"The Way It Sounds": Timbre Models for Analysis and Retrieval of Music Signals

Jean-Julien Aucouturier, François Pachet, and Mark Sandler, *Senior Member, IEEE*

***Abstract*—Electronic Music Distribution is in need of robust and automatically extracted music descriptors. An important attribute of a piece of polyphonic music is what is commonly referred to as "the way it sounds". While there has been a large quantity of research done to model the timbre of individual instruments, little work has been done to analyze "real world" timbre mixtures such as the ones found in popular music. In this paper, we present our research about such "polyphonic timbres". We describe an effective way to model the textures found in a given music signal, and show that such timbre models provide new solutions to many issues traditionally encountered in music signal processing and music information retrieval. Notably, we describe their applications for music similarity, segmentation and pattern induction.**

***Index Terms*—Feature extraction, information retrieval, multimedia database, music, pattern recognition.**

I. Introduction

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 also—and perhaps more importantly—need systems that help them find titles they do not know yet but will probably like.

Many content-based techniques have been proposed recently to help users navigate around large music catalogues. Collaborative filtering [1], for instance, is based on the analysis of large numbers of user profiles. When patterns are discovered in user profiles, corresponding music recommendations are issued to the users. Systems such as Amazon.com exploit these technologies with various degrees of success.

Other content-based management techniques attempt to extract information directly from the music signal. In the context of Mpeg7 in particular, many works have addressed the issues of extracting automatically features from audio signals, such as tempo [2], rhythm, or melodies [3].

In this paper, we propose to go further in the direction of content-based extraction by describing music titles based on their global *timbre quality*. The motivation for such an endeavour is two-fold. First, although it is difficult to define precisely music taste, it is quite obvious that music taste is often correlated with timbre. Some sounds are pleasing to listeners, other are not. Some timbres are specific to music periods (e.g., the sound of Chick Corea playing on an electric piano), others to musical configurations (e.g., the sound of a symphonic orchestra). The second motivation is that timbre similarity is a very natural way to build relations between music titles.

We therefore introduce here a technique to model how a given music title "sounds". More precisely, we do not attempt to label a precise "timbre" in a taxonomy of timbres, e.g., we do not wish to label a piece by Nick Drake as being "soft folk acoustic guitar and a gentle male voice with a bit of cello". However, we want to build models which we are able to compare to one another, in order to yield a measure of timbre similarity. For instance, we may say that the piece by Nick Drake "sounds like" this other acoustic piece by Bob Dylan. We present our hypothesis that timbre is an effective metric in music analysis and information retrieval in the following manner. Section II describes our specific timbre model, based on Mel Frequency Cepstral Coefficients and Gaussian Mixture Models. Then Section III describes our proposal for comparing timbre models. Section IV evaluates this approach in the context of a simple retrieval system. Noting that our work does not seek the best timbre model (if that concept could be defined), merely one that works well, Section V then focuses on using timbre models to segment individual songs, and subsequently, Section VI develops the principle of "texture score" from Section V and describes its application to music similarity issues and to the identification of repeating patterns in music.

II. Modelling Polyphonic Timbre

A. Previous Work About Timbre

A lot of research in music signal processing has dealt with timbre. However most of it has focused on monophonic simple sound samples, notably in the context of Instrument Recognition [4], i.e., identifying if a note, say A4, is being played on a trumpet or a clarinet. In the current state of art, it is generally considered that the timbre of a given instrument resides in the fine dynamics of some local signal features. A typical algorithm would be to cut the signal into short frames, and for each of these frames, to compute a rather high-dimension feature vector describing the temporal and/or spectral characteristics of the signal. Among possible temporal features are rise time, decay, and vibrato,[1] while spectral features can be, e.g.,

Manuscript received July 10, 2003; revised September 2, 2004. The work of M. Sandler was supported in part by the EU-FP6-IST-507142 project Semantic Interaction with Music Audio Contents (SIMAC) The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alexander Loui..

J.-J. Aucouturier and F. Pachet are with the SONY Computer Science Laboratory, 75005 Paris, France (e-mail: jj@csl.sony.fr; pachet@csl.sony.fr).

M. Sandler is with the Department of Electrical Engineering, Queen Mary University of London, London, U.K. (e-mail: mark.sandler@elec.qmul.ac.uk).

Digital Object Identifier 10.1109/TMM.2005.858380

[1] Although vibrato is a frequency modulation, it is generally assessed in the time domain since its variation is slow compared to the usual STFFT frame-rate.

spectral centroid, spectral skewness, or spectral roll-off. Then, to model the timbre of the instrument sample, one generally uses dynamic statistical models such as Markov chains [5], recurrent neural networks or hidden Markov models. These models are relatively complex compared to static models.

B. *Long-Term Statistics Rather Than Local Dynamics*

On the contrary, here, we are concerned with full polyphonic music and complex instrumental textures, for which we want to extract a *global* timbre description. For instance, we are interested in modeling the "timbre" of *The Beatles, Yesterday*: soft electric guitar, Paul McCartney's medium-ranged soft, melancholic voice, gentle brushes from Ringo Starr's drum kit, violin and cello joining on chorus, etc. State of the art source separation algorithms [6] cannot yet separate out individual sources from such a whole polyphonic mix. This means that we cannot use the usual framework to model timbre: the features that we would extract would not represent one given instrument, and the dynamics we would model would be a meaningless mix of the dynamics of all the individual sources, which are not synchronised.

