, si ce n'est le dernier terme du BIC, qui est remplacé par $2d$ (nombre de paramètres d'une loi normale multivariée en dimension d, dont la matrice de covariance est diagonale).

On peut alors détecter un changement de section quand le deuxième modèle est préféré au premier, c'est à dire pour $\Delta BIC > 0$. La position optimale du changement de section correspond à un maximum local de ΔBIC. Un exemple de fonction $d^{BIC}(m_0) = \Delta BIC(m_0)$ calculée pour un enregistrement musical est donné dans la figure 10.4.

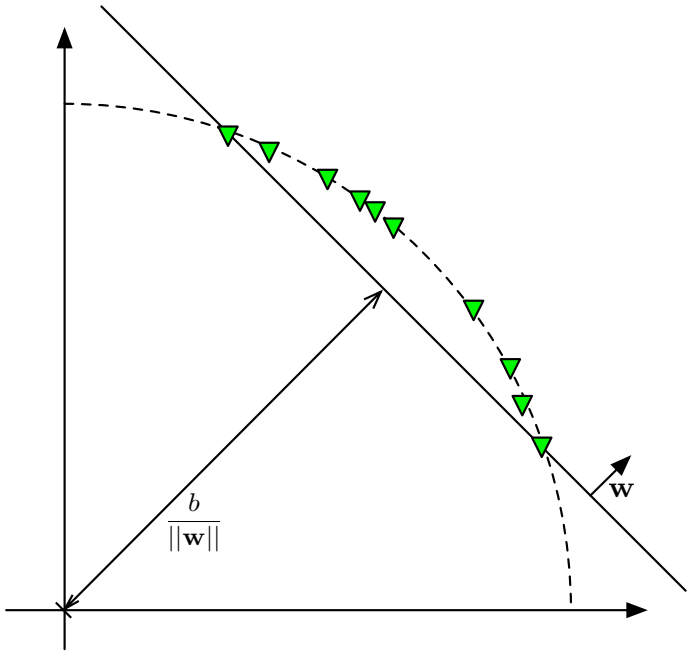


FIG. 10.2 – Séparation par un hyperplan des points sur une hypersphère

10.1.3.2 Machine à vecteurs de support à une classe et détection de nouveauté

Nous présentons dans cette section des méthodes de détection de nouveauté utilisant les machines à vecteurs de support à une classe, principalement une méthode basée sur le rapport de vraisemblance introduite par Canu et Smola dans [CS05], et une méthode dénommée KCD (Kernel Change Detection) utilisant un critère voisin du critère de Fisher, introduite par Desobry et al. dans [DDD05].

Machines à vecteurs de support à une classe Les machine(s) à vecteurs de support à une classe (SVM1C) fournissent une solution au problème suivant : étant donné un ensemble d'apprentissage constitué d'objets décrits par des vecteurs d'attributs réels $(\mathbf{x}_i)_{i\in\{1,\ldots,N\}}$, déterminer une fonction $f(\mathbf{x})$ telle que $f(\mathbf{x}) > 0$ si et seulement si $\mathbf{x}$ est similaire aux éléments de l'ensemble d'apprentissage, où, plus précisément, si $\mathbf{x} \in \mathcal{R}$ où $\mathcal{R}$ est le support de $P(\mathbf{x})$, la plus petite région vérifiant $\int_{\mathcal{R}} p(\mathbf{x})d(\mathbf{x}) = 1$.

À cet effet, considérons tout d'abord une application $\phi : \mathbb{R}^d \mapsto \mathcal{H}$ où $\mathcal{H}$ est un espace de Hilbert (voir annexe B.3), vérifiant la propriété de normalisation suivante : $\phi(\mathbf{x}) \cdot \phi(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) = 1$. Par exemple, le noyau gaussien présenté en B.3.2.2 et utilisé dans toute cette section vérifie cette propriété. Ainsi, dans l'espace $\mathcal{H}$, les points de l'ensemble d'apprentissage sont tous sur une hypersphère de rayon 1. Nous supposons qu'il existe un hyperplan $H(\mathbf{w}, b)$ séparant les points $(\phi(\mathbf{x}_i))_{i\in\{1,\ldots,N\}}$ de l'origine (voir figure 10.2), et nous nous proposons de déterminer celui de marge $\frac{b}{||\mathbf{w}||}$ maximale.

Notons tout d'abord que les hyperplans $H(\mathbf{w}, 0)$ constituent une solution dégénérée, inintéressante. Sans perte de généralité (quitte à normaliser et changer le signe de $\mathbf{w}$), nous pouvons imposer $b = 1$. Le problème d'optimisation correspondant est alors :

$$\text{minimiser} \quad \frac{1}{2}||\mathbf{w}||^2 \tag{10.5}$$

$$\text{sous contraintes} \quad \phi(\mathbf{x}_i) \cdot \mathbf{w} \geq 1 \tag{10.6}$$

L'introduction de multiplicateurs de Lagrange et l'expression des conditions de Karush-Kuhn-Tucker (annexe B.1.2) permet la formulation du problème dual :

$$\text{minimiser} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i \tag{10.7}$$

$$\text{sous contraintes} \quad \alpha_i \geq 0 \tag{10.8}$$

Cette forme se prête à la *ruse du noyau* (annexe B.3), puisque $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$. La fonction de décision s'y prête également et devient :

$$f(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i K(\mathbf{x}_i, \mathbf{x}) + 1 \tag{10.9}$$

En présence de données bruitées, la séparation n'est pas toujours possible. Une solution consiste à utiliser comme dans l'annexe B.2.2 des variables de marge autorisant la violation de certaines des contraintes. Le problème d'optimisation dual prend alors la forme :

$$\text{minimiser} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i \tag{10.10}$$

$$\text{sous contraintes} \quad 0 \leq \alpha_i \leq C \tag{10.11}$$

Et se résoud par les mêmes méthodes que celles décrites dans l'annexe B.2.3. Quelques bornes relatives à la capacité de cette méthode à estimer le support d'une distribution à partir d'exemples sont données dans [SPST+99].

Application directe à la détection de nouveauté Les SVM1C peuvent alors être directement appliquées à la détection de nouveauté par la méthode suivante :

- Une SVM1C est apprise sur les données passées $S_1(m_0)$, produisant une fonction de décision $f_{m_0}(\mathbf{x})$.
- La nouveauté de la fenêtre future est mesurée par la fraction de vecteurs dissimilaires aux vecteurs de S_1 qu'elle contient. La dissimilarité est mesurée par le signe de $f_{m_0}(\mathbf{x})$.

Le critère obtenu est ainsi :

$$d^{frac}(m_0) = \frac{1}{2}\sum_{\mathbf{x}\in S_2(m_0)}(1 - \operatorname{sgn} f_{m_0}(\mathbf{x})) \tag{10.12}$$

Rapport de vraisemblance Une interprétation des SVM1C en termes d'estimation des paramètres d'une distribution exponentielle généralisée au maximum a posteriori est donnée par Canu et Smola dans [CS05]. Le résultat essentiel est que si l'on admet que les éléments de S_i sont distribués selon :

$$P_i(\mathbf{x};\boldsymbol{\theta}) = \exp\left(\phi(\mathbf{x})\cdot\boldsymbol{\theta} - g(\boldsymbol{\theta})\right) \tag{10.13}$$

Où $\phi(\mathbf{x})$ est une statistique exhaustive de x et $g(\boldsymbol{\theta})$ une fonction assurant la normalisation de $P_i(\mathbf{x};\boldsymbol{\theta})$, alors une estimée de $P_i(\mathbf{x})$ est :

$$\hat{P}_i(\mathbf{x}) = \mu(\mathbf{x}) \exp\left(\sum_{m=1}^{N} \alpha_m^i K(\mathbf{x}, \mathbf{x}_m^i) - k_i\right) \tag{10.14}$$

Où k_i et $\mu(\mathbf{x})$ assurent la normalisation, $\mathbf{x}_m^i$ sont des vecteurs de S_i, et α_m^i sont les multiplicateurs de Lagrange associés, obtenus par apprentissage d'une SVM1C sur S_i.

Il est alors possible de définir le rapport de vraisemblance entre les deux hypothèses :

– Les éléments de S_1 et S_2 sont distribués selon P_1 et P_2 (respectivement).

– Les éléments de S_1 et S_2 sont distribués selon une même distribution P_1.

Ce rapport de vraisemblance, évalué sur une fenêtre centrée en m_0 est supérieur à 1 quand m_0 est une frontière de segment. Son expression est :

$$R = \frac{\prod_{\mathbf{x}\in S_1} P_1(\mathbf{x}) \prod_{\mathbf{x}\in S_2} P_2(\mathbf{x})}{\prod_{\mathbf{x}\in W} P_1(\mathbf{x})} = \frac{\prod_{\mathbf{x}\in S_2} P_2(\mathbf{x})}{\prod_{\mathbf{x}\in S_2} P_1(\mathbf{x})} \tag{10.15}$$

En utilisant les estimées $\hat{P}_1(\mathbf{x})$ et $\hat{P}_2(\mathbf{x})$, nous obtenons :

$$\log R = \left(\sum_{\mathbf{x}\in S_2} \sum_{m=1}^{N} \alpha_m^2 K(\mathbf{x}, \mathbf{x}_m^2) - k_2\right) - \left(\sum_{\mathbf{x}\in S_2} \sum_{m=1}^{N} \alpha_m^1 K(\mathbf{x}, \mathbf{x}_m^1) - k_1\right) \tag{10.16}$$

Le premier terme mesure la performance de la SVM1C sur son propre ensemble d'apprentissage. Nous pouvons le supposer nul (du moins, ses variations sont faibles et ont peu d'incidence). Nous en déduisons ainsi la fonction de détection simplifiée suivante :

$$d^{LLR}(m_0) = -\sum_{\mathbf{x}\in S_2(m_0)} \sum_{m=1}^{N} \alpha_m^{(1,m_0)} K(\mathbf{x}, \mathbf{x}_m^{(1,m_0)}) \tag{10.17}$$

Un exemple de fonction de détection produite est donné dans la figure 10.4.

Une variante du critère de Fisher (KCD) Dans l'espace des attributs transformés $\mathcal{H}$, les vecteurs d'attributs de S_i sont placés sur une hypersphère de rayon 1, et séparés de l'origine avec la marge maximale par un hyperplan H_i. L'intersection de l'hyperplan et de l'hypersphère définit un cercle $\mathcal{C}_i$, et forme une calotte de sommet $\mathbf{c}_i$. Nous pouvons en outre considérer un point quelconque $\mathbf{p}_i$ sur le cercle $\mathcal{C}_i$ (voir figure 10.3).

En s'inspirant du critère de Fisher[1], Desobry et al. proposent dans [DDD05] le critère de dissimilarité suivant entre S_1 et S_2, rapport entre une mesure de la dispersion inter-classe (mesurée par la longueur de l'arc joignant $\mathbf{c}_1$ et $\mathbf{c}_2$), et une mesure de la dispersion intra-classe (mesurée par les longueurs des arcs joignant $\mathbf{c}_i$ et $\mathbf{p}_i$) :

$$\mathcal{D} = \frac{\widehat{\mathbf{c}_1\mathbf{O}\mathbf{c}_2}}{\widehat{\mathbf{c}_1\mathbf{O}\mathbf{p}_1} + \widehat{\mathbf{c}_2\mathbf{O}\mathbf{p}_2}} \tag{10.18}$$

Des considérations géométriques permettent alors de calculer $\mathcal{D}$ à partir des matrices de Gram $\mathbf{K}_{ij}$ telles que l'élément en ligne m et colonne n soit $K(\mathbf{x}^i(m), \mathbf{x}^i(n))$, et des vecteurs $\boldsymbol{\alpha}_i$ contenant les multiplicateurs de Lagrange obtenus par apprentissage d'une SVM1C sur S_i :

$$\widehat{\mathbf{c}_1\mathbf{O}\mathbf{c}_2} = \arccos\left(\frac{\boldsymbol{\alpha}_1^T \mathbf{K}_{12} \boldsymbol{\alpha}_2}{\sqrt{\boldsymbol{\alpha}_1^T \mathbf{K}_{11} \boldsymbol{\alpha}_1}\sqrt{\boldsymbol{\alpha}_2^T \mathbf{K}_{22} \boldsymbol{\alpha}_2}}\right) \tag{10.19}$$

$$\widehat{\mathbf{c}_i\mathbf{O}\mathbf{p}_i} = \arccos\left(\frac{1}{\sqrt{\boldsymbol{\alpha}_i^T \mathbf{K}_{ii} \boldsymbol{\alpha}_i}}\right) \tag{10.20}$$

On notera $d^{KCD}(m_0)$ la valeur prise par ce critère évalué sur la fenêtre glissante centrée en m_0. Un exemple est donné dans la figure 10.4.

[1] En fait, Pour $L \to \infty$ et un noyau gaussien, le critère proposé tend vers le critère de Fisher calculé dans l'espace $\mathcal{H}$.

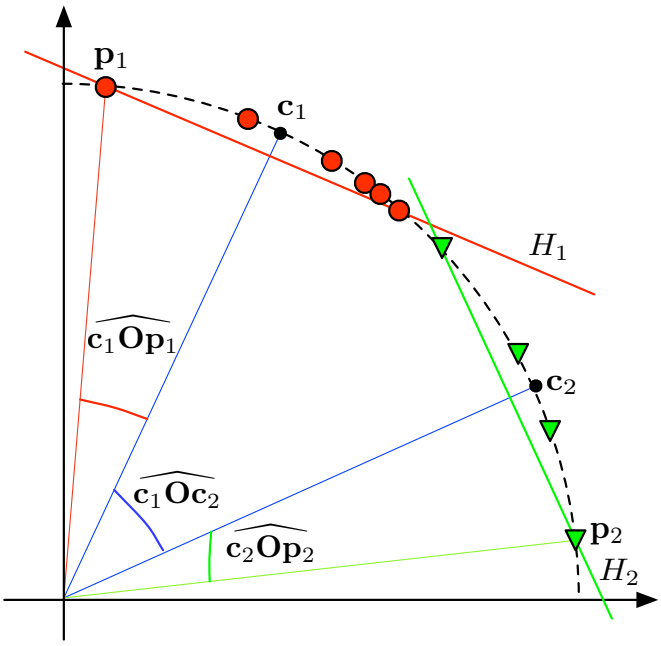


FIG. 10.3 – Principe de l'algorithme KCD

Méthode efficace de calcul Observons tout d'abord que les trois critères présentés dans cette section (fraction d'éléments hors support, rapport de vraisemblance, KCD) ne dépendent que des vecteurs dont les multiplicateurs de Lagrange associés sont non-nuls, réduisant considérablement la charge en calculs de ces méthodes.

Nous observons également qu'en raison de l'utilisation d'une fenêtre glissante, l'évaluation d'un de ces critères en deux points m_0 et $m_0 + 1$ successifs demande la résolution du problème de minimisation quadratique sous contrainte présenté dans les équations 10.11 pour des ensembles d'apprentissage $S_1(m_0)$ and $S_1(m_0 + 1)$ ayant $L - 1$ vecteurs en commun. Cette propriété permet un gain substantiel lors des calculs. En effet, l'apprentissage du SVM1C pour la fenêtre $S_1(m_0+1)$, s'effectue par une méthode itérative semblable à celle décrite dans l'annexe B.2.3. Durant la phase d'initialisation, les multiplicateurs de Lagrange associés aux vecteurs qui étaient déjà dans $S_1(m_0)$ sont préservés, tandis que le multiplicateur de Lagrange associé au vecteur entrant $\mathbf{x}(m_0 + L + 1)$ est initialisé à 0. Ainsi, dans le cas où ni le vecteur entrant, ni le vecteur sortant ne sont des vecteurs de support, la procédure de résolution itérative est directement initialisée avec la solution optimale. Il est également possible de préserver le contenu du cache utilisé dans diverses implémentations logicielles pour limiter le nombre d'évaluations de la fonction noyau. Afin de mesurer les apports de cette méthode d'adaptation, nous avons considéré un problème de segmentation d'une séquence de 1600 vecteurs de dimension d variable (données synthétiques, correspondant à 8 sections distinctes). Les paramètres choisis sont les suivants : $C = 5$, $\sigma = 1$ (paramètre du noyau gaussien normalisé). Les temps de calculs de la fonction de détection (en secondes) mesurés sur une machine dotée d'un processeur Core Duo cadencé à 2 GHz, utilisant la boîte à outils Matlab SimpleSVM [LCV+03] et sa forme modifiée pour utiliser l'adaptation sont donnés dans la table 10.3.

L	d	**Sans adaptation (s)**	**Avec adaptation (s)**
10	10	35.7	**7.0**
10	100	36.3	**7.2**
100	10	197.2	**23.1**
100	100	241.5	**26.8**

TAB. 10.3 – Temps de calcul des fonctions de détection avec et sans résolution adaptative des SVM à 1 classe

10.1.3.3 Distances probabilistes dans un espace de Hilbert à noyau reproduisant (RKHS)

Une autre façon de mesurer la nouveauté des vecteurs de S_2 par rapport aux vecteurs de S_1 est d'utiliser une mesure de similarité entre les distributions P_1 et P_2, estimées à partir des éléments de S_1 et S_2. La distance de Bhattacharyya ou la divergence de Kullback-Leiber symétrisée sont des exemples de telles mesures de similarité entre distributions (dites distances probabilistes) :

$$d_B(P_1, P_2) = \int_{\mathbf{x}\in\mathbb{R}^d} \sqrt{p_1(\mathbf{x})p_2(\mathbf{x})}d\mathbf{x} \tag{10.21}$$

$$d_{sKL}(P_1, P_2) = \int_{\mathbf{x}\in\mathbb{R}^d} (p_1(\mathbf{x}) - p_2(\mathbf{x})) \log\left(\frac{p_1(\mathbf{x})}{p_2(\mathbf{x})}\right) d\mathbf{x} \tag{10.22}$$

Dans le cas gaussien, c'est à dire pour $P_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, il existe des expressions analytiques de ces distances :

$$\begin{aligned} d_B(P_1, P_2) &= \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left[\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\right]^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\log\frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}} \\ d_{sKL}(P_1, P_2) &= \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\operatorname{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1 - 2\mathbf{I}) \end{aligned}$$

Cependant, l'hypothèse de gaussianité des données n'est pas valable dans notre cas, et la difficulté de l'estimation des paramètres à partir d'un nombre réduit d'observations, pour d grand, a déjà été discutée dans la section 10.1.3.1. À l'opposé, nous pourrions utiliser des estimateurs non-paramétriques (estimateurs de Parzen par exemple [DHS01]) mais les calculs des distances requièrent, dans ce cas, une intégration numérique qui s'avérerait trop coûteuse lorsque d est grand.

Une solution proposée par Zhou et Chellappa dans [ZC06] consiste à projeter les données dans un espace de Hilbert $\mathcal{H}$ muni d'un noyau reproduisant $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. L'hypothèse de gaussianité des données projetées $(\phi(\mathbf{x}_i))$ est alors plus réaliste[2]. Nous pouvons ainsi estimer $\boldsymbol{\mu}_i$ et $\boldsymbol{\Sigma}_i$ selon :

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{L}\sum_{\mathbf{x}\in S_i} \phi(\mathbf{x}_i) = \boldsymbol{\Phi}_i\mathbf{s} \tag{10.23}$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{1}{L}\sum_{\mathbf{x}\in S_i} (\phi(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_i)(\phi(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_i)^T = \boldsymbol{\Phi}_i\mathbf{J}\mathbf{J}^T\boldsymbol{\Phi}_i^T \tag{10.24}$$

[2]Cette supposition courante, selon laquelle une projection d'un espace de dimension finie réduite vers un espace de dimension grande ou infinie gaussianise les données, est à la base de méthodes comme l'analyse en composantes principales à noyaux, ou l'analyse discriminante linéaire de Fisher à noyaux. Quelques justifications théoriques sont données dans [HL06].

Où $\mathbf{\Phi}_i$ contient tous les vecteurs de S_i, $\mathbf{e}$ est un vecteur unitaire, $\mathbf{s} = \frac{1}{L}\mathbf{e}$, $\mathbf{J} = L^{-1/2}\left(\mathbf{I} - \mathbf{s}\mathbf{e}^T\right)$. Malheureusement, $\hat{\mathbf{\Sigma}}_i$ n'est pas de rang plein, donc non inversible dès que $\dim \mathcal{H} > L$. Zhou et Chellappa proposent donc d'approximer $\hat{\mathbf{\Sigma}}_i$ par la matrice suivante :

$$\mathbf{C}_i = \mathbf{\Phi}_i \mathbf{J} \mathbf{Q}_i \mathbf{Q}_i^T \mathbf{J}^T \mathbf{\Phi}_i^T + \rho \mathbf{I} \tag{10.25}$$

$\mathbf{C}_i$ a les trois propriétés suivantes :

- Elle est régularisée. D'une part, la matrice $\mathbf{Q}_i$, de dimension $L \times r$, avec $r << L$, limite le nombre de degrés de libertés de $\mathbf{C}_i$; et d'autre part, le coefficient ρ joue un rôle similaire au coefficient de rétrécissement (*shrinkage*) utilisé pour estimer des matrices de covariances de grande taille (voir [DHS01] pp 113–114).
- Elle est inversible (en utilisant la formule de Woodbury).
- Son inverse dépend de la quantité $\mathbf{Q}_i^T \mathbf{J}^T \mathbf{\Phi}_i^T \mathbf{\Phi}_i \mathbf{J} \mathbf{Q}_i$. Or, $\mathbf{\Phi}_i^T \mathbf{\Phi}_i$ est la matrice de Gram $\mathbf{K}_{ii}$, et peut être directement calculée à partir des données, sans projection.

$\mathbf{Q}_i$ est choisie pour que $\mathbf{C}_i$ soit une approximation de $\hat{\mathbf{\Sigma}}_{\boldsymbol{i}}$, au sens où ces deux matrices ont les mêmes valeurs propres principales et vecteurs propres associés. Le calcul de $\mathbf{Q}_i$ repose sur l'analyse des r valeurs propres dominantes de la matrice $\mathbf{J}^T \mathbf{K}_{ii} \mathbf{J}$. Les calculs détaillés de l'approximation de la matrice de covariance et des distances probabilistes sont présentés dans [ZC06].

Comme précédemment, le fait que les calculs s'effectuent sur des fenêtres glissantes permet une implémentation particulièrement rapide. D'une part le calcul complet des matrices de Gram $\mathbf{K}_{ii}$ n'a pas à être effectué, seules les dernières lignes et colonnes sont à calculer à chaque décalage de la fenêtre d'observation. Ensuite, la décomposition de $\mathbf{J}^T \mathbf{K}_{ii} \mathbf{J}$ en ses r plus grandes valeurs propres se fait typiquement par des méthodes itératives (méthode d'Arnoldi, comme utilisé dans la fonction `eigs` de Matlab) qui convergent plus rapidement lorsqu'elles sont initialisées par une approximation des vecteurs propres à extraire ; une telle approximation pouvant alors être fournie par la décomposition effectuée à l'étape précédente.

Un exemple de fonctions de détection produites par ces méthodes, pour les distances de Bhattacharyya et la divergence de Kullback-Leibler symétrisée sont données dans la figure 10.4.

10.1.4 Évaluation des approches proposées pour la segmentation

Les méthodes présentées dans cette section étant nouvelles ou n'ayant pas été appliquées au problème de la segmentation de signaux de musique, nous les évaluons séparément ici. À ces fins, une base de données de 100 signaux de musique (`Music-100`, décrite dans l'annexe D.2) a été constituée et annotée.

Les fonctions de détection correspondant à toutes les méthodes décrites précédemment ont été calculées pour chacun des signaux, avec un jeu d'attributs complet (ceux décrits en 10.1.1), et les attributs sélectionnés listés dans la table 10.2. Les paramètres spécifiques à chaque méthode – paramètre du noyau gaussien, paramètre C des SVM1C, dimension r ont été choisis par validation croisée – une moitié de la base a été utilisée pour déterminer les paramètres maximisant les performances, et ces paramètres ont été utilisés pour effectuer la segmentation sur l'autre moitié. Le seul paramètre fixé une fois pour toutes est la taille de la fenêtre d'observation, $L = 64$ (correspondant à une durée d'observation de 8 secondes).

Les fonctions de détection obtenues présentant de larges variations de dynamique, elles ont été post-traitées selon les méthodes décrites en 4.2.2, avec pour paramètres $W_l = 40$ s et $W_s = 4$ s correspondant respectivement aux tailles maximales et minimales des sections à détecter. Le seuil de détection τ a été fixé à 70 valeurs différentes dans l'intervalle $[-2, 5]$. Les segmentations produites pour chaque valeur de τ ont été évaluées selon les mesures de rappel et de précision :

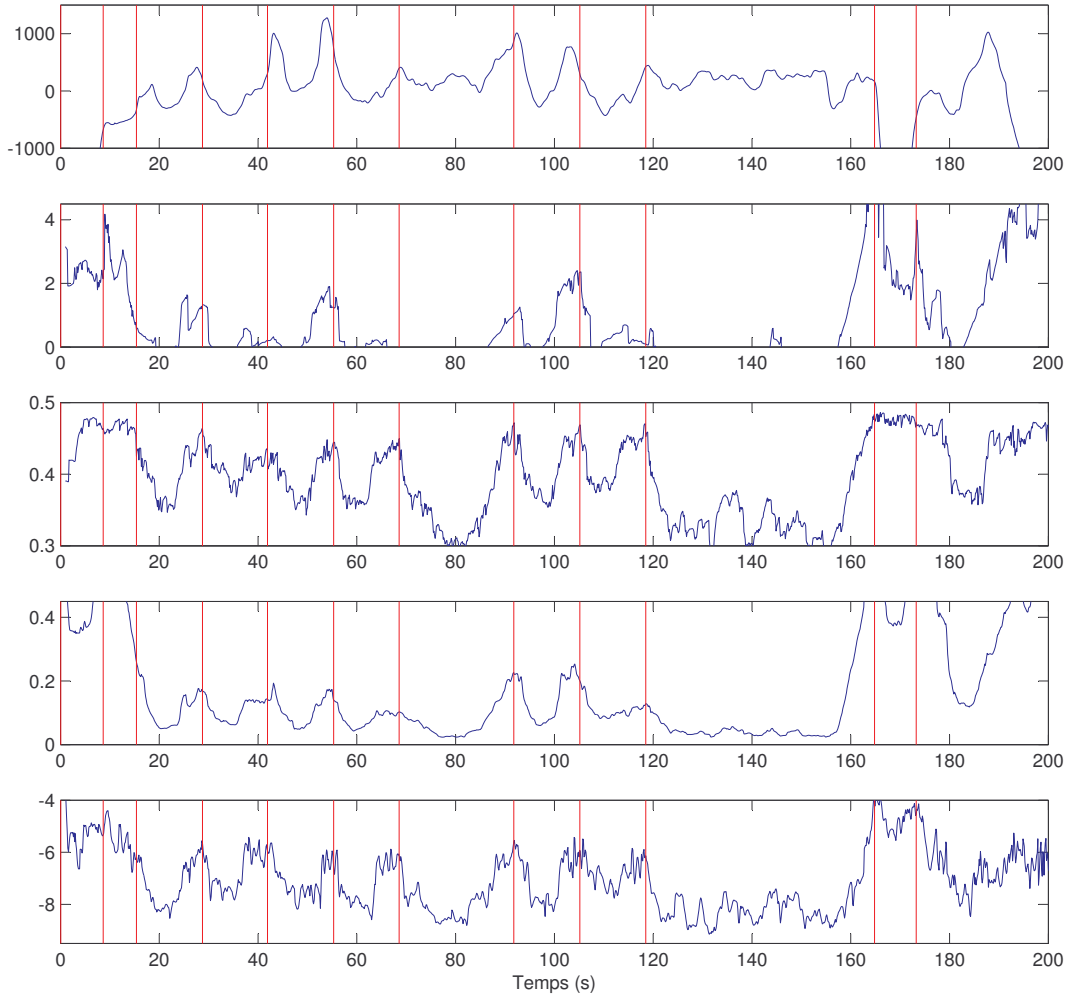


FIG. 10.4 – Fonctions de détection de nouveauté calculées (Saint Étienne – *Split Screen*). De haut en bas : BIC, rapport de vraisemblance avec SVM1C, KCD, divergence de Kullback-Leibler symétrisée et distance de Bhattacharyya. Les changements manuellement annotés sont représentés par des lignes rouges.

Algorithme	**F-mesure**(1) (%)
Distance de Bhattacharyya dans un RKHS	**74**
Divergence de Kullback-Leibler dans un RKHS	68
Critère de Fisher induit par SVM1C (KCD)	72
Rapport de vraisemblance avec SVM1C	67
Critère d'information Bayésien	59

TAB. 10.4 – F-mesure, avec un seuil $\tau = 1$, pour la tâche de détection de frontières de segments dans la base `Music-100`

$$\text{précision}(\tau) = \frac{\text{Nombre de frontières correctement détectées}}{\text{Nombre de frontières détectées}} \tag{10.26}$$

$$\text{rappel}(\tau) = \frac{\text{Nombre de frontières correctement détectées}}{\text{Nombre de frontières à détecter}} \tag{10.27}$$

$$\text{F-mesure}(\tau) = \frac{2 \cdot \text{précision}(\tau) \cdot \text{rappel}(\tau)}{\text{précision}(\tau) + \text{rappel}(\tau)} \tag{10.28}$$

Une erreur égale à 2 s au plus est tolérée entre la position d'une section et un pic dans la fonction de détection. Les courbes de rappel/précision déduites sont données dans la figure 10.5. De plus, la F-mesure, calculée selon la dernière expression, est donnée pour la valeur typique $\tau = 1$, dans la table 10.4.

Les meilleures performances sont obtenues avec la distance de Bhattacharyya dans un RKHS. La divergence de Kullback-Leibler dans un RKHS offre également de bonnes performances pour des taux de rappel faibles. Au delà, l'algorithme KCD offre une meilleure précision. Le rapport de vraisemblance calculé à partir des sorties de SVM1C est un critère globalement moins performant.

Les résultats obtenus avec le BIC sont plus mauvais. Cela peut s'expliquer dans notre cas par la non-gaussianité des données. Une solution classique pour gérer la non-gaussianité des données tout en utilisant le BIC pourrait être de modéliser P_1 et P_2 par des mélanges de gaussiennes. Cependant, l'accroissement du nombre de paramètres causé par ce changement ne permet pas une estimation robuste. Une approche plus robuste et compatible avec la petite taille des fenêtres d'observation consisterait à apprendre des modèles de mélanges de gaussiennes génériques (définis par exemple pour chaque genre ou type de formation instrumentale) et à les adapter aux données observées. Cependant, cette approche, qui serait équivalente à une méthode de segmentation par classification, serait incapable de traiter des genres ou des instrumentations inconnues. L'échec de tels modèles génératifs souligne la robustesse et la pertinence des méthodes à noyaux pour les problèmes où les données observées sont en nombre insuffisant, bien que de grande dimensionnalité.

Nous donnons également dans la figure 10.6 les courbes rappel/précision obtenues avec la meilleure méthode (distance de Bhattacharyya dans un RKHS) et la pire (BIC), avec différents jeux d'attributs : les attributs utilisés dans une étude préliminaire [GR06a], constitués des MFCC, moments spectraux et du taux de passage par zéro ; l'ensemble des 70 attributs candidats considérés, et l'ensemble des attributs sélectionnés en 10.1.2. Nous notons d'abord que dans tous les cas, l'ensemble exhaustif d'attributs introduit en 10.1.1 permet une meilleure segmentation que la paramétrisation simple utilisée en [GR06a]. Dans le cas du BIC, la réduction de la dimensionnalité par sélection d'attributs conduit à de meilleures performances. Dans le cas de la distance de Bhattacharyya, le jeu d'attributs sélectionnés offre des performances similaires au jeu d'attributs complet. Il semble donc que la sélection d'attributs n'est avantageuse en termes de performances que pour les méthodes fragiles face à la "malédiction de la dimensionnalité", ce qui n'est pas le cas des méthodes de segmentation à noyaux. La sélection d'attributs n'est cependant pas inutile, puisqu'elle peut dans ce cas être vue comme un moyen de réduire le coût en calculs de la procédure de segmentation sans impact sur les performances.

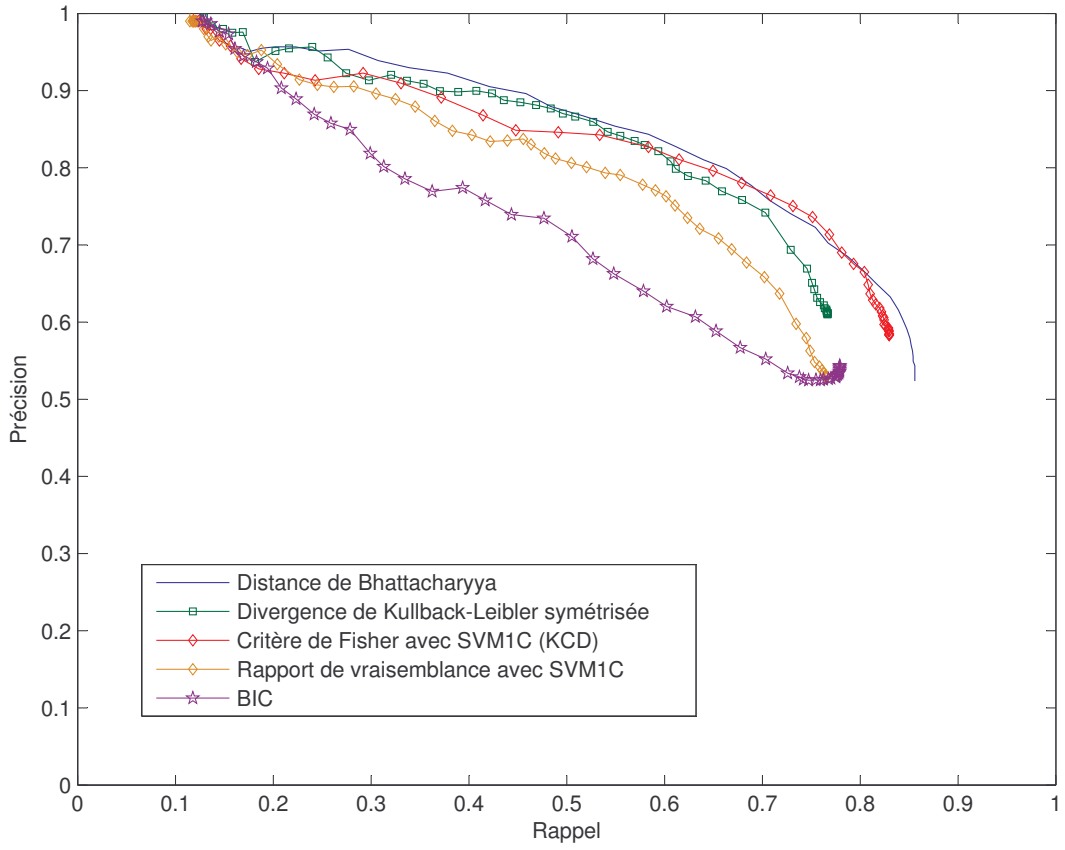


FIG. 10.5 – Courbes rappel/précision pour la tâche de détection de frontières de segments dans la base `Music-100` : Comparaison des algorithmes

10.2 Extraction de la structure des séquences vidéo

Sont présentées ici les approches retenues pour la segmentation d'une séquence vidéo aux niveaux suivants : plans, séquences, et mouvements.

10.2.1 Segmentation en plans

Pour un état de l'art des systèmes dédiés à la segmentation d'une séquence vidéo en plans, le lecteur est invité à consulter le rapport de la dernière évaluation TRECVid [OIKS06], où sont décrits des systèmes capables de détecter à la fois les transitions brutales entre plans (*hard cuts*), et des transitions progressives tels que zooms, effacements ou fondus enchaînés. Ces deux familles de transitions posent des problèmes différents. Discriminer un *hard cut* d'un changement rapide dans l'image (flash, changement d'illumination ou mouvement brusque de caméra) est difficile. Par ailleurs, sur un horizon d'observation temporel court, les transitions progressives résultent en des changements minimes dans l'image, et sont de fait difficiles à détecter.

