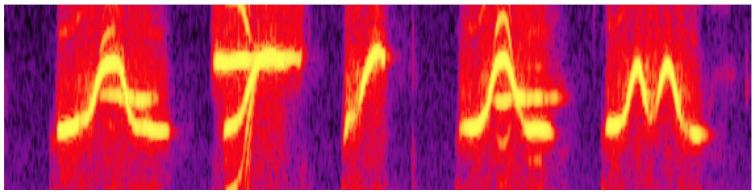
Mémoire de stage de recherche ATIAM

Développement de nouvelles mesures de similarité pour sons environnementaux basé sur le modèle cortical STRF

Auteur :
Edgar Hemery

Encadrant :
Jean-Julien Aucouturier

31 juillet 2013



Remerciements

Je voudrais exprimer ma sincère reconnaissance à mon encadrant, Jean-Julien Aucouturier pour tout ce qu'il m'a appris, pour sa grande disponibilité et sa pédagogie exemplaire. Ce fut un réel plaisir de travailler sous sa direction et de profiter de son expertise scientifique. Ses conseils ont été vraiment encourageants et pertinents dans le cadre de ce stage, mais le resteront aussi par la suite.

Merci également à tous les membres de l'équipe Perception & Design Sonore qui m'ont permis de travailler dans d'excellentes conditions pendant ces cinq mois de stage. Je remercie particulièrement Olivier Houix de nous avoir prêté son disque dur, grâce auquel j'ai pu trouver toutes sortes de sons.

Merci aux ATIAMs 13' qui ont su me trouver pour "la pause café".

Enfin, merci à ma mère pour la relecture attentive et compréhension de ce mémoire.

Table des matières

Résumé

Ce travail s'inscrit dans la lignée des travaux de la communauté MIR (*Music Information Retrieval*) sur les mesures algorithmique de similarité acoustique entre des sons. L'originalité de ce travail est de se baser sur le modèle STRF (*spectro-temporal receptive fields*), un modèle neuro-inspiré et qui donne des valeurs réparties dans quatre domaines physiques : le temps, la fréquence (comme le classique spectrogramme), mais aussi les *scales* (comme les MFCCs) et les *rates* (taux de modulation temporelle). Nous montrons que le modèle STRF est en fait une méta-représentation à partir de laquelle la plupart des autres représentations acoustiques classiques du MIR peuvent être dérivées.

En appliquant des méthodes de réduction de dimension et des techniques fondamentales du traitement du signal et du machine learning, nous construisons et évaluons au total 88 algorithmes différents pour mesurer des distances entre les STRFs de paires de sons. Notre meilleur algorithme, baptisé "cepstral dynamic frequency warping" donne une précision à 90.9% sur notre base de données de sons courts environnementaux, dépassant de prés de 20% la seule proposition existante qui utilise les STRFs (Patil et al., 2012), et avoisinant le résultat d'un algorithme appliquant une distance de kernel gaussien sur des MFCCs qui est de 92,3%. Ce travail fournit donc une alternative complétement neuro-inspirée aux classiques MFCCs (beaucoup critiqués pour ne pas l'être), premier pas important pour établir un dialogue interdisciplinaire entre MIR et les sciences cognitives.

Chapitre 1

Introduction

1.1 La similarité : un problème à la base de l'analyse de contenu

L'étude de la similarité prend racine dans la psychologie (Tversky, 1977), la recherche d'information (*information retrieval*) et l'épistémologie. Utilisée pour la perception et la catégorisation depuis les années 90 (Goldstone et al., 1991; Deutsch, 1999), elle permet d'obtenir une mesure quantitative sur la ressemblance entre les *items* des bases de données et de les classer selon certains critères.

Dans le domaine musical, la recherche de similarité entre deux enregistrements, basée sur l'analyse de leurs contenus audio est, depuis les années 2000, la tâche principale des systèmes de *Music Information Retrieval* (MIR). Connaître la similarité existante entre plusieurs sons permet de les comparer pour ensuite les classifier et faciliter la recherche entre de grandes bases de données audio. Cela est devenu indispensable pour la distribution digitale de la musique sur internet : les utilisateurs veulent retrouver rapidement des chansons qu'ils connaissent déjà, mais aussi en découvrir d'autres grâce à des systèmes de recommandation (Celma and Lamere, 2011).

Afin d'extraire l'information nécessaire à l'étude de similarités ou de dissimilarités entre sons, nous créons des vecteurs caractéristiques (*feature vectors* en anglais) qui donnent une représentation de ces derniers, faisant ressortir les ressemblances ou différences significatives. On représente ainsi les trames d'un signal échantillonné en feature vectors dans un espace abstrait à plusieurs dimensions dans lequel on effectue la mesure de distance ou la tâche de

classification. L'expérience dans le domaine du MIR montre que le choix des caractéristiques audio, ou *features* dépend du type de sons que l'on compare, mais aussi surtout de ce que l'on veut comparer dans les sons : leurs timbres (Blum and Wold, 1996; Logan and Salomon, 2001; J.-J. and Pachet, 2002; Berenzweig et al., 2003), leurs "genres" (Tzanetakis and Essl, 2001), leurs harmonies (Gómez, 2006) etc. La première motivation est donc de choisir les features, qui permettent d'exprimer toute la variabilité des données de la façon la plus compacte possible.

Il n'existe pas une façon unique de classer les sons, mais bien au contraire, autant de façons que d'aspects sur lesquels peut s'arrêter le jugement perceptif. Cela peut être un paramètre basique comme le tempo, le barycentre spectral (*spectral centroid*), la largeur de bande, le niveau sonore, le pitch, mais également un paramètre plus complexe comme le timbre ou encore le genre et l'humeur générale ressentie (Shao et al., 2008a; Jun et al., 2009). Globalement, la perception de la similarité en musique est multi-dimensionnelle et de ce fait, trouver une représentation qui peut retranscrire tous ces aspects permettrait de créer un excellent système de classification. Toutefois, une représentation de trop haute dimension est problématique car elle est trop sensible aux changements locaux, c'est à dire qu'elle se prête mal à l'apprentissage. Ce problème, bien connu en reconnaissance de formes, sous le terme de malédiction des hautes dimensions (*curse of dimensionality*) et décrit par Bishop (2006)), montre que le nombre de données d'apprentissage augmente exponentiellement avec la dimension de l'espace dans lequel elles sont représentées. Ainsi, si nous avons plus de dimensions que de données, nous risquons de faire de mauvaises modélisations. Réduire la dimension des données permet donc de trouver un espace vectoriel dans lequel les distances sont robustes.

La recherche sur la similarité des sons a été menée dans trois domaines : la représentation symbolique des sons, la représentation acoustique, et l'information donnée par l'homme (*collaborative tagging*) (Berenzweig et al., 2004). La représentation symbolique, comme celle des partitions MIDI (Shao et al., 2008b), est bien appropriée à la musique ; cependant, si l'on possède seulement un fichier audio et que l'on cherche à analyser des sons courts, il faut opter pour une représentation acoustique. C'est le problème qui nous intéresse dans ce mémoire.

1.2 Problème de la représentation

1.2.1 Mel-Frequency-Cepstrum-Coefficient (MFCC)

Des décennies de recherche sur la parole ont abouti à des systèmes qui extraient les caractéristiques (features) et modèles utilisés pour la reconnaissance automatique de la parole. Les MFCCs sont unanimement acceptés comme étant les features dominantes pour cette tâche depuis des décennies comme l'on fait (par ex., Rabiner and Juang (1993); Young et al. (1999)).

Le procédé, pour obtenir les MFCCs, est le suivant : on divise un signal sonore en trames de quelques millisecondes (typiquement 10 à 30ms) en le multipliant par une fonction fenêtre d'observation (ex : fenêtre de Hamming) ; cela permet de réduire les effets de bord responsables d'éventuels artefacts spectraux lorsque l'on calcule ensuite la Transformée de Fourier Discrete (TFD). Une fois le spectre de chaque trame obtenu, il faut le traiter en prenant le logarithme de la TFD afin de lisser le spectre pour en extraire l'enveloppe spectrale. L'étape suivante, la transformation en "Mels", sert à simuler la résolution fréquentielle de l'oreille humaine en faisant ressortir les fréquences les plus significatives pour la perception humaine. Ainsi, les intervalles de fréquence sont espacés de manière à ce qu'il y ait plus de résolution fréquentielle dans les basses fréquences que les fréquences hautes. Cet espacement correspond à l'échelle de "Mel" (Mel scale), échelle logarithmique dans laquelle un changement à l'octave supérieur est un doublement de la fréquence mel, contrairement à la perception de la fréquence en hertz, qui n'est plus linéaire mais logarithmique à partir de 1000 hertz. Enfin, la dernière étape consiste à décorréler les composantes des spectres "mel" (25 en général) en utilisant une transformée en cosinus discrète (TCD). Cette transformée créée des coefficients réels, contrairement à la TFD.

$$M_{mel} = 2595 \log_{10} (1 + F_{hz}/700) \tag{1.1}$$

Il a été montré que dans le cas de la musique, les spectres *mel* s'adaptent bien malgré la très grande proportion de fréquences hautes (Logan, 2000).

Beaucoup de travail a été effectué avec les MFCCs depuis les 10 dernières années car cette représentation est extrêmement compacte et donne des pourcentages de reconnaissance très satisfaisants pour de nombreuses bases de données. Ce sont les features employés par la plupart des groupes travaillant sur la similarité audio (Blum and Wold (1996), Aucouturier and Pachet

(2004), Logan (2000),etc...). Les résultats du dernier Music Information Retrieval Evaluation eXchange, qui est associé à la conférence ISMIR montrent que les MFCCs sont toujours les features les plus utilisés et les plus performants dans les systèmes de classification. Face aux diverses tâches à accomplir en similarité musicale, il faut savoir choisir les bonnes *features* qui nous permettront d'exploiter et d'interpréter au mieux les résultats. La littérature dans ce domaine est vaste, et il existe de nombreux modèles d'indexation utilisant des représentations MFCC, de détection d'onset et de changement d'accords. Il n'est pas nouveau de mélanger des features, comme dans Tzanetakis and Cook (2002), où un seul vecteur contient tous les features précédemment évoqués pour représenter un son. L'essentiel est de capturer l'essentiel de l'information d'un échantillon sonore et de le représenter en un vecteur le plus compact possible. Nous mesurons ensuite la similarité ou la distance entre sons, deux à deux, et remplissons ainsi des matrices de similarité ou de distance (c.à.d de dissimilarité).

1.2.2 Représentation du temps

Ayant un vecteur de caractéristiques (*feature vector*) par trame temporelle, chaque son est donc représenté par une matrice dont la dimension correspond à son nombre de trames multiplié par le nombre de dimensions dans lesquelles les trames sont exprimées. Si l'on veut comparer deux sons qui n'ont pas la même longueur (ce qui est presque toujours le cas), se pose alors le problème de comparer des données de dimensions différentes. Ce problème est en fait celui de la représentation du temps, et il y a eu de nombreuses solution proposées depuis les débuts de la recherche en MIR.

Une première solution, la plus simple, est de remplacer la série temporelle de feature vectors par une statistique issue de cette série. Par exemple, on peut réduire un morceau à son feature vector moyen. C'est la stratégie "historique" employée par Blum and Wold (1996) pour le système MuscleFish. Chaque morceau est ainsi représenté par un unique feature vector, que l'on peut comparer à celui des autres morceaux par exemple par distance euclidienne.

Une deuxième solution consiste à utiliser une distance se prêtant à la comparaison de séries temporelles. Contrairement à la stratégie "MuscleFish", ces distances feront typiquement la différence entre une série de feature vector et la même série présentée en sens inverse. Une distance possible est la distance de Levenshtein ou *dynamic time warping* (Bellman, 2003) (voir

2.3.3).

Si l'on visualise chaque son comme un nuage de points à N dimensions, il est également concevable de le modéliser avec une distribution statistique. Les modèles de mélange de gaussiennes (*Gaussian Mixture Model*), expliqués dans la partie 2.2.2, sont souvent utilisés afin de modéliser et comparer des distributions de points. Cette technique est très efficace pour la mesure de similarité de chansons (Logan, 2000). Dans leur approche "bag-of-frames" (BOF), Aucouturier et al. (2007) modélisent une distribution globale des features (MFCC) avec des GMMs. Cette méthode est très efficace pour classifier des scènes sonores (*soundscapes*). Elle est aussi très performante dans l'analyse de la musique polyphonique ; son succès est tel que un quart des systèmes proposés à la conférence ISMIR utilise cette approche depuis sa création [1]. De plus, le type de descriptions qu'elle permet d'extraire est très large, allant du genre au language chanté. La particularité du bag-of-frames est qu'il ne tient pas compte de l'ordre temporel puisque les trames temporelles sont considérées comme des points dans un espace vectoriel.

