The influence of Polyphony on the Dynamical Modelling of Musical Timbre.

Jean-Julien Aucouturier [a,*], Francois Pachet [b]

[a] *Ikegami Laboratory, Department of General Systems Studies, Graduate School of Arts and Sciences, The University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan*

[b] *SONY Computer Science Laboratory, 6 rue Amyot. 75005 Paris, France*

Abstract

This letter addresses the problem of pattern recognition of polyphonic musical timbre. Frame-level dynamics of audio features are particularly difficult to model, although they have been identified as crucial perceptive dimensions of timbre perception. Recent studies seem to indicate that traditional means to model data dynamics, such as delta-coefficients, texture windows or Markov modelling, do not provide any improvement over the best static models for real-world, complex polyphonic textures of several seconds' length. This contradicts experimental data on the perception of individual instrument notes. This letter describes an experiment to identify the cause of this contradiction. We propose that the difficulty of modelling the dynamics of full songs results either from the complex structure of the temporal succession of notes, or from the vertical polyphonic nature of individual notes. We discriminate between both hypothesis by comparing the performance of static and dynamical algorithms on several specially designed datasets, namely monophonic individual notes, polyphonic individual notes, and polyphonic multiple-note textures. We conclude that the main cause of the difficulty of modelling dynamics of real-world polyphonic musical textures is the polyphonic nature of the data.

Key words: music, timbre, dynamics, polyphony, similarity, segmentation
PACS: 43.60.Bf, 43.60.Cg, 43.60.Lq

* Corresponding author.
Email addresses: aucouturier@gmail.com (Jean-Julien Aucouturier), pachet@csl.sony.fr (Francois Pachet).

1 Introduction

1.1 Sound textures

Timbre is defined by the American Standards Association (ASA, 1960) as "that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar". It notably describes the quality of a musical note which distinguishes different musical instruments.

The exploding field of Electronic Music Distribution (EMD) is in need of powerful content-based management systems to help the end-users navigate huge music title catalogues, much as they need search engines to find web pages in the Internet. Not only do users want to find quickly music titles they already know, but they alsoand perhaps more importantlyneed systems that help them find titles they do not know yet but will probably like. The global "sound" or timbre of a piece of music seems an important component of such Music Information Retrieval (MIR) systems. Music listeners are sensitive to timbres that specific to e.g. music periods (e.g. the sound of Chick Corea playing on an electric piano), musical configurations (e.g. the sound of a symphonic orchestra), or musical genres (e.g. heavily saturated electric guitar).

Most of the studies on musical instrument discrimination (Herrera-Boyer et al., 2003) have focused on sound samples corresponding to clean recordings of a unique note, played by a single instrument. However, this approach seems little suited to the modelisation of real-world, complex polyphonic textures of several seconds' length, of which the music recommendation industry is in demand. Psychoacoustic investigations on monophonic timbre discrimination (Grey, 1977; Iverson and Krumhansl, 1993; McAdams et al., 1995) show that precise time attributes, such as the attack time of a note, are crucial dimensions of a meaningful perceptual timbre space. However, such descriptors are practically impossible to compute from complex polyphonic textures, since the different sound sources are typically not synchronized. A sound segment corresponding to a given note of a given instrument is likely to be superimposed with other notes with various time offsets. For instance, the attack of a piano note extracted from the recording of a jazz trio may superimpose with the decay of a double-bass note, and its steady-state may be similarly corrupted by several drum onsets.

1.2 Traditional modelling

The lack of psychophysical models for the timbre perception of polyphonic textures has led researchers to take a pragmatic approach to build the much-

needed automatic systems able to model and compare sound textures. The signal is cut into short overlapping frames (typically 50ms with a 50% overlap), and for each frame, a feature vector is computed. Features usually consist of a generic, all-purpose spectral representation such as Mel Frequency cepstrum Coefficients (MFCC), a particular encoding of the spectral envelope widely used in the speech recognition community (Rabiner and Juang, 1993). The features are then fed to a statistical model, such as a Gaussian Mixture Model (GMM), which models their global distributions over the total length of the extract. Global distributions can then be used to compute decision boundaries between classes (to build e.g. a genre classification system such as Tzanetakis and Cook (2002)) or directly compared to one another to yield a measure of timbre similarity (Aucouturier and Pachet, 2004). Note that an alternative approach to the problem of polyphony is the one taken by Essid et al. (2005), which applies successive monophonic recognition algorithms to eliminate the components of a texture.