While it is very hard to keep track of individual spectral shapes in the signal, a polyphonic signal still has a specific spectral shape of its own. Fig. 1 shows the superposition of the spectrums of 500 adjacent 50-ms frames of a polyphonic texture. One can see that these 25 seconds of music generate a very definite spectral envelope, and not a constant amplitude, noise-like superposition as one could have thought. This global shape turns out to be quite specific to a given texture, just like the spectral envelope of 2 s of trumpet in instrument recognition systems, only on a larger scale (e.g., 1 min of sound). But contrary to instrument recognition, we can only (or need only to) use static models: we are trying to capture a global and statistically emerging shape, not fine local dynamics.

The next two sections present the modeling algorithm. We proceed just like in Fig. 1. We cut the signal into frames, compute the spectral envelope of each frame, and "average" all the envelopes to describe the signal globally. In practice, we model the distribution of the envelopes as a mixture of Gaussian distributions, which is more precise and allows richer applications such as segmentation.

C. *Spectral Envelope Extraction*

The musical signal is cut into 2048 points frames (50 ms), and for each frame, we compute the short-time spectrum. We then model its spectral envelope, i.e., the curve in the frequency-magnitude space that "envelopes" the peaks of the short-time spectrum, using Mel Frequency Cepstrum ([7]). The cepstrum is the inverse Fourier transform of the log-spectrum after a nonlinear frequency warping onto a psychoacoustic frequency scale (the Mel scale)

$$c_n = \frac{1}{2\pi} \times \int\limits_{\omega=-\pi}^{\omega=+\pi} \log\left(S(e^{j\omega})\right) \cdot e^{j\omega \cdot n} d\omega \qquad (1)$$

The c_n are called Mel Frequency Cepstrum Coefficients (MFCCs). In practice, we use the discrete cosine transform (DCT) instead of the inverse FFT: this guarantees that the output values are real and decorrelated. The low order MFCCs account for the slowly changing spectral envelope, while the higher order ones describe the fast variations of the spectrum. Section IV gives complete details about the choice of an appropriate number of coefficients.

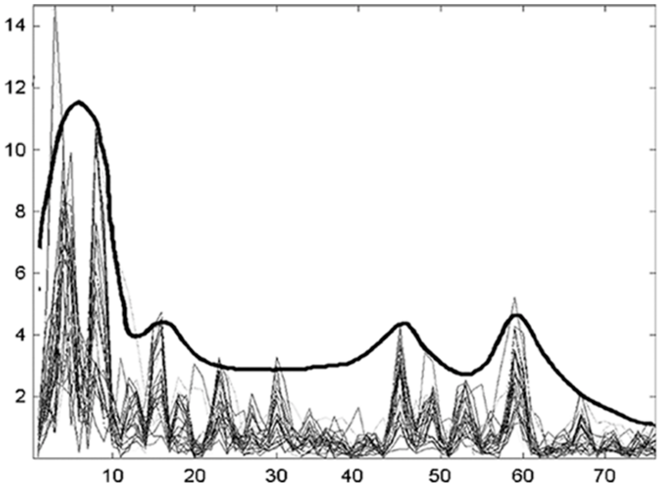


Fig. 1. Emergence of a global spectral shape for polyphonic textures (plot of amplitude against frequency in Hertz).

D. *Modeling*

We model the distribution of each song's MFCCs as a mixture of Gaussian distributions over the space of all MFCCs. A Gaussian mixture model (GMM) [8] estimates a probability density as the weighted sum of M simpler Gaussian densities, called components or states of the mixture

$$p(F_t) == \sum_{m=1}^{M} \pi_m \mathrm{N}(F_t, \mu_m, \Gamma_m) \qquad (2)$$

where F_t is the feature vector observed at time t, N is a Gaussian PDF with mean μ_m, covariance matrix Γ_m, and π_m is a mixture coefficient (also called state prior probability).

We initialize the GMM's parameters by k-mean clustering, and train the model with the classic E-M algorithm [8]. Fig. 2 shows a three-dimensional (3-D) projection of a typical feature space (which is originally dimension 8). The dots represent MFCCs and the ellipsoids are the projection of the Gaussian distributions in the trained GMM.

In Fig. 2, we use mixtures of $\mathrm{M} = 3$ Gaussian distributions. A complete discussion about the choice of an appropriate M is to be found in Section IV.

III. Comparing Timbre Models

In the previous section, we have presented how to model the global timbre of a piece of music. We present here a first application, which is also a good way to evaluate the models: comparing the timbre models of different songs to compute their "timbral similarity".

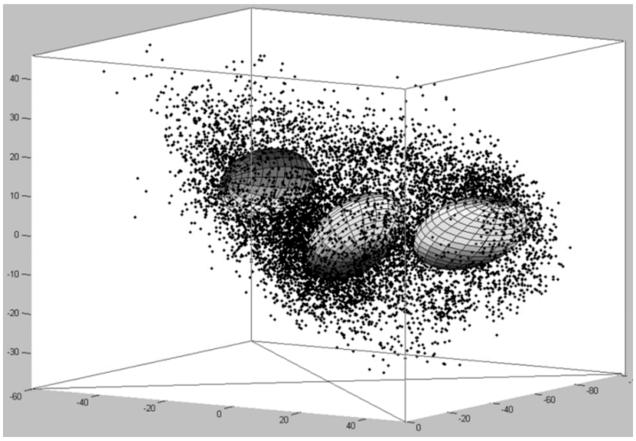
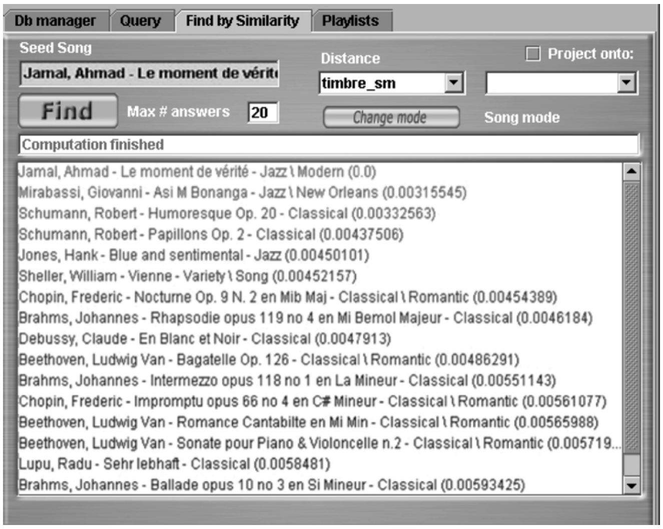