Dans le cas des clips vidéos, deux observations facilitent cette tâche de segmentation. Premièrement, nous avons observé que sur notre corpus `Video-100` (décrit en annexe D.3) 91% des transitions entre plans sont des *hard cuts*, sans doute parce qu'elles permettent un style de montage très rythmé. Nous pouvons donc obtenir de bonnes performances même en ignorant les autres transi-

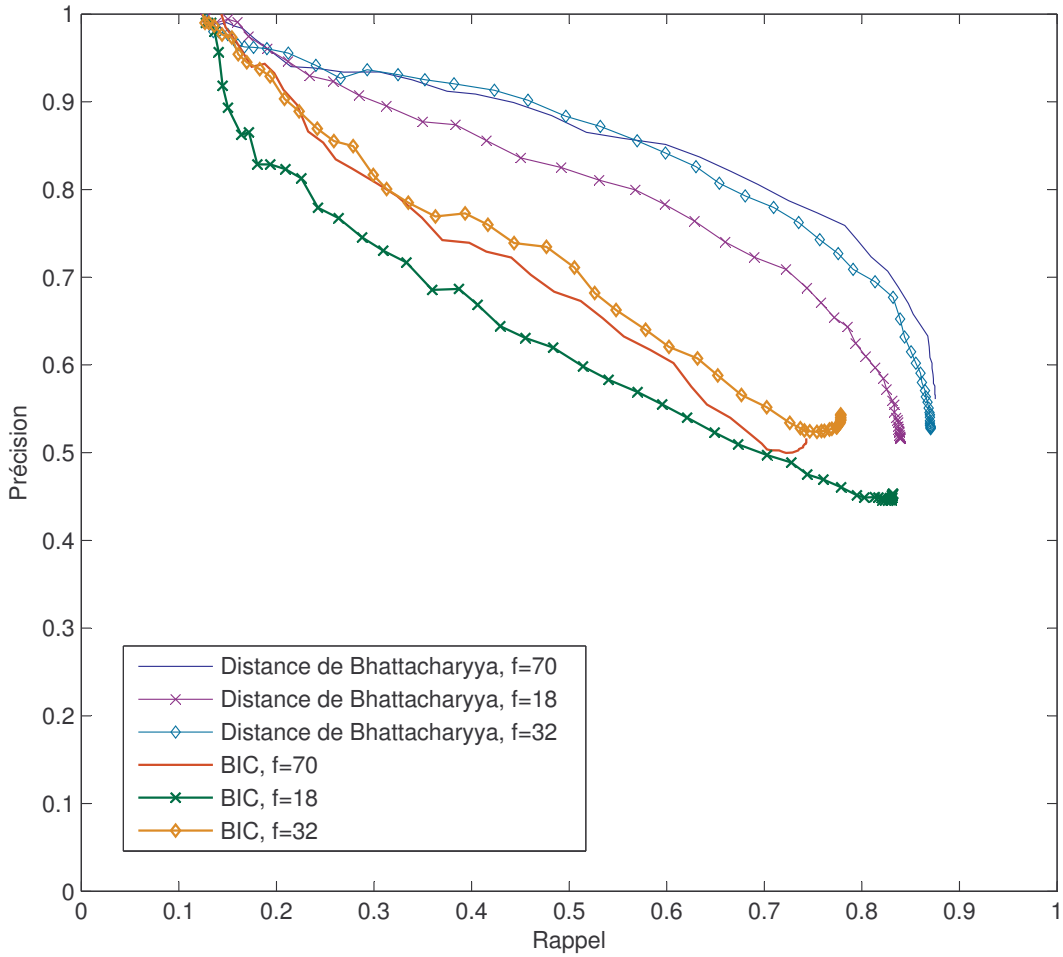


FIG. 10.6 – Courbes rappel/précision pour la tâche de détection de frontières de segments dans la base `Music-100` : Comparaison des ensembles d'attributs

tions. Deuxièmement, pour l'application qui nous intéresse, les changements rapides d'illumination, les flashs, ou mouvements de caméra ne doivent pas être vus comme des faux positifs, puisque de tels événements peuvent être synchrones avec la musique, et doivent donc être détectés. Inversement, les transitions progressives sont moins localisées dans le temps, et leur synchronie avec des événements audio sont plus difficiles à mesurer.

Nous utilisons en conséquence un détecteur de *hard cuts* simple, basé sur la distance entre des attributs de couleur et luminosité entre trames adjacentes. Pour chaque trame, trois histogrammes à 16 classes sont construits à partir des composantes YUV de chacun des pixels de l'image ; produisant un vecteur de 48 attributs, $\mathbf{x}^v(m)$. La fonction de détection des *hard cuts* est alors définie comme :

$$d_s(m) = ||\mathbf{x}^v(m) - \mathbf{x}^v(m-1)||_1 = \sum_{i=1}^{48} |\mathbf{x}_i^v(m) - \mathbf{x}_i^v(m-1)| \tag{10.29}$$

10.2.2 Segmentation en séquences

Nous nous proposons maintenant de segmenter la vidéo en séquences, une séquence étant constituée de plusieurs plans décrivant la même scène. Dans le cas d'un clip vidéo, ces plans peuvent correspondre à différents cadrages du chanteur, par exemple, ou à une alternance entre différents plans montrant chacun des musiciens – tandis que des formes de séquences plus typiques du cinéma se retrouvent dans les clips à contenu narratif.

10.2.2.1 Clustering des trames

Une approche directe consisterait à utiliser les méthodes de détection de nouveauté présentées dans la partie 10.1.3 à une suite de vecteurs d'attributs extraits de chacune des trames. Cependant, une telle approche est trop sensible aux changements brusques causés par les changements de plan au sein d'une même séquence. Il apparaît nécessaire d'effectuer la détection de changement de plan sur une représentation de niveau supérieur de la vidéo. Ainsi, la méthode que nous proposons repose sur un clustering préalable des différentes trames, afin d'obtenir une représentation de la vidéo sous la forme d'une suite d'entiers $y(m)$ indiquant l'indice du cluster auquel est attribuée la trame m. Les *clusters* de trames peuvent ainsi, par exemple, regrouper des trames tirées de plans tournés dans les mêmes décors ; ou bien des trames de plans montrant un même musicien – ils pourraient directement être utilisés pour construire un résumé vidéo comme décrit par Yahiaoui et al. dans [YMH01].

Les méthodes de clustering classiques comme les k-moyennes ou le clustering agglomératif [DHS01] ne prennent pas en compte la dimension temporelle des séquences vidéo, en particulier la contrainte selon laquelle deux images adjacentes dans la séquence sont très probablement associées au même groupe. Une manière d'effectuer un clustering en imposant des contraintes temporelles consiste à apprendre par l'algorithme de Baum-Welch [Rab89] les paramètres d'un HMM à partir de la suite de vecteurs d'attributs extraits de la séquence à segmenter.

Nous utilisons ici les attributs de couleur et de luminosité décrits précédemment. Certains de ces attributs étant corrélés, nous appliquons au préalable une PCA aux vecteurs $\mathbf{x}^v(m)$ observés, et retenons les composantes principales concentrant 90% de la variance (voir section 4.3.2.2). Le nombre moyen d'attributs transformés retenus à l'issue de cette étape est de 27 sur notre base de données.

Pour l'apprentissage du HMM, la matrice de transition $\mathbf{A}$ est initialisée à $A_{ij} = \frac{1}{R}$ où $R = 16$ désigne le nombre d'états. Nous n'imposons ainsi aucune topologie particulière sur le HMM appris, les transitions entre tous les états étant autorisées.

Chaque distribution des vecteurs de paramètres associée à un état du modèle est modélisée par une gaussienne multivariée de matrice de covariance diagonale. Leurs moyennes et covariances sont initialisées à l'aide d'un premier clustering obtenu par l'algorithme des k-moyennes. 10 itérations de l'algorithme de Baum-Welch sont effectuées, produisant ainsi, en plus d'une matrice A_{ij} modélisant les transitions entre groupes de trames, un étiquetage de la séquence par la suite $y(m)$ indiquant le groupe (l'état) auquel appartient la m-ième trame.

Soit une fenêtre glissante $W(m_0)$, centrée en m_0 et longue de $2L+1$ trames. Si cette fenêtre ne recouvre qu'une seule séquence, les trames qu'elle englobe ne proviennent que d'un nombre réduit de groupes (par exemple, 3 groupes si la séquence alterne entre un plan sur le chanteur principal, les danseurs et le décor). Par contre, si cette fenêtre chevauche plusieurs séquences, elle contiendra des trames issues d'un plus grand nombre de groupes. L'entropie de la distribution des états observée sur la fenêtre $W(m_0)$ offre une mesure de dispersion robuste permettant de distinguer ces deux situations :

$$H(m_0) = \sum_{k=1}^{R} -\hat{p}_{m_0}(y=k)\log_2 \hat{p}_{m_0}(y=k) \tag{10.30}$$

$$\text{avec } \hat{p}_{m_0}(y=k) = \frac{\sum_{m=m_0-L}^{m_0+L} \delta^k_{y(m)}}{2L+1} \tag{10.31}$$

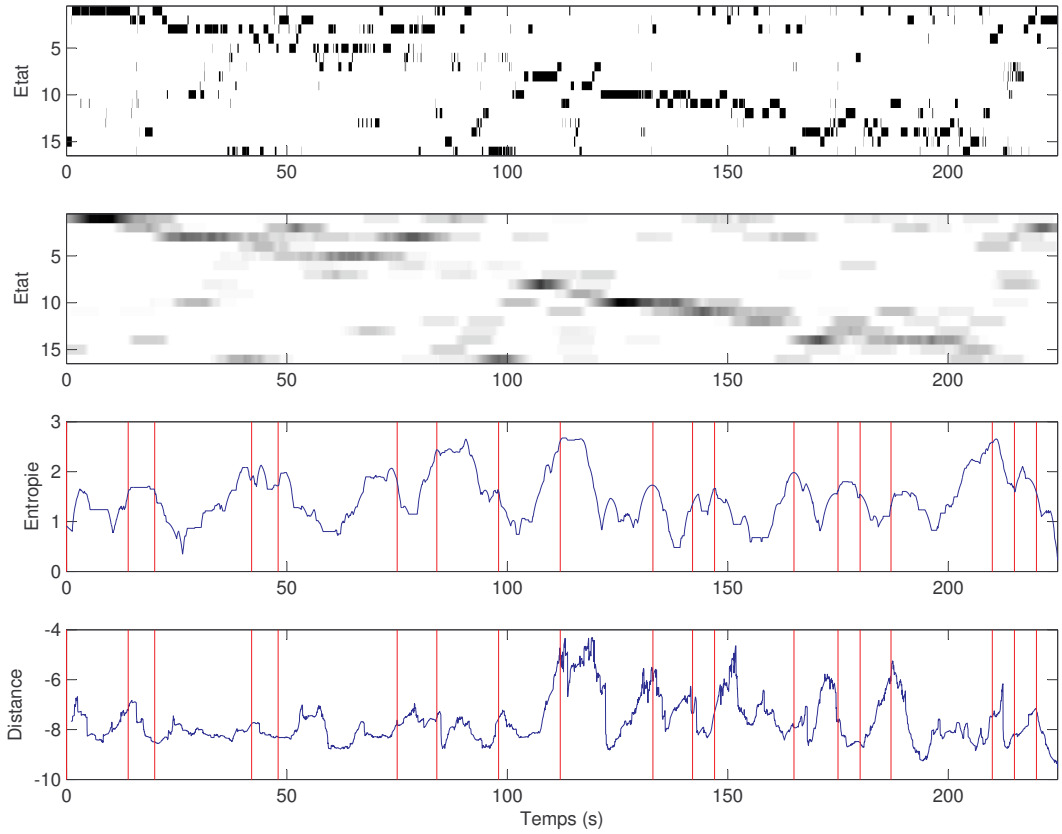


FIG. 10.7 – Segmentation en séquence (Daft Punk – *Burnin'*). De haut en bas : suite d'états du HMM, distribution des états sur une fenêtre d'observation glissante, et entropie de cette distribution. En bas : fonction de détection utilisant la distance de Bhattacharyya, calculée directement sur les vecteurs d'attributs

Les maxima locaux de $H(m_0)$ indiquent ainsi les frontières de séquence. Un exemple est donné dans la figure 10.7. Nous observons, sur l'exemple donné, que l'emploi de méthodes de détection de nouveauté produit des pics dans la fonction de détection correspondant à des changements brusques survenant au sein d'une même section. Par contraste, tous les maxima de la fonction de détection basée sur l'entropie de la distribution des états du HMM correspondent à des changements de section. Notons cependant que dans le cas de séquences très courtes, ces changements rapides ne peuvent pas être détectés et se manifestent par des plateaux dans la fonction de détection.

10.3 Détection d'événements dans une séquence vidéo

Au niveau le plus bas, un plan peut être segmenté en actions ou événements élémentaires, en détectant les instants associés à des modifications de l'intensité du mouvement. Par analogie avec la détection d'onsets sur les signaux audio, nous cherchons à mesurer le "pouls" d'une séquence vidéo.

Bien qu'il existe des systèmes de suivi et d'analyse des mouvements adaptés à une variété de tâches (voir état de l'art au chapitre 6), de tels systèmes ne peuvent être utilisés que dans des environnements bien contrôlés, avec des caméras fixes. Nous ne pouvons utiliser ici que des critères de mouvement les plus génériques, pour lesquels il est nécessaire de trouver un compromis entre

FIG. 10.8 – De gauche à droite : deux trames successives d'une séquence vidéo ; et le champ de vecteurs de mouvement estimé

FIG. 10.9 – Champ de vecteurs de mouvement sur une zone non-texturée

les estimateurs de flot optique (coûteux en calcul mais robustes), et les méthodes se basant sur la différence entre trames successives (peu robustes).

Une méthode particulièrement intéressante et peu coûteuse consiste à extraire une information de mouvement dans le domaine compressé en considérant directement la représentation MPEG de la séquence vidéo. En effet, dans les flux vidéos MPEG, la redondance temporelle est éliminée en codant certaines trames (dites trames P) par leur différence avec la trame précédente, avec compensation du mouvement. Les trames P sont découpées en blocs de 16×16 pixels, dits macroblocs. Soit $\mathbf{I}(x, y, m)$ un bloc d'image de 16×16 pixels centré en (x, y) dans la trame m. Un macrobloc $\mathbf{I}(x, y, m)$ peut être de deux types :

Macrobloc P Il est dans ce cas codé comme la différence $\mathbf{\Delta}(x, y, m) = \mathbf{I}(x, y, m) - \mathbf{I}(x - \delta_x, y - \delta_y, m - 1)$, où $\delta_x(x, y, m)$ et $\delta_y(x, y, m)$ sont choisis pour minimiser $||\mathbf{\Delta}(x, y, m)||_2$. Le vecteur $\mathbf{u}(x, y, m) = \begin{bmatrix} \delta_x(x, y, m) \\ \delta_y(x, y, m) \end{bmatrix}$ peut alors s'interpréter comme un vecteur de mouvement mesurant la vitesse instantanée du bloc $\mathbf{I}(x, y, m)$.

Macrobloc I Il est alors codé de façon absolue, sans référence à une trame précédente.

Un exemple de trame P avec ses vecteurs de mouvement est donné dans la figure 10.8. Les vecteurs extraits de cette façon peuvent être particulièrement bruités sur les régions non-texturées (voir figure 10.9). Pour plus de robustesse, nous effectuons une segmentation grossière de l'image en régions texturées/non-texturées en considérant comme non-texturées les régions dont les coefficients DCT correspondant à des fréquences élevées sont nuls. Les vecteurs de mouvement dans les régions non-texturées sont ignorés.

Soit $N(x, y, m) = ||\mathbf{u}(x, y, m)||_2$ le champ scalaire des normes des vecteurs de mouvement. $N(x, y, m)$ est filtré par un filtre médian de taille 3×3 pour le lisser, produisant un champ $N'(x, y, m)$; et une mesure d'activité de mouvement est alors extraite selon :

$$A(m) = \sqrt{\sum_{\mathbf{I}(x,y,m) \text{ non-texturé}} N'(x, y, m)^2} \tag{10.32}$$

Si la m-ième trame est une trame I, la valeur de $A(m)$ est interpolée linéairement à partir de $A(m-1)$ et $A(m+1)$. Dans les vidéos traitées, le schéma d'alternance des trames (dépendant du codage) est tel qu'une trame I survient toutes les 18 trames.

Soulignons que $A(m)$ diffère du descripteur d'activité de mouvement MPEG-7 [JD01; PD03] en deux points. D'une part, ce dernier descripteur est défini comme l'écart type des valeurs prises par $N(x, y, m)$, de manière à compenser les mouvements constants de caméra (travellings par exemple). Dans notre application, mesurer de tels mouvements est intéressant car ils peuvent être synchrones à la musique. D'autre part, le descripteur de mouvement MPEG-7 est quantifié sur une échelle subjective à 5 valeurs.

De manière à détecter les changements brusques dans la fonction $A(m)$, nous considérons sa dérivée $dA(m)$, obtenue par filtrage par un filtre dérivateur d'ordre 5 (voir section 4.2).

10.4 Conclusion

Nous avons présenté dans cette section les différents outils de segmentation des flux audio et vidéo utilisés par la suite pour définir les mesures de synchronie des changements.

Le problème de la détection des changements de section dans les signaux de musique a été résolu par des outils statistiques de détection de nouveauté. Des outils récents, basés sur les méthodes à noyaux pour efficacement gérer la dimensionnalité et la non-gaussianité des données, se sont montrés plus efficaces que des mesures classiques comme le BIC lors de nos évaluations. Nous avons par ailleurs proposé l'utilisation de distances probabilistes dans un RKHS comme mesure de similarité entre données passées et futures. La distance de Bhattacharyya s'avère être la plus efficace pour la segmentation. Nous avons également présenté quelques stratégies pour implémenter efficacement ces méthodes. De manière à réduire encore le coût en calcul de ces méthodes, il est souhaitable de réduire la dimensionnalité des données à traiter. Un moyen d'y parvenir est de sélectionner les meilleurs attributs. Nous avons proposé dans ce chapitre une procédure de vote sélectionnant les attributs les plus fréquemment capables de discriminer les trames de deux segments adjacents.

Les méthodes retenues pour la segmentation à bas et moyen niveau du flux vidéo (mouvements et plan) sont classiques, et ont été choisies à partir d'observations relatives aux documents à traiter (clips vidéo). La segmentation à haut niveau (séquences) du flux vidéo est effectuée par clustering des trames observées à l'aide d'un HMM ; et en considérant l'entropie de la distribution des états du HMM sur une fenêtre d'observation glissante. Les performances de ces méthodes n'ont malheureusement pas pu être évaluées en tant que telles sur notre base de données. Elles seront cependant utilisées dans le chapitre suivant, détaillant quelques unes de leurs applications.

Publications liées à ce chapitre

Les méthodes de segmentation présentées dans ce chapitre ont fait l'objet d'un article [GER07], étendant des résultats préliminaires publiés dans [GR06a].

CHAPITRE **11**

Mesures de corrélation entre flux audio et vidéo

Nous nous intéressons dans ce chapitre à diverses applications exploitant des mesures de corrélation entre les structures extraites au chapitre précédent. Le calcul de ces corrélations est d'abord présenté dans la section 11.1. Une première application à la recherche de musique d'accompagnement par la vidéo est décrite dans la section 11.2.1 ; elle est évaluée sur une base de données de clips vidéos. Les dépendances entre les mesures de corrélation présentées et le *genre visuel* sont discutées dans la section 11.2.2. Une dernière application envisageable, traitée en 11.2.3, est la resynchronisation des flux audio et vidéo.

11.1 Mesures de corrélation des flux audio et vidéo structurés

Les systèmes de segmentation présentés au chapitre précédent produisent tous des fonctions de détection dont les pics signalent des événements d'intérêt : jeu d'une note, changement de section, changement dans l'intensité de mouvement, changement de plan et de séquence. Il serait possible de seuiller ces fonctions de détection afin d'obtenir une segmentation proprement dite. Une mesure de synchronie ou de corrélation entre les segmentations obtenues consisterait alors à compter le nombre d'opérations élémentaires (fission, fusion, déplacement de frontières) nécessaires pour faire coïncider deux segmentations. Nous n'avons pas suivi cette approche pour plusieurs raisons. Tout d'abord, elle demande le réglage d'un seuil de décision, qui peut supprimer des changements peu marqués mais néanmoins significatifs. Deuxièmement, elle ne prend pas en compte l'intensité de chacun de ces événements. Enfin, elle ne prend pas non plus en compte l'incertitude temporelle relative à la localisation d'un événement (s'agit-il d'un pic, d'une bosse ou d'un plateau dans la fonction de détection ?).

Pour ces trois raisons, nous mesurons directement les corrélations à partir des fonctions de détection, plutôt que sur les segmentations/structures qu'on en déduirait. Soient $d_o(m)$ la fonction de détection produite par le détecteur d'onsets ; $d_c(m)$ la fonction de détection des changements de section dans la musique (voir section 10.1.3) ; $d_m(m)$ la fonction de détection obtenue par différentiation d'une mesure d'activité de mouvement ; $d_s(m)$ la fonction de détection du détecteur de *hard cuts* (voir section 10.2) ; et enfin $d_q(m)$ la fonction de détection des changements de séquence. Toutes ces fonctions de détection sont normalisées et compressées par suppression d'une tendance médiane, et division par une mesure locale d'échelle, comme décrit dans la section 4.2.2. Elles sont également toutes rééchantillonnées à une fréquence commune de 25 Hz qui correspond au nombre de trames par seconde des séquences vidéos utilisées lors de l'évaluation.

11.1.1 Alignement local des fonctions de détection

Lorsque deux événements se produisant dans les flux audio et vidéo (par exemple, un changement de section dans la musique et un changement de plan) sont simultanés, à la trame m, leurs fonctions de détection possèdent toutes deux un pic en m. Cependant, des changements perçus comme simultanés peuvent en réalité différer d'un léger délai – qui peut être aussi bien présent dans le document original (erreur ou imprécision lors du montage), que dû au procédé de détection (délai dans les détecteurs). Ainsi, avant tout calcul des mesures de corrélation, les fonctions de détection sont alignées de manière à maximiser leur corrélation.

Soient $d_a(m)$ et $d_b(m)$ deux fonctions de détection qu'on cherche à aligner. L'alignement consiste à chercher une fonction de déformation temporelle $\phi(m)$ maximisant un critère donné entre $d_a(\phi(m))$ et $d_b(m)$. Soulignons ici que ne sont autorisées que des déformations temporelles limitées, la contrainte $m - 2 \leq \phi(m) \leq m + 2$ étant imposée. Il existe une méthode d'alignement local explicitement conçue pour maximiser la corrélation entre deux trains d'impulsion : le *Correlation Optimized Warping* [NCS98]. Cette méthode est cependant trop coûteuse en calculs pour nos expériences qui requièrent le calcul de plusieurs dizaines de milliers d'alignements. Nous avons donc simplement utilisé une déformation temporelle dynamique – *Dynamic Time Warping* (DTW) [Kru83]. La recherche du chemin d'alignement optimal a été contrainte au voisinage de la diagonale (à ± 2 trames) ; et la valeur absolue a été utilisée pour comparer les points à aligner.

11.1.2 Mesures de corrélation considérées

Différentes mesures issues des statistiques ou de la théorie de l'information peuvent être utilisées pour mesurer la corrélation entre des fonctions de détection $d_a(m)$ et $d_b(m)$.

En particulier, si l'on suppose que les séquences $d_a(m)$ (respectivement $d_b(m)$) se composent de réalisations indépendantes, identiquement distribuées d'une variable aléatoire A (resp. B), on peut définir :

Le coefficient de corrélation de Pearson, défini comme :

$$\rho(A, B) = \frac{\mathbb{E}[(A - \mathbb{E}[A])(B - \mathbb{E}[B])]}{\sqrt{\mathbb{E}[(A - \mathbb{E}[A])^2]\mathbb{E}[(B - \mathbb{E}[B])^2]}} \tag{11.1}$$

Empiriquement, si l'on suppose les fonctions de détection centrées :

$$\rho(A, B) = \frac{\sum_{i=1}^{M} d_a(m) d_b(m)}{\sqrt{\left(\sum_{i=1}^{M} d_a(m)^2\right)\left(\sum_{i=1}^{M} d_b(m)^2\right)}} \tag{11.2}$$

Notons que dans le cas où $d_a(m)$ (respectivement $d_b(m)$) a été seuillée pour obtenir une fonctions de détection $d'_a(m)$ (respectivement $d'_b(m)$) prenant la valeur 1 si m est une frontière de segment et 0 sinon ; On a $\mathbb{E}[A'] \approx 0$, $\mathbb{E}[B'] \approx 0$ et le numérateur de $\rho(A, B)$ correspond alors au nombre de changements co-occurrents observés, tandis que son dénominateur correspond à la moyenne géométrique du nombre de segments dans les deux flux comparés. On retrouve alors le critère de co-occurence utilisé en 8.2.1.2.

L'information mutuelle, définie dans le cas discret par :

$$I(A, B) = \sum_a \sum_b P(A = a, B = b) \log \frac{P(A = a, B = b)}{P(A = a)P(B = b)} \tag{11.3}$$

Pour permettre le calcul de cette quantité, les valeurs prises par $d_a(m)$ et $d_b(m)$ sont quantifiées optimalement en 32 valeurs à l'aide de l'algorithme de Lloyd-Max.

Puisque les flux audio sont segmentés à 2 niveaux, et les flux vidéo à 3 niveaux, 6 mesures de corrélation audiovisuelles peuvent être définies. Pour chacune d'entre elles, les deux mesures possibles (coefficient de corrélation de Pearson ou information mutuelle) sont envisagées. Nous avons choisi celle maximisant les performances de notre système dans l'expérience de recherche par le contenu décrite en 11.2.1.

Sont ainsi définies les 6 mesures suivantes :

$$C_{onsets/plans} = \rho(d_o, d_s) \tag{11.4}$$
$$C_{sections/plans} = \rho(d_c, d_s) \tag{11.5}$$
$$C_{onsets/sequences} = \rho(d_o, d_q) \tag{11.6}$$
$$C_{sections/sequences} = \rho(d_c, d_q) \tag{11.7}$$
$$C_{onsets/mouvement} = I(d_o, d_m) \tag{11.8}$$
$$C_{sections/mouvement} = \rho(d_c, d_m) \tag{11.9}$$

11.2 Applications

Nous détaillons dans cette section quelques applications des mesures de corrélation définies précédemment.

11.2.1 Requêtes de modalités croisées

Nous nous intéressons dans cette expérience au problème de la recherche, dans une base de données de fichiers musicaux, d'une musique d'accompagnement illustrant une séquence vidéo donnée. L'évaluation des résultats est difficile, et les quelques solutions proposées dans la littérature (voir section 9.1.2) se contentent généralement d'une évaluation subjective. Le protocole que nous proposons ici tente de fournir une mesure objective de la qualité des résultats.

Nous considérons en effet dans cette expérience une base de données de 100 clips vidéos (nommée par la suite `Video-100` et décrite dans l'annexe D.3). Ces vidéos proviennent de différentes sources : 25 clips de haute qualité esthétique tirés de [Jon03; Gon03; Div02], et 75 autres clips vidéos représentatifs de divers styles utilisés des années 80 à nos jours. Toutes les vidéos sont encodées au format MPEG-2, avec une résolution de 320×240 pixels à 25 trames/seconde. Les flux audio et vidéo de chacun des clips sont dissociés, pour former une base de données $(V_i)_{i\in\{1,\ldots,100\}}$ de séquences vidéo, et une base $(A_j)_{j\in\{1,\ldots,100\}}$ de signaux audio.

Étant donné un critère de corrélation C et une séquence vidéo V_i, nous nous proposons de rechercher les séquences A_j de la base de données l'accompagnant le mieux au sens du critère de corrélation considéré. Pour un seuil θ donné, nous définissons l'ensemble $R_i(\theta)$ des indices des signaux de musique les plus corrélés avec la requête vidéo V_i :

$$R_i(\theta) = \{j, C(A_j, V_i) > \theta\} \tag{11.10}$$

S'il est possible d'évaluer subjectivement la qualité de l'association entre V_i et les éléments de R_i, une mesure objective peut être obtenue en supposant que A_i, la musique originale pour laquelle a été réalisée la séquence vidéo V_i, doit se trouver dans R_i. Nous pouvons alors définir des mesures de rappel et de précision, par analogie avec l'évaluation des systèmes de recherche de documents :

$$\text{Precision}_i(\theta) = \begin{cases} \frac{1}{\#R_i(\theta)} & \text{si } i \in R_i \\ 0 & \text{si } i \notin R_i \end{cases} \tag{11.11}$$

$$\text{Rappel}_i(\theta) = \begin{cases} 1 & \text{si } i \in R_i \\ 0 & \text{si } i \notin R_i \end{cases} \tag{11.12}$$

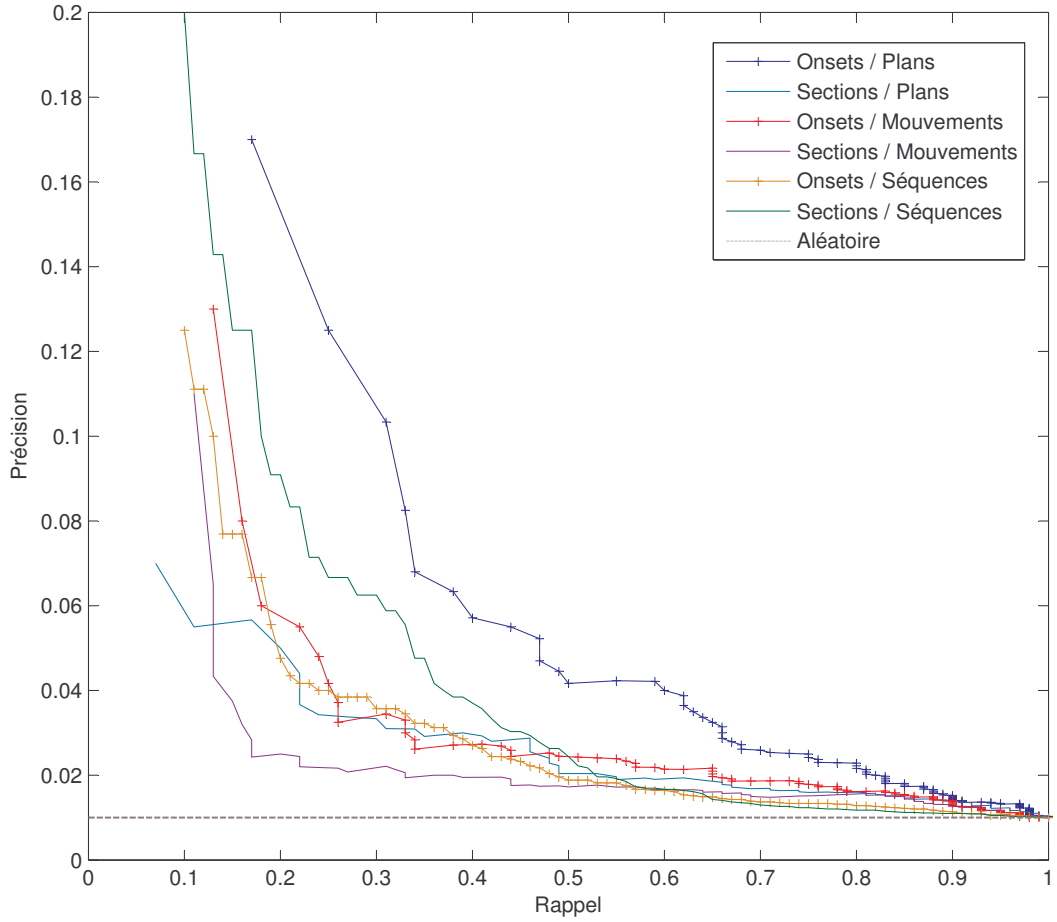


FIG. 11.1 – Courbes rappel/précision pour l'expérience de recherche d'accompagnement musical à partir d'une séquence vidéo

Pour une valeur de θ donnée, des scores globaux de rappel et de précision sont obtenus en moyennant les scores $\text{Precision}_i(\theta)$ et $\text{Rappel}_i(\theta)$, $i \in \{1, \ldots, 100\}$. Les courbes correspondantes sont données dans la figure 11.1.

La décroissance très rapide de ces courbes suggère que les mesures de corrélation présentées ne sont efficaces que sur une fraction de la base de données. Sur ce sous-ensemble, les meilleures performances sont obtenues en considérant la synchronie des changements de plans et des onsets – la structure de la matrice $a_{ij} = C(V_i, A_j)$ est montrée dans la figure 11.2. Une autre mesure produisant des résultats acceptables est la synchronie des changements de section dans la musique, et des changement de séquences vidéo. Parmi les corrélations utilisant une mesure de mouvement, la plus pertinente est la corrélation entre mouvements et onsets de notes. Les corrélations impliquant des éléments de niveaux très différents (mouvement et sections, séquences et onsets) sont parmi les moins performantes.

Si l'on se restreint au tiers de la base de données offrant les meilleurs résultats, avec la meilleure méthode (synchronie onsets/plans) l'accompagnement audio original se retrouve toujours parmi les 11 premiers résultats. Les seules expériences similaires effectuées dans la littérature sont celles de Yang et Brown [YB04] : pour une base de 100 fichiers audio et 5 séquences vidéo, l'accompagnement musical considéré comme le plus pertinent est classé en première position dans tous les cas. La nature des documents utilisés n'est cependant pas explicitée.

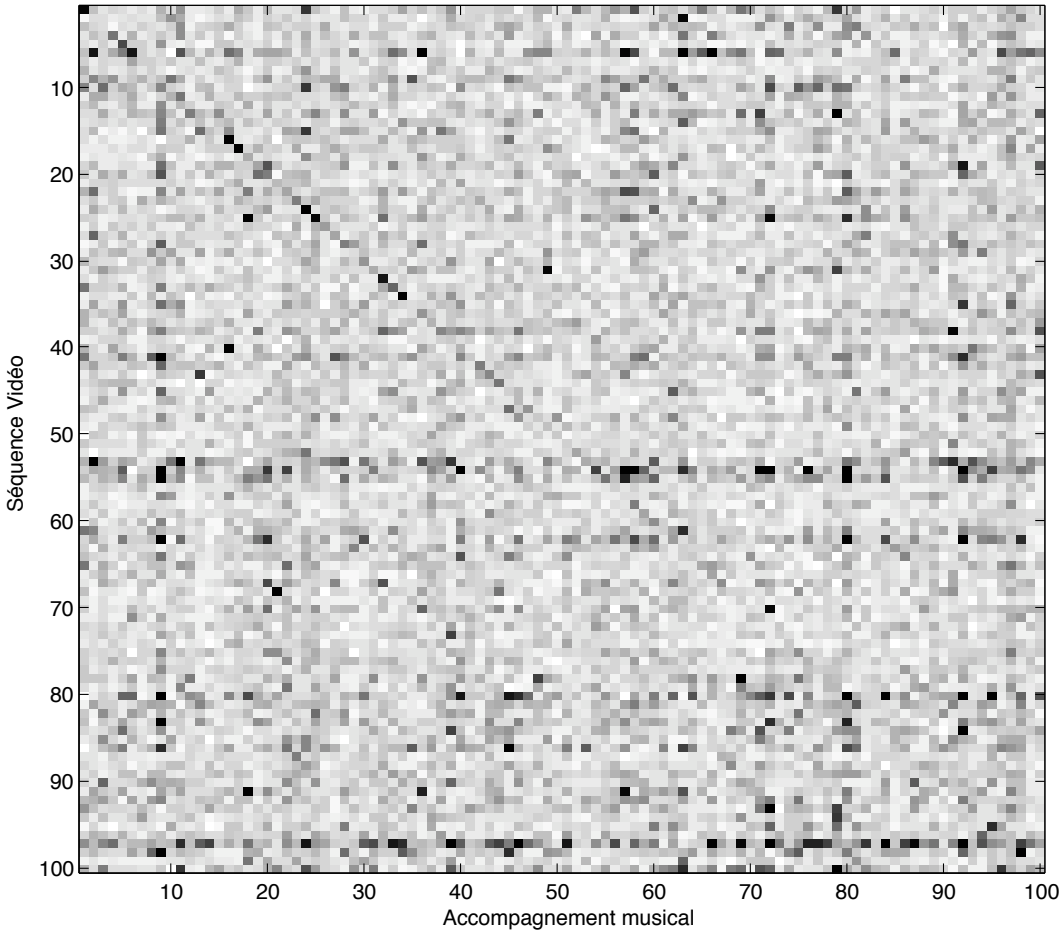


FIG. 11.2 – Matrice de synchronie entre les flux audio et vidéo, pour la mesure de synchronie onsets/plans

Nous soulignons également que la métrique utilisée ici est "sévère" au sens où tout accompagnement musical A_j est considéré comme incompatible avec la séquence vidéo V_i dès lors que $i \neq j$. Or, les paires (V_i, A_j), avec $i \neq j$ et $C(V_i, A_j)$ élevé obtenues lors des expériences ne correspondent pas toujours à des erreurs, et conduisent souvent à des résultats intéressants et étonnants. En particulier, si les oeuvres musicales A_k et A_j ont des tempi similaires, et si le clip vidéo réalisé pour A_k est édité au tempo, il apparaîtra comme synchrone avec A_j. Cela suggère une application intéressante et inattendue, la génération de *mashups* audio/vidéos, documents audiovisuels remplaçant la bande son d'un clip vidéo par une autre oeuvre musicale pour produire des effets intéressants ou humoristiques. Dans, ce cas, on considère pour une séquence vidéo V_i l'accompagnement A_{j^*} avec $j^* = \mathrm{argmax}_{j \neq i} C(V_i, A_j)$. De tels *mashups* incluent par exemple un morceau de rock progressif aux changements de sections trop graduels pour être détectés (Stereolab - *Jenny Ondioline*) sur les images d'un clip vidéo ne contenant qu'une seule séquence (Kylie Minogue - *Come Into My World*) ; ou plusieurs exemples de morceaux pop dont le tempo et la structure (y compris l'alternance des parties chantées et des soli de guitare) coïncident, se traduisant par une certaine interchangeabilité des images des musiciens.

Genre visuel	Rang moyen de l'original
Narration	23
Visuels abstraits	19
Danse	13
Musiciens	11
VJing et sampling vidéo	6

TAB. 11.1 – Influence du *genre visuel* sur les résultats de l'expérience de recherche de musique par la vidéo

11.2.2 Corrélations et *genre visuel*

Nous nous intéressons maintenant à l'apport de ces corrélations pour la tâche de classification des clips selon leur *genre visuel*. À cet effet, les clips de la base sont classés manuellement selon les 5 catégories suivantes (Quand plusieurs catégories peuvent être utilisées pour un même clip vidéo, la catégorie représentative du plus grand nombre de plans a été choisie) :

Narration Le clip vidéo possède une trame narrative et une chronologie – il serait ainsi possible de situer chacune des séquences de la vidéo sur un axe chronologique.

Musiciens Le clip vidéo montre essentiellement les musiciens jouant, sous forme de séquence vidéo ou d'animation.