1.3 The paradox of dynamics

The type of algorithm described above has led to some success but a previous study of the authors on timbre similarity (Aucouturier and Pachet, 2004) shows that it seems to be bounded to moderate performance. Most notably, as we know report in this section, classical pattern recognition extensions that take the data dynamics into account have surprisingly failed to improve the precision of the models.

The prototypical algorithm described above (GMMs of MFCCs) does not take any account of time scale greater than the frame size. Frames are modelled without any account of their ordering in time [1]. It is a common strategy to try and modify this prototypical algorithm so as to take the dynamics of the data into account. Modifications may occur at the feature level, by e.g.

- Tapped delay line: consecutive feature vectors can be stacked into n-times larger feature vectors, before sending them to the statistical model, thus constructing a flat spatial embedding of the temporal sequence (see e.g. Scaringella and Zoia (2005)).
- Derivation: Furui (1986) showed that speech recognition performance can be greatly improved by adding time derivatives to the basic static features. Delta Coefficients are computed using the following formula:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (f_{t+\theta} - f_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2} \quad (1)$$

[1] Note that we are discussing here the ordering in time of *frames* in a sequence, not the reordering of the *samples* within each frame.

where d_t is a delta coefficient at time t, computed using a time window Θ. The same formula can be applied to the delta coefficients to obtain the acceleration coefficients. The resulting modified feature set contains, for each frame, the static feature values and their local delta values.

- Texture windows: local static features (typically extracted every 50 ms) can be averaged over larger-scale windows (typically several seconds), in an attempt to capture the long-term nature of sound textures, while still assuring that the features be computed on small stationary windows. Several statistics can be used on such so-called *texture windows*, e.g. mean, standard deviation, skewness, range, etc. (see e.g. Tzanetakis and Cook (2002))
- Modulation spectrum: Another strategy to characterize the dynamics of the static features is to compute features on the signal constituted by the static feature sequence (which is considered to be sampled at the original frame-rate). For instance, a high-resolution STFT can be taken on large frames (of several seconds' duration), and the low-frequency variations of the features (e.g. $[1 - 50Hz]$) are taken as features instead of the original ones (see e.g. Peeters et al. (2002)).

Modifications can also occur at the modelling level, to account for the dynamics of static features. Such dynamic models include recurrent neural networks (R-NN), which are typically 2 layer networks with feedback from the first layer output to the first layer input, and hidden Markov models, which can be defined as a set of GMMs (also called states) linked to one another by a matrix of transition probabilities.

Table 1 shows the type of performance achieved by some of such static and dynamic approaches as measured in Aucouturier and Pachet (2004). It reports on R-precision scores achieved by the best found static algorithm (GMMs with 50 gaussian components), as well as a number of extensions to this algorithm aiming at better modelling the dynamics of the data (see Appendix A for more details about the corresponding experimental methodology).

It appears that no precision is gained by computing first order derivatives of the features (so called "delta coefficients"), computing mean and variance of the features on intermediate-size "texture windows" or using dynamical models such as hidden Markov models (HMM, see Rabiner (1989)). In Aucouturier et al. (2006), we described a technique inspired by image texture similarity which used co-occurrence matrices on a quantized space of MFCCs to model the second-order statistics of timbre textures. Table 1 shows that, identically, this is at best equivalent to simpler and static histograms. The poor improvement, if any, achieved by state-of-art dynamical models has recently been confirmed by Scaringella and Zoia (2005) on the related task of musical genre classification.

This is a surprising observation. Static models consider all frame-permutations

Table 1
R-precision scores achieved for the best algorithm as well as a number of extensions aiming at better modelling the dynamics of the data. Results reproduced from Aucouturier and Pachet (2004); Aucouturier et al. (2006). "Random" corresponds to the chance-based strategy of drawing a random set of nearest neighbor for each seed song. See Appendix A for details.

