Improving Timbre Similarity : How high's the sky ?

Jean-Julien Aucouturier Francois Pachet

Abstract—**We report on experiments done in an attempt to improve the performance of a music similarity measure which we introduced in [2]. The technique aims at comparing music titles on the basis of their global "timbre", which has many applications in the field of Music Information Retrieval. Such measures of timbre similarity have seen a growing interest lately, and every contribution (including ours) is yet another instantiation of the same basic pattern recognition architecture, only with different algorithm variants and parameters. Most give encouraging results with a little effort, and imply that near-perfect results would just extrapolate by fine-tuning the algorithms' parameters. However, such systematic testing over large, inter-dependent parameter spaces is both difficult and costly, as it requires to work on a whole general meta-database architecture. This paper contributes in two ways to the current state of the art. We report on extensive tests over very many parameters and algorithmic variants, either already envisioned in the literature or not. This leads to an improvement over existing algorithms of about 15% R-precision. But most importantly, we describe many variants that surprisingly do not lead to any substancial improvement. Moreover, our simulations suggest the existence of a "glass ceiling" at R-precision about 65% which cannot probably be overcome by pursuing such variations on the same theme.**

I. Introduction

The domain of Electronic Music Distribution has gained worldwide attention recently with progress in middleware, networking and compression. However, its success depends largely on the existence of robust, perceptually relevant music similarity relations. It is only with efficient content management techniques that the millions of music titles produced by our society can be made available to its millions of users.

A. Timbre Similarity

In [2], we have proposed to computing automatically music similarities between music titles based on their global timbre quality. The motivation for such an endeavour was 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). In any case, listeners are sensitive to timbre, at least in a global manner.

The second motivation is that timbre similarity is a very natural way to build relations between music titles. The very notion of two music titles that "sound the same" seems to make more sense than, for instance, query by humming. Indeed, the notion of melodic similarity is problematic, as a change in a single note in a melody can dramatically impact the way it is perceived (e.g. change from major to minor). Conversely, small variations in timbre will not affect the timbre quality of a music title, considered in its globality. Typical examples of timbre similarity as we define it are :

- a Schumann sonata ("Classical") and a Bill Evans piece ("Jazz") are similar because they both are romantic piano pieces,
- A Nick Drake tune ("Folk"), an acoustic tune by the Smashing Pumpkins ("Rock"), a bossa nova piece by Joao Gilberto ("World") are similar because they all consist of a simple acoustic guitar and a gentle male voice, etc.

B. State of the Art

Timbre Similarity has seen a growing interest in the Music Information Retrieval community lately ([4], [5], [7], [12], [13], [18], [22], [25], [32]).

Each contribution often is yet another instantiation of the same basic pattern recognition architecture, only with different algorithm variants and parameters. The signal is cut into short overlapping frames (usually between 20 and 50ms and a 50% overlap), and for each frame, a feature vector is computed, which usually consists of Mel Frequency cepstrum Coefficients (MFCC, see section II for more details). The number of MFCCs is an important parameter, and each author comes up with a different number: 8([2]), 12([12]), 13 ([4]),14([18]), 19([22]), 20([7].

Then a statistical model of the MFCCs' distribution is computed. K-means are used in [4], [7], [13], [22], [25], and GMMs in [2], [7], [18]. Once again, the number of kmean or GMM centres is a discussed parameter which has received a vast number of answers : 3 ([2]), 8 ([7]), 16 ([4], [7], [22]), 32([7], [18]), 64([7]). [25] uses a computationally simpler histogram approach computed from Bark Loudness representation, and [12] uses a supervised algorithm (tree-based vector quantizer) that learns the most distinctive dimensions in a given corpus.

Finally, models are compared with different techniques: sampling ([2]), Earth Mover's distance ([4], [7], [22]), Asymptotic Likelihood Approximation ([7]).

All these contributions (including ours) give encouraging results with a little effort and imply that near-perfect results would just extrapolate by fine-tuning the algorithms' parameters.

We should make clear here that this study is only concerned with timbre similarity, and that we do not claim that its conclusions extend to music similarity in general (whatever this may mean), or related tasks like classification or identification. Recent research in automatic genre classification and artist identification, for instance, have

Jean-Julien Aucouturier is assistant researcher in Sony Computer Science Laboratory, Paris, France (Phone: (+33)144080511, Email: jj@csl.sony.fr)

Francois Pachet is researcher in Sony Computer Science Laboratory, Paris, France (Phone: (+33)144080516, Email: pachet@csl.sony.fr)

shown that the incorporation of other features such as beat and tempo information ([30]), singing voice segmentation ([17], [6]) and community metadata ([33]) could improve the performance. However, such techniques are not explored here as they go beyond the scope of timbre perception.

C. Evaluation

This article reports on experiments done in an attempt to improve the performance of the class of algorithms described above. Such extensive testing over large, dependent parameter spaces is both difficult and costly.