Danse Le clip vidéo contient essentiellement des scènes de danse (danseurs, chanteur principal).

Visuels abstraits Le clip vidéo est une séquence de plans fixes ou de séquences vidéos ne décrivant aucune activité liée au jeu ou à l'écoute de musique. L'association avec la musique se fait à un niveau sémantique supérieur (lien avec l'atmosphère du morceau ou ses paroles).

VJing, sampling vidéo Le clip vidéo est construit à partir de séquences courtes éditées et déclenchées pour s'accorder au rythme et aux *samples* utilisés dans la musique.

Nous cherchons ici à identifier pour quelles catégories les corrélations définies dans ce chapitre sont significatives. Pour chaque flux vidéo V_i, les flux audio A_j sont classés selon leur corrélation avec V_i. Soit r_i le rang assigné au flux audio original qui accompagnait la vidéo V_i. De faibles valeurs de r_i indiquent que la relation entre la musique et l'image est suffisamment forte pour permettre la sélection de la musique à partir de l'image. La table 11.1 liste la moyenne de r_i pour chacun des 5 genres définis.

Nous observons que les mesures de corrélation présentées sont les plus efficaces pour les vidéos de la catégorie VJing/sampling vidéo. Plus généralement, les corrélations offrent de bonnes performances pour les vidéos montrant des activités reliées à la musique (jeu ou danse) ; tandis que dans les vidéos narratives ou utilisant des visuels abstraits, les images et la musique ne peuvent être mis en correspondance que sur des critères sémantiques de plus haut niveau, qui échappent à nos mesures de corrélation.

11.2.3 Resynchronisation audio/vidéo

Pour un document musical audiovisuel (V_i, A_i) donné, nous considérons la somme $S_i(\tau)$ des mesures de corrélation données en 11.1.2, évaluées entre le flux vidéo et le flux audio retardé d'un délai τ. Un exemple de courbe S_i est représenté dans la figure 11.3 pour un clip vidéo montrant des activités musicales (jeu et danse). Le pic observé en $\tau = 0$ suggère de maximiser la somme des mesures de corrélation, ou la plus significative d'entre elles, pour resynchroniser des flux audio et vidéo.

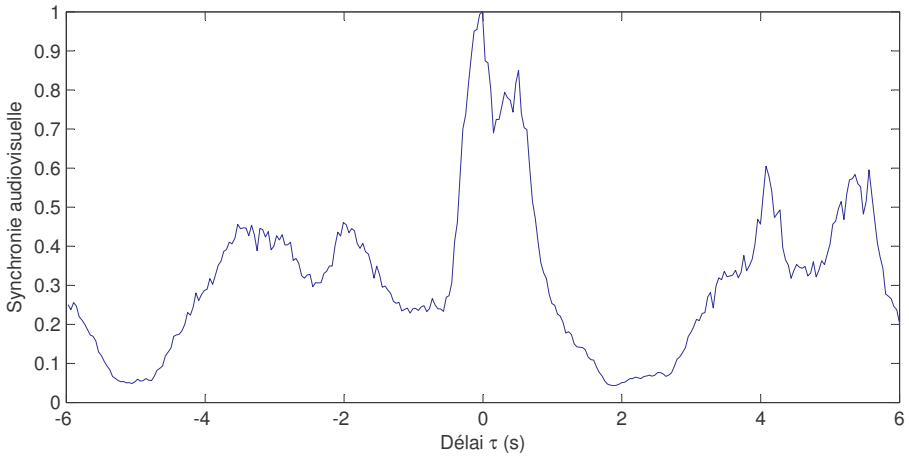


FIG. 11.3 – Influence du retard entre la musique et l'image sur la somme des mesures de corrélation (Herbie Hancock - *Rockit*)

11.3 Conclusion

Nous avons présenté dans ce chapitre des mesures de corrélation (plus précisément, de synchronie des changements) entre les flux audio et vidéo, calculées à partir des fonctions de détection obtenues au chapitre précédent pour la segmentation. Pour plus de robustesse, nous suggérons d'utiliser directement les fonctions de détection, sans seuillage préalable, et de leur appliquer une DTW pour compenser de légers décalages entre phénomènes qui sont pourtant perçus comme simultanés.

Trois applications ont été par la suite présentées. Tout d'abord la recherche d'un signal de musique accompagnant le mieux une séquence vidéo donnée. Nous observons que sur une fraction (environ un tiers) de la base de clips vidéos considérés, les mesures de corrélation peuvent efficacement être utilisées pour retrouver l'oeuvre musicale originale pour laquelle la séquence vidéo a été réalisée. Les corrélations les plus pertinentes sont alors la synchronie des changements de plan et des onsets de notes ; et la synchronie des changements de séquence et des changements de section dans la musique. Sur le reste de la base de données, les mesures de corrélation sont globalement peu efficaces.

Ce manque d'efficacité rend-il les mesures proposées inutiles ? Une deuxième expérience montre que les performances de ces mesures sont en fait dépendantes du *genre visuel* : les mesures de corrélation définies sont efficaces pour les clips vidéo montrant des activités musicales (danse, musiciens). Cela suggère d'utiliser une telle mesure de performance comme attribut dans des systèmes de classification du genre musical. Faute de temps, l'étude de tels systèmes n'a malheureusement pas pu être réalisée dans le cadre de cette thèse.

Enfin, sur les clips vidéo montrant des activités musicales, nous avons observé que les mesures de corrélation décroissent rapidement lorsqu'on désynchronise les flux audio et vidéo, suggérant la maximisation de ces mesures pour la resynchronisation des contenus multimédia.

Publications liées à ce chapitre

Les résultats présentés dans ce chapitre sont décrits dans [GER07]. Ils complètent les résultats d'expériences préliminaires réalisées sur une base de données plus réduite, publiés dans [GR06a].

Conclusion de la partie III

Nous avons dans cette partie tenté de combler le fossé entre les systèmes d'analyse de scènes musicales audiovisuelles qui ne peuvent s'appliquer qu'à des séquences vidéos enregistrées dans des circonstances bien contrôlées et qui en extraient une information spécifique, et les systèmes génériques d'indexation de séquences vidéo, qui ne tirent pas parti de leur contenu musical.

Un exemple de problème intermédiaire, impliquant à la fois analyse musicale et analyse de séquences vidéo, est celui de la découverte des relations liant la musique à l'image. Nous avons montré au chapitre 9 que certaines relations d'association entre musique et image se manifestent par des structures identiques dans le contenu musical et dans la séquence d'images. Nous nous sommes dès lors proposés d'extraire de telles structures et de les comparer. Différentes méthodes de segmentation ont été discutées dans le chapitre 10, en particulier pour la segmentation en sections des signaux de musique. Pour résoudre ce problème, nous avons privilégié divers algorithmes de détection de nouveauté à noyaux, certains connus et conçus spécifiquement pour cette tâche, d'autres (distances probabilistes) issus d'autres domaines. Les performances offertes par ces méthodes se sont démarquées nettement d'une approche classique – le critère d'information bayésien. Nous pensons que de telles méthodes sont particulièrement pertinentes pour les tâches de segmentation, où les données disponibles sont peu nombreuses, mais de grande dimensionnalité. Le problème de la segmentation en plans et séquences de clips vidéo a également été abordé, mais n'a pas pu faire l'objet d'une évaluation.

Nous avons par la suite défini des mesures de corrélation mesurant la synchronie des changements à diverses échelles (notes/sections pour la musique ; mouvements/plans/séquences pour la vidéo) entre les flux audio et vidéo. Pour plus de robustesse, ces mesures sont directement calculées sur les fonctions de détection produites par les modules de segmentation. Plusieurs applications possibles ont été proposées pour ces mesures. Nous avons tout d'abord évalué leur utilité pour une tâche de recherche d'accompagnement musical à partir d'une séquence vidéo. Un protocole expérimental original employant des clips vidéos est utilisé. Il permet une mesure objective, certes sévère, de la pertinence des accompagnements musicaux retrouvés. Les résultats montrent la validité des mesures proposées pour une fraction (environ 1/3) de la base de données. L'analyse des erreurs commises par le système suggère également une application inattendue : la génération de *mashups* audio/vidéos, identifiant des contenus audiovisuels aux structures similaires dont les bandes sonores peuvent être échangées. Au delà, les mesures proposées sont incapables de saisir les relations d'association purement sémantiques entre un contenu audio et vidéo, par exemple la relation entre les paroles d'une chanson et sa narration visuelle. Cet échec suggère d'utiliser les mesures de corrélation définies pour discriminer les clips vidéos illustrant des activités musicales (danse, jeu des instruments) d'autres *genres visuels*. Dans notre base de données, nous observons que le rang du document original obtenu dans l'expérience de requête de musique par la vidéo est dépendant du *genre visuel*, suggérant son utilité comme attribut dans un système de classification. Une autre application envisageable est d'utiliser ces mesures à des fins de resynchronisation.

Nous espérons que cette première proposition incite à explorer le terrain quasi-vierge entre les domaines du *Music Information Retrieval* et de l'indexation vidéo.

CHAPITRE **12**

Perspectives

Pour faire suite aux bilans proposés en guise de conclusion de chacune des parties de ce manuscrit, nous livrons dans ce dernier chapitre quelques directions de recherche pour prolonger nos travaux, pour chacun des différents thèmes abordés.

12.1 Analyse des signaux percussifs

12.1.1 Transcription de la piste de batterie

À court terme Nous avons présenté au chapitre 5 de nouvelles méthodes de séparation de sources pour la batterie, certaines ayant été développées ultérieurement à nos expériences de transcription réalisées au chapitre 4. Il serait souhaitable d'évaluer les gains de performance obtenus en utilisant ces méthodes de séparation en lieu et place du pré-traitement décrit au chapitre 3.

Nous avons souligné au chapitre 4 la nécessité de disposer de meilleurs attributs pour discriminer les différentes frappes de la batterie – nous avons en effet constaté qu'en présence d'un accompagnement et pour les attributs que nous avons définis, certaines des classes n'étaient pas séparables. Nous avons vu également que les attributs les plus discriminants étaient des attributs spécifiques au problème (énergie en sortie de bancs de filtres adaptés). D'autres attributs spécifiques pourraient être calculés en considérant les coefficients produits par une décomposition non-négative sur un dictionnaire de densités spectrales de puissance comme utilisé en 5.3. De tels attributs permettraient de réconcilier les approches *Séparer et Détecter* et *Segmenter et Reconnaître* : plutôt que de simplement détecter des pics dans les enveloppes temporelles extraites par NMF (ou ISA), on pourrait utiliser ces enveloppes à la fois pour détecter les onsets, et pour en extraire des attributs utilisés en classification.

Enfin, nous n'avons pu comparer nos résultats qu'à un nombre réduit de méthodes adverses. Nous espérons que la diffusion publique de la base ENST-drums permettra dans un avenir proche de disposer de mesures de performances comparables pour tous les systèmes de transcription de la piste de batterie proposés dans la littérature.

À moyen terme Nous avons évoqué au chapitre 5 la dualité entre le problème de la séparation et de la transcription de la piste de batterie – l'un étant plus aisément résolu connaissant une solution, même approximative, de l'autre. Une voie de recherche intéressante serait d'évaluer des méthodes itératives, réalisant séquentiellement transcription et séparation. Un effort particulier devra être mené pour démontrer la convergence d'une telle démarche[1]

[1] L'accumulation d'erreurs pourrait en effet faire converger le système vers un résultat tel que, sur la durée d'une séquence, seulement un type de frappe est transcrit – par exemple, pour la caisse claire, les cross-sticks sont transcrits mais pas les frappes normales. Nous avons observé de tels comportement après plusieurs itérations d'ADAMAST [YGO04a].

Deux difficultés ont été rencontrées au chapitre 4, lors de la mise en oeuvre de méthodes supervisées et non-supervisées pour la correction des erreurs de transcription. La première était le manque de fiabilité des probabilités a posteriori fournies par les classifieurs. Ce problème ne pourra être résolu qu'en utilisant des attributs plus robustes et discriminants. Le deuxième problème est le coût en calculs prohibitif de la méthode de réduction de la complexité présentée en 4.5.3. Bien qu'elle semble prometteuse, cette approche ne pourra réellement porter ses fruits que si des méthodes d'optimisation plus efficaces que les algorithmes évolutionnaires peuvent être mises en oeuvre – que ces méthodes soient exactes ou qu'il ne s'agisse que d'heuristiques. Nous pensons qu'une solution à ce problème pourrait avoir d'autres applications en communications (comment modifier un message le plus légèrement possible pour en faciliter le codage ?). Nous sommes cependant pessimistes quant à l'existence d'une solution de complexité polynomiale.

À long terme Deux outils nous ont fait défaut dans nos travaux, et leur existence nous aurait fait suivre une toute autre approche. Le premier est une représentation des signaux permettant l'estimation jointe de la transcription et du timbre de chacun des instruments – ou, de façon équivalente, du signal séparé de la piste de batterie. Une telle représentation permettrait d'éviter l'estimation séquentielle d'un élément par rapport à l'autre (telle qu'elle est réalisée dans des méthodes comme ADAMAST [YGO04a], ou telle que nous l'avons suggéré). La NMF ou l'ISA ne sont des solutions que partiellement satisfaisantes, car la représentation des sources par des profils spectraux et des enveloppes temporelles ne permet pas une resynthèse de signaux de qualité, et requiert plusieurs heuristiques de sélection des composantes et de détection des pics pour produire une partition.

Le second outil qui nous a fait défaut est un modèle génératif des signaux produits par l'ensemble des instruments de la batterie, offrant un bon compromis entre expressivité et solvabilité. Un tel modèle, utilisé en conjonction avec des modèles des signaux produits par les autres instruments (de tels modèles existent déjà, voir par exemple [DGI06] ou [VR04b]) permettraient de réaliser la séparation et la transcription par estimation de ses paramètres à partir du signal observé. Nous pensons cependant que la présence de composantes à la fois déterministes et stochastiques dans les signaux des instruments à percussion, et leur instationnarité, rend la formulation d'un tel modèle difficile. Une première simplification consisterait à modéliser séparément les composantes stochastiques et déterministes des signaux, et à réaliser l'estimation sur ces deux modèles, isolément.

12.1.2 Séparation de la piste de batterie

À court terme L'évaluation que nous avons menée dans cette thèse pourra être approfondie. Le suivi d'un protocole de validation croisé rigoureux permettra de s'assurer que nos modèles n'ont pas réalisé de surapprentissage.

Un effort tout particulier devra être apporté quant à la définition de meilleures métriques pour la séparation de sources percussives – nous avons déjà livré quelques pistes en 5.4.2 : rapport masque à interférence/distorsion/bruit, mesures distinctes sur les transitoires et les parties stables du signal ou critères de percussivité. Dans l'attente de meilleures métriques, les tests d'écoute restent la solution la plus fiable pour évaluer nos méthodes – de tels tests devront ainsi être menés.

À moyen terme Quelques problèmes relatifs à la séparation harmonique/bruit, qui est à la base de plusieurs méthodes décrites dans ce manuscrit, restent irrésolus. En particulier, nous avons du avoir recours à un ajustement manuel de l'ordre (nombre de sinusoïdes à extraire) dans chacune des bandes, les méthodes d'estimation de l'ordre n'étant pas adaptées aux signaux non-stationnaires. La pré-segmentation du signal, avant sa décomposition, pourrait apporter des réponses : elle permettrait non seulement d'utiliser des critères d'ordre sur des segments homogènes, mais aussi de gagner en précision dans le suivi de l'espace signal, en le ré-initialisant par une EVD complète après chaque frontière de segment. Cependant, cela requiert une pré-segmentation du signal, et nous avons vu que les méthodes les plus robustes de détection d'onsets se basent... sur une décomposition harmonique/bruit. Cela suggère encore une fois une approche itérative : segmentation grossière du signal, séparation des composantes harmonique/bruit sur les segments homogènes générés, et utilisation

de cette séparation pour la suite des traitements (détection d'onsets, séparation), et pour raffiner la segmentation initiale.

Enfin, les méthodes de séparation de sources présentées au chapitre 5 pourraient être améliorées de diverses façons. Nous avons déjà évoqué dans la conclusion de ce chapitre l'intérêt éventuel d'une procédure d'adaptation, dans le cadre du filtrage pseudo-Wiener. D'autres améliorations consisteraient à traiter de façon distincte les parties stochastiques et harmoniques (ce que nous faisons déjà, d'une certaine façon, en enrichissant le dictionnaire de d.s.p de la batterie avec l'estimation de la composante stochastique du signal), ou à imposer des contraintes temporelles dans la décomposition – en disposant de sous-dictionnaires de d.s.p séparément appris sur les attaques et les parties entretenues des signaux, avec une contrainte de parcimonie imposant à un seul de ces sous-dictionnaires d'être utilisé.

12.1.3 Application de l'analyse de la piste de batterie dans les signaux de musique

À court terme Nous regrettons que peu d'efforts aient été faits pour intégrer des systèmes de transcription ou de séparation de piste de batterie dans des applications logicielles utiles au musicien – à l'exception de l'outil de remixage proposé par Yoshii et al. dans [YGO05], et de notre moteur de recherche de boucles de batterie décrit dans [GR05b; GR05e]. Des applications intéressantes à développer autour de nos travaux incluraient, par exemple, un système de recherche par le contenu de boucles rythmiques, capable de transcrire des boucles où jouent la basse, et/ou d'autres instruments mélodiques ; ou un système de recherche d'oeuvres musicales par le rythme.

À moyen terme Le développement d'un module de remixage de la batterie pouvant être intégré à un lecteur de musique demandera sans doute plus d'efforts, puisqu'il faudra résoudre le problème de la non-causalité et du coût en calculs des traitements décrits. Certains d'entre eux (séparation harmonique/bruit, détection d'onsets) causent une latence modérée, de l'ordre de quelques centaines de millisecondes. Par contre, les méthodes utilisant la NMF ou l'ISA demandent que l'intégralité du signal à traiter soit connu à l'avance. Une direction de recherche intéressante vers la (quasi) causalité consisterait à étudier des formes adaptatives des algorithmes de NMF.

À long terme Nous espérons que l'amélioration des performances des systèmes de transcription de la piste de batterie permettra, à long terme, leur intégration dans les logiciels d'édition audio, afin d'offrir des moyens intuitifs et puissants d'éditer des enregistrements musicaux "sémantiquement", et non plus comme de simples signaux.

12.2 Analyse audiovisuelle du jeu de la batterie

À court terme Nous avons proposé dans la section 8.3 diverses variantes du système de transcription audiovisuelle de séquences de batterie, et discuté leur applicabilité à différents scénarios d'usage. Une implémentation de toutes ces variantes devra être réalisée, et une évaluation rigoureuse, dans toutes les combinaisons de conditions énumérées, devra alors être conduite. Les suggestions données dans la table 8.4 pourront ainsi être infirmées ou confirmées expérimentalement. Un effort particulier devra être fait sur le choix des classifieurs pour les méthodes utilisant des classifieurs locaux : quelles classifieurs sont les plus efficaces lorsque les ensembles d'apprentissage sont réduits et de grande dimensionnalité ?

À moyen terme La procédure itérative de transcription décrite en 8.3.2 devra également être évaluée. Il serait en particulier intéressant d'étudier la (non-)convergence de cette procédure : les erreurs de transcription ou de segmentation tendent-elles à se propager, ou observe-t-on la convergence ? Nous suggérons qu'une telle méthode n'est réellement efficace que si les classifieurs utilisés

pour l'initialiser sont suffisamment robustes – cette méthode ne pourrait donc porter ses fruits qu'à long terme.

Nous avons vu que sous réserve de l'intervention d'un opérateur humain, la tâche de segmentation et d'association régions/instruments est facilitée. Ce scénario est plausible pour les applications à l'interaction musicien/machine ou l'apprentissage. Pourrait-on développer des systèmes commerciaux de capture du jeu et d'aide à l'apprentissage ? Le problème des coûts en calculs des méthodes utilisées n'a pas été abordé dans ce manuscrit – nous sommes très loin du temps réel. Néanmoins, il serait nécessaire de cerner d'abord les besoins des utilisateurs de tels systèmes : au cours d'une enquête informelle, des musiciens débutants ont évoqué l'intérêt qu'ils auraient à utiliser un système leur permettant de filmer leur jeu, et de le visualiser frappe par frappe. Un tel système est possible, en dirigeant la lecture de la vidéo par le résultat d'une segmentation audio.

Une autre application connexe que nous n'avons pas évoquée dans cette thèse est l'opération inverse de la transcription audiovisuelle : la synthèse de séquences vidéo de jeu de batterie, à des fins de visualisation, à partir d'un signal audio. Une telle synthèse pourrait être effectuée en transcrivant la séquence que l'on souhaite illustrer, à l'aide du système de transcription audio décrit au chapitre 4 par exemple, et en assemblant des segments d'une séquence vidéo préalablement indexée (par le système de transcription audiovisuelle décrit au chapitre 8). Les critères utilisés pour la recherche des segments pourraient être la continuité avec les segments voisins (continuité des images et des vecteurs de mouvement), et le contenu musical (frappes jouées dans le segment). La recherche de l'assemblage optimal pouvant se faire par programmation dynamique, par analogie avec les systèmes de synthèse concaténative de la parole.

À long terme Dans l'idéal, la transcription audiovisuelle devrait pouvoir être effectuée sur un document audiovisuel musical quelconque. Cela ouvre donc de nouveaux problèmes à résoudre. Tout d'abord, des méthodes de segmentation insensibles à la couleur devront être développées. De telles méthodes pourraient utiliser des attributs de texture (non considérés dans nos travaux, sauf par le biais du critère de variance), et utiliser un modèle a priori de la disposition des éléments de la batterie. Le problème du suivi des régions segmentées lorsque la caméra est en mouvement devra être résolu. Une piste intéressante consiste à utiliser des contours actifs (*snakes*), initialisés sur une segmentation de la première trame, ou à apparier les segmentations produites pour chaque trame. Il serait également possible de mettre en correspondance les trames successives de la séquence, et compenser ainsi le mouvement de la caméra en formant une séquence de trames déformées, montrant la scène sous un angle fixe – dans ce cas, toutes les méthodes présentées dans ce manuscrit peuvent s'appliquer.

Le problème du suivi des baguettes devra lui aussi être résolu par de nouvelles méthodes : L'échec des critères géométriques utilisés dans nos expériences préliminaires, et la faible robustesse du critère arrière-plan/avant-plan utilisé suggèrent des approches très différentes. Une solution non considérée dans cette thèse pourrait s'avérer fructueuse : elle consisterait à définir un modèle paramétrique du corps du batteur. Ce modèle permettrait le suivi des mouvements du musicien, et les paramètres extraits seraient génériques et indépendants du batteur, une étape supplémentaire en direction d'un modèle générique du jeu de l'instrument – même si nous pensons que formuler un tel modèle en des termes autres que ceux de paramètres de haut niveau est difficile.

12.3 Analyse de documents audiovisuels musicaux

À court terme Faute de temps, nous n'avons pu évaluer les méthodes de segmentation vidéo proposées. Un premier effort d'annotation devra donc être mené pour permettre cette évaluation. Il serait également intéressant de conduire des tests subjectifs pour évaluer la qualité des *mashups* produits lors des requêtes d'enregistrements musicaux à partir de séquences vidéo ; en les comparant en particulier à des paires musique/vidéo formées aléatoirement.

Nous n'avons pas non plus pu comparer les techniques de segmentation de signaux de musique proposées à d'autres systèmes décrits dans la littérature. L'intérêt des méthodes que nous avons employées semble cependant avéré, puisqu'elles ont été appliquées avec succès au problème connexe

de la transcription de flux radiophoniques dans [RRE07]. Certaines des méthodes présentées pourraient être réconciliées : on pourrait en effet définir un critère semblable au BIC utilisant les estimées régularisées des matrices de covariances dans un RKHS comme en 10.1.3.3.

Terminons enfin par une application non évoquée dans ce manuscrit : la recherche d'une séquence vidéo illustrant au mieux une oeuvre musicale. Elle peut être effectuée par les mêmes méthodes, et évaluée par le même protocole que sa réciproque traitée en 11.2.1.

À moyen terme Nous avons opposé dans la section 9.2 l'analyse des associations image/musique utilisant un modèle esthétique explicite, et notre approche implicite basée sur la synchronie des changements, condition nécessaire à la perception d'une relation d'association. Une voie intermédiaire consisterait à extraire indépendamment un ensemble d'attributs des flux audio et vidéo, en considérant à la fois des attributs de bas niveau (intensité sonore, luminosité, teinte) et de haut niveau (instrumentation, présence ou absence de voix chantée ou tempo pour la musique ; présence de visage, classification scène intérieure/extérieure ou détection d'objets et de concepts pour la vidéo). D'une part, les attributs de haut niveau extraits, aussi bien à partir de la vidéo que de la musique, fourniraient un niveau supplémentaire de segmentation "sémantique" – en détectant des changements dans les concepts ou mots-clés extraits, plutôt qu'à partir d'attributs de bas niveau. D'autre part, les corrélations entre toutes les paires d'attributs audio/vidéo extraites pourraient à la fois servir les mêmes buts que les mesures de synchronie des changements introduites dans ce manuscrit ; mais auraient en plus, par le biais de méthodes comme l'analyse des corrélations canoniques, un pouvoir explicatif permettant de répondre à des questions du type "Qui illustre quoi dans ce clip vidéo ?", "Quel personnage est le chanteur ?" ou "À quels concepts est associé le refrain ?". De plus, de tels modèles d'association pourraient être appris sur une base de données de clips vidéos, et être utilisés dans des applications de recherche d'accompagnement musical ou d'illustration vidéo.

Un problème évoqué au chapitre 11 mais non traité est celui de la classification automatique d'un clip vidéo selon son *genre visuel*. Si nous pouvons d'ores et déjà affirmer que les corrélations (ou plutôt une mesure de leur pertinence pour une tâche de recherche de musique par l'image, ou d'image par la musique) sont des attributs intéressants pour une telle classification, nous n'avons pas poursuivi cette voie. D'autres attributs, comme évoqué plus haut (détection de concept, détection de visage...) seraient à considérer, et des méthodes d'apprentissage statistique pourraient alors être mises en oeuvre.

À long terme Le problème de l'analyse des relations musique/image a été considéré parce qu'il est représentatif du type d'applications qui peuvent être développées en hybridant systèmes d'indexation vidéo et d'indexation musicale. Cependant, d'autres applications hybrides pourraient également être considérées : localisation des musiciens dans une scène (quelques éléments ont été donnés dans la section 8.2.1.2), identification automatique de l'artiste en utilisant les modalités audio et vidéo, ou comme évoqué plus haut classification d'un clip vidéo selon son *genre visuel*.

Quatrième partie

Annexes - Boîte à outils

ANNEXE **A**

Palette d'attributs

Nous détaillons dans cette annexe différents paramètres de signaux audio, utilisés à diverses reprises dans ce manuscrit. Nous appellerons $x(n)$, $n \in \{0, \ldots, N-1\}$ le signal observé sur la fenêtre considérée, et $X(k)$ sa transformée de Fourier discrète sur $2K = 16384$ points obtenue après fenêtrage de $x(n)$ par une fenêtre de Hann, et extension par des zéros.

A.1 Paramètres de distribution de l'énergie

Puissance totale du signal Définie comme le logarithme de la racine carrée de la valeur moyenne du carré du signal sur l'intégralité de la fenêtre d'observation (lRMS).

$$lRMS_t = 20 \log_{10} \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x(n)^2} \tag{A.1}$$

Puissance du signal en sortie de filtres adaptés Tanghe et al. décrivent dans [TDB05] trois filtres adaptés au contenu spectral des signaux de grosse caisse, caisse claire, et hi-hat. Ce sont des filtres de Butterworth passe-bande, dont les bandes passantes sont respectivement centrées en 50 Hz, 200 Hz, et 10 kHz. Leurs réponses en fréquence sont données dans la figure A.1.

Si l'on note h_{bd}, h_{sd}, h_{hh} leurs réponses impulsionnelles (infinies), les attributs calculés sont alors :

$$lRMS_{bd} = 20 \log_{10} \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x * h_{bd})(n)^2} \tag{A.2}$$

$$lRMS_{sd} = 20 \log_{10} \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x * h_{sd})(n)^2} \tag{A.3}$$

$$lRMS_{hh} = 20 \log_{10} \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x * h_{hh})(n)^2} \tag{A.4}$$

On définit également les attributs suivants, mesurant la proportion de la puissance totale en sortie de chacun des filtres, ainsi que des rapports de puissance :

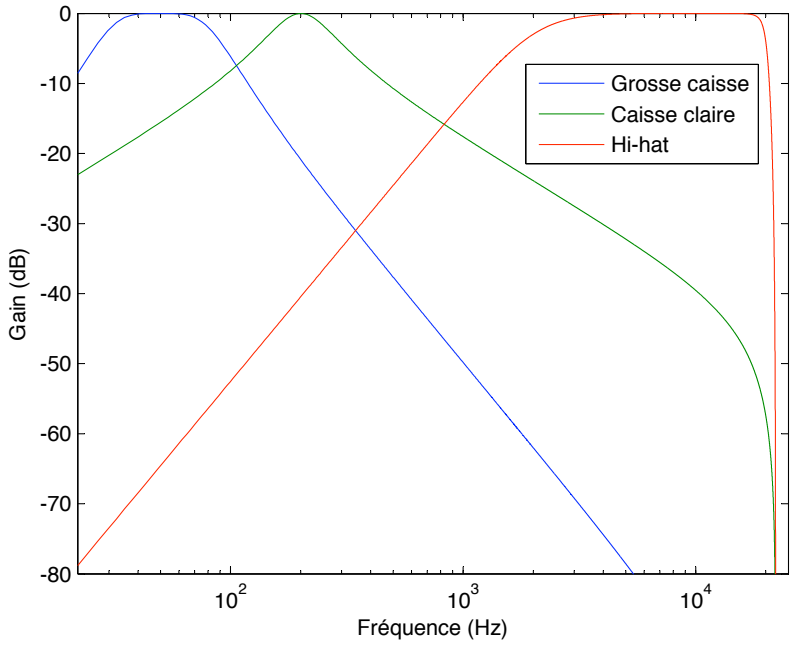


FIG. A.1 – Filtres passe-bande adaptés définis par Tanghe et al

$$lRMSrel_{bd} = lRMS_{bd} - lRMS \quad (A.5)$$
$$lRMSrel_{sd} = lRMS_{sd} - lRMS \quad (A.6)$$
$$lRMSrel_{hh} = lRMS_{hh} - lRMS \quad (A.7)$$
$$lRMSrel_{bd,sd} = lRMS_{bd} - lRMS_{sd} \quad (A.8)$$
$$lRMSrel_{sd,hh} = lRMS_{sd} - lRMS_{hh} \quad (A.9)$$
$$lRMSrel_{hh,bd} = lRMS_{hh} - lRMS_{bd} \quad (A.10)$$

Puissance du signal en sortie d'une décomposition adaptée Dans [GR04], nous décrivons un découpage empirique du spectre en bandes de fréquences (donné dans la table A.1), chaque instrument de la batterie occupant typiquement une de ces bandes. 8 attributs sont définis à partir de la puissance dans chacune de ces bandes :

$$lRMS_{gband,i} = 10\log_{10}\frac{1}{N}\sum_{k=k^i_{min}}^{k^i_{max}} |X(k)|^2 \quad (A.11)$$

Où k^i_{min} et k^i_{max} sont respectivement les limites inférieures et supérieures de la i-ième bande.

Rapports d'énergie entre octaves adjacentes Ces attributs, non spécifiques à la batterie, ont été introduits par Essid et al. dans [ERD06b] sous le nom d'Octave Band Signal Intensity Ratios (OBSIR). Leur avantage est de permettre la description approximative de la distribution des harmoniques des signaux de musique en s'affranchissant de l'étape d'estimation de la fréquence fondamentale. Ils consistent à analyser le signal par un banc de filtres en bandes d'octaves (les réponses en fréquence des 8 filtres sont données dans la figure A.2), et à mesurer le rapport d'énergie entre

Frontières de la bande (Hz)	Instrument
$[10, 70]$	Grosse caisse
$[70, 130]$	Tom basse, certaines grosses caisses
$[130, 300]$	Tom medium, caisse claire
$[300, 800]$	Tom alto, timbre de la caisse claire
$[800, 1500]$	Claps, cloches, timbre de la caisse claire
$[1500, 5000]$	Cymbales, timbre de la caisse claire
$[5000, 10000]$	Cymbales, timbre de la caisse claire
$[10000, 15000]$	Cymbales, timbre de la caisse claire

TAB. A.1 – Découpage empirique du spectre et éléments de la batterie associés

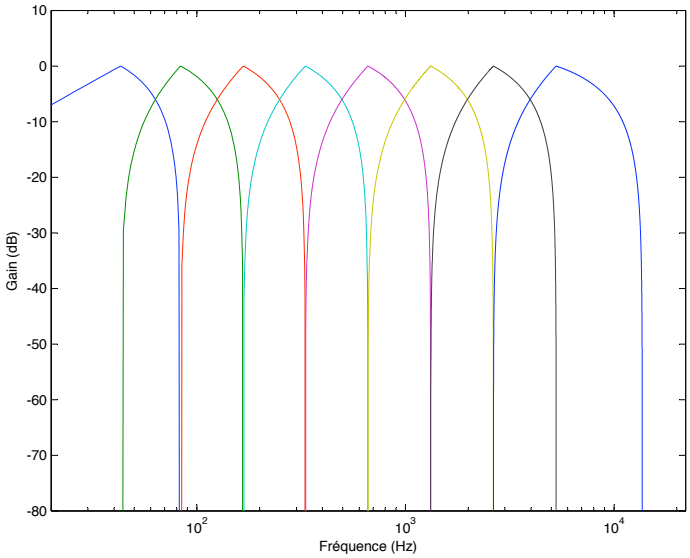


FIG. A.2 – Banc de filtre en bandes d'octave utilisé pour le calcul des attributs OBSIR

deux bandes adjacentes :

$$OBSI_i = 10 \log_{10} \frac{1}{N} \sum_{k=k'^i_{min}}^{k'^i_{max}} |X(k)|^2 \tag{A.12}$$

$$OBSIR_i = OBSI_{i+1} - OBSI_i \tag{A.13}$$

Énergie en sortie d'un banc de filtres en demi-tons Ces attributs[1] mesurent l'énergie E_k^t dans chacune des bandes d'un banc de filtres à 12 voies, chaque filtre ayant une réponse fréquentielle $H_k(f), k \in \{0, \ldots, 11\}$ définie par :

[1] De tels attributs ne sont pas utilisés pour la reconnaissance de frappes de batterie, mais pour la segmentation de documents musicaux.

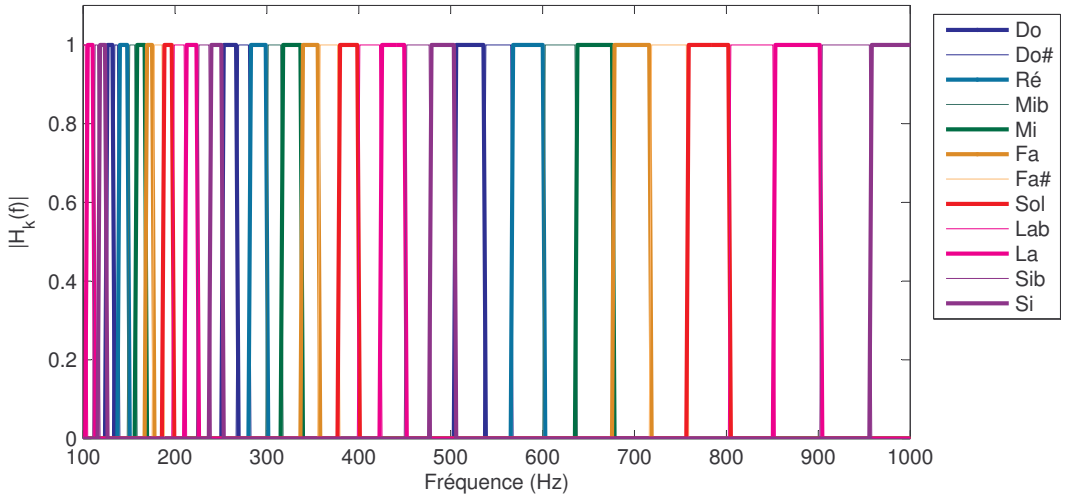


FIG. A.3 – Banc de filtres en demi-tons

$$H_k(f) = \begin{cases} 1 & \text{si } k \equiv \left[\frac{12}{\log 2} \log \frac{f}{f_C}\right] \mod 12 \\ 0 & \text{sinon} \end{cases} \tag{A.14}$$

Où f_C désigne la fréquence de la note Do (l'octave n'importe pas). Le filtre $H_k(f)$ est ainsi "accordé" sur le k-ième demi-ton de la gamme tempérée (voir figure A.3). De tels filtres étant difficiles à synthétiser, le calcul de l'énergie est directement effectué dans le domaine fréquentiel. Ils ont été introduits dans [BW01] pour des applications de détection de refrain.