Algorithm	*R*-Precision
Random	0.12
Best (20 MFCC + 50-state GMM)	0.65
Delta $\Theta = 10$	0.60
Acceleration $\Theta = 10$	0.610
Delta $\Theta = 2$	0.62
Delta $\Theta = 5$	0.62
Acceleration $\Theta = 5$	0.62
Acceleration $\Theta = 1$	0.63
Delta $\Theta = 1$	0.63
Texture Window $w_t = 10$	0.64
Texture Window $w_t = 20$	0.65
HMM (5 states)	0.62
HMM (10 states)	0.63
HMM (20 states)	0.62
HMM (30 states)	0.44
Co-occurrence matrix	0.44
Histogram	0.50

of the same audio signal as identical, while this has a critical influence on their perception. Moreover, as mentionned earlier, psychophysical experiments such as (McAdams et al., 1995) have established the importance of dynamics, notably the attack time and fluctuations of the spectral envelope, in the perception of individual instrument notes.

1.4 Hypothesis testing

There are 3 main hypothetic causes that explain the difficulty of modelling dynamics in the case of polyphonic timbre textures:

H1 Either the dynamics of timbre frames are impossible to capture at the time-scale of an individual note, e.g. because there is too much timing variation from note to note. This is improbable, as success in doing so has been reported notably in instrument note recognition (Dubnov and Fine, 1999; Eronen, 2003).

H2 Either it is the dynamics of *polyphonic* timbre frames that is difficult to model. We have already noted the problems of spectral masking and asynchronicity of several concurrent sound sources, and how they defeat naïve analysis generalized from the monophonic case.

H3 Or it is the dynamics of *successive notes* at the time-scale of a phrase or a full song that are difficult to model. It is unclear e.g. whether a HMM attempts to capture fine-grained dynamics such as the succession of transient and steady-state inside individual notes or rather longer-term structure like the succession of different instrument timbres.

In this letter, we propose to discriminate between these 3 hypotheses by testing the performance of dynamical algorithms on two databases of individual sound samples:

- one composed of clean, monophonic individual instrument notes (DB1)
- the other obtained from a polyphonic, real-world recording (DB2)

The comparison of the performance of dynamical modelling against static modelling in both contexts has the potential to disprove some of the above hypotheses. Evidence that dynamical modelling performs constantly worse than static modelling on both databases would support **H1** (and be at odds with previous findings from the literature). **H2** will be supported should dynamical modelling perform better than static on the monophonic dataset (DB1), but not on the polyphonic one (DB2): this would mean that polyphony ruins attempts at modelling dynamics even within the constrained time-scale of individual notes. Finally, evidence that dynamical models overperforms static models for both databases, but not for textures made of successive notes (which is our starting-point observation), would indicate that the critical factor is the existence of the longer-term structure of e.g. phrase rhythm and instrument changes (**H3**).

2 Methods

2.1 Databases

Table 2 and 3 describe the contents of both databases. DB1 was obtained as an extract of the IRCAM "Studio On Line" database, made available in

the context of the Cuidado European Project (Vinet et al., 2002). It contains 710 short sound samples, categorized in 16 classes corresponding to the instrument used for their recording. DB2 consists of sound samples obtained from the automatic analysis of the song *The Beatles - Let it be*, using a segmentation algorithm described below (Section 2.2). The process yielded 595 samples, which were manually clustered and categorized into 16 categories, corresponding to the different mixtures of sound sources occurring in the song (a few samples were discarded because they were either too short, or difficult to categorize).

Table 2
Composition of DB1.

Class	Size
Accordion	37
Alto violin	51
Bass	50
Bassoon	39
Cello	48
Clarinet	44
Flute	38
Guitar	66
Harp	40
Horn	78
Oboe	36
Sax	32
Trombone	38
Trumpet	32
Tuba	35
Violin	46
Total	710

Table 3
Composition of DB2.

Class	Size
Drums	59
Electric Guitar	74
Electric Piano	13
Organ	27
Organ & Drums	16
Piano	95
Piano & Tchak	26
Tutti 1	70
Voice & Bass & Drums	35
Voice & Organ & Drums	11
Voice & Piano	58
Voice & Piano & Choir	6
Voice & Piano & Organ	5
Voice & Piano & Tchak	19
Voice & Tutti1	53
Voice & Tutti1 & Elec. Guitar	21
Total	588

We remark that both databases have the same number of classes, and roughly the same size, which makes their comparison quite reliable. However, DB2 being obtained with automatic segmentation, samples from the same category may have quite different durations. This may be detrimental to dynamical algorithms, which may match samples of the same duration across different categories. To control this effect, we create a third database, DB3, which contains the same samples as DB2, but sub-categorizes the timbre categories

according to the samples' duration: samples that were categorized as e.g. *Piano* in DB2 are categorized in DB3 as one of {*Piano_100*, *Piano_200*, ...}, where {100,200,...} denotes the duration (in ms) of the sample (averaged to the nearest multiple of 100ms). The typical sample duration being between 50ms and 1 sec., this creates up to 10 time-indexed sub-classes per original instrument class in DB2.