D'autres propositions, plus minoritaires, visent à représenter l'évolution (ou "modulation") temporelle des feature vectors en en calculant la transformée de Fourier sur plusieurs secondes. C'est le sens, par exemple, des Fluctuation Patterns de Pampalk (2006) ou, à l'IRCAM, du *Modulation Spectrum* de Peeters (2004).

1.3 Motivation pour une nouvelle approche neuro-inspirée :

La multidisciplinarité entre les sciences informatiques, le traitement du signal, et les sciences cognitives ont un réel succès dans le domaine de l'image. En effet, les domaines de la vision par ordinateur et du traitement de l'image s'inspirent de modèles neuroscientifiques depuis les années 50 (Kuffler, 1953). Le concept des champs réceptifs (*receptive fields*) est bien établi et des neuroscientifiques comme Haldan Keffer Hartline (Hartline, 1939) ont réussi à caractériser les stimulis encodés par les neurones sensoriels depuis les années 40. La compréhension des neurones dans le cortex visuel est intégrée depuis plusieurs décennies (Blakemore and Campbell, 1969). Puisque le système de

1. voir par ex. les soumissions du concours MIREX 2012 : `http://www.music-ir.org/mirex/wiki/2012:MIREX_Home`

vision humain est notre point de référence, il est naturel que l'on cherche à l'émuler. De plus, le système visuel cortical est maintenant assez bien compris et modélisé. Il y a un réel intérêt computationnel à faire du bio-inspiré, et les modèles consituent l'état de l'art en reconnaissance de formes et vision par ordinateur (Serre et al., 2007).

Dans le domaine du signal sonore, pourtant, jusqu'à présent toutes les tentatives de faire du bio-inspiré n'ont rien donné (voir Aucouturier and Bigand (2012)) car on a besoin d'une meilleure représentation - or, si la connaissance du cortex visuel a précédé de beaucoup celle du cortex auditif, nous avons maintenant une bonne compréhension de la réponse des neurones, dans les voies auditives, appelés champs réceptifs spectro-temporels (*spectro-temporal receptive fields*). Il y a depuis une dizaine d'années un modèle cognitif computationnel (Patil et al., 2012) comme nous allons le voir. Il n'y a donc plus vraiment d'excuses à ne pas utiliser ces modèles en MIR (Aucouturier and Bigand, 2013).

D'autre part, il y aurait un intérêt scientifique à rapprocher les disciplines du traitement du signal et des sciences cognitives. En effet, les GMMs de MFCC ne permettent pas le dialogue entre le *Music Information Retrieval*, la psychologie cognitive et la neuroscience (pour une discussion récente de ce problème, voir par exemple Aucouturier and Bigand (2012)).

1.3.1 Objectif du stage :

Dans le cadre du stage, nous tentons de développer une mesure de similarité acoustique à partir des STRFs. Les STRFs, acronyme de "Spectro-Temporal Receptive Fields" ou " Champs réceptifs Spectro - Temporels ", sont à la base d'un modèle neuro-computationnel qui stimule le processus observé dans le thalamus et le cortex auditif primaire pendant l'écoute chez le mammifère et plus précisément chez l'homme. Le modèle STRF donne des valeurs énergétiques réparties sur quatre domaines physiques : le temps et la fréquence (comme un spectrogramme), mais également les " rates " (taux de modulation temporelle) et les " scales " (échelles de modulation fréquentielle, ou quéfrence cepstrale). En se basant sur les traitements cognitifs, nous avons une représentation plus générale que celle du spectre (temps-fréquence) et que le cepstre (temps-quéfrence). De plus, le modèle STRF semble pouvoir expliquer les jugements psychoacoustiques du timbre mieux que le spectre (Patil et al., 2012).

Notre objectif est de construire des fonctions de similarité basées sur

la représentation STRF et de tenter de reproduire voir surpasser les performances des algorithmes non biologiquement inspirés, de type GMM de MFCC. Pour cela, nous explorerons une large classe de stratégies de représentation du temps (dont celles vues en Section 1.2.2) basée sur cette représentation. Nous généraliserons ensuite ces méthodes sur tous les domaines des STRFs : si un son peut être vu comme la série temporelle de features dans les domaines fréquence × rate × scale, il peut être vu, de façon équivalente, comme une série en fréquence de features dans les domaines temps × rate × scale, ou une série en rate de features dans les domaines temps × fréquence × scale, etc. Ainsi, nous verrons que la représentation STRF est en fait une méta-représentation de la plupart des autres représentations audio : le spectre intègre les domaines rate et scale pour obtenir une représentation en temps et fréquence ; le cepstre intègre les domaines fréquence et rate pour obtenir une représentation en temps et scales ; les *fluctuations Patterns* de Pampalk (2006) sont une représentation fréquence × rate (une *fluctation pattern* est une matrice à deux dimensions où les lignes correspondent à des bandes fréquentielles et les colonnes à des modulations de fréquences entre 0 et 10 hertz), etc. Toutes ces propositions antérieures peuvent donc être vues comme des "cas particuliers" du modèle général STRF ; nous les testerons comme telles dans ce mémoire, et nous essayerons de dériver de nouvelles représentations sur le même mode : par exemple, est-ce qu'une représentation fréquence × scale (c.-à-d. qui intégre temps et rate) est intéressante ? Est-il pertinent de faire des séries fréquentielles plutot que temporelles ? Des GMMs dans le domaine scale plutot que le temps ? etc.

Précisons pour finir que ce travail se donne comme problème la simulation de jugements humains de similarité entre sons courts environnementaux (utilisant pour se faire des bases de données disponibles dans l'équipe PDS de l'IRCAM - Houix et al. (2011)). Cependant, les résultats que nous allons présenter restent informatifs, nous le pensons, pour d'autres problèmes comme les sons musicaux courts (Patil et al., 2012), les textures environementales longues (ou *soundscapes*) et les morceaux de musique longs.

Chapitre 2

Algorithmes

2.1 Une nouvelle représentation audio neuro-inspirée : le modèle STRF

L'équipe de Shihab Shamma, professeur à l'Université de Maryland, s'est inspirée des recherches neuro-physiologiques qui analysent l'architecture corticale, hiérarchique à la Serre et Poggio Serre et al. (2007) reposant notamment sur l'enregistrement de milliers de neurones dans le cortex primaire du furet. Cette recherche sur les STRFs, menée depuis plus de dix ans par plusieurs chercheurs à travers le monde a abouti en novembre 2012 à un papier intitulé "Music in Our Ears : The Biological Bases of Musical Timbre" Patil et al. (2012). Posant les bases neuro-computationnelles du modèle STRF, cette équipe constituée de Kailash Patil, Daniel Pressnitzer (ENS, ex-IRCAM), Shihab Shamma et Mounya Elhilali a créé une toolbox de fonctions Matlab qui permet d'obtenir une représentation de type "spectrogramme cortical" à partir des sons. Ils ont montré que les caractéristiques spectro-temporelles de cette représentation permettent la reconnaissance de timbres musicaux avec 98.7% de précision (Patil et al., 2012) et si l'on applique l'ingénieurie inverse, la re-synthèse des sons à partir de l'activation des neurones dans le cortex auditif primaire (Pasley et al., 2012). Ce modèle se différencie donc des autres descriptions spectro-temporelles du son telles que les Mel-Frequency Cepstral Coefficients (MFCC) du fait qu'il est directement inspiré par la réalité biologique.

2.1.1 Le modèle cortical :

Les STRFs sont les fonctions de transfert des réponses neuronales mesurées dans le cortex auditif primaire A1. Le cortex auditif primaire, consitué de deux parties, que l'on désigne A1 et A2, est responsable de l'intégration temporelle des sons, c'est à dire de l'ordonnancement des mots ou sons entendus (Carrasco and Lomber, 2009). Des expériences où l'on a retiré le cortex auditif chez l'animal montrent qu'il ne parvient plus à discerner des sons complexes de même fréquence et de structure temporelle différente. A1 ne répond pas à des sons simples de type sinusoïdal, mais à des sons complexes modulants en temps et en fréquence, que l'on appelle *ripples*. A2 est bien moins compris et ne semble pas être aussi simplement structuré que A1, il réagit à des structures temporelles plus longues et est sensible à des motifs sonores plus complexes. Les filtres spectro-temporels, modulants de facon similaire aux neurones de A1 ont été mesurés sur le cortex primaire auditif du furet grâce à la méthode du *ripple analysis* (Shamma et al., 1995).

Les caractéristiques physiologiques des neurones sensoriels ont été mesurées chez le furet pour la similarité entre leur cortex auditif et celui de l'homme. Cependant, le système auditif périphérique humain précédant le traitement du cortex auditif, nous appliquons une première transformation sur les signaux acoustiques, imitant la transformation faite par la cochlée. Cette transformation est simulée par une banque de 128 filtres passe-bandes asymétriques et à Q-constant qui sont également répartis sur une échelle logarithmique couvrant 5.3 octaves.

Le modèle cortical analyse ensuite le contenu spectro-temporel du spectrogramme auditif en utilisant une banque de filtres, centrés sur chaque fréquence de l'axe tonotopique, et qui modélisent les champs réceptifs neurophysiologiques. Chaque filtre est accordé à un rate spécifique, correspondant à une modulation temporelle (en hertz) et un scale spécifique, correspondant à une modélisation spectrale (en cycle/octave). Ces modulations spectro-temporelles, respectivement *rates* et *scales* sont obtenues en appliquant une transformée de Fourier rapide (FFT) sur l'axe des fréquences puis du temps sur le spectrogramme auditif. La FFT appliquée sur l'axe des fréquences, résultant en un cepstre, est une application connue en traitement du signal. Cependant, la modulation sur l'axe du temps, donnant les rates mesurés en hertz est plus inhabituelle, rappelant les *fluctuation patterns* de Pampalk (2006) ou le modulation spectrum de Peeters et al. (2002).

Les filtres spectro-temporels, respectivement h_r (*rates*) et h_s (*scales*), ont

une allure d'ondelettes, convoluées par la suite avec le spectrogramme pour modéliser un STRF. Un STRF est donc centré sur une bande fréquentielle précise de l'axe tonotopique, mais aussi sur un rate et un scale spécifique.

Mathématiquement, ces filtres peuvent être mis en équation de la façon suivante :

$$STRF = h_{IRT}(t) * h_{IRS}(x) \tag{2.1}$$

avec :

$$h_{IRS}(w, \Omega, \phi) = h_s(x; \Omega) \cos\phi + \hat{h}_s(x; \Omega) \sin\phi \tag{2.2}$$

et

$$h_{IRT}(t, w, \theta) = h_t(t; w) \cos\theta + \hat{h}_s(t; w) \sin\theta \tag{2.3}$$

La réponse de chaque STRF dans le modèle est donné par l'équation :

$$r_{\pm}(t, f; w, \Omega; \theta, \phi) = z(t, f) *_{t,f} STRF_{\pm}(t, f; w, \Omega; \theta, \phi) \tag{2.4}$$

ou $*_{t,f}$ est une convolution en temps et fréquence, Ω et w les scales et rates respectivement. θ et ϕ sont les phases caractéristiques qui déterminent le degré d'asymétrie sur les axes temps et fréquence respectivement (observable sur la figure 2.1).

Ces ripples (ou ondulations) représentées par $h_{IRS}(t)$ et $h_{IRT}(t)$, mettent en évidence les champs excitatifs et inhibiteurs du cortex auditif en le stimulant avec des signaux dont les enveloppes spectro-temporelles modulent de façon sinusoïdale sur un bruit à large bande. Afin de mieux se rendre compte du rôle de ces filtres, nous avons visualisé à l'aide de Matlab, les descriptions spectro-temporelles de certains sons. La figure suivante représente l'activation des filtres h_s et h_r pour un son de rugissement de lion (voir figure 2.1). Il faut préciser que dans chaque case de cette figure, toutes les fréquences sont intégrées sous forme de spectrogramme auditif tandis que les champs réceptifs sont sélectifs sur une fréquence tonotopique donnée dans le modèle STRF. Cette figure permet d'observer une représentation des rates et des scales.