Subjective evaluations are somewhat unreliable and not practical in a systematic way: in the context of timbre similarity, we have observed that the conditions of experiment influence the estimated precision a lot. It is difficult for the users not to take account of a priori knowledge about the results. For instance, if the nearest neighbor to a jazz piece is also a piece by another famous jazz musician, then the user is likely to judge it relevant, even if the two pieces bear no timbre similarity. As a consequence of this, a same similarity measure may be judged differently depending on the application context.

Objective evaluation is also problematic, because of the choice of a ground truth to compare the measure to. In [1], we have projected our similarity measure on genre metadata to study its agreement to the class information, using the Fisher coefficient. We concluded that there were very little overlap with genre clusters, but it is unclear whether this is because the precision of the timbre similarity is poor, or because timbre is not a good classification criteria for genre. Several authors have studied the problem of choosing an appropriate ground truth : [22] considers as a good match a song which is from the “same album”, “same artist”, “same genre” as the seed song. [25] also proposes to use “styles” (e.g. Third Wave ska revival) and “tones” (e.g. energetic) categories from the All Music Guide AMG [1]. [7] pushes the quest for ground truth one step further by mining the web to collect human similarity ratings.

The algorithm used for timbre similarity comes with very many variants, and has very many parameters to select. At the time of [2], the systematic evaluation of the algorithm was so unpractical that the chosen parameters resulted from hand-made parameter twitching. In more recent contributions, such as [4], [25], our measure is compared to other techniques, with similarly fixed parameters that also result from little if any systematic evaluation. More generally, attempts at evaluating different measures in the literature tend to compare individual contributions to one another, i.e. particular, discrete choices of parameters, instead of directly testing the influence of the actual parameters. For instance, [25], [5] compares the settings in [22](19 MFCCs+16Kmeans) to those of [2](8 MFCCs+3GMM).

Finally, conducting such a systematic evaluation is a daunting task, since before doing so, it requires building a general architecture that is able to :

- access and manage the collection of music signals the measures should be tested on
- store each result for each song (or rather each duplet of songs as we are dealing with a binary operation $dist(a, b) = d$ and each set of parameters
- compare results to a ground truth, which should also be stored
- build or import this ground truth on the collection of songs according to some criteria
- easily specify the computation of different measures, and to specify different parameters for each algorithm variant, etc...

In the context of the European project Cuidado, the music team at SONY CSL Paris has built a fully-fledged EMD system, the Music Browser ([24]), which is to our knowledge the first system able to handle the whole chain of EMD from metadata extraction to exploitation by queries, playlists,etc. Metadata about songs and artists are stored in a database, and similarities can be computed on-the-fly or pre-computed into similarity tables. Its open architecture makes it easy to import and compute new similarity measures. Similarity measures themselves are objects stored in the database, for which we can describe the executables that need to be called, as well as the arguments of these executables. Using the Music Browser, we were able to easily specify and launch all the simulations that we describe here, directly from the GUI, without requiring any additional programming or external program to bookkeep the computations and their results.

This paper contributes in two ways to the current state of the art. We report on extensive tests over very many parameters and algorithmic variants, some of which have already been envisioned in the literature, some others being inspired from other domains such as Speech Recognition. This leads to an absolute improvement over existing algorithms of about 15% R-precision. But most importantly, we describe many variants that surprisingly do not lead to any substancial improvement of the measure's precision. Moreover, our simulations suggest the existence of a “glass ceiling” at R-precision about 65% which probably cannot be overcome by pursuing such variations on the same theme.

II. Framework

In this section, we present the evaluation framework for the systematic exploration of the parameter space and variants of the algorithm we introduced in [2]. We first describe the initial algorithm, and then describe the evaluation process.

A. The initial algorithm

Here we sum up the original algorithm as presented in [2]. As can be seen in Figure 1, it has a classical pattern recognition architecture. The signal is first cut into frames. For each frame, we estimate the spectral envelope by computing a set of Mel Frequency Cepstrum Coefficients. The cepstrum is the inverse Fourier transform of

[1] www.allmusic.com

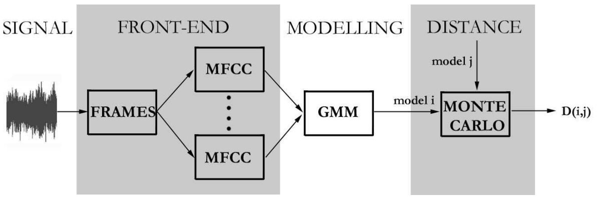

Fig. 1. The initial algorithm has a classical pattern recognition architecture.

the log-spectrum $\log \mathcal{S}$.

$$c_n = \frac{1}{2\pi} \int_{\omega=-\pi}^{\omega=\pi} \log \mathcal{S}(\omega) \exp j\omega n \, \mathrm{d}\omega \tag{1}$$

We call mel-cepstrum the cepstrum computed after a non-linear frequency warping onto a perceptual frequency scale, the Mel-frequency scale ([27]). The c_n are called Mel frequency cepstrum coefficients (MFCC). Cepstrum coefficients provide a low-dimensional, smoothed version of the log spectrum, and thus are a good and compact representation of the spectral shape. They are widely used as feature for speech recognition, and have also proved useful in musical instrument recognition ([10]).