2.2 Segmentation

We give here an overview of the segmentation algorithm used to populate DB2. The algorithm is described in more details in Aucouturier et al. (2004). The signal is cut into frames (2048 points at 44100Hz), on which we compute the short-term spectrum. The spectrum itself is processed by a Mel filter bank of 20 bands. Each band's energy is weighted according to the frequency response of the human ear (Schroeder et al., 1979). Finally, the energy is summed across all bands. Change detection is done by smoothing the energy profile by a zero-phase filtering by a Hanning window of size S_w, and looking for all the local maxima of the smooth version. The segment boundaries are the deepest valleys in the raw energy profile between 2 adjacent peaks in the smooth profile. The size of the convolution window S_w can be adapted to the local shape of the energy profile, by an analysis based on long-term Fourier transform. This algorithm was used in Aucouturier et al. (2004) to build an automatic synthesizer from an existing music recording, and a variant thereof in Jehan (2004) to analyze the metrical structure of musical pieces.

2.3 Algorithms

17 algorithmic variants were implemented for each database and their results compared. Table 4 describes the parameters of each variant.

The use of dynamic information is embodied by 3 algorithmic variants based on Dynamic Programming (DP, see e.g. Crochemore and Rytter (1994)). DP is typically used for aligning or computing the distance between 2 sequence of symbols, such as text, protein or DNA sequences. It was also used e.g. in Smith et al. (1998) for comparing sequences of musical notes. DP relies on a symbol-distance, which measures the distance between duplets of symbols in the alphabet (which may have infinite size), and an edit-operation cost, which penalizes alignment changes in the sequence (e.g., deletion, insertion, substitution). In our case, we compare sequences of MFCC frames, using the euclidean distance as symbol distance, and compare 3 values for the edit cost $\{10, 100, 1000\}$. The smaller the edit-cost, the more tolerant the measure is to modifications of the time arrangement of successive MFCC frames. This

Table 4
Description of the algorithms used to compare monophonic and polyphonic sample similarity

DynProg MFCC (edit 10)	Dynamic Programming Comparison of MFCC frames (using an edit cost of 10)
DynProg MFCC (edit 100)	Same as above with edit cost of 100
DynProg MFCC (edit 1000)	Same as above with edit cost of 1000
MFCC_MPEG-7_RMS 1G	Monte-Carlo KL comparison of Gaussian Mixture model (using 1 gaussian component) of feature vectors composed of MFCCs (dim 20), MPEG-7 Spectral descriptors (dim 7) and RMS value (dim 1)
MFCC_MPEG-7_RMS 2G	Same as above with 2 gaussian components
MFCC_MPEG-7_RMS 3G	Same as above with 3 gaussian components
MFCC_MPEG-7_RMS 4G	Same as above with 4 gaussian components
MPEG-7 1G	Monte-Carlo KL comparison of Gaussian Mixture model (using 1 gaussian component) of feature vectors composed of MPEG-7 Spectral descriptors (dim 7)
MPEG-7 2G	Same as above with 2 gaussian components
MFCC_MPEG-7 1G	Monte-Carlo KL comparison of Gaussian Mixture model (using 1 gaussian component) of feature vectors composed of MFCCs (dim 20) and MPEG-7 Spectral descriptors (dim 7)
MFCC_MPEG-7 2G	Same as above with 2 gaussian components
MFCC_RMS 1G	Monte-Carlo KL comparison of Gaussian Mixture model (using 1 gaussian component) of feature vectors composed of MFCCs (dim 20) and RMS value (dim 1)
MFCC_RMS 2G	Same as above with 2 gaussian components
MFCC 1G	Monte-Carlo KL comparison of Gaussian Mixture model (using 1 gaussian component) of feature vectors composed of MFCCs (dim 20)
MFCC 2G	Same as above with 2 gaussian components
Mean MFCC Euclidean	Euclidean comparison of the mean of feature vectors composed of MFCCs (dim 20)
Jehan Euclidean	Euclidean comparison of the concatenation of the mean of feature vectors composed of MFCCs (dim 20), and a set of global temporal shape descriptors (dim5)

makes it possible to align close MFCC frames at different positions within the samples, i.e. to match sound samples of the same timbre, but with very different duration. However, this also increases the number of false positives. DP can be viewed as a manual equivalent of decoding the sequence with an a priori trained HMM. Note however that HMM-based similarity (as used for full songs in Aucouturier and Pachet (2004)) was impossible to use in the context of short samples, because of the lack of training data: a typical sample has a duration of 200 ms, which amounts to 10 frames.