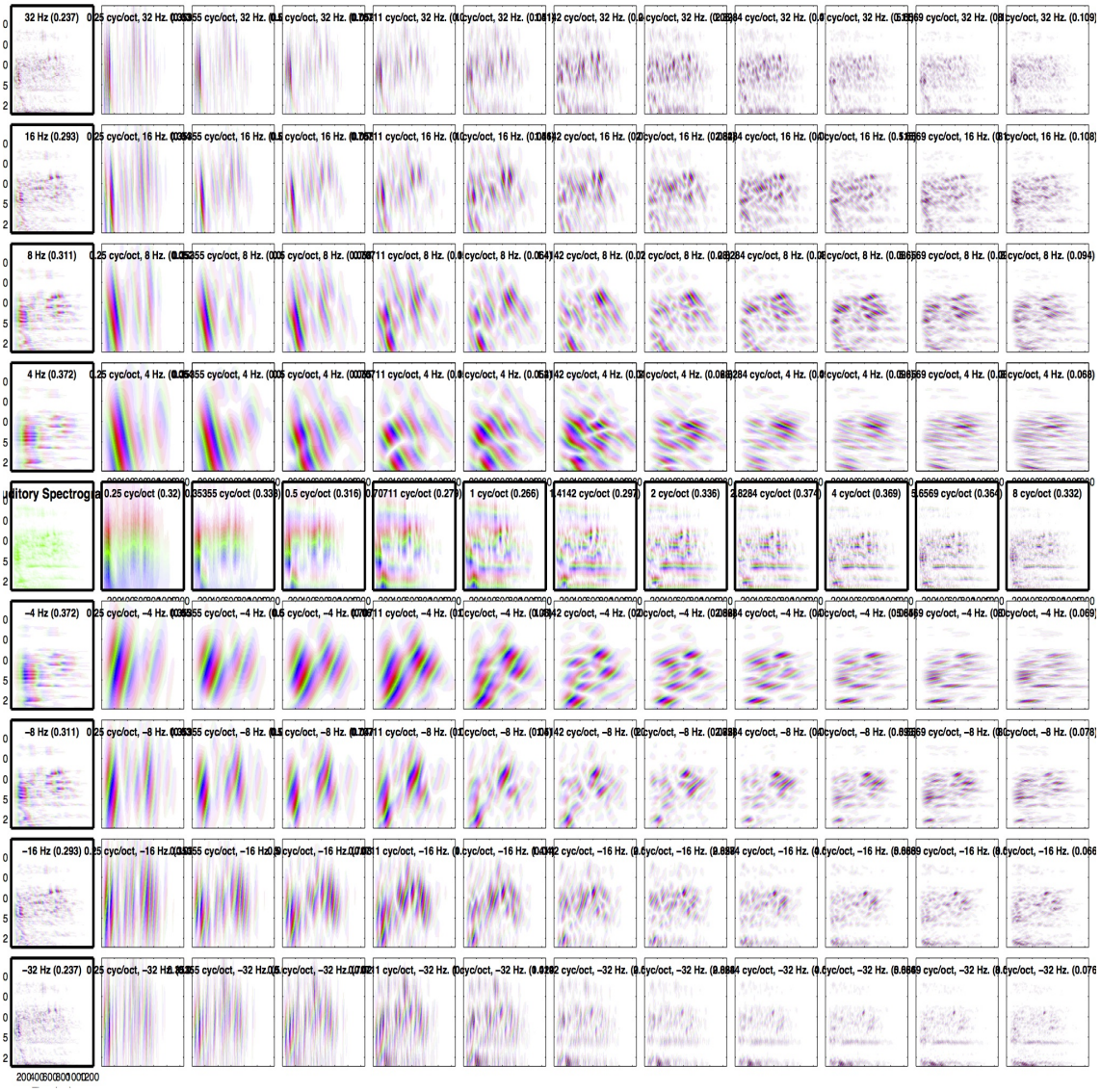


FIGURE 2.1 – Représentation corticale d'un son de rugissement de lion.

La figure 2.2 est une description plus rigoureuse du modèle STRF. Nous prenons ici le spectrogramme auditif d'une gamme de Shepard et y appliquons toutes les transformations qui arrivent dans le thalamus et cortex auditif pour une valeur de rate et scale. Nous choisissons une gamme de Shepard comme exemple, car ayant 12 sons d'une seconde chacun, nous

pouvons observer une périodicité de rate = 1Hz dans le temps et de scale = 1 cycle/octave dans les fréquences. Les différentes opérations sont détaillées ici et fléchées dans l'ordre chronologique.

- Premièrement, le modèle simule la transformation effectuée par la cochlée, représenté par un spectrogramme auditif. L'échelle des fréquences est logarithmique et il y a 24 canaux par octave (SR=24 c/o). Le temps est échantillonné à 125 hertz.
- Nous appliquons une transformée de Fourier sur chaque trame temporelle par rapport aux fréquences, résultant en un "cepstrum", en *scale* dont l'unité est le cycle/octave (c/o).
- Nous appliquons ensuite une transformée de Fourier sur chaque trame "cepstrale" par rapport au temps et obtenons un spectre de fréquence en *rate*. Le spectrogramme résultant de ces deux transformées de Fourier est en scale (c/o) vs. rate (Hz).
  Remarque : comme dans le modèle de Shamma, nous avons gardé toutes les fréquences dans la seconde transformée, c.à.d. les rates négatifs allant de -SR/2 à 0 et les rates positifs allant de 0 à SR/2.
- Nous pouvons maintenant filter le spectrogramme avec les filtres STRF : nous multiplions les trames de scale avec H_r qui est la projection du filtre STRF sur l'axe des rates. H_r est un filtre type passe-bande centré sur une fréquence de coupure donnée (ici, $r_c = 1$ Hz).
- L'étape suivante consiste à appliquer une transformée inverse sur l'axe des rates pour retourner à une représentation scale (c/o) vs. time (trames). Nous multiplions alors les trames temporelles avec H_s, la projection du filtre STRF sur l'axe des scales. Ici, H_s est centré sur la fréquence de coupure $s_c = 1$ c/o.
- Nous faisons une seconde transformée de Fourier inverse, cette fois sur l'axe des scales. Nous retournons ainsi à une représentation fréquence (Hz) vs. temps (trames).

Dans cette dernière représentation, chaque trame de fréquence correspond à la sortie d'un seul neurone centré sur une fréquence particulière de l'axe tonotopique, répondant à une seule valeur de rate et une seule valeur de rate.

Les opérations décrites sont réitérées pour chaque STRF. Patil et al. (2012) proposent d'utiliser 22 valeurs de rates, 11 de scales, donc 242 valeurs de STRFs par fréquence de l'axe tonotopique. L'axe de fréquence étant échantillonné à 128 valeurs, nous avons donc au total 128×242=30,976 valeurs par trame temporelle avec cette représentation.

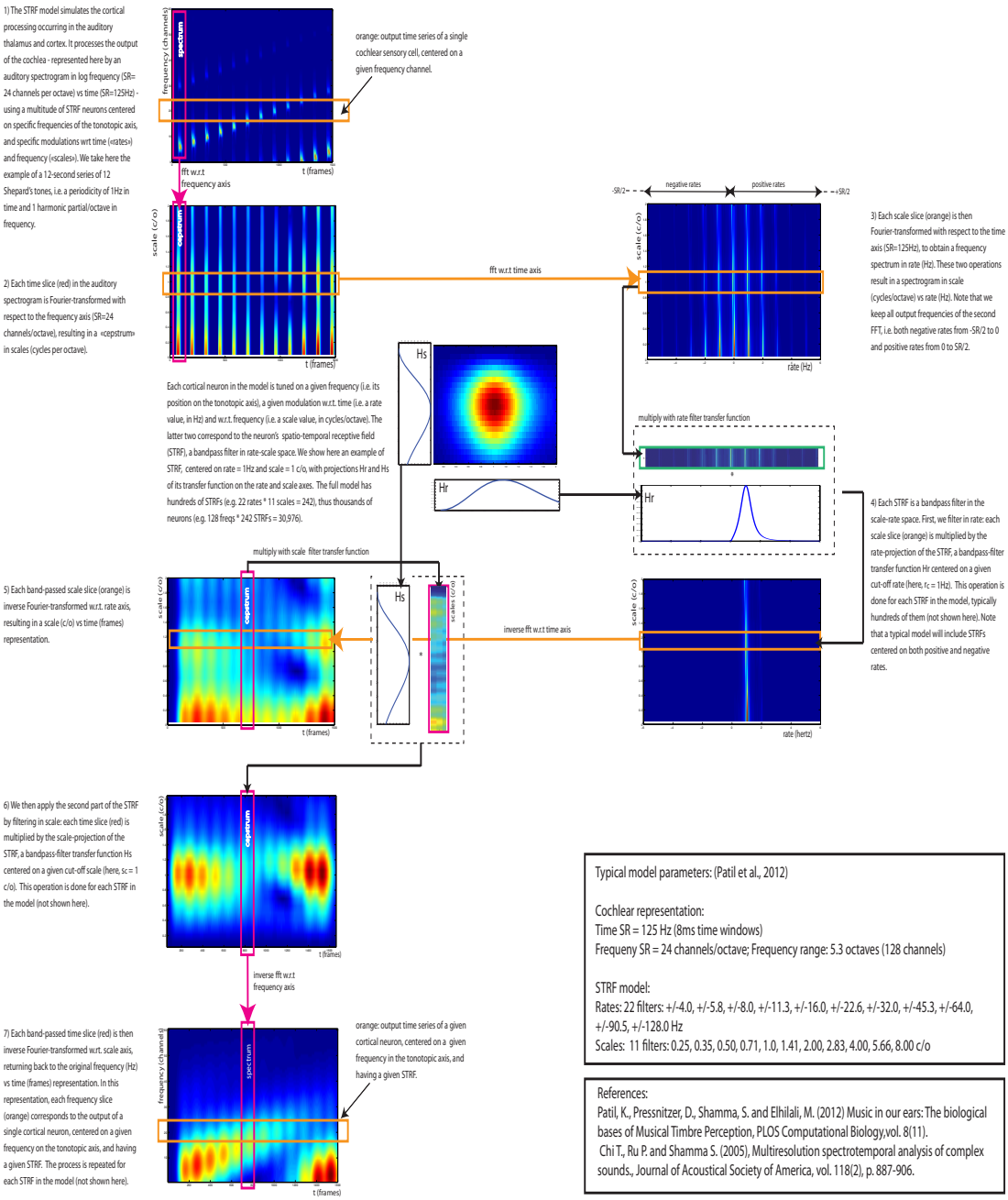


FIGURE 2.2 – Shamma's STRF model : a signal processing primer.

2.2 Calculer une similarité à partir de cette représentation

L'objet de cette recherche étant de comparer les représentions corticales entre sons environnementaux, nous avons fait un algorithme qui mesure la distance entre les valeurs données par les STRFs et constitue ainsi des matrices de dissimilarités.

En prenant les paramètres donnés par Patil et al. (2012), un STRF donne, comme nous l'avons vu, une valeur (une sorte d'énergie) pour 22 valeurs de rates différents, 11 valeurs de scales différents, 128 valeurs de fréquences différentes, et 250 positions temporelles par secondes (une trame=4ms). Si l'on garde dans notre représentation ne serait-ce que trois domaines (rate, scale, fréquence), chaque son est représenté par un feature vector de $128 \times 22 \times 11 = 30{,}976$ valeurs, c'est à dire que l'espace des features est de dimension 30,976.

Le problème qui se pose est de réduire le nombre de ces dimensions. Par exemple, si on "écrase" en moyennant le domaine du temps et des fréquences, chaque son est représenté par 22 valeurs de rate pour chacun des 11 scales, alors les features sont plus que de dimension $22 \times 11 = 242$, ce qui pourrait nous permettre de mesurer des distances. Cependant, les informations perdues au niveau du temps et des fréquences sont peut être nécessaires afin de comparer les sons. Il faut donc repenser quelle dimensions peuvent être réduites et comment.

On essaie donc diverses combinaisons de stratégies sur les STRFs dont les briques sont décrites dans les sections suivantes.