A.2 Paramètres cepstraux

Coefficients cepstraux en Échelle de Mel (MFCC) Les coefficients cepstraux, obtenus par transformée de Fourier inverse du logarithme du module de la transformée de Fourier, sont traditionnellement utilisés en traitement de la parole, puisqu'ils permettent une séparation aisée des contributions des cordes vocales (source/excitation représentée sous forme d'un peigne dans les coefficients élevés) et du conduit vocal (filtre représenté dans les quelques premiers coefficients). Si le modèle source/filtre utilisé en traitement de la parole ne s'applique pas aux signaux de musique polyphoniques, les coefficients cepstraux gardent cependant un pouvoir descriptif intéressant pour les signaux de musique. En effet, on peut considérer qu'ils fournissent une version lissée et compacte de la densité spectrale de puissance, mesurant la distribution globale de l'énergie. Par rapport aux coefficients cepstraux classiques les MFCC emploient une échelle de fréquence perceptuelle non-linéaire, l'échelle des fréquences Mel, ou une de ses approximations. Cette échelle permet de définir un nombre réduit B de bandes critiques (en général plusieurs dizaines de bandes). L'échelle de fréquence que nous avons utilisée, qui est celle de l'`Auditory toolbox`[2], compte 13 bandes linéairement espacées de 0 à 1000 Hz, et 27 bandes logarithmiquement espacées au delà, soit $B = 40$. Les MFCC sont calculés en intégrant l'énergie dans chacune de ces bandes, par sommation du module du spectre $|X(k)|$ multiplié par des fenêtres de pondération triangulaires $t_i(k)$

[2]Les différents choix d'échelles de fréquence propres à chaque implémentation et boîte à outils logicielle n'ont que peu d'influence sur la valeur des MFCC principaux, se référer à [SSLS06] pour une étude de l'influence de l'implémentation sur les coefficients calculés.

centrées sur chaque frontière de bande i, produisant B coefficients e_i. Les coefficients MFCC sont ensuite obtenus par transformée en cosinus discrète inverse :

$$e_i = 20 \log_{10} \sum_{k=0}^{K-1} |X(k)| t_i(k) \tag{A.15}$$

$$c_k = \sum_{i=0}^{B-1} e_i \cos\left(k\left(i+\frac{1}{2}\right)\frac{\pi}{B}\right) \tag{A.16}$$

Cette transformée peut s'interpréter soit comme une transformée temps-fréquence, par analogie avec l'analyse cepstrale classique, soit comme une approximative d'une transformée de Karhunen-Loeve visant à décorréler les coefficients e_i et en réduire la dimensionnalité [Log00]. Les premiers coefficients c_k sont les plus significatifs, nous en avons retenu 13.

Dans notre implémentation, les MFCC sont calculés sur des fenêtres glissantes de 23ms, avec un chevauchement entre fenêtres tel que 100 vecteurs de 13 coefficients c_k sont calculés par seconde. Si l'on note $c_k(m)$ la valeur prise par le coefficient c_k durant la trame m, les attributs finalement calculés sont la moyenne et l'écart-type des coefficients $c_k(m)$ et de leurs dérivées premières $\Delta c_k(m) = c_k(m) - c_k(m-1)$ et secondes $\Delta^2 c_k(m) = c_k(m) - 2c_k(m-1) + c_k(m-2)$ sur la fenêtre d'observation :

$$\mu MFCC_k = \frac{1}{M}\sum_{m=0}^{M-1} c_k(m) \tag{A.17}$$

$$\sigma MFCC_k = \frac{1}{M}\sqrt{\sum_{m=0}^{M-1} \left(c_k(m) - \mu MFCC_k\right)^2} \tag{A.18}$$

$$\mu \Delta MFCC_k = \frac{1}{M-1}\sum_{m=1}^{M-1} \Delta c_k(m) \tag{A.19}$$

$$\sigma \Delta MFCC_k = \frac{1}{M-1}\sqrt{\sum_{m=1}^{M-1} \left(\Delta c_k(m) - \mu \Delta MFCC_k\right)^2} \tag{A.20}$$

$$\tag{A.21}$$

$$\mu \Delta^2 MFCC_k = \frac{1}{M-2}\sum_{m=2}^{M-1} \Delta^2 c_k(m) \tag{A.22}$$

$$\sigma \Delta^2 MFCC_k = \frac{1}{M-2}\sqrt{\sum_{m=2}^{M-1} \left(\Delta^2 c_k(m) - \mu \Delta^2 MFCC_k\right)^2} \tag{A.23}$$

Précisons que tous les paramètres intervenant dans le calcul des MFCC (nombre et limites des filtres, nombre de coefficients c_k retenus, longueur des fenêtres d'observation, filtre dérivateur utilisé pour le calcul des $\Delta MFCC$) correspondent à des valeurs typiques ou par défaut des implémentations logicielles utilisées. Dans [DTB+05], Degroeve et al. décrivent une procédure d'optimisation par recuit simulé de ces différents paramètres, afin de maximiser les performances d'un système de classification de sons percussifs. Les auteurs rapportent que les gains de performances obtenus sont significatifs, bien que minimes (quelques dixièmes de points). Cependant, aucun contrôle n'a été fait quant au pouvoir de généralisation d'une telle approche – il est probable que les paramètres optimaux obtenus par cette méthode soient fortement dépendant de l'ensemble d'apprentissage considéré. De manière à éviter les problèmes de surapprentissage, nous avons évité l'emploi de telles optimisations.

A.3 Paramètres spectraux

Moments spectraux Les moments spectraux permettent de résumer en quelques indicateurs la forme et la position du spectre. Le spectre $|X(f)|$ est normalisé et considéré comme une distribution de probabilité $dp_X(f) = |X(f)|df$, dont on calcule les moments d'ordre i $\mu_i = \int f^i dp_X(f)$. Une estimation empirique de ces moments peut être obtenue par :

$$\mu_i = \frac{\sum_{k=0}^{K-1} f_k^i |X(k)|}{\sum_{k=0}^{K-1} |X(k)|} \tag{A.24}$$

Des moments d'ordre μ_i, on déduit les moments centraux selon :

$$\mu_1^c = \mu_1 \tag{A.25}$$
$$\mu_2^c = \mu_2 - \mu_1^2 \tag{A.26}$$
$$\mu_3^c = \mu_3 - 3\mu_1\mu_2 + 2\mu_1^3 \tag{A.27}$$
$$\mu_4^c = \mu_4 - 4\mu_1\mu_3 + 6\mu_1^2\mu_2 - 3\mu_1^4 \tag{A.28}$$

Les paramètres spectraux utilisés sont alors finalement :

Le centroïde spectral (ou centre de gravité du spectral) fournissant une mesure de brillance du spectre :

$$S_{cntr} = \mu_1^c \tag{A.29}$$

L'étendue spectrale (ou rayon de giration spectral) fournissant une mesure de la compacité du spectre :

$$S_{sprd} = \sqrt{\mu_2^c} \tag{A.30}$$

L'asymétrie spectrale (skewness) qui fournit une mesure de déséquilibre du spectre autour de son centre de gravité :

$$S_{skew} = \frac{\mu_3^c}{(\mu_2^c)^{\frac{3}{2}}} \tag{A.31}$$

La platitude spectrale (kurtosis) qui mesure le caractère "pointu" ou contrasté du spectre :

$$S_{kurt} = \frac{\mu_4^c}{\mu_2^c} - 3 \tag{A.32}$$

Platitude spectrale Un indicateur simple de contraste du spectre, discriminant les spectres constitués de raies et les spectres de bruits (continus), est le rapport entre la moyenne géométrique du spectre de puissance et sa moyenne arithmétique [Pee04]. Pour une spectre uniforme (bruit blanc), ce rapport est maximal et égal à 1 ; il se rapproche de 0 pour les spectres de raies.

$$S_{flat} = \frac{\sqrt[K]{\prod_{k=0}^{K-1} |X(k)|^2}}{\frac{1}{K}\sum_{k=0}^{K-1} |X(k)|^2} \tag{A.33}$$

Fréquence de coupure Nous définissons la fréquence de coupure comme la plus petite fréquence en dessous de laquelle 85% de l'énergie du signal est contenue :

$$F_c = \underset{f}{\operatorname{argmin}} \left\{ f, \sum_{k=0}^{f} |X(k)|^2 \geq 0.85 \sum_{k=0}^{K-1} |X(k)|^2 \right\} \tag{A.34}$$

Coefficients de prédiction linéaires $x(n)$ est modélisé par un processus auto-régressif d'ordre $p = 6$, dont les coefficients sont obtenus en résolvant les équations de Yule-Walker :

$$\begin{bmatrix} r_x(0) & \dots & r_x(p) \\ r_x(1) & \dots & r_x(p-1) \\ \vdots & & \vdots \\ r_x(p) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} a'_0 \\ a'_1 \\ \vdots \\ a'_p \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{A.35}$$

Où $r_x(k)$ désigne les valeurs de la fonction d'autocorrélation de $x(n)$. Les 6 coefficients utilisés comme attributs, fournissant une approximation de l'enveloppe spectrale, sont alors :

$$AR_i = \frac{a'_i}{a'_0} \tag{A.36}$$

A.4 Paramètres temporels

Facteur de crête Il mesure l'impulsivité du signal par le rapport entre son maximum et sa puissance :

$$Crest = \frac{\max_n |x(n)|}{\sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x(n)^2}} \tag{A.37}$$

Centroïde temporel Le centroïde temporel fournit une indication sur la distribution de l'énergie sur la durée de la fenêtre d'observation. Un centroïde temporel faible traduira des événements impulsifs et brefs dont l'énergie est très localisée dans le temps, au début de la fenêtre d'observation.

$$T_{cntr} = \frac{\sum_{n=0}^{N-1} n x(n)^2}{\sum_{n=0}^{N-1} x(n)^2} \tag{A.38}$$

Moments de la forme d'onde La variance T_{var}, l'asymétrie T_{skew}, et la platitude T_{kurt} de la distribution des échantillons sur la fenêtre d'observation sont calculées à partir des moments :

$$\mu_i = \frac{1}{N}\sum_{n=0}^{N-1} x^i(n) \tag{A.39}$$

Taux de passage par zéro Ce paramètre mesurant la fréquence à laquelle le signal change de signe donne une mesure approximative du caractère bruité du signal. D'extrêmement bas niveau, il est peu robuste. On peut extraire une forme plus robuste de taux de passage par zéro en pré-traitant le signal par l'opération d'effondrement suivante :

$$x_e(n) = \begin{cases} 0 & \text{si } |x(n)| < \tau \\ x(n) - \tau \operatorname{sgn} x(n) & \text{sinon} \end{cases} \tag{A.40}$$

Un tel prétraitement est également traditionnellement utilisé en traitement de la parole pour permettre une estimation robuste de la fonction d'autocorrélation.

Les taux de passage par zéro sont définis par :

$$ZCR = \frac{1}{2(N-1)}\sum_{n=1}^{N-1} \operatorname{sgn} x(n) - \operatorname{sgn} x(n-1) \tag{A.41}$$

$$ZCR_r = \frac{1}{2(N-1)}\sum_{n=1}^{N-1} \operatorname{sgn} x_e(n) - \operatorname{sgn} x_e(n-1) \tag{A.42}$$

Paramètres d'enveloppe d'amplitude L'enveloppe d'amplitude du signal $x(n)$ est estimée par :

$$e(n) = (|x + j\mathcal{H}(x)| * h)(n) \tag{A.43}$$

Où $\mathcal{H}$ désigne la transformée de Hilbert, $x(n) + j\mathcal{H}(x)(n)$ est la représentation analytique de $x(n)$, dont le module fournit une estimation de l'enveloppe d'amplitude, et h est un filtre passe-bas dont la réponse impulsionnelle est une demie fenêtre de Hann. Cette enveloppe d'amplitude est modélisée par une exponentielle décroissante Ae^{-Bn}. Les paramètres $\hat{A}, \hat{B}$ sont choisis afin de minimiser l'erreur quadratique moyenne entre les valeurs observées $\log e(n)$ et les valeurs prédites $-\hat{B}n + \log \hat{A}$. Les attributs correspondant aux deux paramètres estimés sont nommés T_A et T_B.

Moments de l'enveloppe d'amplitude L'enveloppe d'amplitude $e(n)$ du signal est estimée comme précédemment. Les moments suivants sont calculés :

$$\mu_i = \frac{1}{N} \sum_{n=0}^{N-1} e^i(n) \tag{A.44}$$

Ces moments sont utilisés pour calculer la moyenne E_{mean}, la variance E_{var}, l'asymétrie E_{skew}, et la platitude E_{kurt} de la distribution des échantillons de l'enveloppe.

A.5 Paramètres psychoacoustiques

Sonie spécifique relative Cet attribut décrit dans [Pee04] mesure la distribution relative de l'énergie en prenant en compte une échelle psychoacoustique. Les sonies spécifiques sont des mesures de sonie sur chacune des 24 bandes critiques de l'échelle de Bark :

$$Ld_i = \left(\frac{1}{K} \sum_{k=B_i}^{B_{i+1}} |X(k)|^2 \right)^{0.23} \tag{A.45}$$

Où (B_i, B_{i+1}) désignent les frontières de la i-ème bande de Bark. La sonie spécifique relative consiste à normaliser la sonie relative par la sonie totale :

$$Ldr_i = \frac{Ld_i}{\sum_{b=1}^{24} Ld_b} \tag{A.46}$$

Acuité L'acuité peut être vue comme une version perceptuelle du centre de gravité spectral, utilisant l'échelle de Bark au lieu d'une échelle fréquentielle linéaire, et la sonie au lieu de la puissance. Sa formulation par Zwicker dans [Zwi77] est :

$$Acu = 0.11 \frac{\sum_{b=1}^{24} bLd_b w(b)}{Ld} \tag{A.47}$$

$$\text{avec : } w(b) = \begin{cases} 1 & \text{si } b < 15 \\ 0.066e^{0.171b} & \text{sinon} \end{cases} \tag{A.48}$$

Étendue Cet attribut introduit dans [Pee04] mesure la distance entre la plus grande valeur de la sonie spécifique et la sonie totale. Cette distance est faible pour les signaux dont l'énergie est localisée dans une bande critique, forte pour les signaux dont l'énergie est répartie sur plusieurs de ces bandes.

$$Et = \left(\frac{Ld - \max_b Ld_b}{Ld} \right)^2 \tag{A.49}$$

ANNEXE **B**

Machines à vecteurs de support (SVM)

Les succès rencontrés lors de l'application des SVM à un grand nombre de tâches de classification supervisée – catégorisation automatique de textes, reconnaissance de visages, diagnostics médicaux, reconnaissance des instruments de musique – en ont fait une méthode de classification discriminative très populaire. Cette méthode de classification étant à diverses reprises utilisée dans cette thèse, nous en effectuons ici une présentation détaillée. La section B.1 en livre une formulation simple, qui suit celle de [Bur98] ou de [SS02], en se plaçant du point de vue de la recherche d'un hyperplan séparateur optimal. Nous livrons également une interprétation géométrique du problème d'optimisation dual. Cette interprétation nous permet d'aborder le cas non linéairement séparable dans B.2. Dans la section B.3 nous expliquons comment des noyaux peuvent être utilisés pour réaliser des surfaces de décision non-linéaires, ou pour exploiter une connaissance a priori sur la structure des données à traiter. Nous concluons dans la section B.4 par la présentation de méthodes permettant de "probabiliser" les sorties des SVM, afin de les utiliser non plus uniquement à des fins de décision, mais d'estimation de probabilités a posteriori.

B.1 Principe, primal et dual

B.1.1 Principe

Soit un ensemble d'apprentissage constitué de vecteurs d'attributs réels étiquetés en deux catégories $(\mathbf{x}_i, y_i)_{i\in\{1,\ldots,N\}}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$. Nous considérons pour l'instant que cet ensemble d'apprentissage est linéairement séparable, c'est à dire qu'il existe au moins un hyperplan $H(\mathbf{w}, b)$ de normale $\mathbf{w} \in \mathbb{R}^d$ et de distance algébrique à l'origine $\frac{b}{||\mathbf{w}||}$:

$$H(\mathbf{w}, b) = \{\mathbf{x}, \mathbf{x}\cdot\mathbf{w} + b = 0\} \tag{B.1}$$

Vérifiant, $\forall i$:

$$\begin{array}{lll} \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 & \text{si} & y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 & \text{si} & y_i = -1 \end{array} \tag{B.2}$$

Notons qu'il est toujours possible de mettre à l'échelle les valeurs de $\mathbf{w}$ et de b de manière à ce qu'il existe au moins deux points pour lesquels l'égalité est vérifiée. Disposant d'un tel hyperplan séparateur, la règle de classification suivante peut alors être utilisée pour classer un vecteur $\mathbf{x}$:

$$y = \operatorname{sgn}(\mathbf{x}\cdot\mathbf{w} + b) \tag{B.3}$$

Parmi les nombreux hyperplans séparateurs possibles (voir figure B.1), lequel donne lieu à la meilleure règle de décision ? Intuitivement, le meilleur hyperplan séparateur en termes de pouvoir de généralisation et de robustesse au bruit est celui "collant" le moins possible aux exemples de l'ensemble d'apprentissage. Appelons d^+ (resp. d^-) la distance du ou des exemple(s) positif(s)

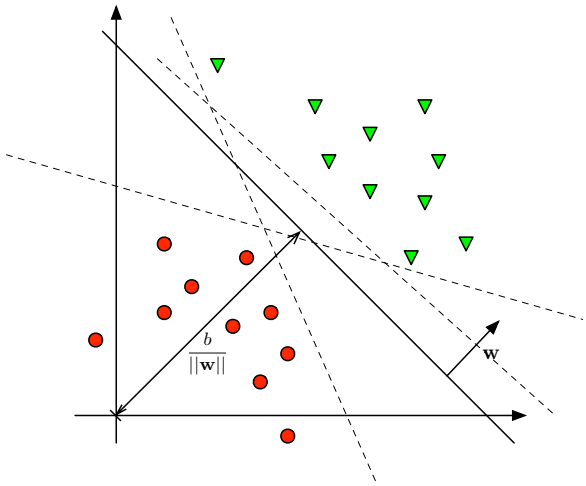


FIG. B.1 – Un exemple d'hyperplan séparateur. D'autres hyperplans séparateurs sont représentés en pointillés

(resp. négatifs) le(s) plus proche(s) de l'hyperplan séparateur à cet hyperplan. Nous rappellons que la distance d'un point $\mathbf{x}$ à un hyperplan paramétré par $(\mathbf{w}, b)$ est $\frac{|\mathbf{x}\cdot\mathbf{w}+b|}{||\mathbf{w}||}$. Alors :

$$d^+ = \min_i \left\{ \frac{|\mathbf{x}\cdot\mathbf{w}+b|}{||\mathbf{w}||}, y_i = +1 \right\} \tag{B.4}$$

$$d^- = \min_i \left\{ \frac{|\mathbf{x}\cdot\mathbf{w}+b|}{||\mathbf{w}||}, y_i = -1 \right\} \tag{B.5}$$

Or, nous avons vu que les exemples positifs et négatifs vérifient les inégalités B.2, atteintes pour au moins un exemple positif et négatif. Ces exemples sont ainsi sur les hyperplans $H^+ : \mathbf{x}\cdot\mathbf{w} + b = +1$ et $H^- : \mathbf{x}\cdot\mathbf{w} + b = -1$. Dès lors, $d^+ = d^- = \frac{1}{||\mathbf{w}||}$. La marge, que nous souhaitons maximiser, est ainsi égale à $d = d^+ + d^- = \frac{2}{||\mathbf{w}||}$. Nous pouvons de façon équivalente minimiser son inverse, ou le carré de son inverse. La recherche de l'hyperplan optimal correspond ainsi au problème d'optimisation suivant (dit primal) d'une forme quadratique sous contraintes linéaires :

$$\text{minimiser} \quad \frac{1}{2}||\mathbf{w}||^2 \tag{B.6}$$

$$\text{sous contraintes} \quad y_i(\mathbf{x}_i\cdot\mathbf{w} + b) \geq 1 \tag{B.7}$$

B.1.2 Résolution du primal

Un tel problème d'optimisation est typiquement résolu en introduisant des multiplicateurs de Lagrange $\alpha_i \geq 0$ pour chacune des N contraintes. Le Lagrangien correspondant est alors :

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} (\alpha_i y_i(\mathbf{x}_i\cdot\mathbf{w} + b) - \alpha_i) \tag{B.8}$$

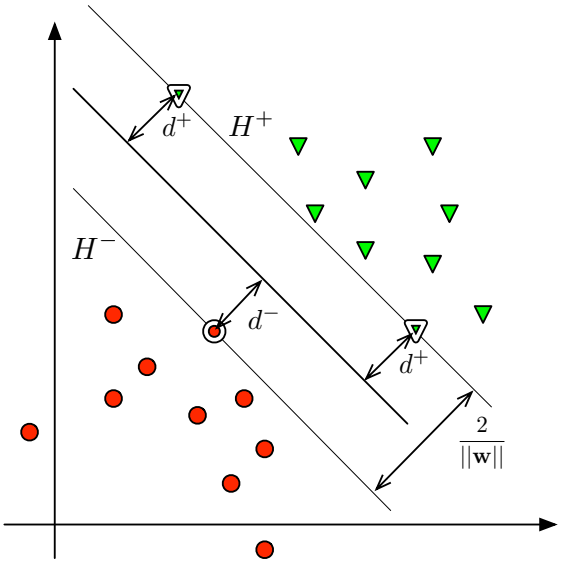


FIG. B.2 – Marge d'un hyperplan séparateur et vecteurs de support

Une solution est alors obtenue en minimisant le Lagrangien $L(\mathbf{w}, b, \boldsymbol{\alpha})$ par rapport à $\mathbf{w}$ et b, et en le maximisant par rapport à $\boldsymbol{\alpha}$. Les conditions de Karush-Kuhn-Tucker (KKT) sont des conditions nécessaires[1] vérifiées par la solution $(\mathbf{w}, b, \boldsymbol{\alpha})$. Elles s'écrivent [SS02] :

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \tag{B.9}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{B.10}$$

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \tag{B.11}$$

$$\alpha_i \geq 0 \tag{B.12}$$

$$\alpha_i(y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1) = 0 \tag{B.13}$$

La dernière condition impose que les éléments de l'ensemble d'apprentissage $\mathbf{x}_i$ pour lesquels la contrainte $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$ n'est pas saturée ont des multiplicateurs de Lagrange nuls associés $\alpha_i = 0$. Puisque l'équation de l'hyperplan séparateur est donnée par :

$$H : \mathbf{x} \cdot \mathbf{w} + b = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b \tag{B.14}$$

On en déduit que cet hyperplan n'est déterminé que par les éléments de l'ensemble d'apprentissage saturant la contrainte $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$, autrement dit, les éléments de l'ensemble d'apprentissage appartenant aux hyperplans H^+ et H^-. Ces éléments marginaux peuvent être vus comme les plus difficiles à classer, et portent le nom de vecteurs de support. Nous insistons sur cette première propriété intéressante des SVM : leur solution ne dépend que des exemples d'apprentissage les plus difficile à classer, et est parcimonieuse dans le sens où elle ne fait intervenir, en termes de calcul, que des produits scalaires avec un nombre limité d'exemples d'apprentissage (par contraste avec

[1] En fait, les conditions de KKT sont ici à la fois nécessaires et suffisantes puisque le critère à minimiser et les contraintes sont convexes.

des méthodes de classification comme les K plus proches voisins qui nécessitent de comparer un exemple à classer avec l'intégralité de l'ensemble d'apprentissage).

B.1.3 Dual

Il est possible de réécrire le Lagrangien en exploitant les égalités données par les conditions de KKT :

$$L_D(\boldsymbol{\alpha}) = \underbrace{\frac{1}{2}||\mathbf{w}||^2}_{\frac{1}{2}\mathbf{w}.\mathbf{w}} - \sum_{i=1}^{N}(\alpha_i y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - \alpha_i) \tag{B.15}$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + \sum_{i=1}^{N}\alpha_i \tag{B.16}$$

Cette réécriture du Lagrangien permet la formulation du problème d'optimisation dual suivant :

$$\text{maximiser} \qquad L_D(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + \sum_{i=1}^{N}\alpha_i \tag{B.17}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N}\alpha_i y_i = 0, \qquad \alpha_i \geq 0 \tag{B.18}$$

Cette formulation duale a les deux mérites suivants :
- Elle ne fait plus intervenir les paramètres de l'hyperplan $\mathbf{w}$ et b. Il s'agit de directement déterminer les multiplicateurs de Lagrange intervenant dans la fonction de décision.
- La forme à maximiser et les contraintes ne font intervenir les exemples d'apprentissage que sous la forme de produits scalaires $\mathbf{x}_i \cdot \mathbf{x}_j$. L'intérêt de cette propriété sera illustré dans la section B.3.

Cependant, cette formulation semble a priori moins intuitive : que représentent géométriquement les multiplicateurs de Lagrange ? Nous nous inspirons ici de [BB00] et [CB99] pour fournir une interprétation géométrique des multiplicateurs de Lagrange intervenant dans le dual.

B.1.4 Interprétation géométrique du dual

Commençons par l'observation suivante : Pour $(\mathbf{w}, b)$ donné, les $(\lambda\mathbf{w}, \lambda b)$, $\lambda \neq 0$ définissent tous le même hyperplan. Autrement dit, les solutions du problème de recherche d'un hyperplan optimal sont définies à une constante multiplicative près. Puisque $\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i$, $\boldsymbol{\alpha}$ est lui aussi défini à une constante multiplicative non nulle près. Ainsi, si un problème d'optimisation a pour solution $\lambda\boldsymbol{\alpha}$, où $\boldsymbol{\alpha}$ est la solution du problème dual, nous pouvons sans perte de généralité le résoudre en lieu et place du dual – sa solution ne correspondra qu'à une paramétrisation différente du même hyperplan séparateur de marge optimale.

Posons $\boldsymbol{\alpha}' = \frac{2}{\sum_{i=1}^{N}\alpha_i}\boldsymbol{\alpha}$. Le dual se réécrit alors :

$$\text{maximiser} \qquad -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha'_i\alpha'_j\mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + 2 \tag{B.19}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N}\alpha'_i y_i = 0, \qquad \sum_{i=1}^{N}\alpha'_i = 2, \qquad \alpha'_i \geq 0 \tag{B.20}$$

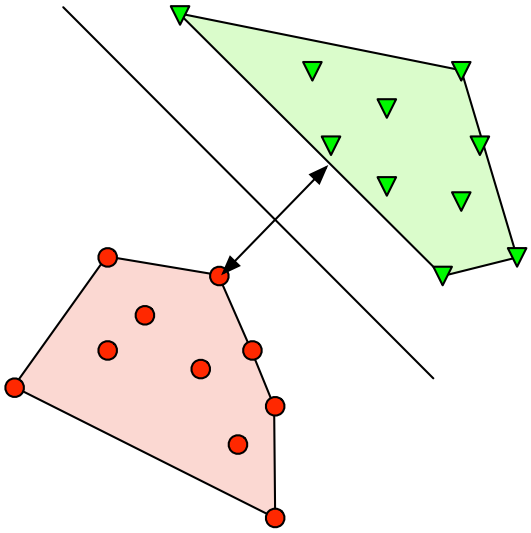


FIG. B.3 – Plus court segment joignant les enveloppes convexes des exemples positifs et négatifs

Ou plus simplement[2] :

$$\text{minimiser} \qquad \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha'_i \alpha'_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j \tag{B.23}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N} \alpha'_i y_i = 0, \qquad \sum_{i=1}^{N} \alpha'_i = 2, \qquad \alpha'_i \geq 0 \tag{B.24}$$

Comment interpréter cette version mise à l'échelle du dual ? Revenons au problème de la recherche de l'hyperplan séparateur optimal. Cet hyperplan [BB00] est la médiatrice du plus court segment joignant les enveloppes convexes des exemples positifs et négatifs (figure B.3).

Utilisons cette formulation en terme d'enveloppes convexes pour déterminer l'hyperplan optimal. Si $A = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ est un ensemble de points de $\mathbb{R}^d$, $\mathcal{C}(A)$ son enveloppe convexe, alors $\mathbf{z} \in \mathcal{C}(A)$ si et seulement si il existe γ vérifiant :

$$\sum_{i=1}^{N} \gamma_i \mathbf{z}_i = \mathbf{z}, \qquad \sum_{i=1}^{N} \gamma_i = 1, \qquad \gamma_i \geq 0 \tag{B.25}$$

Ainsi, la recherche du plus court segment d'extremités x^+ et x^- joignant les enveloppes convexes des exemples positifs et négatifs correspond au problème d'optimisation suivant :

[2] Il est possible d'arriver directement à cette version mise à l'échelle du dual en écrivant les contraintes de séparation sous la forme :

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +\rho \quad \text{si} \quad y_i = +\rho \tag{B.21}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -\rho \quad \text{si} \quad y_i = -\rho \tag{B.22}$$

Avec $\rho \geq 0$. Cette nouvelle contrainte fait apparaître un multiplicateur de Lagrange supplémentaire δ, se traduisant par une condition de KKT additionnelle.

$$\text{minimiser} \qquad ||\mathbf{x}^+ - \mathbf{x}^-||^2 \tag{B.26}$$

$$\text{sous contraintes} \qquad \begin{array}{lll} \sum_{i=1,y_i=+1}^{N} \gamma_i^+ \mathbf{x}_i = \mathbf{x}^+, & \sum_{i=1,y_i=+1}^{N} \gamma_i^+ = 1, & \gamma_i^+ \geq 0 \\ \sum_{i=1,y_i=-1}^{N} \gamma_i^- \mathbf{x}_i = \mathbf{x}^-, & \sum_{i=1,y_i=-1}^{N} \gamma_i^- = 1, & \gamma_i^- \geq 0 \end{array} \tag{B.27}$$

Posons $\alpha_i = \begin{cases} \gamma_i^+ & \text{si } y_i = +1 \\ \gamma_i^- & \text{si } y_i = -1 \end{cases}$. Le problème d'optimisation se réécrit alors de la façon suivante :

$$\text{minimiser} \qquad \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j \tag{B.28}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad \sum_{i=1}^{N} \alpha_i = 2, \qquad \alpha_i \geq 0 \tag{B.29}$$

C'est le dual mis à l'échelle. Nous avons ainsi vu qu'une formulation géométrique différente du problème de la recherche de l'hyperplan séparateur optimal mène directement au dual. Les multiplicateurs de Lagrange $\boldsymbol{\alpha}$ s'interprètent alors simplement comme des poids, définissant deux points des enveloppes convexes des exemples positifs et négatifs.

B.2 Cas non linéairement séparable

B.2.1 Vision géométrique intuitive

Nous nous intéressons maintenant au cas où l'ensemble d'apprentissage est non linéairement séparable, par exemple en raison de la présence d'exemples bruités ou erronés. Géométrique, deux ensembles de points sont non linéairement séparables si leurs enveloppes convexes s'intersectent. Un remède à cette non-séparabilité consiste à faire "fondre" les enveloppes convexes des deux ensembles, en considérant des enveloppes convexe μ réduites $\mathcal{C}_\mu$. Si $A = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ est un ensemble de points de $\mathbb{R}^d$, alors $\mathbf{z} \in \mathcal{C}_\mu(A)$ si et seulement si il existe γ vérifiant :

$$\sum_{i=1}^{N} \gamma_i \mathbf{z}_i = \mathbf{z}, \qquad \sum_{i=1}^{N} \gamma_i = 1, \qquad \mu \geq \gamma_i \geq 0 \tag{B.30}$$

Pour $\mu = 1$, nous retrouvons la formulation classique. Quand μ décroît vers zéro, nous diminuons progressivement l'influence des points marginaux, et l'enveloppe convexe μ réduite se condense vers l'intérieur (voir figure B.4). La recherche d'un hyperplan séparateur à marge maximale entre les enveloppes convexes réduites correspond alors très simplement au problème d'optimisation suivant que nous appelons μ-SVM :

$$\text{minimiser} \qquad \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j \tag{B.31}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad \sum_{i=1}^{N} \alpha = 2, \qquad \mu \geq \alpha_i \geq 0 \tag{B.32}$$

Le paramètre ajustable $0 \leq \mu \leq 1$ réalise alors un compromis entre généralisation et prise en compte exhaustive de l'ensemble d'apprentissage.

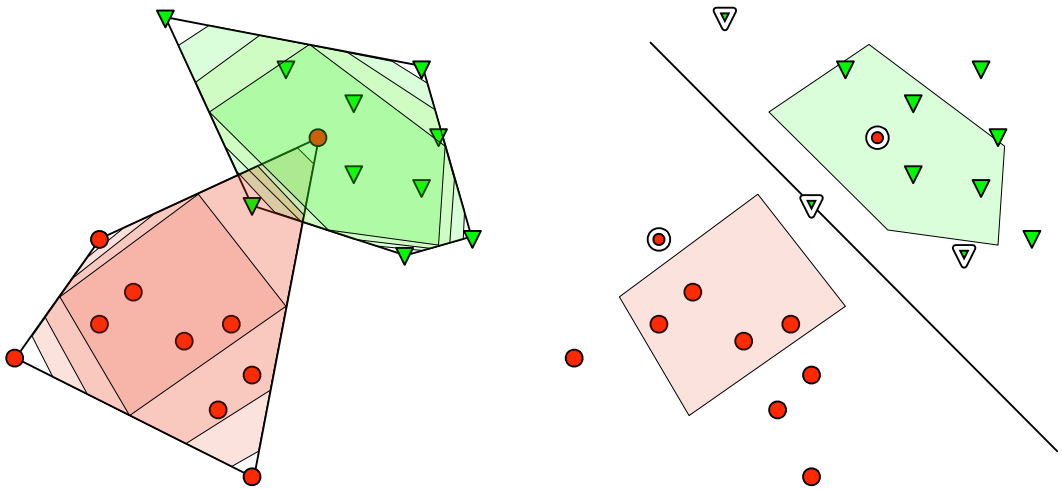


FIG. B.4 – Enveloppes convexes μ-réduites pour $\mu = 0.8$, $\mu = 0.6$, $\mu = 0.5$. Hyperplan séparant les enveloppes convexes 0.5-réduites et vecteurs de support

B.2.2 C-SVM

Nous dérivons maintenant une autre formulation du cas non-séparable linéairement. Rappelons que dans le cas linéairement séparable, nous avons, $\forall i$:

$$\begin{array}{lll} \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 & \text{si} & y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 & \text{si} & y_i = -1 \end{array} \tag{B.33}$$

Dans le cas non-linéairement séparable, ces contraintes ne peuvent être satisfaites pour les points marginaux. Elles sont alors relaxées en introduisant des termes de marge $\xi_i \geq 0$. Ces termes représentent, intuitivement, le degré avec lequel la i-ème contrainte de séparabilité est violée.

$$\begin{array}{lll} \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \xi_i & \text{si} & y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 + \xi_i & \text{si} & y_i = -1 \end{array} \tag{B.34}$$

De manière à privilégier les solutions violant le moins possible les contraintes, un terme de coût $C \sum_{i=1}^{N} \xi_i$ pénalisant les solutions violant trop les contraintes est introduit dans le critère à optimiser. Le problème d'optimisation devient alors :

$$\text{minimiser} \qquad \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \xi_i \tag{B.35}$$

$$\text{sous contraintes} \qquad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0 \tag{B.36}$$

Un nouveau jeu de multiplicateurs de Lagrange $\boldsymbol{\mu}$ doit être introduit pour la contrainte de positivité de $\boldsymbol{\xi}$. Le Lagrangien s'écrit :

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left(y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\right) - \sum_{i=1}^{N} \mu_i \xi_i \tag{B.37}$$

Les conditions de KKT deviennent :

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \tag{B.38}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{B.39}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = C - \boldsymbol{\alpha} - \boldsymbol{\mu} = 0 \tag{B.40}$$

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i \tag{B.41}$$

$$\alpha_i \geq 0 \tag{B.42}$$

$$\mu_i \geq 0 \tag{B.43}$$

$$\xi_i \geq 0 \tag{B.44}$$

$$\alpha_i(y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i) = 0 \tag{B.45}$$

$$\mu_i \xi_i = 0 \tag{B.46}$$

En réécrivant le Lagrangien en exploitant les égalités données par les conditions de KKT :

$$L_D(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} (C - \alpha_i - \mu_i)\, \xi_i \tag{B.47}$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + \sum_{i=1}^{N} \alpha_i \tag{B.48}$$

Cette réécriture du Lagrangien permet la formulation du problème d'optimisation dual suivant, dit C-SVM :

$$\text{maximiser} \qquad L_D(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + \sum_{i=1}^{N} \alpha_i \tag{B.49}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N} \alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C \tag{B.50}$$

Nous retrouvons une formulation semblable à celle obtenue précédemment, la borne s'appliquant maintenant sur les multiplicateurs de Lagrange non mis à l'échelle. Comme précédemment, le paramètre C exprime un compromis entre généralisation et fidélité à l'ensemble d'apprentissage. Lorsque C est faible, nous nous autorisons de violer plus de contraintes pour maximiser la marge. Lorsque C est élevé, nous pénalisons fortement la violation des contraintes. Une différence notable avec les développements précédents est la plage de variation de ce paramètre. Précédemment, les multiplicateurs de Lagrange étaient normalisés entre 0 et 1 et s'interprétaient comme des poids, μ mesurant la contraction des enveloppes convexes permettant la séparation linéaire. Ici, les multiplicateurs de Lagrange ne sont pas normalisés. Le paramètre C peut prendre alors toute valeur positive réelle.

Une autre interprétation des multiplicateurs de Lagrange se déduit des conditions de KKT :

$$\begin{array}{llll} \text{si } \alpha_i = 0 & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1, & \xi_i = 0 & \text{Hors de la marge, correctement classés} \\ \text{si } 0 < \alpha_i < C & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) = 1, & \xi_i = 0 & \text{Sur la marge} \\ \text{si } \alpha_i = C & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \leq 1, & \xi_i \geq 0 & \text{Hors marge, pouvant être mal classés} \end{array} \tag{B.51}$$

Ainsi, les points dont les multiplicateurs de Lagrange sont non-nuls et inférieurs à C sont les vecteurs de support – ils définissent les exemples de chaque classe les plus difficiles à classer. Les

points dont les multiplicateurs de Lagrange associés sont égaux à C sont les vecteurs de support bornés (ou saturés), ils correspondent à des valeurs aberrantes, erronées, ou à des "exceptions".