We compare these dynamical algorithms to a number of static algorithms, based on combinations of features such as MFCC, energy (root-mean square) or Spectral MPEG-7 descriptors (i.e. Spectral Centroid, Spread, Kurtosis, Skewness, Flatness, Rolloff and Flux). Features are compared using simple average comparison with euclidean distance, or Gaussian Mixture models (i.e. average *and* variance) with up to 4 gaussian components. Note that similarly to HMM-based processing, greater numbers of components (such as 50 as used for full songs) could not be tested because of the lack of training data in individual samples.

Finally, a hybrid algorithm inspired by Jehan (2004), compares a feature vector composed of the average of the MFCCs and a set of global descriptors describing the temporal shape of the samples: normalized loudness at onset and at offset, maximum loudness and relative location of the maximum loudness.

2.4 Evaluation Procedure

The algorithms are compared by computing their precision after 10 documents are retrieved, and their R-precision, i.e. their precision after all relevant document are retrieved. Each value measures the ratio of the number of relevant documents to the number of retrieved documents. The set of relevant documents for a given sound sample is the set of all samples of the same category than the seed (for instance, the precision of a query on a piano sample measures the number of piano samples in the set of its nearest neighbors). This is identical to the methodology used e.g. in Aucouturier and Pachet (2004), and for the results reported in Table 1.

3 Results

Table 5 shows the evaluation scores of the algorithms described above on both databases DB1 and DB2. One can see that dynamic algorithms perform up

Table 5
Comparison of similarity methods for monophonic and polyphonic samples. Best scores for each dataset appear in bold. "Random" corresponds to the chance-based strategy of drawing a random set of nearest neighbor for each seed song (See Appendix A for details).

Method	DB1		DB2		DB3	
	P_{10}	P_R	P_{10}	P_R	P_{10}	P_R
Random	0.26	0.14	0.25	0.18	0.07	0.06
DynProg MFCC (edit 10)	**0.76**	**0.46**	0.44	0.34	**0.24**	**0.22**
DynProg MFCC (edit 100)	**0.73**	**0.46**	0.37	0.27	**0.35**	**0.31**
DynProg MFCC (edit 1000)	**0.70**	**0.44**	0.31	0.17	**0.33**	**0.28**
MFCC_MPEG-7_RMS 4G	0.64	0.34	0.42	0.31	0.12	0.12
MFCC_MPEG-7_RMS 3G	0.63	0.34	0.45	0.32	0.12	0.12
MFCC_MPEG-7_RMS 2G	0.62	0.33	0.47	0.35	0.14	0.13
MFCC_MPEG-7_RMS 1G	0.62	0.35	**0.51**	**0.37**	0.15	0.14
MPEG-7 1G	0.61	0.38	0.36	0.29	0.11	0.11
MFCC_MPEG-7 1G	0.61	0.33	0.47	0.35	0.14	0.13
MPEG-7 2G	0.61	0.38	0.36	0.29	0.11	0.11
MFCC_MPEG-7 2G	0.59	0.31	0.43	0.33	0.12	0.12
Mean MFCC Euclidean	0.58	0.33	**0.50**	**0.39**	0.14	0.13
Jehan Euclidean	0.56	0.32	**0.49**	**0.38**	0.21	0.19
MFCC_RMS 2G	0.56	0.28	0.48	0.35	0.14	0.13
MFCC_RMS 1G	0.55	0.27	**0.50**	**0.37**	0.15	0.14
MFCC 2G	0.51	0.26	0.46	0.32	0.14	0.13
MFCC 1G	0.50	0.26	0.47	0.33	0.15	0.13

to *10% better* (absolute) than static algorithms on the monophonic database. This establishes that the dynamic evolution of instantaneous features are an important factor for timbre similarity. This confirms the findings of both psychophysical experiments on the perception of instrument timbre, and a number of automatic instrument classification systems. The best static performances on DB1 are obtained with fairly involved variants, which typically rely on concatenation of several features, and large Gaussian Mixture Models.