2.2.1 Moyenne sur les domaines STRF

La première approche, qui est la plus triviale, est similaire à celle employée dans Muscle Fish (Blum and Wold, 1996). Les sons de nos bases n'étant pas tous de même longueur, il est normal que les feature vectors changent de taille. Nous sommes donc obligés dans un premier temps de réduire la dimension du temps. La manière la plus simple de le faire est de prendre la moyenne sur toutes les trames temporelles. Ainsi, nous écrasons toute la dynamique sur ce domaine, mais nous pouvons réduire considérablement le problème de dimensionalité et mesurer des distances entre chaque son.

Cette méthode, très directe, réduisant brutalement les dimensions, peut

être appliquée sur les autres domaines STRF. Nous pouvons de cette manière réduire toutes les fréquences, rates et scales respectivement en une seule valeure énergétique.

2.2.2 GMM

Une fois le signal sonore découpé en trames, nous évaluons les features STRF pour chacune d'entre elles, puis modélisons la distribution de toutes les trames en utilisant un modèle mixture gaussien (*Gaussian Mixture Model* - GMM en anglais). Un GMM est un modèle statistique qui estime une densité de probabilité comme une somme pondérée de M fonctions de densités gaussiennes, appelés noyaux. Chaque noyau est décrit par une variance, une moyenne ainsi qu'une amplitude.

$$p(x_t) = \sum_{m=1}^{m=\mathcal{M}} \pi_m \mathcal{N}(x_t, \mu_m, \Sigma_m) \tag{2.5}$$

x_t est un vector de feature, observé à un temps t, N est une fonction de densité de probabilité Gaussienne (loi normale) avec une moyenne σ_m, une matrice de covariance μ_m, π_m le coefficient de pondération du noyau. L'estimation de ces paramètres est effectuée avec l'algorithme espérance-maximisation (algorithme E-M - Bishop (2006)).

Les GMMs réduisent tout un domaine en M points. Si l'on décide de prendre une seule fonction de densité gaussienne, cela se rapproche à prendre la moyenne.

Pour mesurer la distance qui sépare les GMMs, nous utilisons la distance de Kullback-Leilbler (KL) que nous décrivons dans la section 2.3.4.

2.2.3 PCA :

Il existe de multiples techniques de réduction linéaire de dimensions. Nous utilisons ici la plus traditionnelle, à savoir l'analyse en composantes principales (ACP - PCA en anglais) (Shlens, 2005). C'est une manière efficace de transformer des données à grandes dimensions en une représentation réduite en effectuant un changement de base qui permet de mieux représenter la variance des données. L'ACP est utilisée dans de nombreuses formes d'analyse, allant de la neuro-science à l'infographie. L'expérience montre qu'elle permet

de révéler des dimensions cachées et de mettre en évidence la dynamique sous-jacente.

L'ACP sélectionne les directions normalisées dans un espace à M dimensions dans lequel la variance est maximale. Une fois la première direction trouvée p_1, l'algorithme cherche une seconde direction p_2, normale à la première, dont la variance est maximale (donc la deuxième plus grande), etc. Les p's sont ordonnés dans une matrice dont les rangs sont les p_1, p_2, ..., p_m, sont les composantes principales.

2.3 Distances :

Il existe plusieurs façon de mesurer la distance entre nos matrices de similarité. Nous tentons d'appliquer différents types de distances sur nos matrices de dissimilarité. Nous décrivons ici ces distances.

2.3.1 Distance euclidienne :

La distance euclidienne est définie entre deux points x et y ainsi :

$$d(x, y) = \sqrt{(x - y)^2} \tag{2.6}$$

Nous pouvons aussi mesurer la distance entre deux points p et q dans un espace à N dimension :

$$d(A, B) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_i - q_i)^2 + ... + (p_n - q_n)^2} \tag{2.7}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2.8}$$

2.3.2 Distance de Kernel Gaussien :

La distance de kernel gaussien est semblable à la distance euclidienne. Elle nécessite aussi de ne pas avoir de variation de trames temporelles ; il faut donc prendre la moyenne sur le domaine du temps avant de l'utiliser. Sa particularité est qu'elle effectue une transformation de la matrice candidate

en l'optimisant en fonction de la groundtruth : on calcule un vecteur de poids σ_i pour chaque feature vector dont on fait l'apprentissage tel que la distance calculée est la plus proche possible de la groundtruth. En utilisant la méthode de descente de gradiant, on minimise ainsi la différence entre les deux matrices de distance. Il y a donc deux parties pour calculer la distance de kernel : une partie d'apprentissage, où l'on calcule les vecteurs de poids σ_i, en minimisant la fonction de coût :

$$J = -\frac{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(Y(x_i - x_j) - \bar{Y})(K(x_i - x_j) - \bar{K})}{\sigma_K^2} \tag{2.9}$$

où $Y(i,j)$ et $K(i,j)$ sont les composantes des matrices candidates et de référence respectivement. $\bar{Y}$ et $\bar{K}$ en sont les moyennes ; puis le calcul de la distance en utilisant les poids :

$$K(x,y) = \exp(-\sum_{i=1}^{d}\frac{(x_i - y_i)^2}{2\sigma_i^2}) \tag{2.10}$$

2.3.3 Déformation temporelle dynamique :

La déformation temporelle dynamique (*dynamic time warping* - DTW) ou distance de Levenshtein est un algorithme connu dans les sciences informatiques depuis les années 60, qui permet de mesurer la différence entre deux séquences (Muller, 2003). Originellement appliqué aux chaines de caractères, l'algorithme s'applique aussi aux feature vectors. Dans le cas des chaines de caractères, la distance de Levenshtein est le nombre minimal d'insertion, substitution et de suppression nécessaire pour changer un mot en un autre. Ainsi, elle est d'autant plus grande que la différence entre deux chaînes de caractères est grande.

Très utilisée en reconnaissance de la parole, la programmation dynamique permet de prendre en compte l'alignement temporel des phonèmes lorsque ces derniers n'ont pas les mêmes longueurs (Sjölander, 2001). Dans le cas où nous avons des features vectors correspondants à chaque trame temporelle, l'algorithme DTW trouve le "chemin", à travers toutes les distances possible entre trames, qui maximise les correspondances locales entre les trames alignées.

Utilisée pour comparer des séquences de notes (Lemstrom and Laitinen, 2011), la programmation dynamique a également été utilisée pour aligner des

trames dont les MFCCs sont proche mais dont les positions sont éloignées dans Aucouturier et al. (2007). Cela permet de regrouper des sons (ou des trames d'échantillons sonores) de même timbre mais de durées différentes.

2.3.4 Kullback leibler distance :

La distance de Kullback-Leilbler (KL) est la seule distance qui puisse être appliquée aux GMMs. Elle est définie entre deux probabilités de distribution p_A et p_B ainsi :

$$d(A,B) = \int p_a(x) log(\frac{p_b(x)}{p_a(x)}) \tag{2.11}$$

et approximée par :

$$d(A,B) = \frac{1}{n} log(\frac{p_b(x)}{p_a(x)}) \tag{2.12}$$

où n est le nombre d'échantillons.

2.4 Méthodes d'évaluation :

Toute mesure de similarité nécessite d'une matrice de similarité de référence, appelée *groundtruth*, matrice de similarité ou de distance obtenue subjectivement, à partir de laquelle nous nous référons. Le jugement perceptif humain est naturellement notre point de repère et nous pouvons construire des *groundtruth* à partir de plusieurs tests d'écoutes sur différentes populations. Unifier les résultats reste cependant un problème sans solution évidente, sachant que les jugements entre des paires de sons ne sont pas constants chez tous les auditeurs et nous devons consentir au choix le plus représentatif.
Il est aussi risqué de comparer des mesures, obtenues à partir d'analyses sur le signal, à des références basées sur le jugement perceptif humain. Face à ce problème, la meilleure approche est de comparer plusieurs métriques pour la classification, de tester le plus de paramètres différents, puis d'en comparer les résultats afin d'adapter notre méthodologie. La science de la recherche d'information (*information retrieval*) possède de nombreuses métriques différentes ; nous utiliserons ici la R-precision. Cette méthode nécessite une base de données de sons, une requête et une *groundtruth*.

2.4.1 R-précision

La R-précision est la précision à la position r dans le classement d'une requête qui a R résultats pertinents. La précision est égale au rappel (*recall*) à la position r . R est le nombre de plus proche voisin que l'on veut. Le rappel est la fraction de documents trouvés qui sont pertinents à la requête de l'utilisateur.

$$rappel = \frac{(documents\ pertinents) \bigcap (documents\ trouvés)}{document\ trouvés} \tag{2.13}$$

Les résultats de chaque distance calculée sont donc comparés en calculant leur précision après que 10 sons soient trouvés. Chaque valeur donne un ratio du nombre de sons pertinents sur le nombre de sons total trouvés. Les sons "pertinents" sont ceux définis par la groundtruth : tous ceux dont la distance à la cible est égale à 0 pour la DB1 (voir ci-dessous), et les 10 plus proches voisins de la cible pour la DB2. Le calcul de la R-précision a donc besoin en entrée de :

- la matrice de distance d obtenue. Plus la distance est petite, plus le son est proche de son homonyme sur la *groundtruth*.
- la matrice de distance correspondant à la *groundtruth*. Toutes les entrées sont des valeurs entre 0 (distance entre un son et lui même) et 1 (distance entre un son et un autre).

2.5 Bases de données :

2.5.1 DB1

Nous créons une premiere base de données contenant 10 classes de 10 sons similaires. Avec une base de 100 sons, un pour cent de similarité correspond à un son, ce qui est favorable à la métrique.

1. bubble1.wav
2. bubble2.wav
3. bubble3.wav
4. bubble4.wav
5. bubble5.wav
6. bubble6.wav
7. bubble7.wav
8. bubble8.wav
9. bubble9.wav
10. bubble10.wav
11. city@night1.wav
12. city@night2.wav
13. city@night3.wav
14. city@night4.wav
15. city@night5.wav
16. city@night6.wav
17. city@night7.wav
18. city@night8.wav
19. city@night9.wav
20. city@night10.wav
21. citybird1.wav
22. citybird2.wav
23. citybird3.wav
24. citybird4.wav
25. citybird5.wav

26. citybird6.wav
27. citybird7.wav
28. citybird8.wav
29. citybird9.wav
30. citybird10.wav
31. door1.wav
32. door2.wav
33. door3.wav
34. door4.wav
35. door5.wav
36. door6.wav
37. door7.wav
38. door8.wav
39. door9.wav
40. door10.wav
41. flight_info1.wav
42. flight_info2.wav
43. flight_info3.wav
44. flight_info4.wav
45. flight_info5.wav
46. flight_info6.wav
47. flight_info7.wav
48. flight_info8.wav
49. flight_info9.wav
50. flight_info10.wav
51. harbour_dock1.wav
52. harbour_dock2.wav
53. harbour_dock3.wav
54. harbour_dock4.wav
55. harbour_dock5.wav
56. harbour_dock6.wav
57. harbour_dock7.wav
58. harbour_dock8.wav
59. harbour_dock9.wav
60. harbour_dock10.wav
61. pebble1.wav
62. pebble2.wav
63. pebble3.wav
64. pebble4.wav
65. pebble5.wav
66. pebble6.wav
67. pebble7.wav
68. pebble8.wav
69. pebble9.wav
70. pebble10.wav
71. pouringwater1.wav
72. pouringwater2.wav
73. pouringwater3.wav
74. pouringwater4.wav
75. pouringwater5.wav
76. pouringwater6.wav
77. pouringwater7.wav
78. pouringwater8.wav
79. pouringwater9.wav
80. pouringwater10.wav
81. waterways1.wav
82. waterways2.wav
83. waterways3.wav
84. waterways4.wav
85. waterways5.wav
86. waterways6.wav
87. waterways7.wav
88. waterways8.wav
89. waterways9.wav
90. waterways10.wav
91. waves1.wav
92. waves2.wav
93. waves3.wav
94. waves4.wav
95. waves5.wav
96. waves6.wav
97. waves7.wav
98. waves8.wav
99. waves9.wav
100. waves10.wav

2.5.2 DB2

La base de donnée DB2 a été constituée par Olivier Houix de l'équipe PDS (Houix et al., 2011). Elle comprend 60 sons environnementaux courts, qui ont été classés en catégories par 15 non-musiciens, selon une procédure de *free sorting* (cf. une méthode de catégorization). Les participants pouvaient former autant de cluster de sons qu'ils le souhaitaient, et chaque cluster pouvait contenir autant de sons qu'ils le souhaitaient (figure 2.3). La fréquence de co-occurrence de deux sons dans le même cluster, sur l'ensemble des participants, définit une mesure de proximité de ces sons, que l'on utilise pour former une matrice de distance. Contrairement à la base de donnée précédente, cette matrice de distance est continue, et non binaire : les distances entre 2 sons peuvent prendre toutes les valeurs entre 0 et 1, reflétant des degrés variés de co-occurrence dans les clusters de l'expérience de Houix et al. (2011).