Fig. 2. GMM modeling of a distribution of MFCCs (The Beatles—"Let It Be"). The axis correspond to the three first principal components from a set of 12-dim MFCC vectors.



Seed Song: Jamal, Ahmad - Le moment de vérit

Distance: timbre_sm

□ Project onto:

Max # answers: 20

Song mode

Computation finished

Jamal, Ahmad - Le moment de vérité - Jazz \ Modern (0.0)
Mirabassi, Giovanni - Asi M Bonanga - Jazz \ New Orleans (0.00315545)
Schumann, Robert - Humoresque Op. 20 - Classical (0.00332563)
Schumann, Robert - Papillons Op. 2 - Classical (0.00437506)
Jones, Hank - Blue and sentimental - Jazz (0.00450101)
Sheller, William - Vienne - Variety \ Song (0.00452157)
Chopin, Frederic - Nocturne Op. 9 N. 2 en Mib Maj - Classical \ Romantic (0.00454389)
Brahms, Johannes - Rhapsodie opus 119 no 4 en Mi Bemol Majeur - Classical (0.0046184)
Debussy, Claude - En Blanc et Noir - Classical (0.0047913)
Beethoven, Ludwig Van - Bagatelle Op. 126 - Classical \ Romantic (0.00486291)
Brahms, Johannes - Intermezzo opus 118 no 1 en La Mineur - Classical (0.00551143)
Chopin, Frederic - Impromptu opus 66 no 4 en C# Mineur - Classical \ Romantic (0.00561077)
Beethoven, Ludwig Van - Romance Cantabilte en Mi Min - Classical \ Romantic (0.00565988)
Beethoven, Ludwig Van - Sonate pour Piano & Violoncelle n.2 - Classical \ Romantic (0.005719...
Lupu, Radu - Sehr lebhaft - Classical (0.0058481)
Brahms, Johannes - Ballade opus 10 no 3 en Si Mineur - Classical (0.00593425)

Fig. 3. Query by timbral similarity in the Cuidado music browser.

A. State of Art

Among related work in this domain, automatic genre classification ([9]) tries to categorize music titles into genre classes by looking at spectral or temporal signal features. In this approach, the tested song's timbre is matched against pre-computed models of each possible genre. Each genre model averages the timbre of a large number of songs that are known to belong to this genre. There is no matching from one song to another, but rather from one song to a group of songs.

Music title identification or audio fingerprinting ([10]) deals with identifying the title and artist of an arbitrary music signal. This is done by comparing the unlabeled signal's features to a database containing the features of all possible identified songs. In this case, the matching is done from one song to another, but the system only looks for exact matches, not for similarity.

Our approach borrows from both techniques, since it performs approximate matching of one song to another. Since our original formulation of the problem in [11], timbre similarity has seen a growing interest in the Music Information Retrieval community. Each contribution often is yet another instantiation of the same basic pattern recognition architecture, only with different algorithm variants and parameters. For a complete review and comparison of these variants, please refer to [12].

B. Comparing Timbre Models

In order to compare the timbre models of two songs, we use a sampling method to approximate the likelihood of the feature vectors of one song A given the model of another song B. We sample a large number of points S^A from model A, and compute the likelihood of these samples given model B. We then make the measure symmetric and normalize.

$$\begin{aligned} D(A,B) = &\sum_{i=1}^{i=\mathrm{NS}} \log P\left(S_i^A/A\right) + \sum_{i=1}^{i=\mathrm{NS}} \log P\left(S_i^B/B\right) \\ &- \sum_{i=1}^{i=\mathrm{NS}} \log P\left(S_i^A/B\right) - \sum_{i=1}^{i=\mathrm{NS}} \log P\left(S_i^B/A\right) \end{aligned} \tag{3}$$

where NS is the number of samples drawn from each distribution. We have found $\mathrm{NS} = 1500$ to be a sufficient value to obtain good results. This gives a distance measure which is the probability that song A be modeled by model B. Complete details about the algorithm can be found in [12].

In the context of the CUIDADO Music Browser ([13]), we have set up a database of about 20 000 popular music titles, together with metadata extracted automatically through different techniques. Metadata include information about artists, genres, tempo, energy, etc. and the herein discussed timbre models. The user can notably access this database by asking the question: "I like this song. Find me other songs that sound the same". The user selects one song "he likes" in a list, or by typing in some metadata of title, artist, etc. and the system finds out the n closest songs by comparing their timbre models.

Fig. 3 shows a screenshot of the application. The query was "*Ahmad Jamal- L'instant de Vérité*"—a jazz piano solo, and the result lists contains songs of many genres, which all contain romantic-styled piano: New Orleans Jazz (*G. Mirabassi*), Classical piano pieces (*Schumann, Chopin*), and even a "Variety" song (*William Sheller*, a French singer and pianist who had a classical training).