However, dynamic algorithms perform nearly *10% worse* than their static equivalent on the polyphonic database DB2. It also appears that the best

polyphonic performance is achieved with the most simple static algorithms, such as euclidean comparison of the simple average of MFCCs. Notably, while increasing the number of gaussian components for the `MFCC_MPEG-7_RMS` family of algorithms constantly increases the precision on the monophonic dataset (from 0.62% to 0.64 % R-precision), the same operation degrades the precision (from 0.51% to 0.42%) in the polyphonic case.

Results on the duration-indexed version of the polyphonic database (DB3) confirm the fact that dynamical algorithms are helped by keeping the duration constant within a class. Conversely, static algorithms that do not consider duration are penalized by blindly returning samples which may be of the correct DB2 class, but not in the correct DB3 class. However, the performance of dynamical algorithms on DB3 remains more than 25% worse than the static performance on DB2, which shows that, even at constant duration, dynamical algorithms are poor at capturing essential feature dynamics.

4 Conclusions

The observation that dynamic algorithms overperform their static counterparts on DB1, but are ranked in inverse order on DB2 gives strong evidence that polyphony ruins attempts at modelling dynamics even within the constrained time-scale of individual notes (**H2**). This conclusion therefore generalizes all the more so to longer textures (i.e. sequences of notes), and explains the poor performance of dynamical algorithms for the timbre similarity of full songs.

Note that we are only concerned here about measures of timbre similarity, and that we do not claim that our results extend to other types of music similarity (e.g. structural (Dannenberg and Hu, 2003) or rhythmic (Gouyon and Dixon, 2005)) or the modelling of complex, multi-faceted descriptions like musical genre (Aucouturier and Pachet, 2003), which may benefit from information captured by dynamical techniques at the time-scales considered in this study.

Moreover, polyphony seem to make difficult the training of involved static algorithms such as several-component GMMs. These perform less accurately than simplistic euclidean comparison of the mean frame of each segment. As polyphonic samples tend to be longer than monophonic samples, this is not simply an effect of overfitting complex models to too little training data, but a property of the data itself. Polyphony, and notably the quasi-random superposition of asynchronous sources in a given sound sample, probably creates a higher degree of variance from one sound sample to another than in the monophonic case. This effect could probably be limited if more data were available, e.g. in the context of classification where models are trained on a set of several

songs, instead of individual songs as we do here for similarity.

As observed by Flexer et al. (2005), dynamical models tend to increase the likelihood of the training data, without yet increasing the precision of the corresponding task (as we see here). This suggests that the dynamical techniques considered here are able to model some additional statistical information in the sequence of features, but that this information is mostly meaningless for perception. One possible explanation is that human polyphonic listening involves mechanisms of categorization and selective attention to specific sound sources in a texture (e.g."listening to the guitar part"). It is likely that the dynamical factors identified by psychoacoustical research are only meaningful within the stream of a categorized sound source, and not between arbitrary adjacent sound events as modelled by the current frame-based approaches.

Overall, this suggests that the horizontal coding of frames of data, without any account of source separation and selective attention, is a very inefficient representation of polyphonic musical data, and not cognitively plausible. On that respect, more brain-plausible processings such as sparse representations (Georgiev et al., 2005; Daudet, 2006) may provide a fruitful direction for further research.

Acknowledgment

The authors would like to thanks Anthony Beurivé for helping with the implementation of signal processing algorithms and database metadata management. This work has also been influenced by many discussions with Tristan Jehan. This research has been partially funded by the Semantic Hifi IST European project (http://shf.ircam.fr/).

A Appendix: Experimental Methodology for Table 1

The data reported in Table 1 is reproduced from previous work by the authors (Aucouturier and Pachet, 2004; Aucouturier et al., 2006). We give here some details about the corresponding experimental methodology. Please refer to the original papers for complete details.

A.1 Task:

The evaluated task is to use the tested algorithms to build a timbre similarity measure between music titles. For each choice of feature and model (e.g. HMMs of MFCCs), we build a measure of the timbre similarity of songs by comparing timbre models to one another with an appropriate metric (see below for specific details). We then measure the quality of the similarity algorithm by comparing the set of nearest neighbors obtained for each song to a groundtruth, described below.