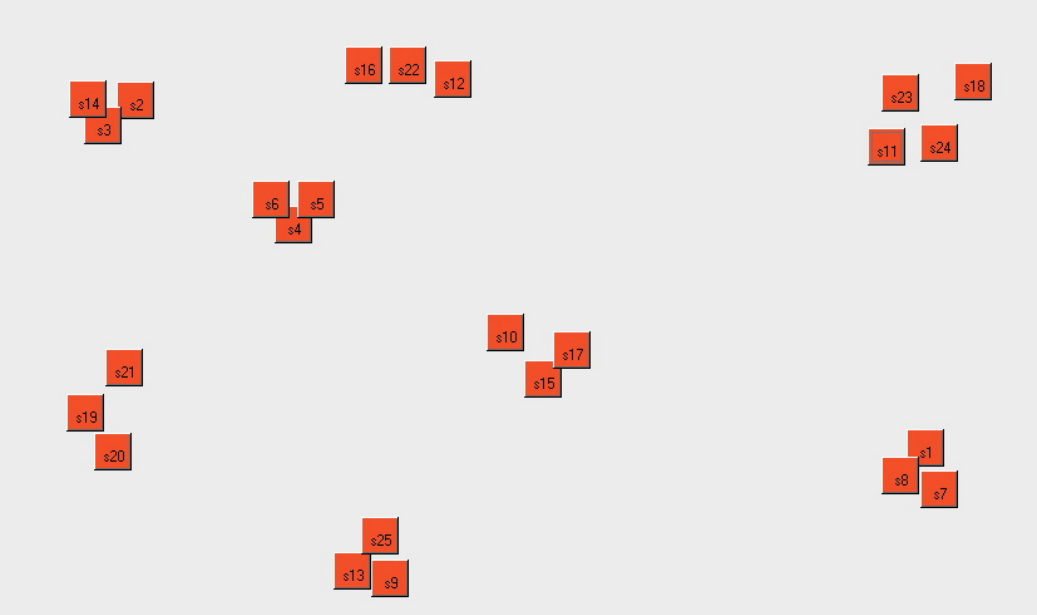


FIGURE 2.3 – Co-occurence

2.6 Equivalence fonctionnelle des domaines STRF :

A la manière des séries temporelles de feature vectors, nous pouvons concevoir d'appliquer les transformations de type GMM, PCA et Kernel Transform sur le domaine des fréquence, des rates et des scales.

Si un son peut-être vu comme la série temporelle de features dans les 3 domaines fréquence × rate × scale, il peut être vu, de façon équivalente, comme une série en fréquence de features dans les domaines temps × rate × scale, ou une série en rate de features dans les domaines temps × frequence × scale, etc. On peut donc imaginer faire des GMMs d'observations temporelles, mais aussi des GMMs d'observations en fréquence, rate, etc. On peut aligner des séries temporelles par DTW, mais aussi des séries fréquentielles par (ce que l'on pourrait appeler) "DFW", des séries de rates par "DRW", etc.

Toutes ces possibilités sont donc fortement combinatoires : on peut combiner PCA, GMM, distances, etc. au sein de chaque permutation possible des domaines. Dans ce très grand nombre de combinaisons, on retombe sur certaines stratégies déjà connues. Par exemple, les GMMs de MFCCs sont

des GMMs appliqués à une série temporelle de feature vectors qui sont en fait des scales (c.à.d des STRFs dont on a intégré la fréquence et le rate). Mais on explorera également des stratégies nouvelles, peu intuitives, mais peut-être aussi (ou plus !) efficaces que l'état de l'art.

Chapitre 3

Résultats

3.1 Expérience préliminaire

3.1.1 Base DB1 et amélioration des valeurs caractéristiques des rates et scales

Jusqu'alors, nous avons simplement ré-utilisé les paramètres des filtres pour les rates et les scales donnés par Patil et al., mais nous réalisons que ces valeures peuvent être améliorées.

Afin de trouver les meilleures valeurs pour les paramètres des rates/scales, nous avons fait un algorithme qui calcule les distances euclidiennes sur DB1 pour différentes densités et plages de rates et de scales. L'idée étant de graduellement augmenter la largeur de l'échelle des rates et des scales respectivement, mais aussi d'augmenter la densité de valeurs prises. Ainsi nous pouvons observer deux graphes, un pour les rates à scale fixé, puis un pour les scales à rate fixé. Nous avons une franche tendance qui montre que lorsque la largeur de bande est maximale, la R-précision augmente. Pour les densités, c'est moins évident, notamment pour les scales comme nous allons le voir.

3.1.2 Evaluation des meilleurs paramètres pour les rates

Nous calculons la R-précision sur notre algorithme de similarité en prenant la moyenne sur le domaine des fréquences, en appliquant une PCA sur les rates et les scales et un GMM sur le temps. Nous calculons la distance

avec la divergence de Kullback-Leibler. Nous nous servons de cet algorithme, sur lequel nous reviendrons dans la partie 3.2, pour comparer les valeurs des rates car il a été a un moment le plus performant. Comme nous l'observons sur la figure 3.1, la meilleure précision ($\sim$0.77 %) apparait en rouge pour une densité à 10 et une largeur de bande allant de 2 à 120 Hz. Il y a 10 valeurs de rates prises régulièrement sur une étendu allant de 2 à 120 hertz.

Cette évaluation nous a permis de nous apercevoir qu'il n'était pas pertinent pour notre probleme de prendre des valeurs négatives contrairement à ce qu'indiquait Patil et al. (2012).

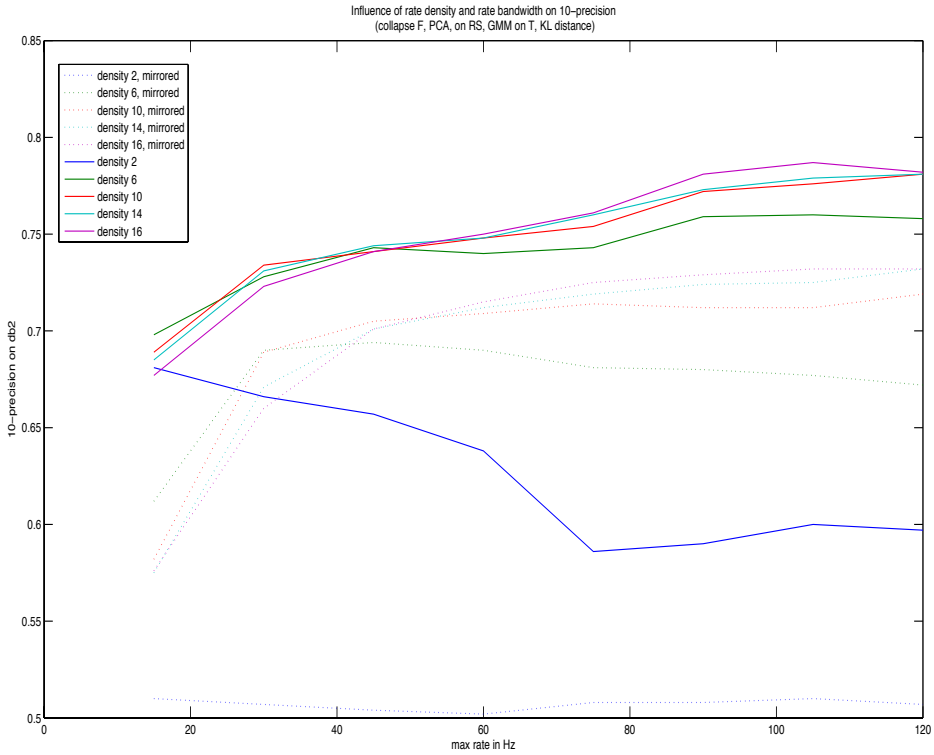


FIGURE 3.1 – Rates.

3.1.3 Evaluation des meilleurs paramètres pour les scales

La même méthode est utilisée pour évaluer les meilleurs scales. Ainsi, nous obtenons une précision de ∼0.80 % pour une densité à 11 et une largeur de bande à 12. Les valeurs vont de 0.15 à 12 c/o. Cela est représenté par la courbe rouge (density 11) sur le graphe 3.2. Nous ne prenons pas la densité qui offre la précision maximale (atteinte à 15), car elle n'est que sensiblement supérieure mais rajoute 5 valeurs par feature vectors et ralongerait conséquemment beaucoup le temps de calcul.

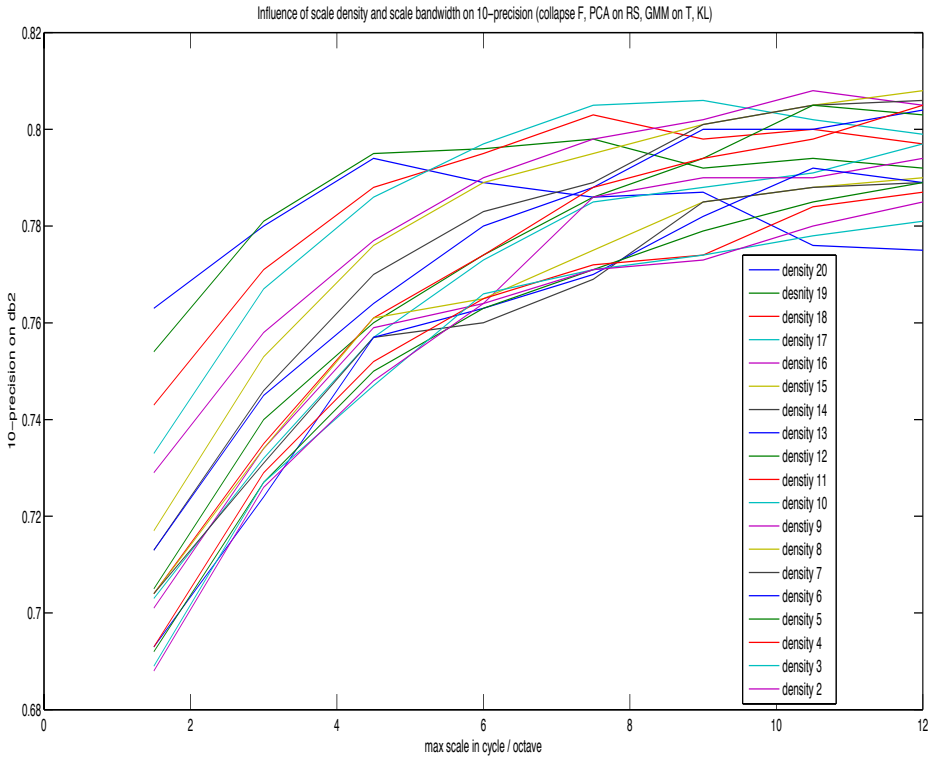


FIGURE 3.2 – Scales.

3.2 Expérience principale

Nous avons conçu notre algorithme de telle sorte qu'il soit le plus modulaire possible. Ainsi, nous pouvons appliquer les différentes transformations sur chacun des domaines STRF successivement et tester toutes les combinaisons possibles. Ceci est précisement l'objectif que nous nous sommes fixé dans l'expérience principale : essayer tous les algorithmes de transformations, calculer toutes les distances et les évaluer avec la métrique R-précision. De ce travail exhaustif découlera le meilleur algorithme pour calculer la similarité dans nos bases de données. Les pictogrammes représentés sur la colonne de gauche dans les figures suivantes représentent les différentes façons de réduire les domaines en les moyennant (ce que nous appelons *feature collapse*). L'étape suivante, non indispensable est la transformation par PCA, GMM, ou Kernel. Enfin, la dernière étape est le calcul de la distance KL, Kernel, Euclidienne ou DP selon les cas.