Nous soulignons qu'il est possible de dériver d'autres paramétrisations géométriques des SVM que les C-SVM ou les μ-SVM. Par exemple, les ν-SVM [CLS05] utilisent une paramétrisation dans laquelle le terme de régularisation ν s'interprète comme la fraction maximale de vecteurs de supports à extraire des exemples d'apprentissage.

B.2.3 Résolution du dual pour les C-SVM

Nous décrivons ici de quelle manière les implémentations logicielles actuelles des SVM, dont celle que nous avons utilisée [CL01], résolvent le problème d'optimisation dual. Le nombre de variables α_i à optimiser est égal à la taille de l'ensemble d'apprentissage N, rendant impossible l'utilisation de solveurs classiques (LOQO par exemple). En particulier, les solveurs classiques chargent en mémoire la matrice $\mathbf{K}_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j$, à la fois de taille rédhibitoire pour de grandes valeurs de N, et dense.

Une solution à ce problème consiste à décomposer l'étape d'optimisation en plusieurs itérations. N'est optimisé à chaque itération qu'un sous ensemble des variables $\alpha_i, i \in V_A$, dit ensemble de travail. Nous notons V_I l'ensemble complémentaire (ensemble des variables inactives). La procédure d'optimisation itérative est décrite dans l'algorithme 7.

Algorithme 7 : Résolution itérative des C-SVM par décomposition

entrées : $C, \boldsymbol{\alpha}, \mathbf{x}, \mathbf{y}, \epsilon$
Initialiser $\boldsymbol{\alpha}^1$ avec une solution des contraintes
$k \leftarrow 1$
tant que $\boldsymbol{\alpha}^k$ *viole les ϵ-conditions de KKT* **faire**
$V_A \leftarrow \{a_1, \ldots, a_M\} \subset \{1, \ldots, N\}$ ensemble de travail intéressant
$V_I \leftarrow \{1, \ldots, N\} \backslash V_A$ variables inactives
Déterminer α^* solution de :

$$\begin{array}{ll} \max\limits_{\alpha_i, i \in V_A} & -\frac{1}{2} \sum\limits_{i \in V_A} \sum\limits_{j \in V_A} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j - \frac{1}{2} \sum\limits_{i \in V_A} \sum\limits_{j \in V_I} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j y_i y_j + \sum\limits_{i \in V_A}^{N} \alpha_i \\ \text{s.c.} & \sum\limits_{i \in V_A} \alpha_i y_i = - \sum\limits_{i \in V_I} \alpha_i y_i, \qquad C \geq \alpha_i \geq 0 \end{array}$$

$\alpha_{a_i}^{k+1} = \alpha_i^*, \forall i \in \{1, \ldots, M\}$
$\alpha_i^{k+1} = \alpha_i^k, \forall i \in V_I$
$k \leftarrow k + 1$
fin
sorties : $\boldsymbol{\alpha}^k$

Le critère d'arrêt consiste à vérifier si les conditions de KKT sont satisfaites, avec une tolérance ϵ. Par exemple l'implémentation SVM^{light} utilise le critère suivant :

$$\begin{array}{ll} \text{si } \alpha_i = 0 & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \epsilon \\ \text{si } 0 < \alpha_i < C & 1 - \epsilon \leq y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \leq 1 + \epsilon \\ \text{si } \alpha_i = C & y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \leq 1 + \epsilon \end{array} \tag{B.52}$$

Reste à définir comment est choisi l'ensemble de travail. Sa taille doit être raisonnable ($M \approx 100$) pour pouvoir traiter le sous-problème correspondant par un solveur classique. De façon plus extrême, la méthode Sequential Minimal Optimization (SMO) [Pla98] n'utilise que deux variables actives, $M = 2$. L'intérêt d'un tel choix est que le problème d'optimisation à deux variables peut être résolu analytiquement en quelques étapes. Le nombre d'itérations sera plus grand, mais chaque itération sera très simple.

Précisons maintenant quel critère est utilisé pour choisir les variables actives. Un critère simple consiste à sélectionner la paire de variables violant au maximum les conditions de KKT, dans chaque direction :

$$a_1 = \underset{i}{\operatorname{argmax}}\{-y_i(\mathbf{x}_i \cdot \mathbf{w} + b), \alpha_i < C, y_i = +1 \text{ ou } \alpha_i > 0, y_i = -1\} \tag{B.53}$$

$$a_2 = \underset{i}{\operatorname{argmax}}\{-y_i(\mathbf{x}_i \cdot \mathbf{w} + b), \alpha_i < C, y_i = -1 \text{ ou } \alpha_i > 0, y_i = +1\} \tag{B.54}$$

D'autres variantes de SMO recensent les variables violant les conditions de KKT, et considèrent successivement, comme ensemble de travail, chacun des couples possibles. Des critères plus efficaces pour des variantes de SMO sont discutés dans [REF05]. Ces critères sont utilisés dans les versions récentes de libSVM [CL01], que nous utilisons.

B.3 SVM à noyaux

Jusqu'ici, nous avons utilisé, pour l'apprentissage et la classification, les vecteurs d'attributs $\mathbf{x}_i$ dans leur espace original $\mathbb{R}^d$. Étudions désormais le cas où l'on applique au préalable aux données une transformation $\phi : \mathbb{R}^d \mapsto \mathcal{H}$ projetant les exemples d'apprentissage vers un espace de Hilbert $\mathcal{H}$ (de dimension supérieure à d, ou de dimension infinie) :

Le problème d'optimisation se réécrit alors en :

$$\text{maximiser} \qquad L_D(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)y_iy_j + \sum_{i=1}^{N}\alpha_i \tag{B.55}$$

$$\text{sous contraintes} \qquad \sum_{i=1}^{N}\alpha_i y_i = 0, \qquad C \geq \alpha_i \geq 0 \tag{B.56}$$

Et la fonction de décision permettant de classifier les exemples devient :

$$\mathbf{y} = \operatorname{sgn}\sum_{i=1}^{N}\alpha_i y_i \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}) + b \tag{B.57}$$

Nous observons que la projection ϕ n'intervient que dans des expressions du type $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$, permettant l'application d'une technique de calcul appelée *ruse du noyau*. Si l'on définit la fonction K par $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$, il n'est pas nécessaire de définir explicitement ϕ et de calculer les projetés des exemples dans $\mathcal{H}$, puisque les calculs ne font intervenir que le noyau $K(\mathbf{x}, \mathbf{y})$. Nous traitons alors le problème de la recherche d'un hyperplan séparateur optimal dans l'espace transformé $\mathcal{H}$, tout en effectuant les calculs dans l'espace original $\mathbb{R}^d$. Ceci est particulièrement intéressant dans les cas où l'application ϕ projette les données dans un espace de dimension infinie. En fait, nous pouvons suivre la démarche inverse : ne pas se soucier de ϕ, et choisir directement une fonction noyau K. K joue alors le rôle de mesure de similarité dans $\mathbb{R}^d$ pertinente pour notre problème.

Nous répondons désormais à deux questions : quel est l'intérêt de rechercher un hyperplan séparateur dans $\mathcal{H}$ plutôt que dans $\mathbb{R}^d$, et quelles fonctions noyaux K pouvons-nous utiliser ?

B.3.1 Séparabilité dans un espace de grandes dimensions

Nous avons traité dans la section B.2 le cas où l'ensemble d'apprentissage n'est pas non-linéairement séparable en raison d'exemples bruités ou erronés. Il existe cependant des problèmes qui sont intrinsèquement non linéairement séparables, comme celui donné dans la figure B.5 où la surface de séparation optimale serait un cercle d'équation :

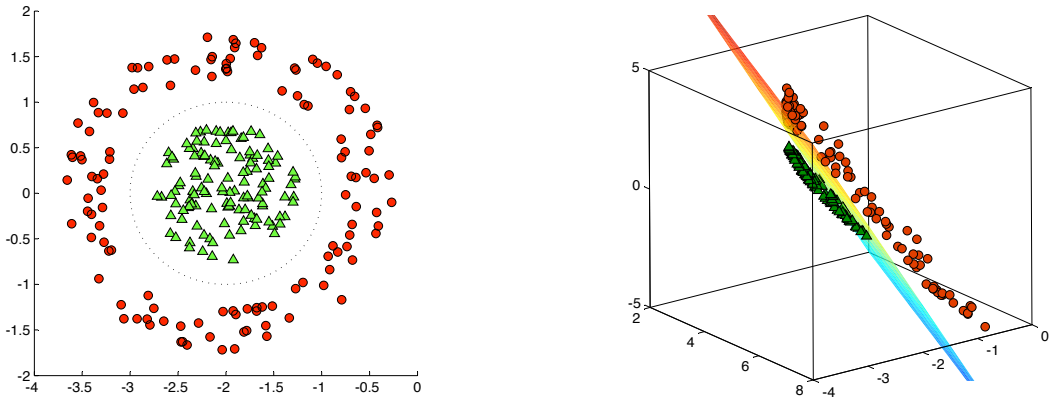


FIG. B.5 – Un problème non-linéairement séparable en dimension 2 le devient en dimension 3 après projection non-linéaire

$$(x_1 + 2)^2 + x_2^2 = 1 \tag{B.58}$$

Une telle surface de séparation ne peut pas être réalisée par un hyperplan en dimension 2. Considérons alors la transformation :

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 \\ x_2^2 \end{bmatrix} \tag{B.59}$$

L'équation du cercle séparateur se réécrit :

$$\begin{bmatrix} x_1^2 \\ x_1 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} + 3 = 0 \tag{B.60}$$

Ainsi, dans l'espace transformé, le problème devient linéairement séparable. Nous avons ici intuité l'équation de la surface de séparation, mais il serait possible de la trouver directement en résolvant le problème d'optimisation B.56 en employant noyau :

$$K(\mathbf{x}_i, \mathbf{x}_j) = x_{i,1}^2 x_{j,1}^2 + x_{i,2}^2 x_{j,2}^2 + x_{i,1} x_{j,1} \tag{B.61}$$

Le rôle de ϕ est ainsi de former des attributs nouveaux permettant une séparation non-linéaire. Une vision duale est de considérer que le noyau K est une mesure de similarité permettant de courber la surface de décision dans l'espace $\mathbb{R}^d$.

B.3.2 Fonctions noyaux

On montre [SS02] que pour une fonction $K(\mathbf{x}, \mathbf{y})$, il existe un espace $\mathcal{H}$ et une fonction ϕ : $\mathbb{R}^d \mapsto \mathcal{H}$ vérifiant :

$$K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y}) \tag{B.62}$$

Si et seulement si, pour toute fonction $g : \mathbb{R}^d \mapsto \mathbb{R}$, $g \in \mathcal{L}^2$:

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) dx dy \geq 0 \tag{B.63}$$

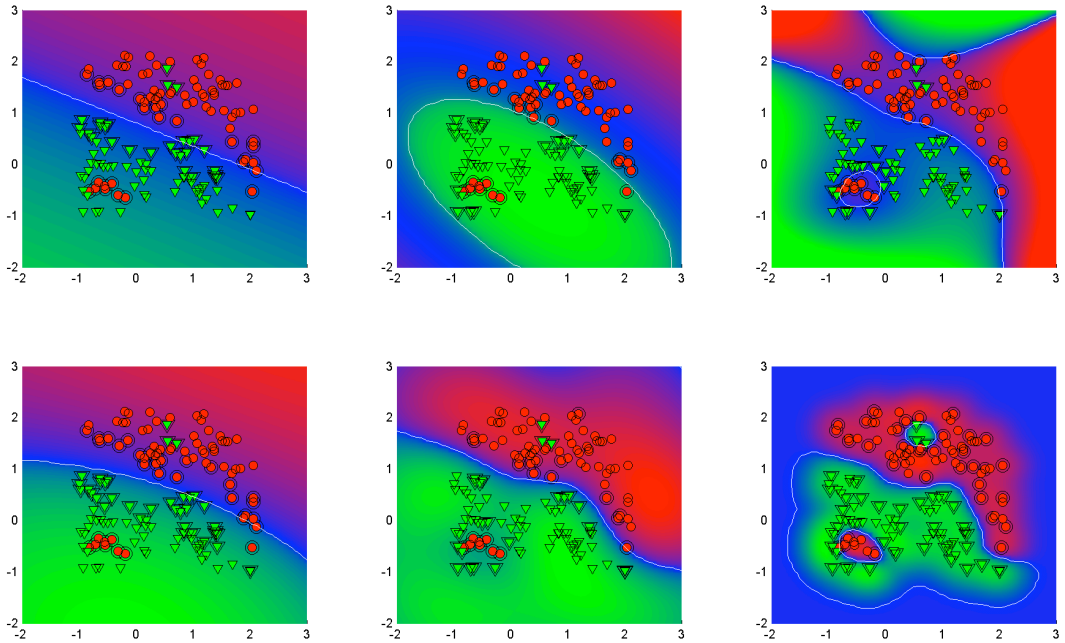


FIG. B.6 – Surface de décision pour différents noyaux : linéaire, polynomial de degré 2, polynomial de degré 4, et Gaussien pour $\sigma = 4$, $\sigma = 1$, $\sigma = \frac{1}{4}$

Dans ce cas, K est un noyau. Cette condition de définie-positivité est connue sous le nom de condition de Mercer. Nous présentons maintenant quelques noyaux communément utilisés.

B.3.2.1 Noyau polynomial d'ordre δ

Ce noyau est défini par :

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^{\delta} \tag{B.64}$$

Les composantes de $\phi(\mathbf{x})$ sont constituées de tous les monômes d'ordre inférieur ou égal[3] à δ. Par exemple, pour $d = 3, \delta = 2$:

$$\phi(\mathbf{x}) = \left[\begin{array}{ccccccccccc} x_1^2 & x_2^2 & x_3^2 & \sqrt{2}x_1 & \sqrt{2}x_2 & \sqrt{2}x_3 & \sqrt{2}x_1x_2 & \sqrt{2}x_1x_3 & \sqrt{2}x_2x_3 & 1 \end{array} \right]^T \tag{B.65}$$

On en déduit dim $\mathcal{H} = C_{\delta+d}^{\delta}$

B.3.2.2 Noyau Gaussien

Ce noyau est défini par :

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2\sigma^2}\right) \tag{B.66}$$

L'intérêt de ce noyau réside dans le paramètre σ permettant de contrôler la forme de la surface de décision, ou la séparabilité des points dans l'espace $\mathcal{H}$ (qui est ici de dimension infinie). Pour σ très

[3] Il est aussi possible de définir un noyau polynomial homogène $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^{\delta}$. Dans ce cas, les composantes de $\phi(\mathbf{x})$ contiennent tous les monômes d'ordre strictement égal à δ. L'intérêt du noyau inhomogène est qu'il inclut, dans l'espace transformé, une "copie" des attributs originaux.

élevé son comportement est similaire au noyau linéaire $K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$. Quand σ décroit la surface de décision se courbe. De façon extrême, il existe ϵ tel que pour $\sigma < \epsilon$, tous les points de l'ensemble d'apprentissage deviennent linéairement indépendants (donc linéairement séparables) dans $\mathcal{H}$. La surface de décision est dans ce cas capable de contourner individuellement tous les exemples de l'ensemble d'apprentissage. Une telle situation est illustrée dans le dernier exemple de la figure B.6.

Parce que son paramètre peut être aisément ajusté (et interprété) pour réaliser un compromis généralisation/apprentissage, c'est ce noyau que nous avons retenu. Nous l'utilisons également sous la forme normalisée suivante (d est la dimension des vecteurs $\mathbf{x}$ considérés) :

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2d\sigma^2}\right) \tag{B.67}$$

B.4 Estimation de probabilités a posteriori à partir de SVM

Jusqu'ici, nous avons utilisé les SVM pour obtenir des fonctions de décision "dures", de la forme :

$$\mathbf{y} = \operatorname{sgn} f(\mathbf{x}_i) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{B.68}$$

Nous nous intéressons maintenant au cas où nous souhaitons obtenir les probabilités a posteriori $p(y|\mathbf{x})$, et non plus seulement la classe y. Disposer de telles probabilités permet, par exemple, d'ajuster le seuil de décision en fonction des coûts associés aux erreurs de type I et II, de permettre la fusion de classifieurs, ou d'utiliser des post-traitements utilisant des connaissances externes (modèles de langage dans notre application). Nous présentons dans cette section deux méthodes pour estimer $p(y|\mathbf{x})$.

B.4.1 Régression logistique à noyaux

Observons d'abord que le problème d'optimisation des C-SVM peut se reformuler en :

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} \max\{0, 1 - y_i f_{\mathbf{w},b}(\mathbf{x}_i)\} + \lambda ||\mathbf{w}||^2 \tag{B.69}$$

Le premier terme mesure le degré de violation des contraintes, ou peut être vu comme une mesure empirique d'erreur de classification selon la fonction de coût $l_c = \max\{0, 1 - y_i f(\mathbf{x}_i)\}$, dite fonction de coût charnière. Le second terme est un terme de régularisation correspondant ici à l'inverse de la marge. Le paramètre λ assurant le compromis entre les deux termes est lié à C (une telle formulation établit le lien entre les SVM et la théorie de la régularisation de Tikhonov). Sous cette formulation, la résolution est difficile, puisque nous sommes en présence d'un terme non-linéaire, mais cette formulation a cependant l'avantage de permettre de suggérer une généralisation des SVM utilisant d'autres fonctions de coût $l(y_i, f(\mathbf{x}_i))$:

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} l(y_i, f_{\mathbf{w},b}(\mathbf{x}_i)) + \lambda ||\mathbf{w}||^2 \tag{B.70}$$

La fonction de coût $l_l(y_i, f_{\mathbf{w},b}(\mathbf{x}_i)) = \log(1 + e^{-y_i f_{\mathbf{w},b}(\mathbf{x}_i)})$ peut par exemple être considérée. Son comportement asymptotique est similaire au coût charnière, laissant supposer des propriétés semblables à celles des SVM. En outre, elle correspond à la fonction de coût utilisée en régression logistique. La fonction $f(\mathbf{x}_i)$ est alors une estimée du logit :

$$f(\mathbf{x}_i) = \log \frac{P(y = +1|\mathbf{x})}{P(y = -1|\mathbf{x})} \tag{B.71}$$

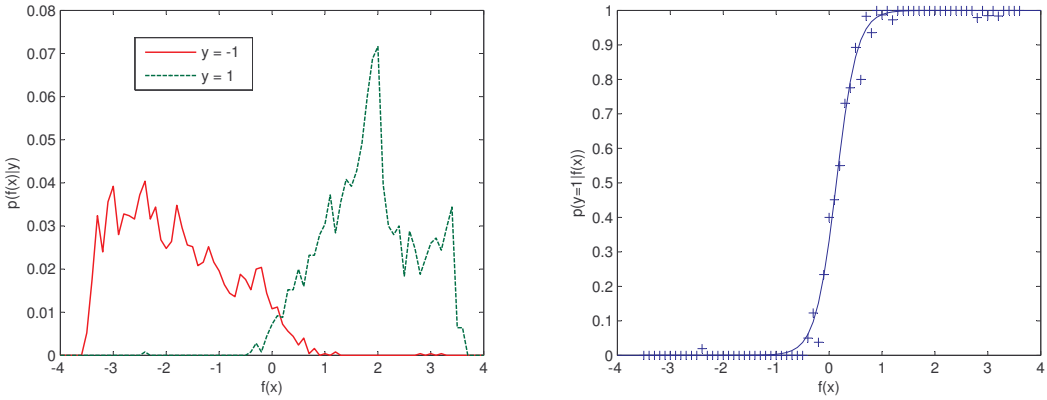


FIG. B.7 – Estimation de probabilités a posteriori à partir de SVM par la méthode de Platt

Et nous en déduisons $P(y = +1|\mathbf{x}) = e^{f_{\mathbf{w},b}(\mathbf{x})}/(1 + e^{f_{\mathbf{w},b}(\mathbf{x})})$.

Le problème d'optimisation correspondant est connu sous le nom de régression logistique à noyaux. Malheureusement, sa résolution est coûteuse en calculs, et les solutions ne peuvent pas être écrites en fonction d'un nombre réduit d'éléments de l'ensemble d'apprentissage. Zhu et Hastie décrivent dans [ZH05] un algorithme permettant d'obtenir des solutions approchées parcimonieuses. Les machines à *vecteurs d'import* apprises par cet algorithme ont des performances en classification semblables aux SVM, mais le coût de leur résolution reste rédhibitoire.

B.4.2 Méthode de Platt

Platt propose dans [Pla00] une méthode empirique permettant d'obtenir des probabilités à posteriori à partir d'une SVM. Supposons qu'on dispose, en plus de l'ensemble d'apprentissage sur lequel f a été appris, d'un ensemble supplémentaire de T exemples étiquetés $(\mathbf{x}_i, y_i)_{i\in\{1,\ldots,T\}}$, $\mathbf{x} \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$. Il est alors possible d'utiliser ces exemples pour estimer $\hat{p}(f(\mathbf{x})|y)$, par exemple par la méthode des fenêtres de Parzen, ou par un simple histogramme. Il est également possible d'utiliser ces exemples supplémentaires, ou l'intégralité de l'ensemble d'apprentissage, pour estimer (par comptage) $\hat{p}(y = 1)$ et $\hat{p}(y = -1)$. La probabilité a posteriori $\hat{p}(y|f(\mathbf{x}))$, sur les exemples supplémentaires, peut être calculée simplement par la règle de Bayes :

$$\hat{p}(y = 1|f(\mathbf{x})) = \frac{\hat{p}(y = 1)\hat{p}(f(\mathbf{x})|y = 1)}{\hat{p}(y = 1)\hat{p}(f(\mathbf{x})|y = 1) + \hat{p}(y = -1)\hat{p}(f(\mathbf{x})|y = -1)} \tag{B.72}$$

La figure B.7 représente $\hat{p}(f(\mathbf{x})|y)$ sur les exemples supplémentaires, et la probabilité a posteriori $\hat{p}(y = 1|f(\mathbf{x}))$. Platt observe qu'empiriquement, sa forme est proche d'une sigmoïde, et propose donc de modéliser la probabilité a posteriori par :

$$p_{A,B}(y = 1|f(\mathbf{x})) = \frac{1}{1 + \exp\left(A + Bf(\mathbf{x})\right)} \tag{B.73}$$

Les paramètres A et B sont choisis pour minimiser $|p_{A,B}(y = 1|f(\mathbf{x})) - \hat{p}(y = 1|f(\mathbf{x}))|^2$, par une méthode d'optimisation classique – algorithme de Marquardt-Levenberg [PTVF92]. Platt suggère également plusieurs stratégies pour obtenir un ensemble d'exemples étiquetés supplémentaires en plus de l'ensemble d'apprentissage (*leave-one-out*, validation croisée). La stratégie que nous avons retenue, qui est la plus efficace en termes de calculs, et qui est rendue possible par la disponibilité d'un grand nombre d'exemples, consiste à utiliser 80% des exemples d'apprentissage pour l'apprentissage de la SVM, et les exemples restants pour l'estimation des paramètres A et B.

Cinquième partie

Annexes - Documents complémentaires

ANNEXE **C**

Autres articles

Cette section reproduit trois articles dont le contenu n'est pas traité en détail dans ce document.

O. Gillet et G. Richard. Indexing and Querying Drum Loops Databases. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing*, 2005.

Ce premier article décrit un système d'indexation et de recherche de courtes séquences rythmiques monophoniques (boucles de batterie). La tâche d'indexation, qui consiste à transcrire chacune des boucles de la base de données, est effectuée par une approche de type *segmenter et classifier* similaire à celle introduite au chapitre 4. Les séquences étant monophoniques et la taxonomie retenue différente, un seul classifieur multi-classes est utilisé (plutôt que plusieurs classifieurs binaires). Des requêtes peuvent être effectuées sur la base indexée en formulant des requêtes vocales à l'aide d'onomatopées (*beatboxing*). À cet effet, un système de reconnaissance vocale multi-locuteur a été développé pour la tâche de transcription des requêtes. Nous proposons enfin un modèle statistique d'interprétation des rythmes sous forme d'onomatopées (modèle présenté plus en détail dans [GR05b]), permettant de calculer un score de similarité entre une requête et chacune des boucles contenues dans la base. Plusieurs autres modalités de requête – requête par l'exemple, exploration cartographique – sont évoquées dans l'article.

O. Gillet et G. Richard. Automatic Transcription of Drum Sequences Using Audiovisual Features. In *Proceedings of the 2005 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'05)*, 2005.

Ce deuxième article présente un système de transcription audiovisuelle de soli de batterie, développé dans une étude préliminaire. Deux approches sont comparées pour la fusion des attributs audio et vidéo : fusion précoce (concaténation des vecteurs d'attributs et construction par PCA d'attributs audiovisuels) ; et fusion tardive (fusion proprement dite par l'opérateur produit, ou fusion par choix du meilleur expert). Les classifieurs utilisés sont des SVM. Une des limites de ce système est que les classifieurs appris ne sont pas universels – ils dépendent de l'angle de prise de vue et de la disposition des différents éléments de la batterie. Cette difficulté nous a poussé a choisir une autre approche, détaillée dans la section 6.3.

O. Gillet et G. Richard. ENST-drums : an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, 2006.

Ce dernier article décrit le contenu et les procédures d'enregistrement, de post-production et d'annotation de la base ENST-drums utilisée tout au long de ce document.

INDEXING AND QUERYING DRUM LOOPS DATABASES

Olivier Gillet and Gaël Richard

GET-TELECOM Paris
37, rue Darreau
75015 Paris, France
[olivier.gillet, gael.richard]@enst.fr

ABSTRACT

Large databases of short drums signals, known as drum loops, are widely used for the composition of modern music. This paper presents a complete and integrated system to index and query such databases. The transcription task necessary to index the database can be performed with a range of different classifiers such as Hidden Markov Models (HMM) or Support Vector Machines (SVM) and achieves a 89.9% correct recognition rate on a simplified taxonomy. Queries can be formulated on this indexed database with spoken onomatopoeia - short meaningless words imitating the different sounds of the drum kit. The syllables of spoken queries are recognized and a relevant statistical model allows the comparison and alignement of the query with the rhythmic sequences stored in the database. This same model can be used to provide a distance measure and allows queries by example. Query results can be graphically displayed and grouped by similarity.

1. INTRODUCTION

Pre-recorded audio databases of drum loops are widely used in the production of modern music, especially in genres such as hip-hop, r'n'b, house, drum'n'bass or techno. These databases, available as collections of CDs or CD-ROMs, gather a large number of short drum signals which are used as a raw material for composition: Either individual notes are extracted and rearranged with music software such as ReCycle, or the whole signal is repeated to build an entire drum track - hence the name, *loop*. Most of the drum-loops collections do not provide any other information than the tempo and style of each loop. As a result, the musician has no other alternative than browsing the entire CD and listening to each individual file. There is therefore a need for more elaborated retrieval and indexing tools that will provide content-based methods in a user-friendly interface, to efficiently search these databases.

An important aspect of such a tool is the necessity to obtain an automatic transcription of the drum loop signals - the indexing stage. Most of the work in the domain of audio transcription is dedicated to melodic instruments (see for instance [8] for a review on instrument recognition), however the transcription of percussive signals (such as drum signals for example) has gained much interest in the past few years. Gouyon & al. [9] evaluated several classifiers and feature sets for natural and electronic drum signals recognition: these approaches proved to be successful but were limited to isolated sounds. A specifity of drum loops signals is that each event can be produced by simultaneous strokes on different instruments (for example bass drum and hi-hat). Another specifity of drum loops is that they contain a succession of events (or strokes). As a consequence, drum loop signals or drum tracks often exhibit a temporal structure.

Similarly to audio indexing, most of the works in music retrieval focus on melody and on query by example. A very popular approach called "Query by humming", aims at retrieving music files from a sung melody. Various systems are already implemented and show promising results ([3], [13]). However, most of them require a high-level representation of the whole searched database, for example as a collection of MIDI files, and only take into account melodic information. In the context of percussive signals where melody is hardly present, a different approach needs to be followed. One of the most natural ways of describing a pure rhythmic content is by means of spoken onomatopoeia - short meaningless words imitating the different sounds of the percussive instruments (drums in this context). The use of spoken onomatopoeia is a rather new approach to drum pattern retrieval which was presented in [5], independently of the works by Nakana & al. [14] and Kapur & al. [10].

This paper details and extends our first works presented in [5]. It is organized as follows. Section 2 presents the overall system architecture of our drum loop retrieval system and describes the new database used in this study. The next section details the different steps of the automatic transcription of drum loops (features extraction, classification) and evaluates the transcription performance. Then, section 3 is dedicated to the spoken onomatopoeia recognition, using a new speaker-independent system. Section 4 describes

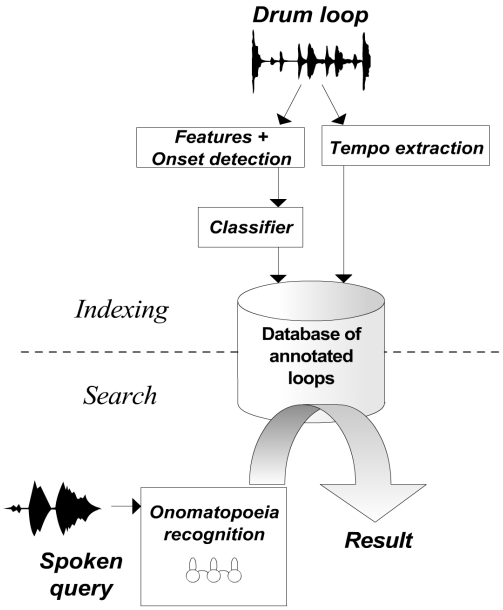


Fig. 1. System architecture

in details the approach followed to align the query with the loops contained in the database, and provides some evaluation results. Following a section dedicated to implementation and applications issues, section 6 suggests some conclusions.

2. SYSTEM ARCHITECTURE, DATABASE AND TAXONOMY

2.1. Components

The overall architecture of the system is depicted in figure 1. The first important component is the automatic drum loops transcription (indexing) module. Each drum loop is individually indexed by segmenting it in successive strokes and by recognizing the instrument or combination of instruments played for each of these strokes. The second important component is the retrieval system: the spoken queries are recognized into a sequence of onomatopoeia, each of them associated to a target drum sound. The indexed database is searched for the drum loops that best correspond to the query.

The rest of this section will focus on the different improvements and extensions of our first transcription system presented in [4] and in [5].

2.2. Drum loops database

Our previous work used a database, B_1 consisting in 315 loops (5327 strokes). We gathered a new collection of loops B_2, containing 128 loops (2685 strokes). This new set includes loops downloaded from the web or extracted from drum solos occuring in songs from the RWC Popular Music Database [6]. The loops from B_1 and B_2 are representative of different styles including rock, funk, jazz, hip-hop, drum and bass and techno and of different recording conditions or production techniques commonly encountered in modern recordings: use of acoustic or electronic drum kits, reverberation or distorsion effects, equalization and compression. The loop duration ranges from two to fifty seconds.

B_1 was manually annotated using eight basic categories: *bd* for bass drum, *sd* for snare drum, *hh* for hi-hat, *clap* for hands clap, *cym* for cymbal, *rs* for rim shot, *tom* for toms-toms and *perc* for all other percussive instruments. When two or more instruments are played at the same time, the event is labelled by all the corresponding categories (for example if bass drum and cymbal are hit simultaneously, both labels are attached to the corresponding stroke). Combinations of up to four simultaneous instruments exist in the database (although they are not frequent). B_2 was semi-automatically annotated by using a SVM classifier trained on B_1 (see [5] for more details about this classifier) - and then by manually correcting the recognition errors.

B_1 and B_2 were finally merged to build up the database used in this work.

2.3. Taxonomy

In theory, $2^n - 1$ combinations are possible by playing simultaneously the instruments from the $n = 8$ basic categories. In our database, after having discarded the combinations occuring less than 40 times, only 18 out of the 255 combinations were observed. The first taxonomy (*detailed taxonomy*) is defined when each stroke is characterized by a distinct label, among the 18 possible combinations. For a better analysis of the results, a simplified taxonomy is also defined: Each segment is annotated with only the most salient instrument, or the two most salient instruments. It is worth precising that the simplified taxonomy is only used to provide an additional interpretation of the results: practically, results for this simplified taxonomy are computed by grouping blocks from the confusion matrix obtained with the detailed taxonomy.

2.4. Segmentation and tempo extraction

The segmentation is obtained by applying an onset detection algorithm based on sub-band decomposition [11]. Concurrently, the overall tempo of the loop is estimated using the algorithm described in [1].

2.5. Features set

The features extracted from the audio signal include:

- **Mean of 13 MFCC** The mean of the Mel Frequency Cepstral Coefficients (MFCC) including c_0, calculated on 20 ms frames with an overlap of 50 % and averaged the coefficients over the stroke duration.

- **4 Spectral shape parameters** defined from the first four order moments.

- **6 Band-wise Frequency content parameters** These parameters correspond to the log-energy in six predefined bands (in Hertz: [10-70] Hz, [70-130] Hz, [130-300] Hz, [300-800] Hz, [800-1500] Hz, [1500-5000] Hz).

To eliminate correlations between some of these 23 parameters, a Principal Component Analysis is performed on the data set. The feature vector used as an input for the classifiers is thus a linear transformation of the features set mentioned above.

2.6. Classifiers

Our first paper [4] presented two classifiers: Hidden Markov Models (HMM) and Support Vector Machines (SVM). HMM took advantage of the short-term time dependencies of drum signals. Considering that the sequence of feature vectors observed is the output of a Hidden markov Model, the transcription task is equivalent to searching the most likely states (strokes) sequence, carried out using the traditional Viterbi algorithm. SVM basically does not take into account time dependencies, but provide very interesting generalization properties. Our article [5] introduced a new model in which time dependencies were taken into account in the SVM model. It consisted practically in replacing the feature vector of one stroke $(f_{1,n}, f_{2,n}, ..., f_{23,n})$ (see section 2.5) by a combined vector containing also the features of the previous stroke $(f_{1,n-1}, f_{2,n-1}, ..., f_{23,n-1}, f_{1,n}, f_{2,n}, ..., f_{23,n})$.

We propose several new improvements to these approaches.

2.6.1. SVM with probabilistic outputs and coupling

Support Vector Machines (see [16] for a detailed presentation) are typically used for discriminating two classes. However, our problem is a multi-class problem, each class being a combination of strokes (for example *bass drum* + *hi-hat* is one class). A classical implementation of SVM for such multiclass problems uses a *one versus one* approach also known as *pairwise classification* ([12]). Following this approach, $\frac{n(n-1)}{2}$ binary classifiers are trained, each of them discriminating a pair of class. If x is the input vector, (i, j) a pair of classes, (x_{ijk}) (resp. (v_{ijk})) the support vectors (resp. the weights), c_{ij} the parameter of the binary SVM classifier trained to discriminate the classes i and j, the decision function commonly used is:

$$f_{ij}(x) = \sum_k w_{ijk} K(x, x_{ijk}) + c_{ij} \quad (1)$$

$$\Omega_{ij}(x) = \begin{cases} i & \text{if } f_{ij}(x) > 0, \\ j & \text{otherwise} \end{cases} \quad (2)$$

To classify a stroke, the decisions of the $\frac{n(n-1)}{2}$ classifiers are aggregated by a simple vote counting (each Ω_{ij} being a vote).

This approach is not fully satisfying for two reasons. Firstly, vote-counting does not take into account the amount of confidence of each individual decision of the pairwise classifications. Secondly, this method does not provide any kind of probabilistic output: thus, it does not enable post-processing - for example, language modeling, or decision fusion.

Our first improvement consists in replacing the "hard" decision function $\Omega_{ij}(x)$ by a probabilistic one, which can be interpreted as a posterior probability $P_{ij}(class = i|x)$. Platt describes in [15] a method to obtain such posterior probabilities. The output of the SVM $f_{ij}(x)$ is mapped to the interval $]0, 1[$ with a sigmoid function: $D'_{ij}(x) = \frac{1}{1+e^{Af_{ij}(x)+B}}$. The parameters A, B are fit using maximum likelihood estimation on a subset of the training data.

The final decision is taken by coupling the pairwise probabilities given by each classifier, in order to compute a global probability for each class. This coupling is performed with the iterative algorithm presented by Hastie and Tibshirani in [7].

As a result, we obtain a posterior probability $P(class = i|x)$ which can be used for an additional post-processing stage, or for direct classification - in this case, the class that maximizes $P(class = i|x)$ is selected.

2.6.2. SVM with language modeling

N-grams Markov models provide an efficient way of modeling context (short-term) dependencies in drum playing ([4]). In these models, a succession of strokes $S_{k-m}, ..S_k$ is associated to each state q_t. Intuitively, the state q_t represents the stroke S_k in the context of $S_{k-m}..S_{k-1}$ at time t. The model is thus clearly context dependent. The transition probabilities from state i to state j are given by (in the case of 3-grams):

$$\begin{aligned} a_{ij} &= p(q_t = j|q_{t-1} = i) \\ &= p(s_t = S_3|s_{t-1} = S_2, s_{t-2} = S_1) \end{aligned}$$

The transition probabilities a_{ij} can be estimated by counting occurrences of each N-gram in the training database.

Traditionally, such models use mixtures of Gaussian distributions to model the observation probability associated

Taxonomy	Detailed	Simplified
HMM, 3-grams, 2 mixtures	60.5% (4.3%)	79.3%
HMM, 4-grams, 2 mixtures	59.5% (3.5%)	77.7%
SVM	70.6% (2.5%)	86.5%
SVM **prob**	70.7% (2.6%)	86.4%
SVM **ctxt**	72.4% (2.7%)	89.1%
SVM **ctxt prob**	72.6% (2.4%)	89.9%
SVM **prob lang**	75.5% (2.8%)	88.0%

Tab. 1. Drum loop transcription results

to each state. Employing such distributions results in overfitting when a large number of mixtures is used; while a smaller number of mixtures cannot efficiently represent the complex decision surface between classes.