A.2 Algorithms:

Except when mentionned, all tested features are modelled with a 50-state GMM, which we found was an optimal number of components in the case of MFCCs. Delta and acceleration coefficients are computed on 20-coefficient MFCC vectors. The reported Θ values are in number of frames, as in Equation 1. For texture windows of size wt frames, we compute the mean and variance of the 20-coefficient MFCCs on running windows overlapping by $wt-1$ frames. GMMs and HMMs are compared to one another by a Monte-Carlo approximation of the Kullback-Leibler distance, using 2000 samples in the case of GMMs and 200 sequences of 100 samples in the case of HMMs. The co-occurrence matrices and histograms, described in Aucouturier et al. (2006), are compared by euclidean distance.

A.3 Ground Truth:

The test database contains 350 song items, extracted from the 15,460-files Cuidado database (Vinet et al., 2002). It is composed of 37 clusters of songs by the same artist, which were refined by hand to satisfy 3 additional criteria:

- First, clusters are chosen so they are as distant as possible from one another.
- Second, artists and songs are chosen in order to have clusters that are "timbrally" consistent (all songs in each cluster sound the same).
- Finally, we only select songs that are timbrally homogeneous, i.e. there is no big texture change within each song.

The database is constructed so that nearest neighbors of a given song should optimally belong to the same cluster as the seed song.

A.4 Evaluation Metric:

We measure the quality of the measure by counting the number of nearest neighbors belonging to the same cluster as the seed song, for each song. More precisely, for a given query on a song $\mathcal{S}_i$ belonging to a cluster $\mathcal{C}_{\mathcal{S}_i}$ of size $\mathcal{N}_i$, the precision is given by :

$$p(\mathcal{S}_i) = \frac{card(\mathcal{S}_k / \mathcal{C}_{\mathcal{S}_k} = \mathcal{C}_{\mathcal{S}_i} and \mathcal{R}(\mathcal{S}_k) \leq \mathcal{N}_i)}{\mathcal{N}_i} \tag{A.1}$$

where $\mathcal{R}(\mathcal{S}_k)$ is the rank of song $\mathcal{S}_k$ in the query on song $\mathcal{S}_i$.

The value we compute is referred to as the R-precision, and has been standardized within the Text Retrieval Community (Voorhes and Harman, 1999). It is in fact the precision measured after R documents have been retrieved, where R is the number of relevant documents (i.e. precision at recall 1). To give a global R-precision score for a given model, we average the R-precision over all queries (i.e. 350, which is the number of songs in the test database).

Further experiments reported in Aucouturier and Pachet (2004) show that, with the similarity measures examined here, the precision decreases linearly with the recall rate. This suggests that the R-precision value is a meaningful metric to compare algorithms to one another. Moreover, the small typical size of the song clusters used as groundtruth makes R-precision a good indication of how the algorithms behave in realistic applications contexts (e.g. "give me the 5 nearest neighbors of this song").

A.5 Baseline performance

The precision of the chance-based strategy of drawing a random set of nearest neighbor for each seed song depends on the groundtruth. For a query on a song $\mathcal{S}_i$ belonging to a cluster $\mathcal{C}_{\mathcal{S}_i}$ of size $\mathcal{N}_i$, the probability to observe n matches (e.g. songs in $\mathcal{C}_{\mathcal{S}_i}$) in a random set of p songs (from a database of size $\mathcal{N}_{tot}$) is given by

$$p(n) = \frac{\binom{\mathcal{N}_i}{n}\binom{\mathcal{N}_{tot}}{p-n}}{\binom{\mathcal{N}_{tot}}{p}} \tag{A.2}$$

and therefore the expected value $\tilde{n}$ in $[1, p]$ is given by

$$\tilde{n} = \frac{\sum_{n=1}^{p} np(n)}{\sum_{n=1}^{p} p(n)} \tag{A.3}$$

and the corresponding precision by

$$p(\tilde{n}) = \frac{\tilde{n}}{min(p, \mathcal{N}_i)} \tag{A.4}$$

This precision can then be averaged over all possible queries in the database. The R-precision is obtained in the above when $p = \mathcal{N}_i$.