Il y a deux stratégies : soit on considère un son comme une série de feature vectors qui peuvent être exprimés sur n'importe lesquel des domaines STRF, et on utilise des méthodes de comparaison de série : GMM ou DP, soit on considère un son comme un seul feature vector, par exemple en faisant la moyenne des vecteurs d'une série ; dans ce second cas, on utilise des méthodes de comparaison de vecteur, comme la distance euclidienne ou les kernels gaussiens. Pour des raisons de clarté, nous séparons le cas de figure "série" en 4 domaines : les séries sur le temps (l'approche classique du MIR depuis 10 ans), les séries sur la fréquence, sur le rate et le scale (ces deux dernières étant complétement originales). Nous décrivons ensuite le second cas de figure où l'on réduit toutes les séries à un seul feature vector.

Nous utilisons la R-précision à R=10 pour évaluer nos algorithmes. Ce choix est bien approprié pour DB1, cependant, il l'est moins pour DB2 puisque cette base est constituée de 60 sons, donc à des classes de moins de 10 sons comme nous l'avons mentionné dans la section 2.5.2. La métrique de distance prenant les 10 plus proches voisins à chaque son, va donc chercher des sons dans d'autres classes de son qui sont les plus proches, induisant des erreurs et faisant baisser la précision.

Nous avons utilisé sur les figures 3.3, 3.4, 3.5, 3.6, 3.7 le code de couleur suivant : le bleu correspond au moins bon résultat et rouge au meilleur. La signification des variations de couleurs entre ces deux extrêmes étant intuitive, nous pouvons avec cette représentation voir d'un coup d'oeil quels sont les meilleurs "chemin" correspondant aux meilleurs algorithmes.

3.2.1 Séries sur le temps

La première série de transformations ne réduit pas le temps. Nous écrasons les autres domaines (F,R,S) individuellement, puis toutes les combinaisons par groupes de 2 puis de 3 (F - R - S - F&R - F&S - R&S - F,R&S). Le temps n'étant pas normalisé dans les *feature collapse* (c.à.d que des sons de durées différentes sont représentés par des séries de longueurs différentes), nous ne pouvons pas calculer de distances euclidiennes ni de kernel gaussien. Nous pouvons cependant utiliser la distance DP avec ou sans PCA, ainsi que la distance KL précédée d'un GMM.

Exemple

Nous décrivons ici le premier trajet afin d'illustrer comment se lisent les figures. Le premier algorithme sur la figure 3.3 est le suivant : on ne réduit aucun domaine (*no collapse*), nous avons donc tous les domaines STRF, soit une représentation en dimension $250 \times 128 \times 10 \times 11 = 3{,}520{,}000$ (250 positions temporelles, 128 valeurs de fréquences, 10 de rate et 11 de scale). Cette série est ensuite soumise à une PCA sur les domaines F,R et S, donc dans un espace de dimension $128 \times 10 \times 11 = 14{,}080$. La PCA réduit la dimension de cet espace de façon à conserver 99,95 % de la variance, la réduisant à une série dans un espace de taille typiquement $\sim$100. Nous modélisons ce nuage de point en dimension $\sim$100 avec une gaussienne sur le domaine temporel (un GMM à une seule gaussienne).

Cas particuliers déjà connus

Le 5^{e} cas de collapse (F & R) qui résulte en une série temporelle de scales est conceptuellement similaire à une série de MFCCs (les scales sont le résultat de la transformée de Fourier du spectre, avec des unités en cycles/octave). Suivi d'un GMM, on retrouve l'algorithme BOF (Aucouturier et al., 2007). Suivi d'une distance DP, on retrouve la proposition de Aucouturier and Pachet (2007).

Le 6^{e} cas de collapse (F & S), i.e. une série temporelle de rates, est conceptuellement similaire au modulation spectrum de Peeters et al. (2002).

Le 7^{e} cas de collapse est un spectrogramme. Le 8^{e} cas, une simple forme d'onde.

Résultats

Tous les résultats sont affichés sur la figure 3.3. Nous avons au total 28 trajets differents dans cette série, chacun étant un algorithme original. Les meilleures précisions, respectivement **88,7 %** et 27,6 % pour DB1 et DB2, sont atteintes en écrasant les rates et les scales (collapse R & S), puis en appliquant un GMM sur le temps, précédé ou non d'une PCA sur le domaine des fréquences. La PCA sur F n'améliore pas la précision, mais ne la baisse pas non plus.

De manière générale, il vaut mieux prendre la moyenne sur R et/ou S que sur F. Il semble donc important de conserver les fréquences. D'autre part, après une PCA, la DP marche beaucoup moins bien que le GMM. Par contre, on obtient de bons résultats avec la DP et/ou un GMM suivi de la distance KL directement.

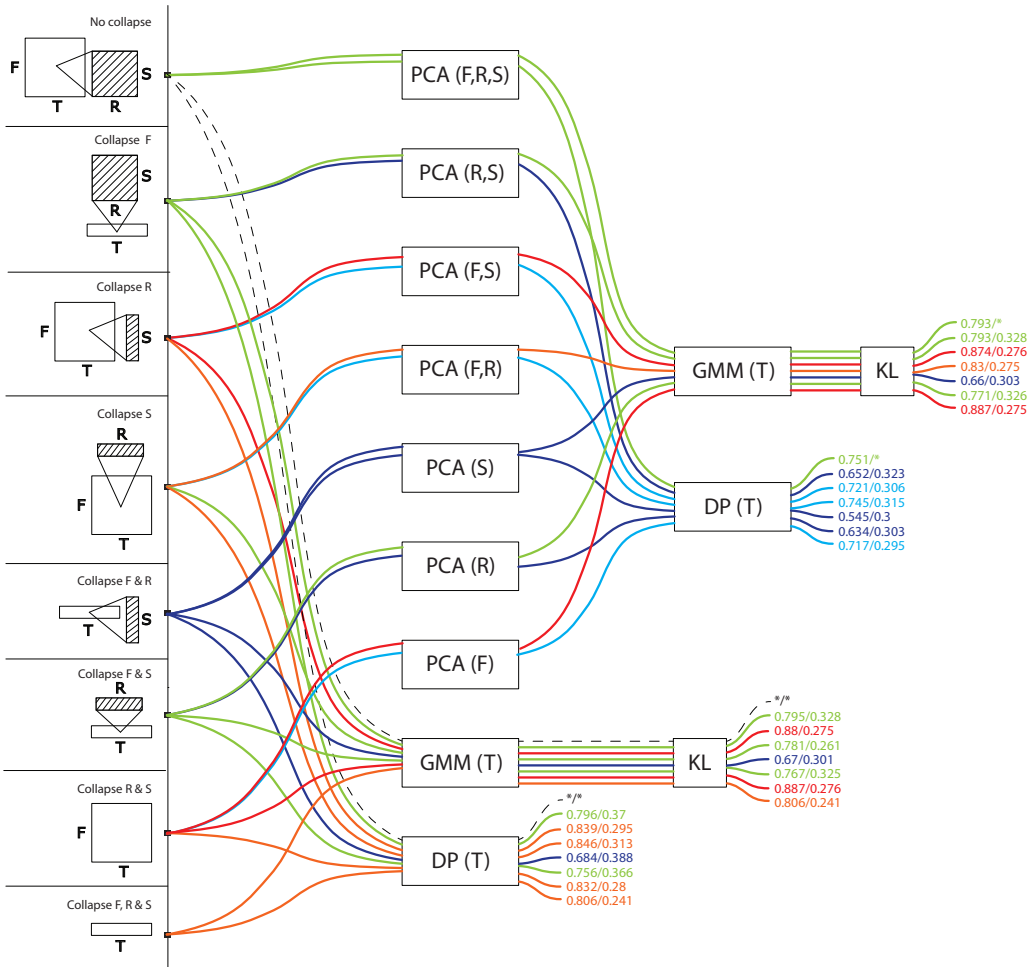


FIGURE 3.3 – Séries sur le temps.

3.2.2 Séries sur la fréquence

Exemple

On réduit la dimension de la matrice de données T×F×R×S en faisant la moyenne selon le domaine T, ce qui donne une matrice de données de dimension F×R×S=14,080. Cette matrice peut être considérée comme une série fréquentielle prenant ses valeurs dans l'espace R×S, c'est à dire une série de 128 trames de taille 10×11. Nous appliquons ensuite une PCA sur les domaines R & S. La série qui

en résulte prend ses valeurs dans l'espace des composantes principales de l'espace R×S. Ce nouvel espace, contenant 99,95% de la variance des valeurs dans R×S, est de dimension inférieure. Cela renvoie une série fréquentielle dont les rates et les scales ont été réduits dans un espace qui représente 99,95% de leur variance. Nous modélisons cette série avec une gaussienne (un GMM à une seule gaussienne) puis mesurons la distance entre chaque son modélisé par un GM à l'aide de la distance KL.

Cas particuliers déjà connus

Le 4e cas de collapse correspond à un calcul de modulation temporelle dans chaque bande de fréquence. C'est l'équivalent des fluctuation patterns de Pampalk (2006).

Résultats

Il y a 14 algorithmes différents dans cette série. Le meilleur apparait lorsqu'on moyenne le temps (collapse T), puis appliquons la distance DP directement sur le domaine des fréquences en gardant tous les rates et scales. Nous obtenons ainsi **90,9 %** de précision sur DB1 et 34,1 % sur DB2. C'est dans cette catégorie qu'on obtient le meilleur résultat des STRFs, que l'on baptise le *Cepstral Dynamic Frequency Warping*. De façon globale, si on doit conserver un seul domaine parmi R ou S, il vaut mieux conserver S que R. Conserver R et S est un peu mieux que S seulement, mais il est nuisible de conserver R si on ne conserve pas S. S étant l'équivalent du cepstrum, on confirme donc que c'est une bonne intuition que le cepstrum soit utilisé tout la communauté MIR depuis 15 ans.

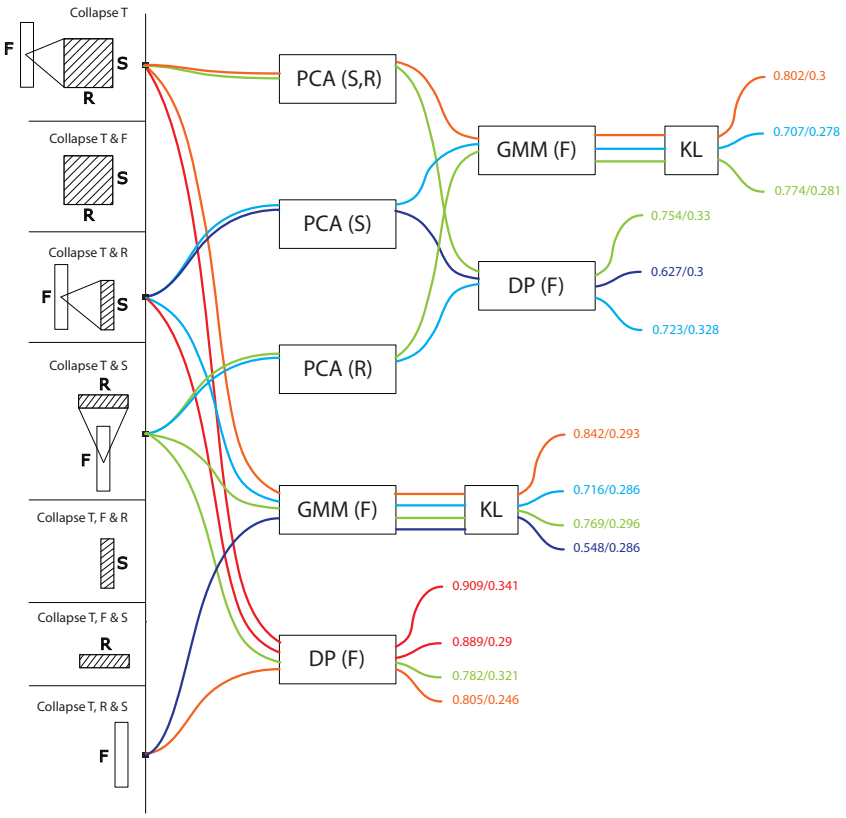


FIGURE 3.4 – Séries sur la fréquence.

3.2.3 Séries sur les rates

Exemple

Le 1er chemin étant très similaire à celui décrit dans l'exemple précédent, nous expliquons le second. On prend ici la moyenne des valeurs sur le domaine du temps (collapse T) et faisons une PCA sur les domaines F & S, donc sur un espace à 128×11=1,408 dimensions. Nous reformattons ensuite ces valeurs en une série de rates, et calculons la distance DP entre tous les vecteurs correspondant à tous les sons.