An alternative approach is to use the probabilistic output of our SVM classifier to estimate the probability that a stroke performed at time t corresponds to a given state of the model. The probabilistic information given by the recognition of each individual stroke with the SVM classifier, and the context information obtained with the language model are both taken into account to choose the most likely sequence of strokes. This is done using the classical Viterbi algorithm.

2.7. Results

A 10-fold cross-validation approach was followed. It consists in splitting the whole database in 10 subsets, training the classifier on nine of them, and keeping the last subset for evaluation. The procedure is then iterated by rotating the 10 subsets used for training and testing. The results are summarized in table 1. Standard deviations were computed using the cross-validation variance estimator $\hat{\theta}_3$ presented in [2] and are given in the table. Modified SVM models have the following labels: **ctxt** when contextual features are used, **prob** when probabilistic outputs and coupling are used, **lang** for language modeling (trigrams).

It can be seen that the best results are obtained with the SVM classifiers. The use of probabilistic outputs and coupling does not significantly improve the performances. It can be explained by the fact that our problem involves a rather large number of classes $N = 18$, allowing a good level of accuracy even with a simple voting scheme. Thus, it seems that the use of SVM with probabilistic outputs and coupling is relevant only when the number of classes is smaller, or when the results need to be post-processed.

The use of SVM with a language-modeling stage increases the recognition performances for the detailed taxonomy; but does not give the best results for the simplified taxonomy. A further analysis of recognition errors shows that language modeling allows a more accurate discrimination of simple and compound strokes (especially the presence or absence of hi-hats), but fails to recognize unusual or rare combinations of strokes. For example, *bass drum* and *bass drum + hi-hat* are less likely to be confused, since the language modeling incorporates information about whether or not a hi-hat is played in the sequence; while *rim shot + hi-hat*, which is much less common than *snare drum + hi-hat*, is very likely to be classified as this first stroke.

Instrument	Onomatopoeia
Bass drum	[pum] / [bum]
Cymbal, hi hat	[ti] / [ts]
Snare drum,	[tʃa]
Snare drum + Bass drum mixture	[ta]
Tom, other percussive instrument	[do] / [dɔm] / [tɔm]

Tab. 2. Language used for spoken queries

3. RECOGNITION OF ONOMATOPOEIA IN SPOKEN QUERIES

3.1. Onomatopeia set

While several rhythmic instruments such as North Indian Tabla have a well-defined set of onomatopoeia (known as *bols* in the case of Tabla) denoting each stroke of the instrument, there is no commonly accepted set of vocables to denote the instruments of the drum kit. This can be explained by the fact that notation plays a more important role than oral tradition in the transmission and teaching of Western popular music.

A possible approach, used by Kapur et al. in their *Bionic BeatBox Voice Processor* [10], is to let the users freely use their own set of onomatopoeia, after having trained the system by providing a few examples of each vocable.

We followed a different approach in which we imposed a set of onomatopoeia to the user. The set chosen for our work is given in the table 2. It has been validated by a perception experiment ([5]) which consisted in randomly playing a drum stroke, and in asking the subjects to pick the onomatopoeia that best described it.

3.2. Recognition of spoken onomatopoeia

In order to train and evaluate the recognition of spoken onomatopoeia in a speaker-independent way, a new database was recorded from 13 speakers, 11 males and 2 females. Most of these speakers practice music regularly, 2 of them practicing electronic music and DJing. The database was recorded according to the following protocol: During an introductory stage, the subject was presented the different instruments of the drum kit and the vocabulary used. During a

first recording stage, a computer animation displayed a random sequence of onomatopoeia, and the subject was asked to pronounce each onomatopoeia as soon as it flashed on the screen. During a second stage, the subject was asked to "perform" or "beatbox" four simple sequences. The voices were recorded using a Shure WH20 headworn directional microphone on an Edirol UA-5 soundcard, at 44.1 kHz.

This corpus was manually segmented and annotated. The annotation includes onomatopoeia ([pum], [ta]...), silences, and a last category for miscellaneous events such as breathes or pops. The entire database contains 1057 utterances.

Training, recognition and evaluation was performed using the HTK Speech Recognition Toolkit. The features used for the recognition are the 13 MFCC + 13 ΔMFCC + 13 $\Delta\Delta$MFCC. Each onomatopoeia is represented by a Bakis (left-right) HMM model with 3 states, at the exception of the silence model which uses 4 states and a different topology. The probability distribution associated to each state is a mixture of 3 gaussians - using a higher number of mixtures resulted in overfitting. These HMM models are trained for each onomatopoeia using the EM algorithm. Given a simple "task grammar" to model the succession of silences and vocal activity (onomatopoeia), all the models were connected to form a network, on which the recognition is performed with the Viterbi algorithm. The output of this query transcription system is a sequence of pairs (t_i, S_i), where S_i is the stroke (or compound stroke, like *bass drum + snare drum*) played at time t_i. This output is post-processed by removing the silence labels, the onomatopoeia shorter than 100ms, and by replacing the recognized onomatopoeia by the rythmic instrument it represents - for example [pum] is replaced by *bass drum*.

3.3. Evaluation

This query recognition system was evaluated using a leave-one-speaker-out validation protocol. This protocol consists in dividing the anotated corpus in $K = 13$ subsets, each subset containing the utterances of a given speaker. The recognition model is trained on $K - 1$ of them, and the last subset is used for evaluation. By rotating each subset, the data recorded for each speaker is used $K-1$ times for training, and once for evaluation.

Once a transcription output was obtained for each of the original utterances, these transcriptions were analyzed and compared to the reference transcriptions. More precisely, the original and output transcriptions were matched using a dynamic programming algorithm. A label insertion or deletion carry a score of 3.3, a label substitution carries a score of 4. The label alignement with the lowest score is found, and the number of substitution (S), insertion (I), deletion (D) errors is counted. Then, the accuracy of the transcription for a total of N onomatopoeia is given by:

$$Accuracy = \frac{N - S - I - D}{N}$$

The accuracy of our speaker independent system is 84.4%.

4. QUERY SCORING AND ALIGNING

4.1. Statistical modeling of interpretation errors

Query by humming systems often use approaches based on string matching. These approaches are not suitable for the scoring of drum queries, for two reason. Firstly, the notion of melody and melodic contour is not relevant when dealing with drum loops. Secondly, most of these approaches are ignoring the rythmic information and only focus on the intervals between notes - a criterion which cannot be defined for drum sounds. On the other hand, tempo or beat histogram features are not sufficient to accurately represent the rythmic information - for example the way snare drums and bass drums are played on downbeats and upbeats.

We consequently chose a novel approach based on a generative statistical model of the loop interpretations. As such, the query task can be reformulated as "find the loop(s) in the database that is (are) most likely a performance with real drums instruments of the interpretation given by the spoken onomatopoeia". This model takes into account the various editing operations likely to occur when a complex rhythmic phrase is interpreted with onomatopoeia: the non-formulation of a stroke contained in the loop (*deletion*), the formulation of a stroke which is not contained in the searched loop (*insertion*), and the approximative formulation (*substitutions*) of a note contained in the searched loop, possibly with timing errors (*alignment*). It allows the computation of the probability that a query is actually a good formulation of one of the loops contained in the database, in other words the likelihood of the interpretation q knowing the loop l. The sequence of editing operations e made by the user when performing the searched loop is considered as a hidden variable:

$$P(q|l) = \sum_e P(q, e|l)$$

Our model is described in details in [5]. It is parametrized by the likelihood of the interpretation of each drum sound b, knowing that it is not present in the loop $P(\{b\}|\emptyset)$ (insertion of strokes not present in the original loop), the likelihood of the deletion of each drum sound a, knowing that it is present in the loop $P(\emptyset|\{a\})$ (non-formulation), a probability distribution for the timing errors $P_a(t)$ from which can be derived the likelihood of a timing error of t between a stroke and its interpretation, and a distribution for the duration of deleted (resp. inserted) strokes $P_d(t)$ (resp. $P_i(t)$). These parameters can be empirically chosen to reflect common mistakes made when vocally performing a rhythm (such as ignoring

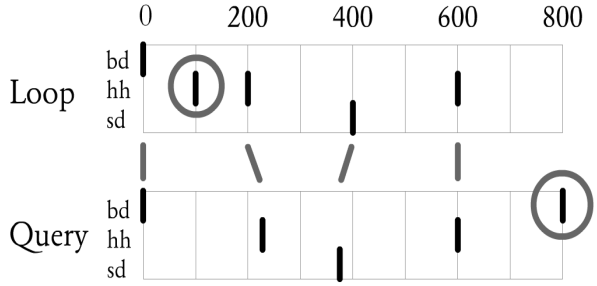


Fig. 2. Interpretation of a loop

e_i	$e_{Li}(l,q)$	$e_{Qi}(l,q)$
align.	$(\{bd\},0)$	$(\{bd\},0)$
deletion	$(\{hh\},100)$	$\emptyset$
align.	$(\{hh\},200)$	$(\{hh\},220)$
align.	$(\{sd\},400)$	$(\{sd\},390)$
align.	$(\{hh\},600)$	$(\{hh\},600)$
insertion	$\emptyset$	$(\{bd\},800)$

Tab. 3. Corresponding editing operations

hi-hats, or snare drum flams), or learned by gathering statistics from original drumloops and their vocal interpretations.

We define $P((t,B)|(u,A))$ as the likelihood that a combination of strokes B at time t is the interpretation of a combination of strokes A occuring at time u. If we consider that time-aligning errors are independent of the confusions between strokes, it can be expressed as: $P((t,B)|(u,A)) = P(B|A)P_a(|t-u|)$, where $P_a(|t-u|)$ is the likelihood of a timing error equal to $|t-u|$ between two events. Using the same notations, $P((t,B)|\emptyset)$ is the likelihood of an insertion of a stroke B, and $P(\emptyset|(u,A))$ is the likelihood of the deletion of a stroke of duration d.

Finally:

$$P(q,e|l) = \prod_i P(e_{Qi}|e_{Li})$$

where the sequences $(e_{Qi})_{i\in[1,E]}$ and $(e_{Li})_{i\in[1,E]}$ describe the alignment resulting from the editing operations e on the loop L and the vocal query Q (refer to figure 2 for an example of interpretation, and the corresponding values of e in 4.1).

The aim of the alignment between the loop and the interpretation is to find the sequence of edit operations e^* maximizing the likelihood of $P(q,e^*|l)$. The search of such an optimal alignment is possible with dynamic programming, and can be efficiently implemented by computing log-likelihoods rather than likelihoods.

4.2. Tempo and loop start alignment

In the maximization computed previously, we assumed that the query was an interpretation of the whole loop. However, it is likely that the query is just an interpretation of a short fragment located at any time offset within the loop. This problem is solved by searching the optimal alignement for a range of time offset and loop durations.

Finally, it is also necessary to deal with the fact that the query is not always formulated at the same tempo as the searched loop. In our previous approach, an optimal alignement was searched for a discreet set of tempo scaling factors, and it resulted in a *tempo independent distance*. The distance used in this article is slightly different since it also incorporates a penalty on the tempo difference: $D = D_{\text{tempo independent}} + C|\log \textit{tempo scaling}|$. The parameter C can be modified to find a trade-off between a tempo independent search based only on the contents of the loop, and a tempo-dependent search that will emphasize on the absolute time structure of the rhythm rather than on its contents.

4.3. Query and comparison

For a query d, given a threshold τ, the matching candidates are:

$$C(\tau,q) = \{L, D(q,L) < \tau\}$$

A model similar to this one can be used to compare two loops from the database. The likelihoods $P(l_1|l_2)$ expressing the substitution cost between two strokes have been symmetrised so that the measure D provided by the recursion can be interpreted as a distance. Not only this allows the grouping of results by similarity, by it also allows query by example - in the case, the example playing the role of the vocal query.

4.4. Evaluation

In order to evalute the query system, the following procedure was iterated $N = 500$ times:

1. A loop l_i was randomly selected from the database.
2. A segment q_i was randomly extracted from this loop; its length varying from 3 to 8 seconds.
3. A query was synthesized by concatenating onomatopoeia contained in a test database (compound of 80 instances of each of the onomatopoeia). This query contains time alignment mistakes, substitutions, deletions and insertions.
4. This query was transcribed by the onomatopoeia recognition system.
5. The loops giving the best score were searched and selected, using a given threshold τ.

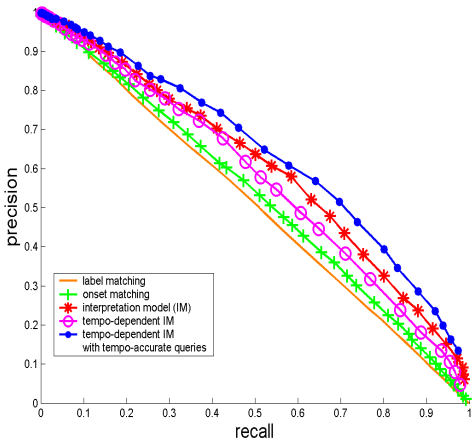


Fig. 3. Precision / Recall curves

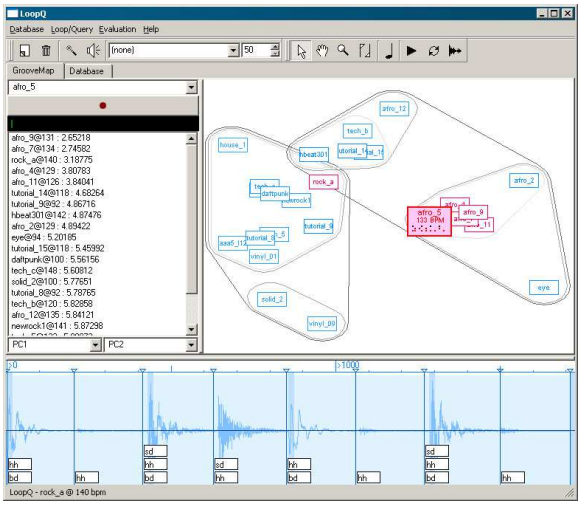


Fig. 4. User interface of the **LoopQ** application

We used the traditional information retrieval performance measures: precision and recall. For each value of the threshold τ, a pair of precision/recall values can be computed by averaging the precision/recall ratios of each single query. Since in our case only one loop is to be retrieved, the recall of a single query is 0 if the loop searched is not present in the set of matches; 1 if it is present. The precision of a single query is 0 if the loop searched is not present in the matches; $1/N$ where N is the number of matches otherwise.

$$Recall(\tau) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{C(q_i,\tau)}(l_i)$$

$$Precision(\tau) = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{1}_{C(q_i,\tau)}(l_i)}{|C(q_i,\tau)|}$$

Several sets of results were obtained, from which precision/recall curves were plotted (figure 3). A first set was obtained using a simple string matching algorithm, that is to say, only the contents of the loop was considered, without regard to the temporal information (label matching). Reversely, the second set was obtained using a distance D taking into account only relative temporal information (onset matching). The third set was obtained with the distance used in our previous work (interpretation model). The fourth set was obtained with a distance taking into account both the rhythmic contents and the tempo information. Finally, the fifth set was obtained using the same protocol and distance as previously, except that the queries were performed at exactly the same tempo as the searched loop.

It can be clearly observed that our interpretation model outperforms label or onset matching approaches. Incorporating tempo information can also improve the overall performance of the retrieval system, provided the queries are formulated at the exact tempo - a condition that can be reasonably satisfied if a click track is played in the background when the user records a query.

5. IMPLEMENTATION

All the modules presented in this paper are integrated in a graphical application, **LoopQ**, developed in C++ with the Qt library. Users can submit vocal queries by clicking the *record* button. The vocal input is subsequently recognized, displayed in the bottom of the screen with tags corresponding to the recognized onomatopoeia, and submitted as a query. At this stage, it is also possible to generate a synthetic drum loop by replacing each onomatopoeia by the corresponding drum sample.

The loops matching the queries are displayed on the left pane, sorted by similarity. The right pane displays the 25 best candidates in a 2D plane. Several axis can be selected to visually group the results: tempo, complexity (number of drum events per second), density (number of drum events per bar), and the 3 first axis obtained by multi-dimensional scaling (MDS) of the resulting data set - using the similarity measure. By default, the first axis obtained by MDS are selected, allowing a visual grouping of similar loops. Each loop is represented by a box containing its name.

Different kind of interactions are possible with this representation. Moving the mouse cursor on a box zooms it, and displays additional information about the loop, such as its tempo and a transcription of its first bar. Clicking on a box plays the corresponding loop. Right-clicking performs a query, using the pointed loop as an example. This allows the user to perform incremental searches and navigate in the database the same way one would follow hyperlinks on

the World Wide Web. An additional interaction mode, the *Jam* mode, specific to DJing uses, allows a continual sound feedback: whenever the mouse cursor hovers over a box, the corresponding loop is continuously played, until another loop is pointed.

6. CONCLUSION AND FUTURE WORK

Content-based indexing and querying systems are necessary to assist composers and DJs, who use large collections of sound files daily. This paper presented an innovative system for indexing and querying drum loops, and its recent improvements. New SVM classifiers, and hybrid approaches using HMM and SVM were experimented, on a larger database, resulting in a 75.5% correct recognition rate for the drum loop transcription task with a detailed taxonomy. Better results could be achieved by using more complex language models than the trigram Markov models presented here - for example by taking into account the cyclic and repetitive characteristics of rhythmic sequences, or by making a better use of time and duration information.

A speaker-independent onomatopoeia recognition front-end has been successfully integrated and gives a 84.4% accuracy. At this stage, further usability experiments should be conducted with drummers and DJs, to evaluate how this recognition front-end deals with the different onomatopoeia used. It is very likely that each drummer or DJ uses his own vocabulary. However, this does not invalidate our intuition that vocal input is one of the most efficient modality to specify rhythmic queries.

Finally, further works will focus on the detection on drum events in polyphonic music signals - our goal being to index not only drum loops, but also the drum tracks of entire songs.

7. REFERENCES

[1] M. Alonso, B. David, and G. Richard. A study of tempo tracking algorithms from polyphonic music signals. In *Proceddings of 4th COST276 Workshop, Bordeaux, France*, march 2003.

[2] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. CIRANO Working Papers 2003s-22, CIRANO, May 2003. available at http://ideas.repec.org/p/cir/cirwor/2003s-22.html.

[3] A. Ghias, J.Logan, D. Chamberlin, and B.C. Smith. Query by humming: Musical information retrieval in au audio database. In *Proceedings of ACM Multimedia'95*, 1995.

[4] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Proceedings of the IEEE ICASSP 2004 Conference*, May 2004.

[5] O. Gillet and G. Richard. Drum loops retrieval from spoken queries. In *Journal of Intelligent Information Systems*, To be published 2005.

[6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 287–288, October 2002.

[7] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*, volume 10, 1998.

[8] P. Herrera, X. Amatriain, E. Battle, and X. Serra. Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proceedings of ISMIR2000*, 2000.

[9] P. Herrera, A. Dehamel, and F. Gouyon. Automatic labeling of unpitched percussion sounds. In *Proceedings of the 114th AES convention*, March 2003.

[10] A. Kapur, M. Benning, and G. Tzanetakis. Query by beatboxing: Music information retrieval for the dj. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, October 2004.

[11] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.

[12] U. H.-G. Kressel. Pairwise classification and support vector machines. In *Advances in kernel methods: support vector learning*, pages 255–268. MIT Press, 1999.

[13] R.J. McNab, L.A. Smith, D. Bainbridge, and I.H. Witten. The new zealand digital library melody index. In *D-Lib Magazine*, 1997.

[14] T. Nakano, J. Ogata, M. Goto, and Y. Hiraga. A drum pattern retrieval method by voice percussion. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, October 2004.

[15] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classiers*, pages 61–74, 2000.

[16] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

AUTOMATIC TRANSCRIPTION OF DRUM SEQUENCES USING AUDIOVISUAL FEATURES

Olivier Gillet and Gaël Richard

GET-ENST (TELECOM Paris)
Signal and Image Processing department
46, rue Barrault, 75013 Paris, France
[olivier.gillet, gael.richard]@enst.fr

ABSTRACT

The transcription of a music performance from the audio signal is often problematic, either because it requires the separation of complex sources, or simply because some important high-level music information cannot be directly extracted from the audio signal. In this paper, we propose a novel multimodal approach for the transcription of drum sequences using audiovisual features. The transcription is performed by Support Vector Machines (SVM) classifiers, and three different information fusion strategies are evaluated. A correct recognition rate of 85.8% can be achieved for a detailed taxonomy and a fully automated transcription.

1. INTRODUCTION

As a consequence of the exponentially growing amount of available digital data, automatic indexing and retrieval of information based on content is becoming more and more important and represent very challenging research areas. Automatic indexing of digital information allows to extract a textual description of this information (i.e. meta data). In the context of music signals, or audiovisual signals of music performances, such a description would ultimately be a complete transcription - in the form of a detailed musical score. Even if promising results have been achieved in the field of music transcription, several problems still need to be addressed in order to design systems powerful enough to obtain a complete and perfect representation of high-level musical information. The transcription task becomes very complex when the problem of source separation arises, especially because the number of sounds played simulatenously remains unknown. Moreover, many parameters related to expressiveness, style or playing technique cannot be easily extracted from the audio signals, but are easier to extract from a video signal of the instrumentist.

In this paper, we describe and evaluate a novel multimodal approach in which video signals recorded by a camera filming a drummer are analyzed in order to enhance the transcription of the performance. This work is a follow-up of a previous study conducted on drum loops transcription [1] where only audio features were used. It is important to note that we ultimately aim at the indexing of existing audiovisual recordings of music performances, a task for which it is impossible to use specific instrumentation such as sensors, or to control the recording conditions in such a way that scene recognition will be performed more easily (for example by using coloured sticks or gloves, or a neutral background). To our knowledge, there is no prior works related to the transcription of music using directly a multimodal approach. However, researches have been carried out in the analysis of the correlation between video and audio sources, for various purposes such as computer human interaction, biometrics, or video indexing. In [2], Smaragdis and Casey present an application of Independent Component Analysis to the extraction of audiovisual features from a video stream, and give a simplified musical example of fingers on a piano keyboard. In [3] Fisher and Darell present various statistical model for joint audio/video analysis, especially for tasks such as speaker localization in video scenes. The computer-vision part our problem has a few similarities with the problem of gesture analysis [4]. In [5], Murphy presents a computer-vision system for tracking a conductor's baton. In [6], Wanderley shows how an expressiveness parameter can be derived from the angle of a clarinet with respect to the performer. Finally, Dahl conducted numerous multimodal experiments showing the relationship between body movements and emotions in marimba performances or the correlation between video features and musical accent [7] in drumming.

The paper is organized as follows. The next section describes the overall system architecture. Section 3 presents the database specifically recorded for this work. Then, section 4 is dedicated to the description of the video features extraction. The different statistical classification approaches tested are presented in section 5. Section 6 discusses the results obtained and, finally, section 7 suggests some conclusions and future directions.

2. SYSTEM ARCHITECTURE

The system aims at transcribing audiovisual drum sequences into a higher level representation consisting of a list of pairs (onset time, instrument of the drum kit played). It is built on a previously developed audio-only transcriber presented in [1].

2.1. Previous audio transcription system

The audio-only transcription system on the top of which the audiovisual extension was built incorporates 3 modules, namely:

- **A segmentation and tempo extraction module.** These parameters were obtained by applying an onset detection algorithm based on sub-band decomposition [8].
- **A features extraction module.** The features extracted from the audio signals include: The **mean of 13 Mel Frequency Cepstral Coefficients** including c_0, calculated on 20 ms frames with an overlap of 50 % and averaged over the stroke duration ; **4 spectral shape parameters** defined from the

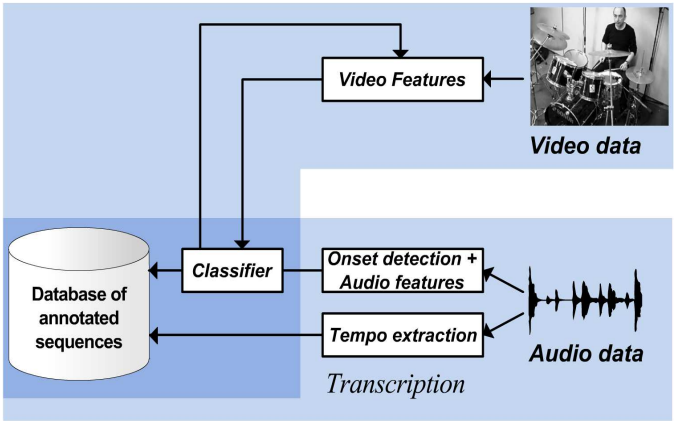


Fig. 1. System architecture

first four order moments ; and **6 Band-wise frequency contents parameters** corresponding to the log-energy in six pre-defined bands (in Hertz: [10-70] Hz, [70-130] Hz, [130-300] Hz, [300-800] Hz, [800-1500] Hz, [1500-5000] Hz).

- **A classification module** for which several classifiers (Hidden Markov Models, Support Vector Machines) were tested.

2.2. Audiovisual transcription system

The extensions and improvements of the previous system which are presented in this work include:

- **A new audiovisual database,** detailed in the next section.
- **A new set of features extracted from the video track.** Because the computation of the video features requires a calibration of the scene, the output of a transcription carried out on the sole audio signal can be used to derive a set of video features that will subsequently enhance the transcription. Alternatively, the user can manually calibrate the system.
- **New classification approaches.** Some of the classifiers presented in our previous work are no longer suitable to the taxonomy and size of the new database. Moreover, several classification and information fusion schemes to deal with the availability of the two audio and video information sources were to be evaluated.

Because audio signals of drum instruments have very sharp onsets, it is easier to detect the start time and duration (T, d) of each stroke in the audio domain than in the video domain.

The overall architecture of the resulting system is depicted in figure 1.

3. DATABASE

Since no audio/video database of drum performances was available, we recorded our own database which consists of 35 sequences containing 2170 strokes. The sequences were played on a drum kit made up of 9 instruments: a bass drum, a snare drum, three toms (high, medium, low), one hi-hat cymbal, two crash cymbals and one ride cymbal. In order to increase the variability of the recorded data, the sequences were performed with two sets of sticks: classic sticks and "bundle sticks" - small wood rods bundled together. Four studio-quality microphones were used: one for the bass drum, one for the snare drum, and two overhead microphones. In the scope of this work, the audio signals were recorded at the stereo output of the mixing desk, at a sample rate of 48 kHz, and converted into mono by combining the right and left channels.

The video signals were recorded with a Canon XL1 professional DV camera. The camera was fixed on a tripod and remained steady during the whole recording. The video was recorded in DV format with a resolution of 720x576, at 25 frames per second. For the purpose of this work, only the luminosity channel of the video was processed. Moreover, since the DV format is interleaved, scanline artifacts were removed with simple spatial filtering. As our goal is the indexing of pre-recorded material, we avoided using any specific sensor or, visual clues such as coloured gloves, sticks or backgrounds to improve the detection, even if the recording conditions for this database were well controlled.

An intermediate annotation was at first obtained with our previous audio based transcription system ; and secondly, this annotation was corrected and refined. It is worth precising that despite the similar instrument set used, the taxonomy used in this work is slightly different and detailed than in [1]. For example, a tom (resp. cymbal) stroke will not be labelled as **tom** (resp. **cymb**) but as **low tom**, **mid tom**, **high tom** (resp. **crash cymbal 1**, **crash cymbal 2**, **ride cymbal**).

As a result, each acoustic event is labelled with the corresponding instrument or combination of instruments when several instruments are played at the same time (for example if the bass drum and the ride cymbal are hit simultaneously, both labels are attached to the corresponding stroke).

4. VIDEO FEATURES

4.1. Masks

We observed that when an instrument of the drum kit is played, two kinds of visual clues can be derived from the video: the motion of the sticks, or any specific gesture the drummer has to perform to hit the instrument (for example, kicking the pedal of the bass drum) ; and the motion of the instrument itself, or the vibration of its membrane.

Thus, two areas of the video images are defined for each instrument: an area in which motion is associated to the gesture performed by the drummer to hit the instrument, and an area in which motion is associated to the vibration of the instrument itself once hit. We subsequently use two 2D weighting masks $M_{gesture}(x, y)$ and $M_{instr}(x, y)$ to represent these areas.

The thresholded difference sequence was used as a simple motion estimator. If $V(x, y, t)$ is the sequence of video images, the thresholded difference sequence $D(x, y, t)$ is given by:

$$D'(x, y, t) = |V(x, y, t) - V(x, y, t-1)| \quad (1)$$

$$D(x, y, t) = \begin{cases} D'(x, y, t) & \text{if } D'(x, y, t) > S, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For each instrument, and each stroke starting at frame T, the duration of which is d frames, two features are computed from the thresholded difference sequence and the weighting masks:

- The intensity of motion in the gesture mask, accross a short time interval centered on the beginning of the stroke.
  $I_{gesture} = \sum_{t \in [T-\delta, T+\delta]} \sum_{x,y} M_{gesture}(x,y) D(x,y,t)$
  Typical value for δ is $\delta = 2$.
- The intensity of motion in the instrument mask, accross the whole duration of the stroke.
  $I_{instr} = \sum_{t \in [T+\delta, T+d-\delta]} \sum_{x,y} M_{instr}(x,y) D(x,y,t)$

This results in a set of 18 features computed for each stroke: The $I_{gesture}$ and I_{instr} features for each of the 9 instruments of the kit.

4.2. Calibration

The system is calibrated by defining the 18 masks. Different calibration schemes are devised:

- *Manual*. A human operator manually defines the image regions corresponding to each instrument of the kit.
- *Automatic*. A transcription is obtained using the audio-only transcription system. This transcription is used to generate a mask, by averaging the difference sequence accross the appropriate interval and all the recognized occurences of each instrument of the kit.

5. CLASSIFICATION

5.1. Information fusion

The fusion of video and audio information is performed by three different fusion approaches:

- **Joint features vectors.** Let x_{audio} (resp. x_{video}) be the audio (resp. video) features vector. Classifiers are trained with joint features vectors:

  $$x_{joint} = [x_{audio}(1)...x_{audio}(25)x_{video}(1)...x_{video}(18)]$$

- **Best of unimodal experts.** Two classifiers are trained, one using the audio features, the other the video features. For each stroke, the output of the classifier giving the best confidence score is kept. For instance, the video classifier is used only when the audio classifier produces an uncertain result. The advantage of this approach is that it allows the use of a larger database for audio transcription, and a smaller, specific database adapted to the current scene and camera angle for the video transcription.
- **Fusion.** As above, two classifiers are trained except that these classifiers produce for each class 2 probabilities:
  $P(class|x_{audio})$, $P(class|x_{video})$.
  Each stroke is labelled with the class that maximizes the product of these two probabilities.

As some of the parameters are correlated, especially when joining video and audio features, a Principal Component Analysis is performed on the fused data set when the joint feature vectors approach is chosen, or on the separate audio and video datasets when another approach is chosen.

5.2. SVM classification

It was shown in [1] that Support Vector Machines (SVM) were well suited for drum loops transcription and are therefore used in this study.

In our work, we use the "one versus one" approach, in which $\frac{n(n-1)}{2}$ binary SVM classifiers are trained, each discriminating between a pair of classes. If x is the input vector, (i,j) a pair of classes, (x_{ijk}) (resp. (v_{ijk})) the support vectors (resp. the weights), c_{ij} the parameter of the binary SVM classifier trained to discriminate the classes i and j, the decision function commonly used is :

$$f_{ij}(x) = \sum_k w_{ijk} K(x, x_{ijk}) + c_{ij} \quad (3)$$

$$D_{ij}(x) = \text{sgn} f_{ij}(x) \quad (4)$$

The input vector x will be classified as i (resp. j) if $f_{ij}(x)$ is positive (resp. negative).

However, to obtain a confidence measure, a specific decision function is defined: the output f_{ij} is mapped to the interval $]0,1[$ with a sigmoid function: $D'_{ij}(x) = \frac{1}{1+e^{Af_{ij}(x)+B}}$

Provided that appropriate values of the parameters A, B are chosen [9], this quantity can be interpreted as an a-posteriori probability $P_{ij}(\text{class} = i|x) = D'_{ij}(x)$. The final output of the classifier is a probability for each class, computed by coupling the pairwise probabilities using the algorithm proposed by Hastie and Tibshirani in [10]. The class assigned to the input x is the one that maximizes the quantity $P(\text{class} = i|x)$, which can be used itself as a probabilistic measure of the accuracy of the classification. This method gives similar results, and a much better ranking function, than more classic approaches using voting and vote counting.

In the scope of this study, a radial basis kernel was chosen: $K(x,y) = \exp^{-\gamma||x-y||^2}$ where γ is equal to the inverse of the number of features. The library LibSVM [11] allowed an easy implementation of these SVM classifiers with a modified output.

6. RESULTS

6.1. Evaluation protocol

Two main experiments were conducted on our dataset. In the first experiment, the video features were computed with a mask manually drawn on the picture. In the second experiment, the video features were automatically computed from an automatic audio-only annotation of the database. Example of computed masks are provided in figure 2. One can also check and correct the automatic transcription used as a preliminary step for the calibration in this second experiment.

For each of these experiments, we compare the recognition rate obtained with different feature sets and fusion schemes. **Blind** is the recognition rate obtained using only audio features. **Deaf** is the recognition rate obtained using only video features. **Joint features**, **Fusion** and **Best expert** are the recognition rates obtained using a combination of video and audio features.

A K-fold cross-validation approach was followed. It consists in splitting the whole database in $K = 5$ subsets, training the classifier on four of them, and keeping the last subset for evaluation. The procedure is then iterated by rotating the 5 subsets used for training and testing.

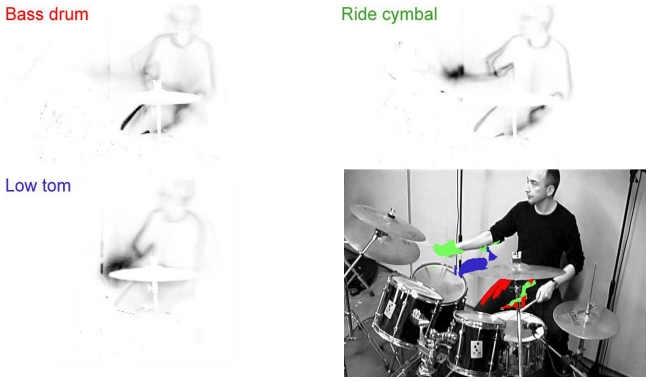


Fig. 2. Examples of computed masks: gesture for bass drum (the pedal is kicked by the right foot), gesture for the cymbal at the right of the drummer, gesture for the low tom at the right of the drummer, and reference image.

	Manual	Automatic
Deaf	67.7%	64.0%
Best expert	82.7%	82.1%
Fusion	84.3%	82.7%
Joint features	**86.7%**	**85.8%**
Blind	81.5%	81.5%

Table 1. Drum instruments recognition results

6.2. Results and discussion

Our classifier using only audio features as presented in [1] managed to cope with a lot of variability in the dataset and complex situations like effects or overlapping strokes. Not surprisingly, it performs well on this simpler dataset, in which only one drum kit is used. Another interesting point is that the set of audio features chosen in our previous work is still relevant for this classification task which uses a more detailed taxonomy.

The increased recognition rate obtained with a combination of audio and video features validates our multimodal approach, however, the **Best expert** strategy in which the most reliable of the information sources is used does not give the best results. This can be explained by the fact that processing the audio and video data in the same classifier allows to take advantage of their correlation. Especially, the PCA step is very important since it forges truly multimodal features.

It is worth precising that these comparisons are relevant only if the variance of the K-fold cross-validation is small enough. However, estimating this variance is difficult. More precisely, because of our limited dataset, there was a high variability in the estimations obtained by the different estimators presented in [12] ; using the estimator $\hat{\theta}_3$, the standard deviation is 2.1%.

7. CONCLUSION AND FUTURE WORK

This paper presented a novel approach to enhance the transcription of drum sequences using audio and video features. The system can work without calibration, even if the best results, a correct recognition rate of 86.7%, are obtained with manual calibration. The overall gain of our multimodal approach, is still limited in the context of the well controlled database used. Future work will in fact consider more complex situations including the transcription of drum signals when other instruments are playing along with the drummer. This could validate the hypothesis that video features will drastically improve the transcription results, in situations when separating the audio sources will become impossible. More robust video features will also have to be tested, as well as sequence models (Hidden Markov Models) based on joint video/audio features.

8. ACKNOWLEDGEMENTS

The authors wish to thank Michel Desnoues for having performed and recorded the sequences used in this work.

9. REFERENCES

[1] O. Gillet and G. Richard, "Automatic transcription of drum loops," in *Proceedings of the IEEE ICASSP 2004 Conference*, May 2004.

[2] P. Smaragdis and M. Casey, "Audio/visual independent components," in *Proceedings of International Symposium on ICA and Blind Source Separation*, april 2003.

[3] J. W. Fisher and T. Darrell, "Signal level fusion for multimodal perceptual user interface," in *Proceedings of Workshop on Perceptive User Interfaces*, october 2001.

[4] M.M. Wanderley and M. Battier, *Trends in Gestural Control of Music*, Ircam - Centre Georges Pompidou, 2000.

[5] D. Murphy, "Tracking a conductor's baton," in *Proceedings of 12th Danish Conference on Pattern Recognition and Image Analysis 2003*, 2003.