The most interesting similarity results are often the most unexpected ones: songs of different artists or genres, but also different dates of production, different cultural backgrounds, etc. For instance:

- Solo piano: "Classical" *Schumann—Horowitz—Kreisleriana, Op 16-5 (sehr langsam)* and "Jazz" *Bill Evans—I loves you Porgy*.
- Orchestral textures: "Classical" *Beethoven—Romanze fur Violine und Orchester Nr. 2 F-dur op. 50* and "Pop" *The Beatles—Eleanor Rigby or* "Musicals" *Gene Kelly—Singin' in the rain*.

These surprising associations provoke an exciting feeling of "discovery". Such similarities, based on our approach of the global "sound" of a piece of music, are very interesting in the context of Music Information Retrieval, because they cannot be assessed by a nonsignal method, contrary to artist and genre similarity.

IV. Evaluation

Using timbre models to assess the timbre similarity between songs is a useful framework to evaluate the quality of the modeling itself. As we will see in the next sections, other applications such as segmentation and structural analysis are difficult to evaluate per se. However, as they are all based on the same instantiation of "timbre", we believe evidence on the "similarity" application is also relevant for other related applications of the same model.

A. Test Database and Evaluation Metric

The question of evaluation is a problem that is hotly debated in the MIR community. The first step toward a unified, standardized evaluation procedure is a common test corpus, which the community has yet to produce, although recent initiatives are making this more of a reality [14].

For our problem, a test database of 350 music titles was constructed as an extract from the Cuidado database. It contains songs from 37 artists, encompassing very different genres and instrumentations. Artists and songs were chosen in order to have clusters that are "timbrally" consistent (all songs in each cluster sound the same). 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 S_i belonging to a cluster C_{S_i} of size N_i, the precision is given by

$$p(S_i) = \frac{\mathrm{card}\,(S_k/C_{S_k} = C_{S_i} \wedge R(S_k) \leq N_i)}{N_i} \qquad (4)$$

where $R(S_k)$ is the rank of song S_k in the query on song S_i.

This value is referred to as the R-precision, and has been standardized within the text retrieval conference (TREC) ([15]). It is, in fact, the precision measured after R documents have been retrieved, where R is the number of relevant documents. To give a global R-precision score for a given model, we average the R-precision over all queries.

B. Results

We use this measure to study the influence of the algorithm's two main parameters.

- The number of MFCCs (N) extracted from each frame of data. The more MFCCs the more precise the approximation of the signal's spectrum, which also means more variability on the data. As we are only interested in the spectral envelopes, not in the finer details, a large number may not be appropriate.
- The number of Gaussian components (M) used in the GMM to model the MFCCs. The more components, the better precision on the model. However, depending on the dimensionality of the data (i.e., N) more precise models may be underestimated.

N and M are not independent: there is an optimal to be found between high dimensionality and high precision of the modeling. To explore the influence of N and M, we make a complete exploration of the associated two-dimensional (2-D) space, with N varying from 10 to 50 by steps of 10, and M from 10 to 100 by steps of 10. These bounds result from preliminary tests showing that the values $\mathrm{N} = 8$ and $\mathrm{M} = 3$ used in [12] are not optimal and that the optimal (N, M) is well above (10,10). Fig. 4 shows the results of the complete exploration of the (N, M) space. We can see that too many MFCCs hurt the precision. When N increases, we model finer spectral variations, which creates unwanted variability in the data. The best R-precision $\mathrm{p} = 0.6358$ is obtained for $\mathrm{N} = 20$ and $\mathrm{M} = 50$.

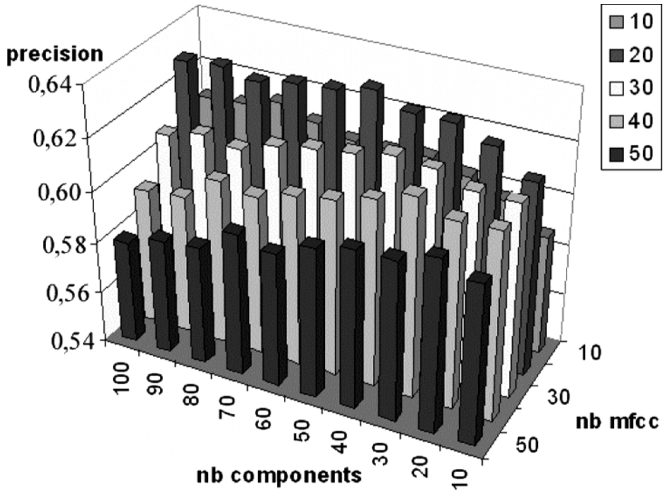


Fig. 4. Exhaustive evaluation of the model parameters showing the influence of the number of MFCCs and the number of Gaussian components on the R-precision of the similarity measure.

While this 63% of precision may appear disappointing, it is important to note that our evaluation criteria necessarily underestimates the quality of the measure, as it does not consider relevant matches that occur over different clusters (false negatives), e.g., Ahmad Jamal being close to Schumann, in the example in Fig. 3. Moreover, in Section III, we described how these false negatives ("surprising matches") can lead to exactly the sort of behavior one wants from a MIR system—helping you find the thing you did not know you wanted. For more evaluation results, see [12].

V. Segmenting Timbres Within One Song

The query by timbre described in the previous sections uses one timbre model for each song. As each model is a mixture of possibly very different gaussian distributions, it can capture several different textures for each song. For instance, *The Beatles—Let it Be* may be represented by one Gaussian for the texture "piano+voice" and another gaussian for the "electric guitar solo" in the middle of the song. It may not be logical to compare such composite models to one another. Indeed the most perfect match to "Let it Be" would be a song which has exactly the same proportion of piano and guitar, and songs which only have a very similar "piano+voice" texture may be ruled further away by the system. Therefore, it would be very interesting to be able to analyze a given song and to segment it into sections of homogeneous timbre (e.g., here extracting the guitar solo in the middle). It is the problem we deal with in this section.