Cas particuliers déjà connus

Rien à notre connaissance.

Résultats

Les meilleures précisions sont atteintes lorsque nous prenons la moyenne sur le temps (collapse T) puis appliquons directement la distance DP sur les rates, donnant **83,3 %** avec DB1. Nous avons un maximum à 29,3 % sur DB2 en faisant la DP sur les rates mais en écrasant le temps et les fréquences préalablement.

Il apparait que les meilleurs résulats sont obtenus lorsque nous conservons F,R & S. Aussi, après une PCA, les GMM/KL sont bien plus performants que la DP, mais curieusement, sans PCA, la DP est plus performante que les GMM/KL.

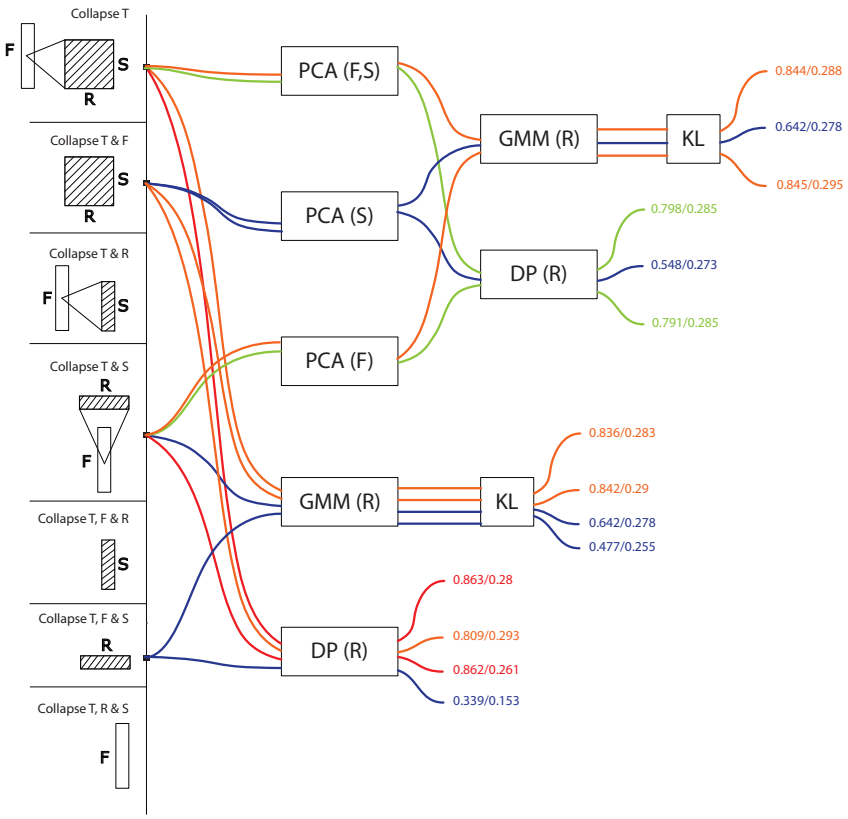


FIGURE 3.5 – Séries sur les rates.

3.2.4 Séries sur les scales

Exemple

Le premier algorithme réduit la dimension de la matrice de données T×F×R×S en faisant la moyenne sur le domaine T, comme nous l'avons fait auparavant. Nous pouvons maintenant considérer la matrice de dimensions F×R×S comme une série de scales, c'est à dire une série de 11 trames de dimension F×R (128×10=1280). Nous appliquons ensuite une PCA sur les domaines F & R, modélisons la

nouvelle série de scales avec un GM puis mesurons la distance entre chaque son modélisé à l'aide de la distance KL.

Cas particuliers déjà connus

Rien à notre connaissance.

Résultats

Parmis les 14 differents algorithmes possible, la meilleure précision sur DB1 est de **83,5 %** en prenant la moyenne sur le temps, une PCA sur les fréquences et les rates puis un GMM sur les scales. La meilleure précision sur DB2 (30,5 %), apparait en prenant la moyenne sur le temps, la fréquence et les rates puis en faisant de la DP sur les scales.

Dans cette série, les meilleurs algorithmes sont obtenus lorsque nous conservons la fréquence (comme dans la série de T). Aussi, appliquer des GMMs sur S donne de meilleures précisions que si l'on mesure la DP sur S.

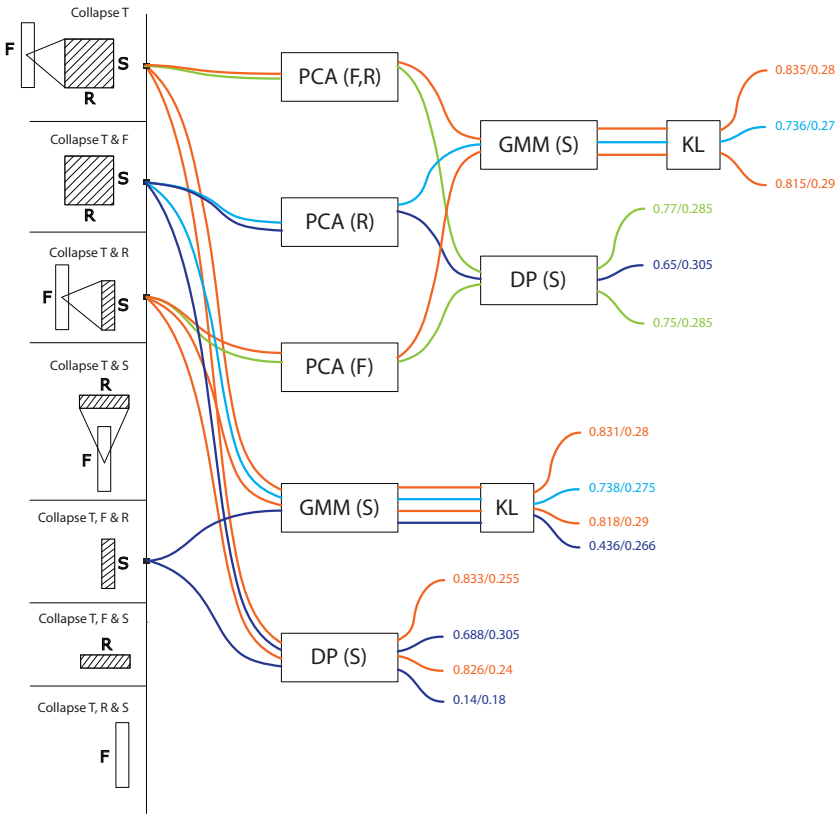


FIGURE 3.6 – Séries sur les scales.

3.2.5 Vecteur

Le dernier cas de figure est lorsque nous transformons les STRFs en vecteurs. Dans ce cas, il nous faut toujours prendre la moyenne sur le temps, afin que tous les sons soient comparables dans le meme espace vectoriel, quelque soit leur durée. Il y a 46 trajets que l'on peut classer dans trois cas de figure :

– Moyenne sur trois domaines :
  Nous prenons la moyenne sur trois domaines, dont le temps. Cela correspond à collapse T,F, & R, collapse T,F, & S et collapse T,R, & S. Ainsi, nous pouvons directement mesurer la distance euclidienne ou la

de Kernel Gaussien qui fonctionne comme la même manière. Nous pouvons aussi appliquer la PCA sur le domaine restant (S dans collapse T,F & R, R dans collapse T,F & S et F dans collapse T,R & S).

- Moyenne sur deux domaines :
  Nous prenons la moyenne sur deux domaines, dont le temps (collapse T & F, collapse T & F et collapse T & S) puis appliquons la PCA sur les domaines restant, respectivement sur R & S après collapse T & F, F & S après collapse T & R et F & R après collapse T & S. La PCA est ensuivie d'une distance euclidienne ou d'une distance de kernel gaussien. Ces représentations étant des matrices de points, il est aussi possible de mesurer la distance euclidienne et de kernel directement.
- Moyenne sur un domaine :
  Le dernier cas de figure est lorsque nous ne prenons la moyenne que sur le temps (collapse T). C'est le cas le plus polyvalent puisque nous pouvons appliquer une PCA sur les trois domaines mais aussi alternativement deux des trois domaines restants (PCA sur R & S ,F & R et F & R). Enfin, nous mesurons les distances euclidiennes et de kernel directement après le collapse ou après la PCA.

Cas particuliers déjà connus

Le premier cas de collapse (collapse T), suivi d'une réduction de dimensions par PCA puis d'une distance de kernel correspond à l'approche de Patil et al. (2012).

Le 4e cas de collapse correspond à un calcul de modulation temporelle dans chaque bande de fréquence. C'est l'équivalent des fluctuation patterns de Pampalk (2006).

Résultats

Le meilleur résulat sur DB1 (**88,5 %**) est obtenu après avoir pris la moyenne sur T, appliqué une PCA sur R & S puis en mesurant avec la distance de kernel. Sur DB2, le meilleur algorithme est de prendre la moyenne sur T, une PCA sur F & S puis mesurer la distance euclidienne, donnant 30,3% de précision.

Les résultats montrent que la PCA augmente la précision dans la majeure partie des cas et que la distance kernel est mieux appropriée que la distance euclidienne. De plus, il est nécessaire de garder R et F, les précisions chutant beaucoup dans le cas contraire.

3.2.6 MFCC sur le temps et sur les vecteurs

Dans le but de comparer nos algorithmes avec ceux de l'état de l'art, nous avons calculé les MFCCs sur DB1 et DB2. Dans le premier cas, nous calculons les MFCCs pour chaque trame temporelle, cela nous donne des résultats médiocres. Dans le second cas, nous prenons la moyenne de toutes les trames temporelles puis calculons les MFCCs. De cette manière, tous les algorithmes sont bons avec une précision majorant à **92,3 %** dans le cas d'une PCA suivie d'une distance de kernel.

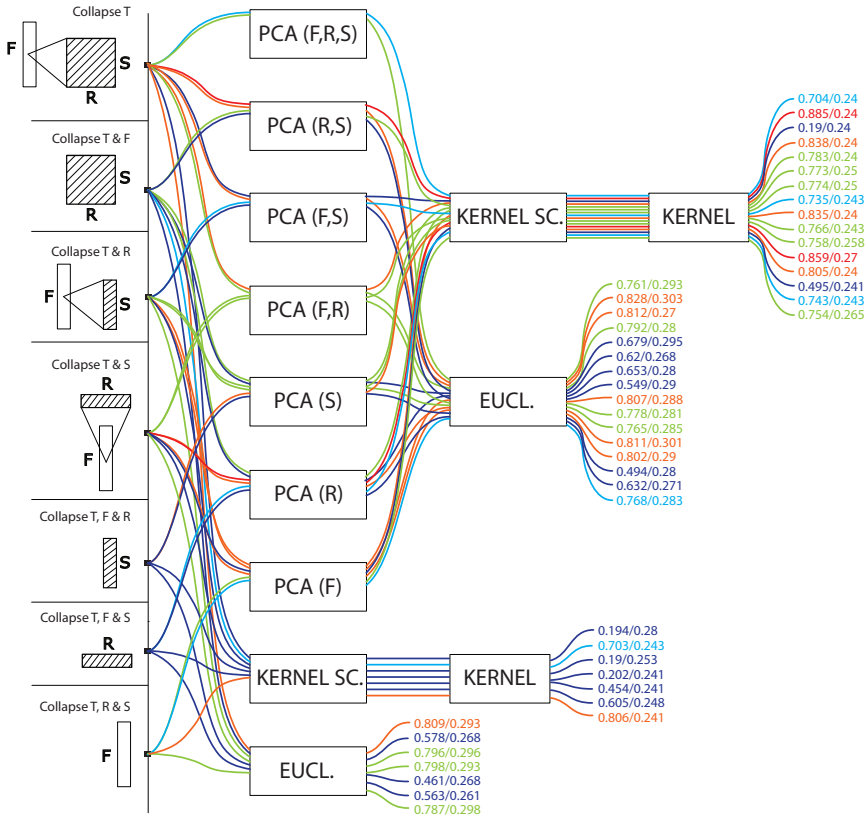


FIGURE 3.7 – Approche “vecteur”.