[6] M. M. Wanderley and P. Depalle, "Gesturally-controlled digital audio effects," in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, December 2001.

[7] S. Dahl, "The playing of an accent - preliminary observations from temporal and kinematic analysis of percussionists," in *Journal of New Music Research*, 2000, vol. 29(3), pp. 225–234.

[8] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.

[9] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classiers*, 2000, pp. 61–74.

[10] Trevor Hastie and Robert Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems*, 1998, vol. 10.

[11] C.C. Chang and C.J. Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] Yoshua Bengio and Yves Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," CIRANO Working Papers 2003s-22, CIRANO, May 2003, available at http://ideas.repec.org/p/cir/cirwor/2003s-22.html.

ENST-Drums: an extensive audio-visual database for drum signals processing

Olivier Gillet and Gaël Richard
GET / ENST, CNRS LTCI, 37 rue Dareau, 75014 Paris, France
[olivier.gillet, gael.richard]@enst.fr

Abstract

One of the main bottlenecks in the progress of the Music Information Retrieval (MIR) research field is the limited access to common, large and annotated audio databases that could serve for technology development and/or evaluation. The aim of this paper is to present in detail the ENST-Drums database, emphasizing on both the content and the recording process. This audiovisual database of drum performances by three professional drummers was recorded on 8 audio channels and 2 video channels. The drum sequences are fully annotated and will be, for a large part, freely distributed for research purposes. The large variety in its content should serve research in various domains of audio signal processing involving drums, ranging from single drum event classification to complex multimodal drum track transcription and extraction from polyphonic music.

Keywords: Research database, Automatic drum transcription, Drum event detection in polyphonic music, Source separation, Multimodal music transcription.

1. Introduction

The field of Music Information Retrieval (MIR) is receiving an ever growing interest from the research community, leading to numerous new approaches and algorithms to solve specific indexing and retrieval problems. However, one of the main bottlenecks in this field is the limited access to common, large and annotated audio databases that could serve for both technology development and evaluation. McGill University Master Samples (MUMS)[1], IRCAM Studio Online collection (SOL) [2], and the University of Iowa Musical Instrument Samples [3] are three examples of such databases. Although they are limited to isolated notes, they are widely used by the community, especially for musical instrument recognition tasks. More recently, a large and remarkable database, the RWC Music Database [4], was built and distributed by the Real World Computing Partnership of Japan. As for percussive instruments and drum processing in particular, no large database is publicly available, although several interesting private databases have been built internally by several teams and used in a recent evaluation campaign. For example, the database used for the MAMI drum transcription project [5] has been used during the latest MIREX campaign.

To cope with the limitations of the previous databases for drum signal processing, a large audiovisual drum database was recorded and fully annotated, in order to cover as many applications as possible in the general framework of automatic drum signal analysis. For this purpose, three professional drummers were recorded on eight audio tracks and simultaneously filmed by two cameras (front and right-side views) which shall allow studies on multimodal music transcription and automatic scene and gesture analysis. This approach overcame two common hurdles in the building of music databases: copyrights - the recorded material is original - and annotation - as the availability of individual tracks and video feedback greatly eases the annotation process. For parts of this database, the drummers played on background music to produce material suitable for studies on drum event detection in polyphonic music or single or multiple sensor audio source separation. A significant part of this database will be publicly released for research purposes while a part of it will remain in our premises and could serve for future evaluation campaigns.

The content of the database is described in section 2. Section 3 details the recording and annotation process. The distribution terms and modalities are given in section 4. Finally, some conclusions and perspectives are given in section 5.

2. Database content

The ENST-Drums database is a large and varied research database for automatic drum transcription and processing. For this database, three professional drummers specialized in different music genres were recorded. The total duration of audio material recorded per drummer is around 75 minutes. Each drummer played his own drum kit, and for each sequence, used either sticks, rods, brushes or mallets to increase the diversity of drum sounds. The drum kits themselves are varied, ranging from a small, portable, kit with two toms and 2 cymbals, suitable for jazz and latin music ; to a larger rock drum set with 4 toms and 5 cymbals.

2.1. Detailed content played by each drummer

For each drummer, five different kinds of sequences were recorded. We underline that for all of these items, the drummers never had to follow a score or imitate a reference pat-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2006 University of Victoria

tern, but rather had to freely interpret the set of constraints given to them. While it made annotation more difficult and cross-checking impossible, this decision ensured that the musicians always played naturally, producing all kinds of combinations and situations likely to be encountered in real drum playing.

2.1.1. Individual strokes or "hits"

The drummers were asked to play sequences of several strokes separated by a few seconds of silence on each element of the drum kit, for each kind of stick available (plain sticks, rods, mallets and brushes).

2.1.2. Phrases

About sixty short drum sequences in various popular styles, without accompaniment, were played by each drummer. Each drummer was given a list of styles: bossa, disco, afro, reggae, jazz, swing, salsa, cha-cha, oriental, rock, blues, metal, hard rock, waltz, funk, country, and was asked to pick his favorites. Due to the different music backgrounds and preferences of the three drummers, only nine of these styles are common to all of them.

For each style, six phrases are played, at different tempi (slow, medium, fast) and at two complexity levels: straight without ornaments, and complex with fill-ins and ornaments. The tempi are not absolute and do not correspond to a given beat per minute (BPM) value, but are rather relative to each genre - e.g., a slow disco phrase would be played at 110 BPM, while a slow Jazz would be played at 70 BPM. Similarly, each drummer interpreted the notion of "complexity" differently, taking into account his preferences and the targeted style.

2.1.3. Soli

Each drummer played a minimum of five soli in the styles of his choice. The instructions given were the following: a typical solo should last about 30s, should use all the drum instruments of the kit and contain some very complex sequences (in terms of number of drum instruments involved, in terms of rhythmic content or/and in terms of tempo).

2.1.4. Accompaniment

Seventeen (17) sequences are played by each drummer on top of a pre-recorded accompaniment extracted from "minus one" CDs [6, 7]. Such CDs are used for the teaching of drumming, and allow students to practice on top of a music accompaniment from which the drum track has been removed. The "minus one" excerpts are about one minute long, cover various styles (blues, twist, metal, funk, celtic...) and are mostly played by acoustic instruments with a few synthetic keyboards. Additionally, twenty-four (24) shorter sequences were also recorded, in which the drummers played on top of pre-recorded synthetic accompaniments generated from MIDI files (the MIDI drum sounds being muted). A summary of the content available for each drummer is given in table 1.

2.2. Video recordings

For each sequence, two video files are available, corresponding to the front (angle 1) and right side (angle 2) views. Examples are shown in figure 1.

Figure 1. Examples of images recorded by camera 1 (top view) and camera 2 (right side view). The numbering used for cymbal events is overlaid on image 2.

2.3. Audio recordings

For each drum sequence played, a number of audio tracks are recorded or generated which allow the tackling of various drum signal processing applications. This leads to ten (or eleven) audio files per sequence. First, 8 monophonic files corresponding to the 8 microphones: bass drum, snare drum, hi-hat, mid tom, low-mid (if available), low tom track, left overhead, right overhead.Then, 3 stereophonic files: a dry stereo mix of the aforementioned tracks, a "wet" stereo mix of the aforementioned tracks (see section 3.4 for the list of processings applied); and finally, a stereo file contains the accompaniment (either "minus one" music background or synthetic MIDI audio files) without drums.

2.4. Annotation

The annotation for each sequence is available as a text file containing a list of $(time, event)$ pairs. Events are identified by the labels listed in table 2. For events associated to cymbals, the number of the cymbal (cymbals are numbered from left to right, from the drummer's point of view, see figure 1) is also added. For example, **rc3** indicates a ride cymbal hit, the 3rd cymbal for this particular drummer.

3. Building the ENST-Drums database

3.1. Audio recording

8 microphones were used to record the performances: A Beyerdynamic M-88 for the bass drum, a Shure SM57 for the snare drum, a Schoeps CMC body with a cardioid capsule for the hi-hat, two Shure SM58 for the mid and low-mid toms, a Sennheiser 441 for the low tom and two Audio-Technica AT4040 for the overheads. The microphones were amplified by 4 Behringer Ultragain Pro Mic2200 dual preamplifiers. The signals were recorded on a Tascam MX2424

Table 1. Number of sequences and events (strokes) recorded per drummer

	Drummer 1		Drummer 2		Drummer 3	
Item	Sequences	Events	Sequences	Events	Sequences	Events
Hits	29	139	31	180	48	283
Phrases	66	5339	74	9305	68	10467
Soli	7	1420	5	1613	5	1983
Accompaniment (Minus one CD)	17	8856	17	8788	17	9382
Accompaniment (MIDI file)	24	8224	24	6274	24	7357
Total	143	23978	151	26160	162	29472

Table 2. Labels used in the annotation

Label	Description	Label	Description
bd	Bass drum	lmt	Low-mid tom
sweep	Brush sweep	mt	Mid tom
sticks	Sticks hit together	mtr	Mid tom, hit on the rim
sd	Snare drum	lt	Low tom
rs	Rim shot	ltr	Low tom, hit on the rim
cs	Cross stick	lft	Lowest tom
chh	Hi-hat (closed)	rc	Ride cymbal
ohh	Hi-hat (open)	ch	Chinese ride cymbal
cb	Cowbell	cr	Crash cymbal
c	Other cymbals	spl	Splash cymbal

digital multitracker, with a resolution of 16 bits and a sampling rate of 44100 Hz. The click and background tracks were played to the drummers through headphones during the recording of the accompaniment sequences.

3.2. Video recording

Two cameras were used for the video recording (see figure 1 for examples of images). The front view (angle 1) was recorded with a Canon XL1 professional DV camera. The camera was fixed on a tripod mounted on a table, for a total elevation of 2.10m. The right side view (angle 2) was recorded by a Sony DCR-TRV30E DV camcorder, mounted on a tripod. Both cameras recorded at a spatial resolution of 720x576, at 25 frames per second, on mini-DV tapes. Though the recording conditions for this database were well controlled, it is important to mention that no visual clues such as coloured gloves, sticks or backgrounds were used.

3.3. Editing and synchronization

About 3 hours of raw audio material was recorded for each drummer. A first stage in the editing process consisted in editing the audio tracks to remove bad takes and long gaps between sequences. This resulted in 9 edited master audio tracks (8 mono tracks corresponding to the 8 microphones, 1 stereo track corresponding to the accompaniment) per drummer.

Then, two master video tracks, one per camera, in DV format, were built by trimming and aligning the video sequences to match the master audio tracks. We did not observe time base drifting, frame loss, or desynchronization between the audio and video tracks recorded by distinct devices. Consequently, no time-stretching had to be performed. The actual alignment was manually performed by matching sharp and short peaks in the master audio tracks signals, and in the audio signals recorded by the cameras' built-in microphones.

3.4. Mixing

Additionally, two stereo audio mixes were made from the master audio tracks. The "dry" mix consisted in simply panning and adjusting the level of each instrument, without any further processing. On the "wet" mix, each instrument was processed by an appropriate equalization and compression. A slight reverberation was added to the result, along with a dynamic processing (Waves L3 Ultramaximizer).

3.5. Annotation

3.5.1. The semi-automatic annotation process

The availability of individual audio tracks eased the annotation process, since each class of drum sound is predominant on the corresponding recording channel. Especially, the bass drum, snare drum, and toms tracks, on which the other instruments of the kit are the most attenuated, could be easily annotated by a same semi-automatic process consisting in detecting all note onsets with the onset detection algorithm presented in [8], building from this onset list a marker file for an audio editor (Wavelab), and finally manually fixing the detection mistakes in the audio editor.

The hi-hat track was annotated using a similar process, but required many more manual corrections, as the snare drum was also present in this track. Moreover, the annotation of this track required the discrimination between closed and open hi-hat strokes. The cymbals were similarly annotated from the pair of overheads. In all cases, a video file adapted to the annotated instrument (angle 1 for cymbals and toms, angle 2 for hi-hat and snare drum) was opened simulatenously, and was extremely helpful in disambiguating strokes.

3.5.2. Special cases

The availability of a video feedback and the mismatch between the audio and video signals we sometimes experienced raised some questions during the annotation process, about which events should be annotated, and which events should not. We encountered:

- Missed strokes, for example when a drummer stretches out his arm to hit a cymbal, but the head of the drum stick

misses the cymbal by a few centimeters. These events were not annotated.

- Moves used purely for time keeping which do not cause any sound, or cause extremely quiet artefacts. For example, one of the drummers tapped the base of the hi-hat pedal on odd beats - which resulted in a slight metallic click very distinct from a *closed hi-hat* sound. These events were not annotated.

- Quiet strokes played periodically for time keeping (for example, played for each quarter note). These events were not annotated.

- Attenuated "Ghost notes" played off-beat and used to create a feeling of "groove", especially in styles such as Funk or Shuffle-Blues. These events were annotated. This latter class of events, which is usually ignored by studies on drum transcription, can be filtered out by computing, for each stroke, its energy, and by removing from the transcription all the strokes whose energy falls below a given threshold, or by clustering the strokes in different classes according to their energy and their position within the metric structure.

3.5.3. Verification

The annotation process (which mostly consisted in correcting the output of the onset detection algorithm) was performed by one individual(the first author of this paper). In order to correct mistakes and to homogenize the handling of the special cases described above, the result of this first annotation step was verified once again by the same annotator. Finally, all the verified annotations, for each instrument, were merged in a single master annotation file per performance, whose format is described in 2.4.

3.6. Segmentation

The final step consisted in segmenting the master files (be it annotations, audio or video tracks) into individual files, in order to isolate each sequence into one individual file. For this purpose, a list of markers defining the beginning and end of each sequence was created from the master tracks. A chain of Python and Sylia (VirtualDub's own scripting language) scripts processed this list and created individual files for each segment.

4. Distribution

A large part of the ENST-Drums database will be freely distributed for research purposes. For this purpose, we have received the acceptance for such a distribution (i.e. limited to research purposes) from the three professional drummers and from PDG Music Publishing, who has edited the "minus one" background music used. The procedure for the distribution is not yet finalized but it should consist in a two step mechanism similar to the one used for the distribution of the RWC Music Database [4]. Firstly, prior to database download, a letter of engagement will need to be signed in which the database usage restriction will be specified. The database web site on which updated information will be posted and from which the database will be downloadable is `http://www.enst.fr/~grichard/ENST-drums/`. At the time of publication, the web site will be fully operational. The remaining part of the database will remain private to serve in particular future evaluation campaigns.

5. Conclusion

In this paper, we provided a detailed description of the ENST-Drums database. This audiovisual database of drum performances is fully annotated and will be, for a large part, freely distributed for research purposes. The large variety of its content should serve research in different domains of audio signal processing involving drums, ranging from single drum event classification to complex multimodal drum track transcription and extraction from polyphonic music.

6. Acknowledgements

The authors wish to acknowledge the support of the French ministry of research (ACI-MusicDiscover project) and of the European Commission under the FP6-027026-K-SPACE contract.

References

[1] F. Opolko J. Wapnick. McGill University Master Samples. http://www.music.mcgill.ca/resources/mums/html, 1987-1989.

[2] G. Ballet, R. Borghesi, P. Hoffmann, and F. Levy. Studio online 3.0: An internet killer application for remote access to ircam sounds and processing tools. In *Proc. of Journes d'Informatique Musicale (JIM'99)*, 1999.

[3] L. Fritts. University of Iowa Musical Instrument Samples. http://theremin.music.uiowa.edu/.

[4] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases. In *Proc. 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 287–288, October 2002.

[5] K. Tanghe, M. Lesaffre, S. Degroeve, M. Leman, B. De Baets, and J.-P. Martens. Collecting Ground Truth Annotations for Drum Detection in Polyphonic Music. In *Proc. 6th Int. Conf. on Music Information Retrieval (ISMIR 2005)*, pages 50–57, September 2005.

[6] E. Thiévon. *Batterie mode d'emploi - Playbacks*. PDG Music Publishing, 2004.

[7] E. Thiévon and P. Argentier. *Drums Training Session - Métier et variété*. PDG Music Publishing, 1999.

[8] M. Alonso, G. Richard, and B. David. Extracting Note Onsets from Musical Recordings. In *Proc. IEEE Int. Conf. Multimedia and Expo*, 2005.

[9] O. Gillet and G. Richard. Automatic Transcription of Drum Loops. In *Proc. 2004 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, May 2004.

ANNEXE D

Corpora utilisés

Artiste	Titre	Artiste	Titre
Aerosmith	Rock this way	Pink Floyd	Money
Architecture in Helsinki	Do the whirlwind	Portishead	Roads
Beats International	Dub be good to me	RWC-MDB-P-2001	N°09
Burnshee Thornside	Can I be a star ?	RWC-MDB-P-2001	N°11
China Chrisis	King in a catholic style	RWC-MDB-P-2001	N°30
Czerkinsky	Natacha	RWC-MDB-P-2001	N°50
Daft Punk	Da funk	RWC-MDB-P-2001	N°52
Depeche Mode	Enjoy the silence	Soulprint	Putrid
Diana Ross	Upside down	Spinecar	Waste away
Drop Trio	Wreck of the zephyr	Stereolab	International colouring contest
Earlimart	The hidden track	Stereolab	Les yper-sounds
Jimi Hendrix	Purple haze	Stevie Wonder	Master blaster
Joan Jett	I love rock'n'roll	Tahiti 80	Better days will come
John B	American girls	The Beastie Boys	That's it that's all
Katerine	Au pays de mon premier amour	The Delgados	Everybody comes down
LCD Soundsystem	Daft punk's playing at my house	The Police	Roxanne
Le tone	Joli dragon	The Rocky Horror Picture Show	Let's do the time warp
Les Rita Mitsouko	Marcia baila	The Selecter	Too much pressure
Lio	Banana split	The Talking Heads	New feeling
M	Onde sensuelle	The unicorns	Tough ghost
MC Solaar	Nouveau western	The Wiseguys	Ooh la la
Minor Threat	Stumped	Thursday Group	Innocent murmur
Morcheeba	Rome wasn't built in a day	Transformer Di Roboter	Hi end
Mouse on Mars	Mine is in yours	Transwave	Malaka dance
Mr Scruff	Spandex man	Very large Array	Magnified
NTM	On est encore là	White Town	Your woman
Paris Combo	Living room	Word Up	Groove me

TAB. D.1 – Corpus `Music-54` pour l'évaluation des pré-traitements d'accentuation de la piste de batterie

Artiste	Titre	Artiste	Titre
13th Floor Elevators	You are gonna miss me	Mu Ziq	The hwicci song
Air	Le soleil est près de moi	My Little Airport	Edward had you ever thought...
Ambulance Ltd	Country gentleman	My Morning Jacket	Wordless chorus
Andrew Bird	Action adventure	Norah Jones	Don't know why
Architecture in Helsinki	Like a call	Of Montreal	I was never young
Architecture in Helsinki	Do the whirlwind	Olano	Latitudes
Asian Dub Foundation	Pknb	Olive	You're not alone
BB King	Aint nobody home	Os Mutantes	Panis et circenses
Bearsuit	On your special day	Paavoharju	Valo tihkuu kaiken lapi
Beats International	Dub be good to me	Perspects	Desire and efficiency
Beck	Loser	PHD	I won t let you down
Belle And Sebastian	Wrapped up in books	Phoenix	If i ever feel better
Bis	We are so fragile	Prefuse 73	Pentagram
Blur	Girls and boys	Ratatat	El pico
Bonobo	Flutter	Sage Francis	Gunz yo
Boy George	Do you really want to hurt me	Saint Etienne	Split screen
Bronski Beat	Small town boy	Salako	Go on then enlighten me
Bubar The Cook	Eat your pitbull	Say Hi To Your Mom	Your brains vs my tractorbeam
Buzzcocks	Love you more	Serge Gainsbourg	Ballade de melody nelson
Camera Obscura	Keep it clean	Sodastream	Horses
China Crisis	King in a catholic style	Soft Cell	Tainted love
DAT Politics	My toshiba is alive	Stereo Total	Musique automatique
Datarock	Computer camp love	Stereolab	Captain easychord
De La Soul	Verbal clap	Stevie Wonder	Sir duke
Dear Nora	The new year	Suburban Prejudice	Anything
Depeche Mode	Dreaming of me	Sufjan Stevens	Jacksonville
Depeche Mode	New life	Sunidhi Chauhan	Dil mein jaagi dhadkhan aise
Digable Planets	Pacifics	Susheela Raman	Trust in me
Earlimart	The hidden track	Tahiti80	Soul deep
Electric Six	Gay bar	That Petrol Emotion	Big decision
Elk City	Love's like a bomb	The Arcade Fire	Neighboorhood 1 - tunnels
Gary Numan	Cars	The Avalanches	Frontier psychiatrist
Ghalia Benali And Timnaa	Awaddu	The Beach Boys	Wouldn't it be nice
John B	American girls	The Decemberists	The soldiering life
Just Brothers	Sliced tomatoes	The Delgados	Everybody come down
K. Kumar - Lata Mangeshkar	Kya yehi pyar hai	The High Llamas	Calloway
Kraftwerk	We are the robots	The High Llamas	Literature is fluff
Lali Puna	Micronomic	The Konki Duet	Imawa mori nona kani
Laura Veirs	Icebound stream	The New Pornographers	From blown speakers
LCD Soundsystem	Daft punk's playing at my house	The Talking Heads	Don't worry about the government
Le Tigre	Viz	The Unicorns	Les os
Lio	Banana split	The Unicorns	Tuff ghost
M	Onde sensuelle	The White Stripes	Dead leaves and the dirty ground
Men Without Hats	Safety dance club mix	Tiger Tunes	Unite
Metric	Combat baby	Transformer Di Roboter	Groundhog eat the girl
Metric	Raw sugar	Transient	Discovery of the symmetric sauce
MIA	Sunshowers	U. Narayan, S. Chauhan	Dhadak dhadak
Minor Threat	Straight edge	Vincent Delerm	Fanny ardant et moi
Modest Mouse	Tiny cities made of ashes	White Town	Your woman

TAB. D.2 – Corpus `Music-100` pour l'évaluation des méthodes de segmentation musicale

Artiste	Titre	Artiste	Titre
A-Ha	Take on me	MC Hammer	Can't touch this
Aphex Twin	Come to daddy	MC Solaar	Bouge de là
Aphex Twin	Ventolin	MC Solaar	Caroline
Aqua	Barbie girl	Metric	Combat baby
Arsenik	Je boxe avec les mots	Metric	Dead disco
Audioslave	Doesn't remind me	MIA	Galang
Autechre	Second bad vilbel	MIA	Sunshowers
Beck	Loser	Michael Jackson	Thriller
Bjork	Hunter	Midnight Oil	Beds are burning
Bjork	Joga	Moloko	Pure pleasure seeker
Bjork	Oh it's so quiet	Moloko	Sing it back
Britney Spears	Baby one more time	Mouse On Mars	Actionist respoke
Bubar The Cook	City endless beat	Mouse On Mars	Distroia
Cibo Matto	Sugar water	Mr Oizo	Flat beat
Cocteau Twins	Song to the siren	Nine Inch Nails	Closer
Coldcut	Timber	Nine Inch Nails	Only
Coldcut	World of evil	Nirvana	Smells like teen spirit
Daft Punk	Around the world	Peter Gabriel	Sledgehammer
Daft Punk	Burnin	Portishead	Only you
Depeche Mode	People are people	Primal Scream	Kowalski
Depeche Mode	Personal jesus	Radiohead	Creep
Devo	Satisfaction	Radiohead	Karma police
Devo	That's good	Radiohead	Paranoid android
Devo	We are devo	REM	Losing my religion
Devo	Whip it	Royksopp	Remind me
Dire Straits	Money for nothing	Run DMC	Walk this way
Dj Shadow	Six days	Sensorama	Star escalator
Eminem	Loose yourself	Shakira	Hips don't lie
Eurythmics	Sweet dreams	Sinead'O'Connor	Nothing compares to you
Frankie Goes to Hollywood	Relax	Squarepusher	Come on my selector
Franz Ferdinand	Take me out	Stereolab	Fluorescences
Gary Numan	Cars	Stereolab	Jenny ondioline
Herbie Hancock	Rock it	Stereolab	The free design
Iam	Je danse le mia	Super Collider	Messagecomin
Jamiroquai	Virtual insanity	The Avalanches	Frontier psychiatrist
Jean Michel Jarre	Zoolookologie	The Beastie Boys	Body movin
Kanye West	Heard'em say	The Beastie Boys	Fight for your right
Katerine	Cent pour cent vip	The Beastie Boys	Intergalactic
Kraftwerk	We are the robots	The Beastie Boys	Sabotage
Kylie Minogue	Come into my world	The Chemical Brothers	Let forever be
LCD Soundsystem	Daft punk's playing at my house	The Chemical Brothers	Star guitar
Len Lye	Free radicals	The Dissociatives	Somewhere down the barrel
Little Computer People	Little computer people	The Postal Service	Against all odds
M	Machistador	The Postal Service	Such great heights
Madness	Our house	The Prodigy	Firestarter
Madonna	Frozen	The White Stripes	Dead leaves and the dirty ground
Madonna	Like a prayer	The White Stripes	Fell in love with a girl
Madonna	Vogue	The White Stripes	The hardest button to button
Mariah Carey	We belong together	TLC	Waterfalls

TAB. D.3 – Corpus `Video-100` de clips vidéos pour l'évaluation des méthodes de corrélation des flux audio et vidéo

Frappe	**Fréq**. (%)
Taxonomie $\{bd, sd\}$	
$\{sd\}$	42.3
$\{bd\}$	29.8
$\{bd, sd\}$	27.9
Taxonomie $\{bd, sd, hh\}$	
$\{sd\}$	23.8
$\{hh\}$	22.1
$\{bd, hh\}$	18.1
$\{hh, sd\}$	12.5
$\{bd\}$	11.6
$\{bd, sd\}$	6.4
$\{bd, sd, hh\}$	5.5
Taxonomie $\{bd, sd, hh, tom\}$	
$\{sd\}$	22.3
$\{hh\}$	20.3
$\{bd, hh\}$	16.4
$\{hh, sd\}$	11.6
$\{bd\}$	10.4
$\{bd, sd\}$	5.7
$\{tom\}$	4.9
...	
Taxonomie $\{bd, sd, hh, cym\}$	
$\{sd\}$	20.4
$\{hh\}$	18.7
$\{bd, hh\}$	14.6
$\{hh, sd\}$	10.2
$\{bd\}$	6.6
$\{bd, cym\}$	6.3
$\{cym\}$	5.9
$\{bd, sd\}$	4.1
$\{bd, hh, sd\}$	3.8
$\{cym, sd\}$	2.7
...	
Taxonomie $\{bd, sd, hh, cym, tom\}$	
$\{sd\}$	19.3
$\{hh\}$	17.3
$\{bd, hh\}$	13.5
$\{hh, sd\}$	9.5
$\{bd\}$	5.9
$\{bd, cym\}$	5.8
$\{cym\}$	5.5
$\{tom\}$	3.9
$\{bd, sd\}$	3.6
$\{bd, hh, sd\}$	3.3
$\{cym, sd\}$	2.6
$\{cym, hh\}$	1.9
$\{bd, cym, hh\}$	1.6
...	

TAB. D.4 – Fréquence des combinaisons de frappes, par taxonomie, dans le corpus ENST-drums. Ne sont listées que les combinaisons les plus fréquentes totalisant 95 des combinaisons observées

Bibliographie

[Aba07] Abaltat. Beat. `http://www.abaltat.com/productsBeat.cfm`, 2007.

[ABDR03] M. Alonso, R. Badeau, B. David, et G. Richard. Musical tempo estimation using noise subspace projections. In *Proceedings of the 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, New Paltz, USA, 2003.

[ADK04] L. Agnihotri, N. Dimitrova, et J. R. Kender. Design and Evaluation of a Music Video Summarization System. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME'04)*, pages 1943–1946, June 2004.

[ADKZ03] L. Agnihotri, N. Dimitrova, J. Kender, et J. Zimmerman. Music videos miner. In *Proceedings of the 11th ACM International Conference on Multimedia*, pages 442–443, 2003.

[Alo06] M. Alonso. *Extraction of Metrical Information from Acoustic Music Signals*. PhD thesis, ENST, 2006.

[AP03] S. A. Abdallah et M. D. Plumbey. Probability as metadata : event detection in music using ICA as a conditional density model. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA'03)*, 2003.

[AP04] S. A. Abdallah et M. D. Plumbey. Polyphonic transcription by non-negative sparse coding of power spectra. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 318–325, 2004.

[App07] Apple. Final cut studio 2 – motion 3. `http://www.apple.com/finalcutstudio/motion/`, 2007.

[ARD05] M. Alonso, G. Richard, et B. David. Extracting Note Onsets from Musical Recordings. In *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME'05)*, 2005.

[ARD07] M. Alonso, G. Richard, et B. David. Accurate tempo estimation based on harmonic + noise decomposition. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007.

[ATD02] A. Albiol, L. Torres, et E. Delp. Combining audio and video for video sequence indexing applications. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo (ICME'02)*, 2002.

[AWWK02] P. S. Aleksic, J. J. Williams, Z. Wu, et A. K. Katsaggelos. Audio-Visual Speech Recognition Using MPEG-4 Compliant Visual Features. *EURASIP Journal on Applied Signal Processing*, 11 :1213–1227, 2002.

[Bad05] R. Badeau. *Méthodes à haute résolution pour l'estimation et le suivi de sinusoïdes modulées. Application aux signaux de musique.* PhD thesis, ENST, 2005.

[BB00] K. P. Bennett et E. J. Bredensteiner. Duality and Geometry in SVM Classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 65–72, 2000.

[BBD02] R. Badeau, R. Boyer, et B. David. EDS parametric modeling and tracking of audio signals. In *Proceedings of the 5th International Conference on Digital Audio Effects (DAFX'02)*, September 2002.

[BBG06] L. Benaroya, F. Bimbot, et R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1) :191–199, January 2006.

[BBHL99] G. Ballet, R. Borghesi, P. Hoffmann, et F. Levy. Studio Online 3.0 : An Internet Killer Application for Remote Access to IRCAM Sounds and Processing tools. In *Proceedings of Journées d'Informatique Musicale (JIM'99)*, 1999.

[BBR07] N. Bertin, R. Badeau, et G. Richard. Blind Signal Decompositions for Automatic Transcription of Polyphonic Music : NMF and K-SVD on the benchmark. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'07)*, 2007.

[BC07] H. Bredin et G. Chollet. Audio-visual speech synchrony measure for talking-face identity verification. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'07)*, 2007.

[BDBG03] L. Benaroya, L. Mc Donagh, F. Bimbot, et R. Gribonval. Non-negative Sparse Representation for Wiener Based source separation with a single sensor. In *Proceedings of the 2003 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003.

[BDDS04] J. P. Bello, C. Duxbury, M. Davies, et M. Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6) :553–556, June 2004.

[BDR05] R. Badeau, B. David, et G. Richard. Selecting the modeling order for the ESPRIT high resolution method : an alternative approach. In *Proceedings of the 2004 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, May 2005.

[Ben03] L. Benaroya. *Séparation de plusieurs sources sonores avec un capteur*. PhD thesis, Université de Rennes 1, 2003.

[BFCL05] D. Barry, D. FitzGerald, E. Coyle, et B. Lawlor. Drum source separation using percussive feature detection and spectral modulation. In *Proceedings of the Irish Signals and Systems Conference (ISSC 2005)*, 2005.

[BG02] M. Bosi et E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Kluwer, 2002.

[BGL07] G. Bascoul, O. Gillet, et G. Laurent. Marginal effects analysis : Identifying the most effective marginal levers in decision making. *Marketing Science*, Soumis, 2007.

[Bil93] J. Bilmes. *Timing is the essence : Perceptual and computational techniques for representing, learning and reproducing expressive timing in percussive rhythm*. PhD thesis, Massachussetts Institute of Technology, Media Laboratory, 1993.

[BJ06] F. Bach et M. Jordan. Learning spectral clustering with application to speech separation. *Journal of Machine Learning Research*, 7 :1963–2001, 2006.

[BKJ05] R. Bencina, M. Kaltenbrunner, et S. Jordà. Improved topological fiducial tracking in the reactivision system. In *Proceedings of the IEEE Internal Workshop on Projector-Camera Systems (PROCAMS'2005)*, 2005.

[BLC04] D. Barry, B. Lawlor, et E. Coyle. Sound source separation : Azimuth discrimination and resynthesis. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX'04)*, October 2004.

[Blo94] I. Bloch. Information Combination Operators for Data Fusion : A Comparative Re-

view with Classification. In *SPIE/EUROPTO Conference on Image and Signal Processing for Remote Sensing*, volume 2315, pages 148–159, Rome, Italy, Septembre 1994.

[Bon02] C. Bond. A new algorithm for scan conversion of a general ellipse. `http://www.crbond.com/papers/ellipse.pdf`, January 2002.

[BOP97] M. Brand, N. Olivier, et A. Pentland. Coupled Hidden Markov Models for Complex Action Recognition. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, page 994, 1997.

[Bra97] M. Brand. Coupled hidden markov models for modeling interacting processes. Technical report, MIT Media Lab Perceptual Computing, June 1997.

[Bre01] L. Breiman. Statistical modeling : The two cultures. *Statistical Science*, 16(3) :199–231, 2001.

[BS03] J. P. Bello et M. Sandler. Phase-based note onset detection for music signals. In *Proceedings of the 2003 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.

[BW01] M. A. Bartsch et G. H. Wakefield. To catch a chorus : Using chroma-based representations for audio thumbnailing. In *Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 15–18, 2001.

[BYB04] A. Ben-Yishai et D. Burshtein. A discriminative training algorithm for hidden markov models. *IEEE Transactions on Speech and Audio Processing*, 12(3) :204–217, 2004.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.

[Cas01] M. Casey. MPEG-7 sound-recognition tools. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6) :737–747, 2001.

[CB99] D. J. Crisp et C. J. C. Burges. A geometric interpretation of ν-SVM classifiers. In *Proceedings of the 12th Conference on Neural Information Processing Systems*, 1999.

[CC98] J. C. C. Chen et A. L. P. Chen. Query by rhythm : an approach for sound retrieval in music databases. In *Proceedings of the IEEE Workshop on Research Issues on Data Engineering*, pages 139–146, 1998.

[CF02] M. Cooper et J. Foote. Automatic Music Summarization via Similarity Analysis. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, 2002.

[CG98] S. S. Chen et P. S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, February 1998.

[Cho05] P. Chordia. Segmentation and Recognition of Tabla Strokes. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, 2005.

[CL01] C. C. Chang et C. J. Lin. LibSVM : a library for Support Vector Machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[Cla] J. Clark. Advanced Programming Techniques for Modular Synthesizers - Chapter 5. Percussions. `http://www.cim.mcgill.ca/~clark/nordmodularbook/nm_percussion.html`.

[CLS05] P. Chen, C. Lin, et B. Schölkopf. A tutorial on ν-support vector machines. In *Applied Stochastic Models in Business and Industry*, volume 21, 2, pages 111–136, 2005.

[CMR+03] A. Camurri, B. Mazzarino, M. Ricchetti, R. Timmers, et G. Volpe. Multimodal Analysis of Expressive Gesture in Music and Dance Performances. In *Proceedings of the 5th International Gesture Workshop*, pages 20–39, April 2003.

[Con06] A. Cont. Realtime multiple pitch observation using sparse non-negative constraints. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*. Victoria, CA., October 2006.

[CS05] S. Canu et A. Smola. Kernel methods and the exponential family. In *Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN'05)*, 2005.

[CSR03] E. Costanza, S. B. Shelley, et J. Robinson. Introducing audio d-touch : A tangible user interface for music composition and performance. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFX'03)*, September 2003.

[CTT05] A. Chaigne, C. Touzé, et O. Thomas. Nonlinear vibrations and chaos in gongs and cymbals. *Journal of Acoustical Science and Technology*, 26(5) :403–409, 2005.

[CVW04] R. Cilibrasi, P. Vitanyi, et R. De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4) :49–67, 2004.

[CW00] M. Casey et A. Westner. Separation of mixed audio sources by independent subspace analysis. In *Proceedings of the International Computer Music Conference (ICMC'00)*, 2000.

[Dah00] S. Dahl. The Playing of an Accent - Preliminary observations from temporal and kinematic analysis of percussionists. In *Journal of New Music Research*, volume 29(3), pages 225–234, 2000.

[Dah04] S. Dahl. Playing the Accent - Comparing Striking Velocity and Timing in an Ostinato Rhythm Performed by Four Drummers. *Acta Acustica united with Acustica*, 90 :762–776, 2004.

[DB97] J. W. Davis et A. F. Bobick. The Representation and Recognition of Action Using Temporal Templates. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[DDD05] F. Desobry, M. Davy, et C. Doncarli. An Online Kernel Change Detection Algorithm. *IEEE Transactions on Signal Processing*, 53(8) :2961–2974, August 2005.

[DDS01] C. Duxbury, M. Davies, et M. Sandler. Extraction of transient content in musical audio using multiresolution analysis techniques. In *Proceedings of the 4th International Conference on Digital Audio Effects (DAFX'01)*, 2001.

[Deu82] D. Deutsch, editor. *The Psychology of Music*, chapter Rhythm and Tempo. Academic Press, 1982.

[DG02] M. Davy et S. Godsill. Detection of abrupt spectral changes using support vector machines : an application to audio signal segmentation. In *Proceedings of the 2002 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, 2002.

[DGI06] M. Davy, S. Godsill, et J. Idier. Bayesian analysis of polyphonic western tonal music. *Journal of the Acoustical Society of America*, 119–4 :2498–2517, April 2006.

[DH04] Pedro F. Daniel et Daniel P. Huttenlocher. Cornell computing and information science. Technical report, Cornell, 2004.