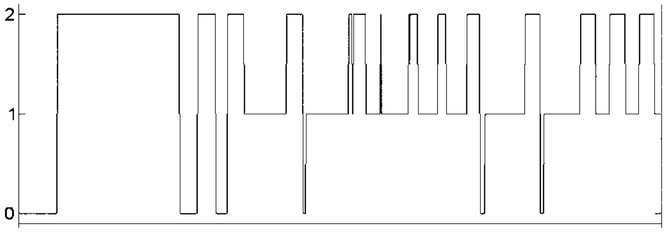


Fig. 5. Segmentation of Bourvil's song. State 0 is {silence}, state 1 is {voice + accordion + accompaniment}, and state 2 is {accordion + accompaniment}, plotted against time.

A. State of Art

In signal processing, segmenting a signal has the very broad meaning of identifying and labeling its different sections of interest. In practice, there are different points of view depending on the scale of the analysis: from smallest to largest, music researchers have called "segmentation" the process of discriminating notes and rests [16], transients and steady parts [17], instruments [4], sources (e.g., different speakers, speech versus music, etc.) [18], [19], or musical structures (verse/chorus, movements...) [20]. As regards the techniques used, we can identify two main types of segmentation algorithms.

- Novelty-based algorithms:

  These algorithms first compute a set of features from the signal cut into frames, and then detect the segment boundaries by looking for abrupt changes in the trajectory of features. The features can be generic MPEG7 features [19] (spectral centroid, spectral skewness, zero-crossing rate, etc.) or specific constructs, e.g., in [16] in-harmonicity to segment transients and vibrato. The change detection algorithms also vary from author to author: constant threshold on the derivative [19], adaptive moving average [16], or correlation with a specially designed kernel [20].

  Novelty-based algorithms have the disadvantage of not providing any "understanding" of the segmentation: They detect boundaries, but do not compare and label the resulting segments. If after four changes, say, we enter a segment (i.e., a note, a timbre, a phrase, an audio class, etc...) that has already occurred before, it will not be identified as being the same.
- Model-based algorithms:

  This second class of algorithms allows such a labeling of the segments, and thus a more "intelligent" segmentation of the signal. The data is first converted into adapted features, just like before. Then, the trajectory of features is matched with a model of each possible type of segment, and each frame is labeled with the model (i.e., the type of segment) that best fits it. As far as we know, all these model-based algorithms have relied so far on a supervised approach, where the different types of segments that can occur are known a priori. Raphael in [16] segments notes in an acoustic performance using a hidden Markov model (HMM) [21] built from the score, which is given *a priori*. Sugiyama in [18] segments audio classes (music/speech...) by first learning HMMs on manually labeled examples of each audio class (a HMM for music, a HMM for speech), and then by decoding the signal with this set of models.

The segmentation algorithm we propose here as a direct application of our timbre models falls in this second-category of algorithms. Foote in [20] also proposes a mixed novelty/model-based scheme: segments are first produced in the first manner by looking at boundaries in the trajectory of MFCCs. Then each of these segments is modeled with a gaussian distribution, compared to the other segment models, and clustered together using singular value decomposition. Contrary to Foote's algorithm, we do not rely on a first-pass boundary detector, but rather learn the segments, as well as their models in a single process.

B. Timbre Model-Based Segmentation

Once we have extracted the timbre model of a given song as described in Section II, the segmentation is simply achieved by labeling each frame with the component it is most probably generated by.

In fact, we can view the E-M algorithm used to fit several Gaussian components to the trajectory of MFCCs as an iterative version of Foote's algorithm: in the E-step, frames are labeled with their most probable segment/model, and in the M-step, we, in turn, use the frames in each segment to build segment models. After a given number of iterations, we can use the learned model to decode the data, i.e., to label each frame with its most probable component index

$$\mathrm{label}(F_i) = \underset{j=1:M}{\arg\max}\,(P(F_i/C_j)) \tag{5}$$

where F_i is a frame of data, and C_j is a Gaussian component from the song's timbre model.

Fig. 5 shows the results of such an analysis on 20 s of music, a 1960's French song by Bourvil ([21]), modeled by a 3-state timbre model. Its instrumentation consists of a male singer accompanied by an accordion, and a discrete rhythmic section. We see that the segmentation is very accurate: we notice the background accompaniment at the end of every sung phrase, sometimes even between the sung words. The accordion introduction appears very clearly, as well as the periodicities of the verse.

C. Evaluation and Further Improvement

While the evaluation of timbre models in the context of "timbre similarity" (Section IV) already gives confidence in the results of the segmentation, it is possible to directly measure the accuracy of the segmentation by listening to the homogeneity of each of the found timbre clusters (i.e., modeled by each component). In the case of the segmentation of the Bourvil song above, each of the three clusters correspond to the instruments being blanked out: one cluster has all frames of voice, another has all frames of accordion, etc. Informal listening tests show that less than 20% of the clusters contain frames from mixed sources. For instance, if we segment a jazz song ("DD Bridgewater—What is this thing called love") with a 50-component timbre model, 17 clusters account for voice frames, 11 clusters for piano frames, ten clusters for percussive frames, three for double bass frames, and nine are clusters containing mixed frames.