References

ASA, 1960. Acoustical terminology, s.1.1-1960. american standards association.

Aucouturier, J.-J., Aurnhammer, M., Pachet, F., 2006. Sound textures: An investigation of image texture analysis for audio timbre similarity. IEEE Transactions on Speech and Audio Processing (submitted).

Aucouturier, J.J and Pachet, F., 2003. Representing Musical Genre: A State of the Art In: Journal of New Music Research, 32(1).

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

Aucouturier, J.-J., Pachet, F., Hanappe, P., 2004. From sound sampling to song sampling. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). Barcelona, Spain.

Crochemore, M., Rytter, W., 1994. Text Algorithms. Oxford University Press.

Dannenberg, R. and Hu, N., 2003. Pattern Discovery Techniques for Music Audio. In: Journal of New Music Research, vol. 32(2), pp. 153-164.

Daudet, L., 2006. Sparse and structured decompositions of signals with the molecular matching pursuit. IEEE Transactions on Speech and Audio Processing(in press).

Dubnov, S. and Fine, S., 1999. Stochastic modeling and recognition of solo instruments using harmonic and multi band noise features. Technical report, HUJI Israel, available from www.cs.huji.ac.il/labs/learning/Papers/icmc99.txt.

Eronen, A., 2003. Musical instrument recognition using ica-based transform of features and discriminatively trained hmms. In: Proceedings of the Seventh International Symposium on Signal Processing and its Applications (ISSPA), Paris, France. pp. 133–136.

Essid, S., Richard, G. and David., B., 2005. Instrument recognition in polyphonic music. In: Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia (USA).

Flexer A., Pampalk E. and Widmer G., 2005. Hidden Markov Models for spectral similarity of songs In: Proceedings of the 8th International Conference on Digital Audio Effects (DAFX'05), Madrid, Spain.

Furui, S., 1986. Speaker-independent isolated word recognition using dynamic

features of speech spectrum. IEEE Transactions on Speech and Audio Processing 34 (1), 52–59.

Georgiev, P. G., Theis, F. and Cichocki, A., July 2005. Sparse component analysis and blind source separation of underdetermined mixtures. IEEE Transactions on Neural Networks 16 (4), 992–996.

Gouyon, F. and Dixon, S., 2005. A review of automatic rhythm description systems In: Computer Music Journal 29(1), pp.34-54.

Grey, J. M., 1977. Multidimensional perceptual scaling of musical timbres. Journal of the Acoustical Society of America, 61:1270–1277.

Herrera-Boyer, P., Peeters, G. and Dubnov, S., 2003. Automatic classification of musical instrument sounds. Journal of New Music Research 32 (2), 3–21.

Iverson, P. and Krumhansl, C. L., 1993. Isolating the dynamic attributes of musical timbre. Journal of the Acoustical Society of America, 94:2595–2603.

Jehan, T., 2004. Perceptual segment clustering for music description and time-axis redundancy cancellation. In: Proceedings of the International Symposium on Music Information Retrieval (ISMIR). Barcelona, Spain.

McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G. and Krimphoff, J., 1995. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. Psychological Research 58, 177–192.

Peeters, G., LaBurthe, A., and Rodet, X., 2002. Toward automatic music audio summary generation from signal analysis. In: Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris (France).

Rabiner, L., 1989. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77 (2).

Rabiner, L. and Juang, B., 1993. Fundamentals of speech recognition. Prentice-Hall.

Scaringella, N. and Zoia, G., 2005. On the modelling of time information for automatic genre recognition systems in audio signals. In: Proc. International Symposium on Music Information Retrieval, London (UK).

Schroeder, M. R., Atal, B. S. and Hall, J. L., 1979. Optimizing digital speech coders by exploiting masking properties of the human ear. Journal of the Acoustical Society of America 66 (6), 16471652.

Smith, L., McNab, R. and Witten, I., 1998. Sequence-based melodic comparison: A dynamic programming approach. Computing in Musicology 11, 101–117.

Tzanetakis, G. and Cook, P., 2002. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing 10 (5).

Vinet, H., Herrera, P. and Pachet., F., 2002. The cuidado project. In: Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris (France).

Voorhes, E. and Harman, D., 1999. Overview of the eighth text retrieval conference. In: Proceedings of the Eighth Text Retrieval Conference (TREC), Gaithersburg, Maryland (USA).