[DHS01] R. Duda, P. E. Hart, et D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[Dig01] Digidesign. Soundreplacer. `http://www.digidesign.com/products/details.cfm?product_id=1059`, 2001.

[Div02] Divers. Visual Niches - Extraordinary Music Videos. DVD, 2002.

[Dix01] S. Dixon. Automatic extraction of tempo and beat from expressive performances. In

Journal of New Music Research, 2001.

[Dru03] Drumagog. Drum replacer 3.0. `http://www.drumagog.com/`, 2003.

[DTB+05] S. Degroeve, K. Tanghe, B. De Baets, M. Leman, et J. P. Martens. A simulated annealing optimization of audio features for drum classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, 2005.

[EA04] D. Ellis et J. Arroyo. Eigenrhythms : Drum pattern basis sets for classification and generation. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, 2004.

[EL03] E. Earl et R. E. Ladner. Enhanced sequitur for finding structure in data. In *Proceedings of the Data Compression Conference*, 2003.

[Ell96] D. Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, MIT, 1996.

[ERD06a] S. Essid, G. Richard, et B. David. Instrument Recognition in Polyphonic Music Based on Automatic Taxonomies. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 14–1, pages 68–80, 2006.

[ERD06b] S. Essid, G. Richard, et B. David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4) :1401–1412, July 2006.

[Ero01] A. Eronen. Automatic musical instrument recognition. Master's thesis, Tampere University of Technology, 2001.

[Ero03] A. Eronen. Musical Instrument Recognition using ICA-based transform of features and discriminatively trained HMMs. In *Proceedings of the 7th International Symposium on Signal Processing and its Applications*, volume 2, pages 133–136, July 2003.

[EW06] D. Ellis et R. Weiss. Model-based monaural source separation using a vector-quantized phase-vocoder representation. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, 2006.

[FCG02] J. Foote, M. Cooper, et A. Girgensohn. Creating music videos using automatic media analysis. In *Proceedings of the 10th ACM International Conference on Multimedia*, pages 553–560, 2002.

[FCL02] D. FitzGerald, E. Coyle, et B. Lawlor. Sub-band independent subspace analysis for drum transcription. In *Proceedings of the 5th International Conference on Digital Audio Effects (DAFX'02)*, 2002.

[FD01] J. W. Fisher et T. Darrell. Signal level fusion for multimodal perceptual user interface. In *Proceedings of the 2001 workshop on Perceptive user interfaces (PUI'01)*, pages 1–7, New York, NY, USA, 2001. ACM Press.

[FDFV00] J. W. Fisher, T. Darrell, W. Freeman, et P. A. Viola. Learning joint statistical models for audio-visual fusion and segregation. In *NIPS*, pages 772–778, 2000.

[FF06] R. Fiebrink et I. Fujinaga. Feature selection pitfalls and music classification. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, 2006.

[Fil06] S. Filippi. Transcription rythmique d'un signal audio de piano à fortes variations de tempo. Master's thesis, Université Paris 7 Denis Diderot, UFR de Mathématiques, 2006.

[Fit04] D. FitzGerald. *Automatic Drum Transcription and Source Separation*. PhD thesis, Dublin Institute of Technology, 2004.

[FL03] D. FitzGerald et B. Lawlor. Independent subspace analysis using locally linear em-

bedding. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFX'03)*, 2003.

[FLC03a] D. FitzGerald, B. Lawlor, et E. Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Proceedings of the Irish Signals and Systems Conference (ISSC 2003)*, July 2003.

[FLC03b] D. FitzGerald, B. Lawlor, et E. Coyle. Prior subspace analysis for drum transcription. In *Proceedings of the 114th AES Convention*, March 2003.

[FM00] I. Fujinaga et K. MacMillian. Real-time recognition of orchestral instruments. In *Proceedings of the International Computer Music Conference*, 2000.

[Foo99] J. Foote. Visualizing music and audio using self-similarity. In *Proceedings of ACM Multimedia'99*, pages 77–87, 1999.

[For73] G. D. Forney. The Viterbi algorithm. In *Proceedings of the IEEE*, volume 61, pages 268–278, march 1973.

[FPF99] A. Fitzgibbon, M. Pilu, et R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5) :476–480, May 1999.

[Fri] L. Fritts. University of Iowa Musical Instrument Samples. `http://theremin.music.uiowa.edu/`.

[GBVF03] R. Gribonval, L. Benaroya, E. Vincent, et C. Févotte. Proposals for performance measurement in source separation. In *Proceedings of the 4th Conference on Independent Component Analysis and Blind Signal Separation (ICA'03)*, April 2003.

[GE03] I. Guyon et A. Elisseeff. An introduction to feature and variable selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[GER07] O. Gillet, S. Essid, et G. Richard. On the correlation of audio and visual segmentations of music videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(2) :347–355, March 2007.

[GH01] F. Gouyon et P. Herrera. Exploration of techniques for automatic labeling of audio drum tracks. In *Proceedings of MOSART : Workshop on Current Directions in Computer Music*, 2001.

[GHC02] F. Gouyon, P. Herrera, et P. Cano. Pulse-dependent analyses of percussive music. In *Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.

[GHD03] F. Gouyon, P. Herrera, et A. Dehamel. Automatic labeling of unpitched percussion sounds. In *Proceedings of the 114th AES convention*, March 2003.

[GHNO02] M. Goto, H. Hashiguchi, T. Nishimura, et R. Oka. Rwc music database : Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, pages 287–288, October 2002.

[Gil03] O. Gillet. Amélioration d'un système de transcription de phrases de Tabla. Rapport de projet 3A, École Nationale Supérieure des Télécommunications, 2003.

[GJ97] Z. Ghahramani et M. I. Jordan. Factorial hidden markov models. *Journal of Machine Learning*, 29(2-3) :245–273, 1997.

[GJCS95] A. Ghias, J.Logan, D. Chamberlin, et B. C. Smith. Query by humming : Musical information retrieval in an audio database. In *Proceedings of ACM Multimedia'95*, pages 231–236, 1995.

[GM94] M. Goto et Y. Muraoka. A sound source separation system for percussion instruments. In *Transactions of the Institute of Electronics, Information and Communication Engi-*

neers, volume J77-D-II, pages 901–911, 1994.

[GM95] M. Goto et Y. Muraoka. A real-time beat tracking system for audio signals. In *Proceedings of the International Computer Music Conference (ICMC'95)*, pages 171–174, 1995.

[Gon03] M. Gondry. The Work of Director Michel Gondry. DVD, 2003.

[GR03] O. Gillet et G. Richard. Automatic labelling of Tabla signals. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, October 2003.

[GR04] O. Gillet et G. Richard. Automatic transcription of drum loops. In *Proceedings of the 2004 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, May 2004.

[GR05a] O. Gillet et G. Richard. Automatic transcription of drum sequences using audiovisual features. In *Proceedings of the 2005 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'05)*, 2005.

[GR05b] O. Gillet et G. Richard. Drum loops retrieval from spoken queries. *Journal of Intelligent Information Systems*, 24(2) :159–177, 2005.

[GR05c] O. Gillet et G. Richard. Drum track transcription of polyphonic music using noise subspace projection. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, September 2005.

[GR05d] O. Gillet et G. Richard. Extraction and remixing of drum tracks from polyphonic music signals. In *Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, October 2005.

[GR05e] O. Gillet et G. Richard. Indexing and querying drum loops databases. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing*, 2005.

[GR06a] O. Gillet et G. Richard. Comparing Audio and Video Segmentations for Music Videos Indexing. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, May 2006.

[GR06b] O. Gillet et G. Richard. ENST-drums : an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, 2006.

[GR07] O. Gillet et G. Richard. Transcription and separation of drum signals from polyphonic music. In *IEEE Transactions on Audio, Speech, and Language Processing, Special Issue on Music Information Retrieval*, (Accepté pour Publication, 2007).

[GWBV02] I. Guyon, J. Weston, S. Barnhill, et V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3) :389–422, 2002.

[Hal01] D. E. Hall. *Musical Acoustics*. Brooks Cole, 3rd edition, 2001.

[Haz05] A. Hazan. Towards automatic transcription of expressive oral percussive performances. In *Proceedings of the 10th international conference on Intelligent user interfaces (IUI'05)*, pages 296–298. ACM Press, 2005.

[HC02] J. Hershey et M. Casey. Audiovisual sound separation via hidden markov models. In *Proceedings of the 15th Conference on Neural Information Processing Systems*, Advances in Neural Information Processing Systems, 2002.

[HHLO83] P. L. Van Hove, M. H. Hayes, J. S. Lim, et A. V. Oppenheim. Signal reconstruction from signed fourier transform magnitude. In *IEEE Transactions on Acoustics Speech and Signal Processing*, volume 31 (5), pages 1286–1293, 1983.

[HL06] S. Y. Huang et Y. J. Lee. Kernel fisher's discriminant analysis in gaussian reproducing kernel hilbert space – theory. Technical report, Academia Sinica, Taiwan, 2006.

[HM00] J. Hershey et J. Movellan. Audio-vision : Using audio-visual synchrony to locate sounds. In *Advances in Neural Information Processing Systems*, pages 813–819. MIT Press, 2000.

[HM03] S. Hainsworth et M. Macleod. Beat tracking with particle filtering algorithms. In *Proceedings of the 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, 2003.

[HO00] A. Hyvärinen et E. Oja. Independent component analysis : Algorithms and applications. *Neural Networks*, 13(4–5) :411–430, 2000.

[HV05] M. Helén et T. Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proceedings of the 13th European Signal Processing Conference*, 2005.

[HW04] K. Hermus et P. Wambacq. Assessment of signal subspace based speech enhancement for noise robust speech recognition. In *Proceedings of the 2004 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, volume 1, pages 945–948, May 2004.

[HYG02] P. Herrera, A. Yeterian, et A. Gouyon. Automatic classification of drum sounds : A comparison of feature selection methods and classification techniques. In *Proceedings of the Second International Conference on Music and Artificial Intelligence (ICMAI'02)*, pages 69–80, London, UK, 2002. Springer-Verlag.

[HYM02] B. Huet, I. Yahiaoui, et B. Mérialdo. Image similarity for automatic video summarization. In *Proceedings of the 11th European Signal Processing Conference (EUSIPCO'2002)*, 2002.

[Hyv99] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. In *IEEE Transactions on Neural Networks*, volume 10(3), pages 626–634, 1999.

[IVWF06] N. Ikizler, J. Vasanth, L. Wong, et D. Forsyth. Finding celebrities in video. Technical Report UCB/EECS-2006-77, University of California Berkeley, 2006.

[JD01] S. Jeannin et A. Divakaran. MPEG-7 Visual Motion Descriptors. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 11, pages 720–724, 2001.

[Joa98] T. Joachims. Making large-scale support vector machine learning practical. In C. Burges A. S. B. Schölkopf, editor, *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.

[Jon03] S. Jonze. The Work of Director Spike Jonze. DVD, 2003.

[Jør02] M. E. Jørgensen. Drumfinder, DSP-project on recognition of drum sounds in drum tracks. http ://www.daimi.au.dk/ pmn/spf02/CDROM/pr4/, 2002.

[JW89] F. Opolko J. Wapnick. McGill University Master Samples. `http://www.music.mcgill.ca/resources/mums/html`, 1987-1989.

[Kam00] I. Kaminskyj. Multi-feature musical instrument sound classifier. In *Proceedings of the Australasian Computer Music Conference*, 2000.

[KBT04] A. Kapur, M. Benning, et G. Tzanetakis. Query by beatboxing : Music information retrieval for the DJ. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, October 2004.

[KKVB+05] A. Kapur, A. Kapur, N. Virji-Babul, G. Tzanetakis, et P. F. Driessen. Gesture-Based Affective Computing on Motion Capture Data. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction, ACII'05*, 2005.

[Kla99] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.

[Kla01] A. Klapuri. Multipitch estimation and sound separation by the spectral smoothness principle. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, USA, 2001.

[Kla03] A. Klapuri. Musical meter estimation and music transcription. In *Proceedings of the Cambridge Music Processing Colloquium*, March 2003.

[Kla04] A. Klapuri. *Signal processing methods for the automatic transcription of music*. PhD thesis, Tampere University of Technology, 2004.

[KPS03] T. H. Kim, S. I. Park, et S. Y. Shin. Rhythmic-Motion Synthesis Based on Motion-Beat Analysis. In *Proceedings of the 30th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH2003)*, 2003.

[Kru83] J. B. Kruskal. An Overview of Sequence Comparison. In David Sankoff et Joseph B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules : The Theory and Practice of Sequence Comparison*, pages 1–44. Addison-Wesley, Reading, MA, 1983.

[Kuh55] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 :83–97, 1955.

[Lar01] J. Laroche. Estimating tempo, swing and beat locations in audio recordings. In *Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'01)*, pages 131–135, 2001.

[Lar04] J. Laroche. Efficient Tempo and Beat Tracking in Audio Recordings. *Journal of the Audio Engineering Society*, 51(4) :226–233, April 2004.

[LCV+03] G. Loosli, S. Canu, S. V. N Vishwanathan, A. J. Smola, et M. Chattopadhyay. Boîte à outils SVM simple et rapide. *Revue d'Intelligence Artificielle*, 2003.

[LE07] A. Lacoste et D. Eck. A supervised classification algorithm for note onset detection. *EURASIP Journal on Advances in Signal Processing*, 2007 :Article ID 43745, 13 pages, 2007. doi :10.1155/2007/43745.

[Lip05] S. D. Lipscomb. The perception of audio-visual composites : accent structure alignment of simple stimuli. *Selected reports in Ethnomusicology*, 12 :37–67, 2005.

[LJ83] F. Lerdahl et R. Jackendoff. *A generative Theory of tonal Music*. MIT Press, Cambridge, 1983.

[Log00] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the 1st International Conference on Music Information Retrieval (ISMIR'00)*, 2000.

[LS01] D. D. Lee et H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.

[LS05] M. Li et R. Sleep. Melody classification using a similarity metric based on kolmogorov complexity. In *Proceedings of the 2nd Conference on Sound and Music Computing*, 2005.

[MAJ04] D. Murphy, T. H. Andersen, et K. Jensen. Conducting Audio Files via Computer Vision. In *Lecture notes in Computer science, LNCS 2915*, 2004.

[Mér95] B. Mérialdo. Modèles probabilistes et étiquetage automatique. *T.A.L, traitement automatique des langues, traitements probabilistes et corpus*, 36 :7–2, 1995.

[MGOR07] K. McGuinness, O. Gillet, N. O'Connor, et G. Richard. Visual analysis for drum sequence transcription. In *Accepté à la 17th European Signal Processing Conference (EUSIPCO'2007)*, 2007.

[Min05] J. Min. *Human Activity Recognition using Motion Trajectories*. PhD thesis, Pennsylvania State University, 2005.

[MIR] MIREX. Results of the MIREX Audio Drum Detection Contest. http ://www.music-

ir.org/evaluation/mirex-results/audio-drum/index.html.

[Mit98] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[MKYH03] P. Mulhem, M. S. Kankanhalli, J. Yi, et H. Hassan. Pivot Vector Space Approach for Audio-Video Mixing. *IEEE MultiMedia*, 10(2) :28–40, Avril–Juin 2003.

[MM99] J. Marques et P. J. Moreno. A study of musical instrument classification using gaussian mixture models and support vector machines. Technical report, Compaq Computer Corporation, 1999.

[MMP02] P. Mitra, C. A. Murthy, et S. K. Pal. Unsupervised Feature Selection Using Feature Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3) :301–312, 2002.

[Mur03] D. Murphy. Tracking a conductor's baton. In *Proceedings of the 12th Danish Conference on Pattern Recognition and Image Analysis*, 2003.

[MW06] S. T. Madsen et G. Widmer. Music complexity measures predicting the listening experience. In *Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC'06)*, 2006.

[NCS98] N. V. Nielsen, J. M. Carstensen, et J. Smedsgaard. Aligning of Single and Multiple Wavelength Chromatographic Profiles for Chemometric Data Analysis Using Correlation Optimised Warping. *Journal of Chromatography A*, 805 :17–35, 1998.

[NMW97] C. G. Nevill-Manning et I. H. Witten. Identifying hierarchical structure in sequences : A linear-time algorithm. *Journal of Artificial Intelligence Research*, 7 :67–82, 1997.

[NMWM94] C. G. Nevill-Manning, I. H. Witten, et D. L. Maulsby. Compression by induction of hierarchical grammars. In *Proceedings of the Data Compression Conference*, pages 244–253, 1994.

[NOGH04] T. Nakano, J. Ogata, M. Goto, et Y. Hiraga. A drum pattern retrieval method by voice percussion. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, October 2004.

[NSK03] M. Nayak, S. H. Srinivasan, et M. S. Kankanhalli. Music Synthesis for Home Videos : An Analogy based Approach. In *Proceedings of the 4th IEEE Pacific-Rim Conference on Multimedia (PCM'01)*, December 2003.

[OIKS06] P. Over, T. Ianeva, W. Kraaij, et A. F. Smeaton. TRECVID 2006 - An Overview. Technical report, National Institute of Standards and Technology (NIST), 2006.

[OPGB05] A. Ozerov, P. Philippe, R. Gribonval, et F. Bimbot. One microphone singing voice separation using source-adapted models. In *Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, Mohonk, NY, USA, 2005.

[Ori01] I. Orife. Riddim : A rhythm analysis and decomposition tool based on independent subspace analysis. Master's thesis, Dartmouth College, Hanover, 2001.

[oW03] University of Waikato. WEKA 3 : Machine Learning Software in Java. `http://www.cs.waikato.ac.nz/ml/weka/`, 2003.

[Pau06] J. Paulus. Acoustic modelling of drum sounds with hidden markov models for music transcription. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, 2006.

[PBR02] G. Peeters, A. La Burthe, et X. Rodet. Toward automatic music audio summary generation from signal analysis. In *Proceedings of the 2nd International Conference on Music Information Retrieval (ISMIR'01)*, 2002.

[PD03] K. A. Peker et A. Divakaran. Framework for measurement of the intensity of mo-

tion activity of video segments. Technical Report TR2003-64, Mitsubishi Electric Research Laboratories, June 2003.

[PDW03] E. Pampalk, S. Dixon, et G. Widmer. Exploring music collections by browsing different views. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, 2003.

[Pee03] G. Peeters. Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization. In *Proceedings of the 115th AES Convention*, October 2003.

[Pee04] G. Peeters. A large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO project. Technical report, IRCAM, 2004.

[PK02] J. Paulus et A. Klapuri. Measuring the similarity of rhythmic patterns. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, 2002.

[PK03a] J. Paulus et A. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo (ICME'03)*, 2003.

[PK03b] J. Paulus et A. Klapuri. Model-based event labeling in the transcription of percussive audio signals. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFX'03)*, September 2003.

[PK06] J. Paulus et A. Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st Audio and Music Computing for Multimedia Workshop (AMCMM'2006)*, 2006.

[Pla98] J. Platt. Fast training of support vector machines using sequential minimal optimization. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.

[Pla00] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classiers*, pages 61–74, 2000.

[PNLM04] G. Potamianos, C. Neti, J. Luettin, et I. Matthews. Audio-visual automatic speech recognition : An overview. In G. Bailly, E. Vatikiotis-Bateson, et P. Perrier, editors, *Issues in Visual and Audio-Visual Speech Processing*, chapter 10. MIT Press, 2004.

[PSH97] V. I. Pavlovic, R. Sharma, et T. S. Huang. Visual interpretation of hand gestures for human computer interaction : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :677–695, 1997.

[PTVF92] W. H. Press, S. A. Teukoslky, W. T. Vetterling, et B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[PV05] J. Paulus et T. Virtanen. Drum transcription with nonnegative spectrogram factorisation. In *Proceedings of the 15th European Signal Processing Conference (EUSIPCO'2005)*, 2005.

[Qui93] R. J. Quinlan. *C4.5 : Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.

[Rab89] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[Rap01] C. Raphael. Automated rhythm transcription. In *Proceedings of the 2nd International Conference on Music Information Retrieval (ISMIR'01)*, 2001.

[RBS06] E. Ravelli, J. P. Bello, et M. B. Sandler. Drum sound analysis for the manipulation of rhythm in drum loops. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, volume 5, pages 233–236,

May 2006.

[RBS07] E. Ravelli, J. P. Bello, et M. Sandler. Automatic rhythm modification of drum loops. *IEEE Signal Processing Letters*, April 2007.

[REF05] C. J. Lin R. E. Fan, P. H. Chen. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6 :1889–1918, 2005.

[Ris02] E. Riskedal. Drum Analysis. Master's thesis, Department of Informatics, University of Bergen, 2002.

[RJ93] L. Rabiner et B. Juang. *Fundamentals of speech recognition*. Englewood Cliffs, NJ, 1993.

[RMK95] C. Ridder, O. Munkelt, et H. Kirchner. Adaptive Background Estimation and Foreground Detection using Kalman Filtering. In *Proceedings of the International Conference on recent Advances in Mechatronics (ICRAM'95)*, pages 193–199, 1995.

[Ros01] T. D. Rossing. Acoustics of percussion instruments : Recent progress. *Journal of Acoustical Science and Technology*, 22, 3 :177–188, 2001.

[Row01] S. T. Roweis. One microphone source separation. In Todd K. Leen, Thomas G. Dietterich, et Volker Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 793–799. MIT Press, 2001.

[RRE07] M. Ramona, G. Richard, et S. Essid. Combined supervised and unsupervised segmentation of radiophonic audio streams. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'07)*, 2007.

[SC03] P. Smaragdis et M. Casey. Audio/visual independent components. In *Proceedings of the 3rd International Conference on ICA and Blind Source Separation*, april 2003.

[Sch85] W. A. Schloss. *On the Automatic Transcription of Percussive Music : From Acoustic Signal to High Level Analysis*. PhD thesis, Stanford University, CA, USA, May 1985.

[Sch98] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1) :588–601, 1998.

[Sep01] J. Seppänen. Tatum Grid Analysis of Musical Signals. In *Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[SG99] C. Stauffer et W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, volume 2, 1999.

[SGH04] V. Sandvold, F. Gouyon, et P. Herrera. Percussion classification in polyphonic audio recordings using localized sound models. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, October 2004.

[SGJS04] D. Sodoyer, L. Girin, C. Jutten, et J. L. Schwartz. Developing and audio-visual speech source separation algorithm. *Speech Communication*, 44 :113–125, 2004.

[SGM98] T. Sonoda, M. Goto, et Y. A. Muraoka. A www-based melody retrieval system. In *Proceedings of the International Computer Music Conference*, pages 349–352, 1998.

[SKSV00] J. Sillanpää, A. Klapuri, J. Seppänen, et T. Virtanen. Recognition of acoustic noise mixtures by combined bottom-up and top-down approach. In *Proceedings of the 10th European Signal Processing Conference (EUSIPCO'2000)*, 2000.

[SKT97] J. Saitoh, A. Kodata, et H. Tominaga. Integrated data processing between image and audio-musical instrument (piano) playing information processing. In *Proceedings of the 6th International Conference on Image Processing and its Applications*, volume 1, pages 432–442, 1997.

[SNI04] T. Shiratori, A. Nakazawa, et K. Ikeuchi. Detecting dance motion structure through music analysis. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, may 2004.

[SPST+99] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, et R. C. Williamson. Estimating the support of a high-dimensional distribution. Technical Report MSR-TR-99-87, Microsoft Research, 1999.

[SS90] X. Serra et J. Smith. Spectral modeling synthesis : a sound analysis/synthesis based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14 (4), 1990.

[SS02] B. Schölkopf et A. J. Smola. *Learning with kernels*. The MIT Press, Cambridge, MA, 2002.

[SSG+02] D. Sodoyer, J. L. Schwartz, L. Girin, J. Klinkisch, et C. Jutten. Separation of audio-visual speech sources : A new approach exploiting the audio-visual coherence of speech stimuli. *EURASIP Journal on Applied Signal Processing*, 11 :1165–1173, 2002.

[SSLS06] K. B. Petersen S. Sigurdssson et T. Lehn-Schiøler. Mel frequency cepstral coefficients : An evaluation of robustness of mp3 encoded music. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, 2006.

[STD+05] D. Van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, et J.-P. Martens. Support vector machines for bass and snare drum recognition. In *Studies in Classification, Data Analysis and Knowledge Organisation*. Springer, 2005.

[SV99] E. D. Scheirer et B. L. Vercoe. SAOL : The MPEG-4 Structured Audio Orchestra Language. *Computer Music Journal*, 23(2) :31–51, 1999.

[SXK03] X. Shao, C. Xu, et M. S. Kankanhalli. Automatically generating summaries for musical video. In 547-550, editor, *Proceedings of the 2003 International Conference on Image Processing*, volume 2, 2003.

[SXK04] X. Shao, C. Xu, et M. S. Kankanhalli. A New Approach to Automatic Music Video Summarization. In *Proceedings of the International Conference on Image Processing*, October 2004.

[Tan05] K. Tanghe. MAMI - software - drum detection console application. `http://www.ipem.ugent.be/MAMI/Public/Software/DrumDetectionCAs/`, 2005.

[Tau91] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11) :1115–1138, 1991.

[TC02] G. Tzanetakis et P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, Vol. 10(5) :293–301, July 2002.

[TDB05] K. Tanghe, S. Degroeve, et B. De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proceedings of the 2005 MIREX evaluation campaign*, 2005.

[TLD+05] K. Tanghe, M. Lesaffre, S. Degroeve, M. Leman, B. De Baets, et J.-P. Martens. Collecting Ground Truth Annotations for Drum Detection in Polyphonic Music. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 50–57, September 2005.

[TM98] C. Tomasi et R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the 1998 IEEE International Conference on Computer Vision*. IEEE Computer Society, 1998.

[TNS04] H. Takeda, T. Nishimoto, et S. Sagayama. Maximum likelihood method for estimating

rhythm and tempo. In *Proceedings of the Internation Symposium on Musical Acoustics (ISMA'04)*, April 2004.

[UD04a] C. Uhle et C. Dittmar. Drum pattern based genre classification of popular music. In *Proceedings of the AES 25th Internation Conference*, 2004.

[UD04b] C. Uhle et C. Dittmar. Further steps towards drum transcription of polyphonic music. In *Proceedings of the 116th AES convention*, May 2004.

[UDS03] C. Uhle, C. Dittmar, et T. Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA'03)*, April 2003.

[UH03] C. Uhle et J. Herre. Estimation of tempo, micro time and time signature from percussive music. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFX'03)*, September 2003.

[Vai93] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[Vir03] T. Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *Proceedings of the 2003 International Computer Music Conference (ICMC'03)*, 2003.

[VR04a] E. Vincent et X. Rodet. Instrument identification in solo and ensemble music using independent subspace analysis. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, 2004.

[VR04b] E. Vincent et X. Rodet. Underdetermined source separation with structured source priors. In *Proceedings of the 5th Symposium on Independent Component Analysis and Blind Signal Separation (ICA2004)*, April 2004.

[WB91] I. H. Witten et T. C. Bell. The zero-frequency problem : Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4) :1085–1094, 1991.

[WCH+05] W. Wang, D. Cosker, Y. Hicks, S. Sanei, et J. Chambers. Video assisted speech source separation. In *Proceedings of the 2005 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, volume 5, pages 425–428, 2005.

[WD01] M. M. Wanderley et P. Depalle. Gesturally-controlled digital audio effects. In *Proceedings of the 5th International Conference on Digital Audio Effects (DAFX'02)*, December 2001.

[WD04] M. M. Wanderley et P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4) :632–644, 2004.

[WE05] I. H. Witten et F. Eibe. *Data Mining : Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[WEBS] J. Weston, A. Elisseef, G. Bakir, et F. Sinz. The Spider Matlab toolbox. `http://www.kyb.tuebingen.mpg.de/bs/people/spider/`.

[WH00] Y. Wu et T. S. Huang. View-independent recognition of hand postures. In *Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, 2000.

[WYC04] J. F. Wang, C. H. Yang, et K. H. Chang. Subspace tracking for speech enhancement in car noise environments. In *Proceedings of the 2004 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, volume 2, pages 789–792, May 2004.

[XKC+04] L. Xie, L. Kennedy, S.-F. Chang, A. Divakaran, H. Sun, et C.-Y. Lin. Discovering meaningful multimedia patterns with audio-visual concepts and associated text. In

Proceedings of the International Conference on Image Processing, 2004.

[YB04] R. Yang et M. S. Brown. Music database query with video by synesthesia observation. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME'04)*, pages 305–308, June 2004.

[YGK+06] K. Yoshii, M. Goto, K. Komatani, T. Ogata, et H. Okuno. An error correction framework based on drum pattern periodicity for improving drum sound detection. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, volume 5, pages 237–240, May 2006.

[YGO04a] K. Yoshii, M. Goto, et H. G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, October 2004.

[YGO04b] K. Yoshii, M. Goto, et H. G. Okuno. Drum sound identification for polyphonic music using template adaptation and matching methods. In *Proceedings of the 2004 Workshop on Statistical and Perceptual Audio Processing*, 2004.

[YGO05] K. Yoshii, M. Goto, et H. G. Okuno. INTER :D : a drum sound equalizer for controlling volume and timbre of drums. In *Proceedings of the 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT'05)*, 2005.

[YMH01] I. Yahiaoui, B. Mérialdo, et B. Huet. Generating summaries of multi-episodes video. In *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo (ICME'01)*, 2001.

[YOI92] J. Yamato, J. Ohya, et K. Ishii. Recognizing Human Action in Time-sequential Images using Hidden Markov Model. In *Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pages 379–385, 1992.

[ZC06] S. Zhou et R. Chellappa. From Sample Similarity to Ensemble Similarity : Probabilistic Distance Measures in Reproducing Kernel Hilbert Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6) :917–929, 2006.

[Zet98] H. Zettl. *Sight, Sound, Motion : Applied Media Aesthetics*. Wadsworth Publishing, 1998.

[ZH00] B. Zhou et J. H. L. Hansen. Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion. In *Proceedings of the International Conference on Spoken Language Processing*, 2000.

[ZH05] J. Zhu et T. Hastie. Kernel Logistic Regression and the Import Vector Machine. *Journal of Computational and Graphical Statistics*, 14(1) :185–205, 2005.

[ZL78] J. Ziv et A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5) :530–536, September 1978.

[ZPDG02] A. Zils, F. Pachet, O. Delerue, et F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. In *Proceedings of the 2nd International Conference on Web Delivering of Music (WEDELMUSIC2002)*, December 2002.

[Zwi77] E. Zwicker. Procedure for calculating loudness of temporally variable sounds. *Journal of the Acoustical Society of America*, 1977.

Bibliographie de l'auteur

Revues internationales

- O. Gillet et G. Richard. Transcription and Separation of Drum Signals from Polyphonic Music. Accepté pour publication dans les *IEEE Transactions on Audio, Speech, and Language Processing, Special Issue on Music Information Retrieval*.

- O. Gillet, S. Essid et G. Richard. On the Correlation of Audio and Visual Segmentations of Music Videos (Invited Paper). *IEEE Transactions on Circuits and Systems for Video Technology*, 17(2) :347–355, 2007.

- O. Gillet et G. Richard. Drum loops retrieval from spoken queries. *Journal of Intelligent Information Systems*, 24(2) :159–177, 2005.

Conférences internationales avec comité de lecture

- O. Gillet et G. Richard. Supervised and unsupervised Sequence Modelling for Drum Transcription. Soumis à *8th International Conference on Music Information Retrieval (ISMIR'07)*, 2007.

- K. McGuinness, O. Gillet, N. O'Connor et G. Richard Visual Analysis of Drum Playing. Accepté à la *15th European Signal Processing Conference (EUSIPCO'2007)*, 2007.

- O. Gillet et G. Richard. ENST-drums : an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, 2006.

- O. Gillet et G. Richard. Comparing Audio and Video Segmentations for Music Videos Indexing. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, May 2006.

- O. Gillet et G. Richard. Indexing and Querying Drum Loops Databases. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing*, 2005.

- O. Gillet et G. Richard. Extraction and Remixing of Drum Tracks from Polyphonic Music Signals. In *Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'05)*, October 2005.

- O. Gillet et G. Richard. Drum Track Transcription of Polyphonic Music using Noise Subspace Projection. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, September 2005.

- O. Gillet et G. Richard. Automatic Transcription of Drum Sequences Using Audiovisual Features. In *Proceedings of the 2005 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'05)*, 2005.

- O. Gillet et G. Richard. Automatic Transcription of Drum Loops. In *Proceedings of the 2004 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, May 2004.

- O. Gillet et G. Richard. Automatic Labelling of Tabla Signals. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, October 2003.

Revues internationales (autres domaines)

- G. Bascoul, O. Gillet, et G. Laurent. Marginal effects analysis : Identifying the most effective marginal levers in decision making. Soumis à *Marketing Science*, 2007.

Index

Acuité, 222
AdaBoost, 155
ADAMAST, 23, 27
Adaptation, 20, 22, 109, 163, 186, 188
ADRes, 36
Apprentissage discriminatif, 64
AR, modèle, 221
Arrière-plan, modélisation de l', 144, 147
Attributs, 42, 59, 178

Baguettes, 6, 32, 147, 152
Banc de filtres, 34, 35
    bandes d'octave, 35
Bhattacharyya, distance de, 143, 187
BIC, 182
Blanchiment, 47
Boucles, 30
Box-Cox, transformation, 62

C4.5, 135
Caisse claire, 32
Canny, algorithme de, 138
Causalité, 29
Clustering, 21, 108, 138, 143, 193
Co-occurences, 157, 198
Code de Huffman, 82
Coefficient de corrélation, 126, 146, 157, 198
Complexité de Kolmogorov, 79
Congas, 14
Contrôle gestuel, 124
Couleur, attributs de, 134, 155, 192
Covariance, matrice de, 45, 187
COW, 198
Crête, facteur de, 221
Cymbale, 32, 80

Démixage, matrice de, 39, 128
Danse, 123
Drum replacement, 30
DTW, 198
Dual, 230

EDS, modèle, 44
Ellipse
    critère morphologique, 138
    dissimilarité, 143
    reconnaissance, 139
ENST-drums, 54, 58, 84, 112, 135, 158
Entropie, 194
Enveloppe, 42, 222
Enveloppe convexe, 226
Étendue, 222
Évolutionnaire, algorithme, 83

F-mesure, 86
Facteur de crête, 42
Fenêtre
    taille variable, 109
Filtre
    adapté, 154, 215
    bilatéral gaussien, 133
    en demis-tons, 217
    en sous-espace, 46
    non-linéaire, 58
    Pseudo-Wiener, 107
    TFS, 104
    Wiener, 107
Fisher, critère de, 65, 185
Fréquence de coupure, 220
Fusion, 52, 58, 68, 89, 125, 155, 157

Genre
    reconnaissance, 30
    visuel, 173, 202
GMM, 144, 190
    apprentissage en ligne, 147
Grammaire hors-contexte, 79
Grosse caisse, 32

HMM, 21, 124, 163, 193
    bi-modaux, 126
    couplés, 125
    factoriels, 127

ICA, 24, 102
    audiovisuelle, 123
    par sous-bande, 39
Indexation vidéo, 171
Information mutuelle, 126, 198
Instruments de musique

reconnaissance, 18
IOI, 16, 70
IRMFSP, 65
ISA, 24
Itération orthogonale, 45

Kuhn-Munkres, algorithme de, 157
Kullback-Leibler, divergence de, 187
Kurtosis, 42

Lagrange, multiplicateurs de, 186, 224, 229
LibSVM, 231

Mashup, 200
Masques TFS, 104
MFCC, 218
MIREX, 27
Moments
de l'enveloppe, 42, 222
spectraux, 220
temporels, 221
MPEG vidéo, 195
MPEG-4, 2
MPEG-7, 2

N-grammes, 29, 74
NMF, 27, 107, 109, 145
Normalisation, 62
Nouveauté, détection de, 180
Noyau, 67, 183, 185, 187, 232, 234

OBSIR, 216
Onsets, 13, 55
Ordre, critère d', 47

Parole
localisation du locuteur, 126
reconnaissance, 125
séparation, 127
PCA, 24, 62, 193
Piano
Gestes, 123
Transcription audiovisuelle, 122
Pics, 58, 152, 154
Platitude, 42, 220
Pré-écho, 109
Précision, 86
Probabilités a posteriori, 236
PSA, 27

Quantification, 70, 72, 157

Régression logistique à noyaux, 235
Régularisation, 82, 188, 235
Résumé audiovisuel, 172
Rappel, 86
Reconnaissance des gestes, 124
Remixage, 30, 112
Requêtes, 3, 30
de modalités croisées, 172, 199
Resynchronisation, 202
RFE-SVM, 66

Sélection d'attributs, 64, 178
Séparabilité, 94, 232
Séparation
aveugle, 24, 102
informée, 27, 102
Séquence, modèle de, 68
SAR, 112
SDR, 112
SEF, 15, 56
Segmentation
en mouvements, 194
en notes, 55
en plans, 191
en régions, 133
en séquences, 193
en sections, 177
Semi-automatique
classification, 161
segmentation, 146
Sequitur, 79
SIR, 41, 112
SMO, 231
Sobel, opérateur de, 138
Sonie spécifique, 222
Stéréo, 31, 36
Stochastique, composante, 43
Structure, 70, 177, 197
Suivi de sous-espace, 46
SVM, 42, 64, 155, 223
à une classe, 183
Synchronie, 173, 197, 198, 202

Tabla, 14, 19
Tatum, 70
Taxonomie, 52
Tempo, 17, 42
TFCT, 15, 56, 109
Toms, 32
Transformée de distance euclidienne, 141
Transitoires, 32, 42

Vecteur de mouvement, 195
Vecteurs de support, 94, 225
Viterbi, algorithme de, 76
Vraisemblance, 77, 83, 182, 184

Witten-Bell, lissage de, 75

ZCR, 221