One concern that arises is the fact that physical sources may be shared over several clusters (which may occur if we use more components than the number of different timbre textures in the song, e.g., $M = 50$ above). We propose two ways to solve this issue. First, one can do a post-processing step where we cluster the high number of components by hierarchical clustering, until a certain cluster width is reached (this is notably reminiscent of Foote's algorithm). Individual Gaussian components can be compared e.g., with the Kullback–Leibler distance. Another way of decreasing the sharing of timbres between clusters, as well as potentially decreasing the number of mixed clusters, is to investigate more refined models, such as hidden Markov models. A HMM can be viewed as the "dynamic" extension of a GMM, where we also model the dynamics over the succession of Gaussian components. In [23], we have shown that such a dynamic modeling sometimes improves the quality of the segmentation. However, a formal comparison between GMM-timbre models and HMM-timbre models remains to be done.

In any case, the fact that some clusters may not represent a meaningful "physical" sound source, either because several clusters account for the same source, or because a given cluster gathers frames of mixed sources, is not necessarily a problem. We look here at a midlevel representation of music, which is useful even if it is not perfectly correlated to human judgments. In the next section, we will analyze the structure of the timbre segmentation we obtain here and show that it is useful for several interesting problems in Music Information Retrieval.

VI. Applications of the Texture Score

The segmentation obtained from our timbre models provides a useful representation for music, just like a very simplified score, a "texture score". In this section, we use this symbolic representation in a Music Information Retrieval perspective to match different performances of the same song, and to find repeating patterns in a song.

A. Similarity by Long-Term Structure

As we have remarked on Fig. 5, the texture score reveals much of the structure of the song: phrases succeed to phrases, common patterns are repeated every verse and chorus, instrument solos stand out clearly and echo the introduction and ending, etc.

One interesting property of the timbre representation is that it is based on spectrum, but is independent of what the spectrums really are: We only look at the succession of the textures, not at the textures themselves. A simple "A-B-A" texture score could correspond to {guitar}—{guitar + voice}—{guitar}, but could also well be {cello}—{cello + violin}—{cello}, etc.

In [24], we have used this property to match different interpretations of the same song (i.e., same long-term structure) which use different instrumentations (i.e., the spectral content of the textures is different). The matching between scores (considered as strings of texture labels) is done with the classic edit-distance algorithm [25]. The edit distance intrinsically copes with noise (similar structures can differ quite a lot locally) and time warping (two different performances with the same structure can have a different rhythm). Additionally, there is a provision to deal with permutations: as the numeration of the textures by the segmentation stage is arbitrary, a texture which is referred to as "1" in one song, could be referred to as "3" in another. The automatic reordering of the textures is dealt with by heuristics on the statistical distribution of the labels: e.g., matching a long series of "11. . .1" (an instrument solo, for instance) in one song to the same series of "33. . .3" in the other.

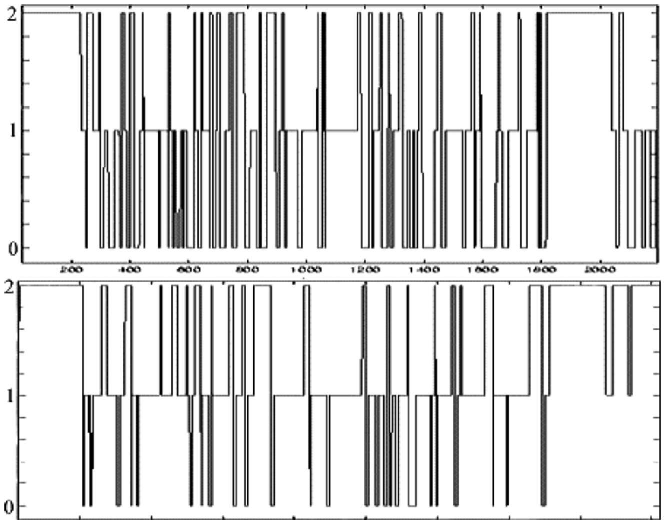


Fig. 6. Comparison of the texture score representations of two different interpretations of the same song.

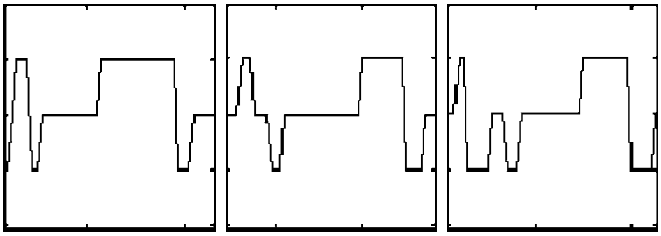


Fig. 7. Three occurrences of a pattern in Bourvil's texture score.

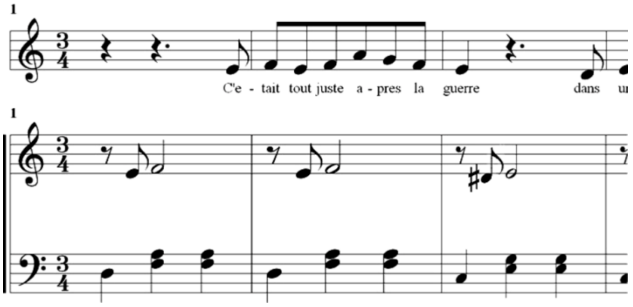


Fig. 8. Transcription of the first occurrence of the pattern.

Fig. 6 shows the texture scores for the beginning of two versions of the same song, with different instrumentation: the first one (Bourvil's song used earlier) is a male singer and an accompaniment based on accordion; the second one has a female singer and violins. Since we have freed ourselves from these spectral differences by using the texture scores, the algorithm is able to notice that the two pieces show some similarity. Details about the results can be found in [24]: tested on a database of songs, the edit distance between "covers" or more generally songs with the same long-term structure (e.g., simple blues music) is generally small, and the distance between different songs is big.