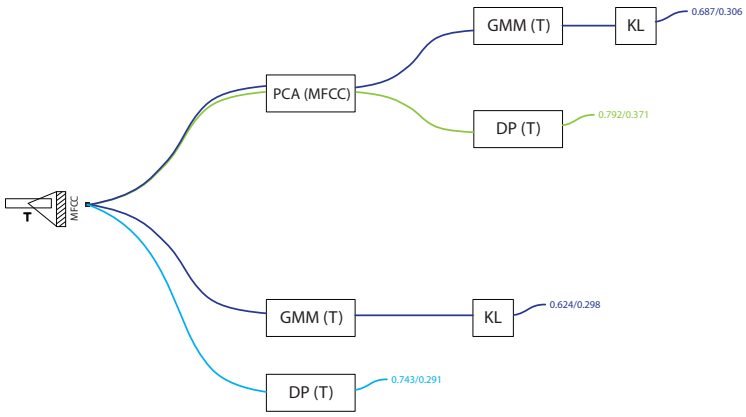


FIGURE 3.8 – Approche série temporelle" basée sur des MFCCs.

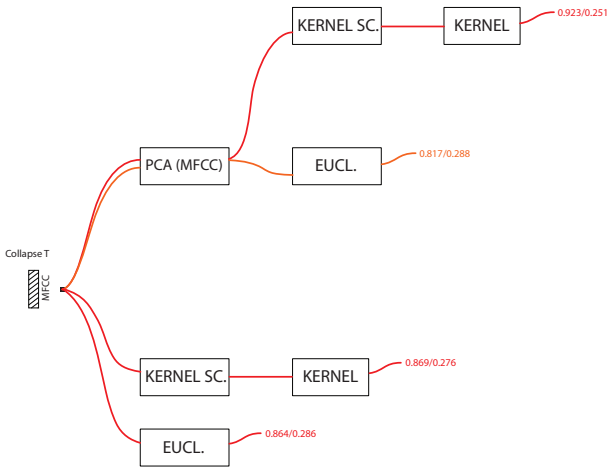


FIGURE 3.9 – Approche “vecteur” basée sur des MFCCs.

Chapitre 4

Conclusion

L'experience principale décrite dans ce mémoire consiste au test systematique de plus de 88 combinaisons de transformations et de distances basées sur le modèle STRF. Ce travail a nécessité l'implémentation de plusieurs techniques fondamentales du traitement du signal et du machine learning (GMM, gaussian kernel, etc.), et a nécessité plus de trois semaines cumulées de temps de calcul distribué sur les ordinateurs de l'IRCAM.

L'unique précédent de ce travail consiste en la proposition de Patil et al. (2012), qui, selon notre nomenclature, consiste en un collapse temporel, une PCA commune sur F,R et S, suivi de l'apprentissage d'un kernel gaussien (collapse T $\rightarrow$ PCA $\rightarrow$ kernel). Cette stratégie a été la seule testée par les auteurs [1], et postulée comme la plus précise. Notre travail montre que, au moins sur le problème defini par notre base DB1, cette approche est loin d'etre optimale (70.4%). Dans la même stratégie "vecteur" (i.e. de ne pas constituer une série de STRF), le simple fait, non testé par Patil et al. (2012) à notre connaissance, de faire une PCA sur R et S seulement, en conservant l'étendu complète des fréquences, amene un gain de précision considérable de 18%. D'encore meilleurs résultats sont obtenus avec des approches moins intuitives, comme celles de considérer les STRF comme une série fréquentielle. La meilleure précision obtenue sur toute l'expérience est de 90.9%, prés de 20% d'amélioration par rapport à l'algorithme de Patil et al. (2012).

L'approche systématique de test de toutes les combinaisons nous a permis de découvir des algorithmes très performants dont personne n'avait eu l'intuition, par exemple l'alignement de séries fréquentielles de scales par dynamic

1. Mounya ElHilali, communication personnelle, Juillet 2013

programming, qu'on pourrait baptiser "cepstral dynamic frequency warping" (cepstral, car agissant sur les scales, et DFW par analogie au DTW qui agit sur une série temporelle). Sa précision est remarquable : p=90,9%.

Il peut apparaitre un peu frustrant que, malgré toute la sophistication de l'approche STRF et le nombre de combinaisons testées dans ce mémoire (88), on ne fasse pas mieux qu'avec des MFCCs : la stratégie consistant à prendre la moyenne des MFCCs dans le temps, et de comparer ce vecteur moyen par simple distance euclidienne, atteint une precision de 86% sur DB1 ; suivi d'une PCA puis d'une distance de kernel gaussien, on obtient les meilleurs résultats de ce mémoire : 92,3% sur DB1. Cependant, si notre algorithme "cepstral dynamic frequency warping" donne seulement 1% de moins, il a plusieurs avantages sur l'utilisation des MFCCs : d'une part, l'algorithme de cepstral DFW ne nécessite pas d'apprentissage, contrairement au kernel gaussien, qui est optimisé par rapport à une groundtruth, et pour lequel on peut craindre en pratique des problèmes de sur-apprentissage (non testés ici). D'autre part, la possibilité que nous venons de révéler de faire aussi bien que les MFCCs avec un algorithme neuro-inspiré ouvre de considérables perspectives d'interaction entre MIR et la psychologie et les neurosciences cognitives. En effet, la non recevabilité des MFCCs en tant que modèle cognitif est un des obstacles relevés par Aucouturier and Bigand (2013) au dialogue interdisciplinaire entre MIR et les sciences cognitives. Nous pouvons donc désormais considérer cet obstacle comme levé.

Même si nos résultats ne surpassent pas pour l'instant l'approche MFCC, nous pouvons tout de même imaginer que cette nouvelle approche est peut-être une manière de "briser" le *glass ceiling* signalé par Aucouturier and Pachet (2004) qui montre que la précision en similarité spectrale est plafonnée tant que nous travaillons avec des variations de MFCC. Nos résultats confirment l'interêt de faire un modèle computationnel neuro-inspiré. De plus, il semblerait que les valeurs de rates et de scales peuvent encore être améliorées et que le modèle STRF peut encore être perfectionné.

Plusieurs pistes existent pour continuer ce travail. D'une part, il faudra tester d'autres bases de données. Comme l'on a pu le voir, les résultats sur DB2 ne sont pas satisfaisants, dû à la grande diversité des sons contenus et sans doute aussi à cause de la métrique non adaptée à la base (N-précision à 10). Il serait interessant d'utiliser le modèle STRF sur d'autres problèmes, comme la musique et les *soundscapes*. D'autre part, nous pourrions aussi comparer plus précisément les résultats avec des données de similarité obtenues directement au niveau du cortex auditif plutot que par des jugements

psychoacoustiques. Cela serait rendu possible par électro-encéphalographie (EEG) avec le paradigme de négativité de discordance (*mismatch negativity* ou MMN), qui permet de mesurer une réponse corticale (un "potentiel évoqué") à un son rare au milieu d'une série de sons standards, réponse dont l'amplitude s'avère proportionnelle à la différence percue entre deux stimuli audio (Toiviainen et al., 1998). Nous avons commencé au cours du stage une collaboration sur ce thème avec la neuroscientifique Anjali Bhatara du LPP de Paris 5, pour préparer une expérience qui utilisera les présents résultats, et qui est plannifiée au cours de l'automne 2013.

Bibliographie

Aucouturier, J. and Bigand, E. (2013). Seven problems that keep mir from attracting the interest of cognition and neuroscience. *Journal of Intelligent Information Systems*.

Aucouturier, J., Defreville, B., and Pachet, F. (2007). The bag-of-frame approach to audio pattern recognition : A sufficient model for urban soundscapes but not for polyphonic music. *Journal of the Acoustical Society of America*, 122(2) :881–91.

Aucouturier, J. and Pachet, F. (2004). Improving timbre similarity : How high's the sky ? *Journal of Negative Results in Speech and Audio Sciences*, 1(1).

Aucouturier, J. and Pachet, F. (2007). The influence of polyphony on the dynamic modelling of musical timbre. *Pattern Recognition Letters*, 28 :654–661.

Aucouturier, J.-J. and Bigand, E. (2012). Mel cepstrum & ann ova : The difficult dialog between mir and music cognition. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*.

Bellman, R. E. (2003). *Dynamic Programming*. Dover Publications, Incorporated.

Berenzweig, A., Ellis, D. P. W., and Lawrence, S. (2003). Anchor space for classi cation and similarity measurement of music. In *IEEE International Conference on Multimedia and Expo*.

Berenzweig, A., Logan, B., Ellis, D., and Whitmann, B. (2004). A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2) :63–76.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blakemore, C. and Campbell, F. W. (1969). On the existence of neurons in the human visual system selectively sensitive to the orientation and size of retinal images. *Journal of Physiology*, 203 :237–260.

Blum and Wold (1996). Content-based classification, search, and retrieval of audio. *MultiMedia IEEE*, 3(3).

Carrasco, A. and Lomber, S. (2009). Evidence for hierarchical processing in cat auditory cortex : nonreciprocal influence of primary auditory cortex on the posterior auditory field. *Journal of Physiology*, 29 :45.

Celma, O. and Lamere, P. (2011). If you like radiohead, you might like this article. *AI Magazine*, 32(3) :57.

Deutsch, D. (1999). *The Psychology of Music*. New York : Academic Press, 2 edition.

Goldstone, R. L., Medin, D. L., and Gentner (1991). Relational similarity and the nonindependence of features in similarity jugements. *Cognitive Psychology*, 23 :222–264.

Gómez, E. (2006). *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain.

Hartline, H. K. (1939). Excitation and inhibition of the "off" response in the vertebrate optic nerve fibers. American Journal of Physiology. *Journal of Neurophysiology*, 126 :527.

Houix, O., Lemaitre, G., and Urdapilleta, I. (2011). A lexical analysis of environmental sound categories. *J. Experimental Psychology : Applied*.

J.-J. and Pachet, F. (2002). Music similarity measures : What's the use ? In *International Symposium on Music Information Retrieval (ISMIR)*, pages 157–163.

Jun, S., jun Han, B., and Hwang, E. (2009). A similar music retrieval scheme based on musical mood variation. In *ACIIDS*, pages 167–172.

Kuffler, S. W. (1953). Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology*, 16 :37–68.

Lemstrom, K. and Laitinen, M. (2011). Transposition and time-warp invariant geometric music retrieval algorithms. In *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo*, ICME '11, pages 1–6.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *Int. Symposium on Music Information Retrieval*.

Logan, B. and Salomon, A. (2001). A music-similarity function based on signal analysis. *International Conference on Multimedia and Expo*.

Muller, M. (2003). *Information Retrieval for Music and Motion*. Springer.

Pampalk, E. (2006). Audio-based music similarity and retrieval :combining a spectral similarity model with information extracted from fluctuation patterns. In *Proceedings of the ISMIR International Conference on Music Information Retrieval (ISMIR'06), Vienna, Austria*.

Pasley, B. N., David, S. V., Mesgarani, N., Flinker, A., Shamma, S. A., Crone, N. E., Knight, R. T., and Chang, E. F. (2012). Reconstructing speech from human auditory cortex. *PLoS Biol*, 10(1) :e1001251.

Patil, K., Pressnitzer, D., Shamma, S., and Elhilali, M. (2012). Music in our ears : The biological bases of musical timbre perception. *PLOS Computational Biology*, 8(11).

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM.

Peeters, G., A., L., and Rodet, X. (2002). Toward automatic music audio summary generation from signal analysis. In *ISMIR*.

Rabiner, L. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, united states ed edition.

Serre, T., Wolf, L., S., B., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3).

Shamma, S., Versnel, H., and Kowalski, N. (1995). Ripple analysis in ferret primary auditory cortex. i. response characteristics of single units to sinusoidally rippled spectra. 1 :233–254.

Shao, B., Li, T., and Ogihara, M. (2008a). Quantify music artist similarity based on style and mood. In *WIDM'08*, pages 119–124.

Shao, B., Li, T., and Ogihara, M. (2008b). Query by humming of midi and audio using locality sensitive hashing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08)*.

Shlens, J. (2005). A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*.

Sjölander, K. (2001). Automatic alignment of phonetic segments.

Toiviainen, P., Tervaniemi, M., Louhivuori, J., Saher, M., Huotilainen, M., and Nâätänen, R. (1998). Timbre similarity : convergence of neural, behavioral and computational approaches. *Music Perception*, 16 :223–241.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84 :327–351.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10 :293–302.

Tzanetakis, G. and Essl, G. (2001). Automatic musical genre classification of audio signals. In *IEEE Transactions on Speech and Audio Processing*, pages 293–302.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1999). *The HTK Book Version 2.2*. Entropic Cambridge Research Laboratory.