B. *Finding Repeating Patterns*

Rather than matching different songs by comparing their texture score, the authors have proposed in [26] to use the texture score to find repeating patterns within one song. In order to discover patterns in the texture score string, one could use dynamic programming as above. In [26], we have also introduced a novel string matching algorithm inspired by an image-processing technique: the Hough Transform.

The pattern analysis on the texture score of this paper's followed example, Bourvil's song "C'était bien" [22] reveals a lot of the structure of the tune. We present here an example of a short pattern found by the system. Its length is relatively small, about 3 s. It occurs 15 times during the song, five times in each occurrence of the verse/chorus unit. Fig. 7 presents three of its occurrences (the first three in the first chorus), and Figs. 8–10 show a transcription of the corresponding music by the first author.

The state sequences shown in Fig. 7 have the same labeling than in Section IV: state 1 is silence, state 2 is {voice+accompaniment}, and state 3 is {accompaniment}. In the transcriptions shown in Figs. 8– 10, the upper staff corresponds to the vocal score, and the two bottom staffs correspond to the accompaniment: accordion, and bass. The drum track has not been transcribed, as it does not influence the segmentation very much.

We can see from the transcriptions in Figs. 8—10 that these three occurrences correspond to the same sequence of scale degrees (2-3-2-3-5-4-3-2), but diatonically transposed to three levels, harmonized in Dm,C,Bb.

Classic pattern induction algorithms would deal with such a pitch similarity by using musical rules to account for transposition, or by just looking at musical contour. In our case, this similarity of the pitches cannot be assessed from the texture score, since it hides all pitch information within the textures. The algorithm thus has discovered some similarity based something else: structure. These occurrences have the same succession of textures. Note that the variations between the occurrences, such as the duration of the textures, correspond to variations of timing and expressivity on the same phrase. This is especially clear about the frames of silence (texture 1), which reveal short pauses between sung words or in the accompaniment.

It is remarkable that melodic phrases and texture timing be so closely correlated, and this suggests that a pitch transcription may not be the only useful notation to understand music. In the context of music processing, this opens the way for alternative, more abstract representations of polyphony, which are easier to generate from raw data, without having to separate sources. The texture score, using our research on timbre modeling, appears to be a good example of such a representation.

One possible application of the pattern discovery algorithm described above is "Audio Thumbnailing". The idea is to provide the user with the main characteristics of a title without playing it entirely. One strategy to extract such a summary is to select the most recurring pattern in the song. This path has notably been followed by Peeters [27] and Bartsch [28]. Our results show that texture scores can be a way to find such large patterns.

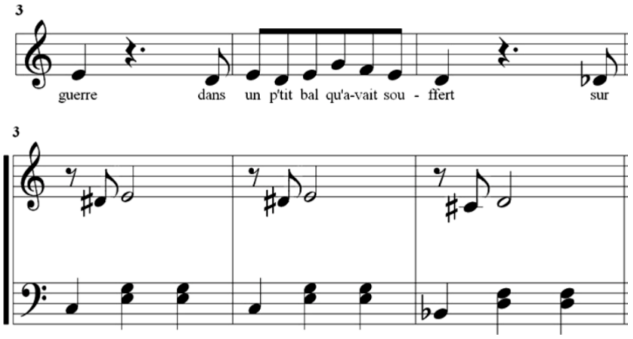


Fig. 9. Transcription of the second occurrence of the pattern.

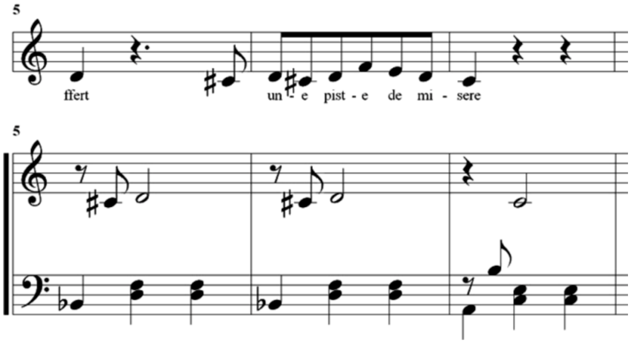


Fig. 10. Transcription of the third occurrence of the pattern.

VII. Conclusion

Electronic Music Distribution is in need of automatic descriptors of the content of a piece of music. In this paper, we have presented our research about "polyphonic timbres", i.e., how to model the global "sound" of a given music title. Mixtures of Gaussian distributions over a space of Cepstral coefficients are an efficient way to model the textures found in a given music signal. Such timbre models provide new solutions to many issues traditionally encountered in music signal processing and music information retrieval. First, they are directly applicable to compute timbre similarity between songs. Second, we have shown that the same approach allows us to segment a piece of music into sections of constant timbre, with a time resolution as small as the duration of a note. The output of such a timbre segmentation, which we call "texture score", gives a lot of information about the musical structure of the songs. It can be used to compute structural similarity between songs, and to extract meaningful recurring patterns within one song.

Acknowledgment

This paper reports on research conducted in two research groups over three years. Early work on segmentation with HMM, structural similarity, and pattern induction was carried in the Digital Music Lab at Queen Mary University of London, London, U.K. (formerly in King's College London). Timbre

models, timbre similarity, and its evaluation originated and are currently being investigated in the Music Team at Sony Computer Science Laboratory, Paris, France, in the context of the Cuidado project.

References

[1] F. Pachet *et al.*, "Musical Datamining for EMD," in *Proc. WedelMusic Conf.*, Firenze, Italy, 2001.

[2] E. Scheirer, "Music-Listening Systems," Ph.D. dissertation, MIT, Cambridge, MA, 2000.

[3] A. Klapuri, "Multiple fundamental frequency estimation by harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 804–816, Nov. 2003.

[4] D. Herrera-Boyer, G. Peeters, and S. Dubnov, "Automatic classification of musical instrument sounds," *J. New Music Res.*, vol. 32, no. 1, pp. 3–21, 2003.

[5] S. Dubnov and S. Fine, "Stochastic Modeling and Recognition of Solo Instruments Using Harmonic and Multi Band Noise Features," Tech. Rep., [Online] Available at: http://www.cs.huji.ac.il/labs/learning/Papers, 1999.

[6] M. Plumbley *et al.*, "Automatic music transcription and audio source separation," *Cybern. and Syst.*, vol. 33, no. 6, pp. 603–627, 2002.

[7] D. Schwarz and X. Rodet, "Spectral estimation and representation for sound analysis-synthesis," in *Proc. Int. Computer Music Conf.*, Beijing, China, 1999.

[8] C. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.

[9] J.-J. Aucouturier and F. Pachet, "Musical genre: A survey," *J. New Music Res.*, vol. 32, no. 1, pp. 83–93, 2003.

[10] E. Allamanche, "Content-based identification of audio material using MPEG-7 low level description," in *Proc. 2nd Int. Symp. on Music Information Retrieval (ISMIR)*, Bloomington, IN, 2001.

[11] J.-J. Aucouturier and F. Pachet, "Music similarity measures: what's the use?," in *Proc. 3rd Int. Symp. Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[12] ——, "Improving timbre similarity: how high's the sky," *J. Negative Results in Speech and Audio Sci.*, vol. 1, 2004.

[13] F. Pachet *et al.*, "Popular music access: The sony music browser," *J. Amer. Soc. Inform. Sci.*, 2003.

[14] M. Goto *et al.*, "RWC music database: popular, classical, and jazz music databases," in *Proc. 3rd Int. Conf. Music Information Retrieval (ISMIR 2002)*, Oct. 2002, pp. 287–288.

[15] E. Voorhes and D. Harman, "Overview of the eighth text retrieval conference," in *Proc. 8th Text Retrieval Conference (TREC-8)*, 2000.

[16] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden Markov models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 360–370, Apr. 1999.

[17] S. Rossignol *et al.*, "Feature extraction and temporal segmentation of acoustic signals," in *Proc. Int. Computer Music Conf.*, Ann Arbor, MI, 1998.

[18] S. Sugiyama *et al.*, "Speech segmentation and clustering based on speaker features," in *Proc. Int. Conf. Acoustics, Speecvh and Signal Processing*, vol. 2, Minneapolis, MN, Apr. 1993, pp. 395–398.

[19] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for browsing and annotation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, Oct. 1999.

[20] J. Foote and M. Cooper, "Media segmentation using self-similarity decomposition," in *Proc. SPIE Storage and Retrieval for Multimedia Databases*, vol. 5021, J. 2003.

[21] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1983.

[22] L. Bourvil, C'etait bien (le petit bal perdu), Lyrics: R. Nyel, Music: G. Verlor, Editions Bagatelle, 1961.

[23] J.-J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," in *Proc. 110th Conv. the AES*, Amsterdam, The Netherlands, 2001.

[24] ——, "Using long-term structure to retrieve music: representation amd matching," in *Proc. 2nd Int. Symp. Music Information Retrieval (ISMIR)*, Bloomington, IN, 2001.

[25] M. Crochemore and W. Rytter, *Text Algorithms*. New York: Oxford Univ. Press, 1994.

[26] J.-J. Aucouturier and M. Sandler, "Finding repeating patterns in acoustical musical signals," in *Proc. AES 22th Int. Conf. Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, 2002.

[27] G. Peeters, A. La Burthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proc. 3rd Int. Symp. Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[28] M. Bartsch and G. Wakefield, "To catch a chorus: using chroma-based representations for audio thumbnailing," in *Proc. 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2001.

Jean-Julien Aucouturier graduated from Ecole Supérieure d'Electricité (France) as an electronics and computer engineer and received the M.Sc. degree in music signal processing from King's College, University of London, London, U.K. He is currently pursuing the Ph.D. degree at the University of Paris, Paris, France.

He is a Research Assistant with Sony Computer Science Laboratory,Paris, where he investigates interaction systems with large music databases.

François Pachet received the Civ.Eng. degree from Ecole des Ponts et Chaussées, Paris, France, and the Ph.D. degree from the University of Paris 6, Paris, France.

He was an Assistant Professor in Artificial Intelligence and Computer Science at the Paris 6 University. In 1997, he set up a music research team at Sony Computer Science Laboratory, Paris, and developed the vision that metadata can greatly enhance the musical experience in all its dimensions from listening to performance. His team conducts research in interactive music listening, computer-aided performance, and musical metadata, and has developed several innovative technologies and award-winning systems (MusicSpace—constraint-based spatialization; PathBuilder—intelligent music scheduling using metadata; the Continuator—interactive music improvization). He has authored more than 60 scientific publications in the fields of musical metadata and interactive instruments.

Mark Sandler (M'88–SM'98) was born in 1955. He received the B.Sc. and Ph.D. degrees from the University of Essex, Colchester, U.K., in 1978 and 1984, respectively.

He is Professor of Signal Processing and Director of the Centre for Digital Music at Queen Mary University of London, London, U.K., where he moved in 2001 after 19 years at King's College, London. He was Founder and CEO of Insonify, Ltd., an internet audio streaming startup for 18 months. He has published over 250 papers in journals and conferences.

Dr. Sandler is a Fellow of the IEE and a Fellow of the Audio Engineering Society. He is a two-time recipient of the IEE A. H. Reeves Premium Prize.