We then model the distribution of the MFCCs over all frames using a Gaussian Mixture Model (GMM). A GMM estimates a probability density as the weighted sum of $\mathcal{M}$ simpler Gaussian densities, called components or states of the mixture. ([8]):

$$p(\mathcal{F}_t) = \sum_{m=1}^{m=\mathcal{M}} \pi_m \mathcal{N}(\mathcal{F}_t, \mu_m, \Sigma_m) \tag{2}$$

where $\mathcal{F}_t$ is the feature vector observed at time t, $\mathcal{N}$ is a Gaussian pdf with mean μ_m, covariance matrix Σ_m, and π_m is a mixture coefficient (also called state prior probability). The parameters of the GMM are learned with the classic E-M algorithm ([8]).

We can now use these Gaussian models to match the timbre of different songs, which gives a similarity measure based on the audio content of the music. The timbre models are meant to integrate into a large scale meta-database architecture, hence we need to be able to compare the models themselves, without storing the MFCCs. In [2], we use a Monte Carlo approach to approximate the likelihood of the MFCCs of one song A given the model of another song B: we sample a large number of points $\mathcal{S}^{\mathcal{A}}$ from model A, and compute the likelihood of these samples given Model B. We then make the measure symmetric and normalize :

$$D(\mathcal{A},\mathcal{B}) = \sum_{i=1}^{i=\mathcal{DSR}} \log \mathcal{P}(\mathcal{S}_i^{\mathcal{A}}/\mathcal{A}) + \sum_{i=1}^{i=\mathcal{DSR}} \log \mathcal{P}(\mathcal{S}_i^{\mathcal{B}}/\mathcal{B}) - \sum_{i=1}^{i=\mathcal{DSR}} \log \mathcal{P}(\mathcal{S}_i^{\mathcal{A}}/\mathcal{B}) - \sum_{i=1}^{i=\mathcal{DSR}} \log \mathcal{P}(\mathcal{S}_i^{\mathcal{B}}/\mathcal{A}) \tag{3}$$

The precision of the approximation is clearly dependent on the number of samples, which we call Distance Sample Rate (DSR).

B. Test Database, Ground Truth and Evaluation metric

A test database of 350 music titles was constructed as an extract from the Cuidado database [24] (which currently has 15,000 mp3 files). It contains songs from 37 artists, encompassing very different genres and instrumentations. Table I shows the contents of the database[2]. While the size of test database may appear small, we would like to stress the very heavy computational load of computing a large number of n^2 similarity matrices, some of which resulting from intensive, non optimized algorithms (e.g. HMMs with Viterbi decoding for each duplet of song). This has prevented us from increasing the size of database any further. The computation of a single similarity matrix on the full Cuidado database (15,000 songs) can represent up to several weeks of computation, and this study relies on more than a hundred of such matrices.

Artists and songs were chosen in order to have clusters that are "timbrally" consistent (all songs in each cluster sound the same). Hence, we use a variation on the "same artist/same album" ground truth as described in section I, which we refine by hand by selecting the test database according to subjective similarity ratings.

Moreover, we only select songs that are timbrally homogeneous, i.e. there is no big texture change within each song. This is to account for the fact that we only compute and compare one timbre model per song, which "merges" all the textures found in the sound. In the case of more heterogeneous songs (e.g. Queen - Bohemian rapsody), a segmentation step could increase the accuracy of the measure, but such techniques are not considered in this study (see for instance [11]).

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 :

$$p(\mathcal{S}_i) = \frac{card(\mathcal{S}_k/\mathcal{C}_{\mathcal{S}_k} = \mathcal{C}_{\mathcal{S}_i} and \mathcal{R}(\mathcal{S}_k) \leq \mathcal{N}_i)}{\mathcal{N}_i} \tag{4}$$

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

This framework is very close to traditional IR, where we know the number of relevant documents for each query. The value we compute is referred to as the R-precision, and has been standardized within the Text REtrieval Conference (TREC) [31]. 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.

III. Finding the best set of parameters for the original algorithm

As a first evaluation, we wish to find the best set of parameters for the original algorithm described above. We

[2] The "descriptions" were taken from the AMG (www.allmusic.com)

TABLE II
Influence of signal's sample rate

SR	R-Precision
11kHz	0.488
22kHz	0.502
44kHz	0.521

explore the space constituted by the following parameters :

- Signal Sample Rate (SR): The sample rate of the music signal. The original value in the system in [2] is 11KHz. This value was chosen to reduce the CPU time.
- Number of MFCCs (N): The number of the MFCCs 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, faster details like pitch, a large number may not be appropriate. The original value used in [2] is 8.
- Number of Components (M): The number of gaussian components 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. the number of MFCCs), more precise models may be underestimated. The original value is 3.
- Distance Sample Rate (DSR): The number of points used to sample from the GMMs in order to estimate the likelihood of one model given another. The more points, the more precision on the distance, but this increases the required CPU time linearly.
- Alternative Distance : Many authors ([22], [7]) propose to compare the GMMs using the Earth Mover's distance (EMD), a distance measure meant to compare histograms with disparate bins ([28]). EMD computes a general distance between GMMs by combining individual distances between gaussian components.
- Window Size :The size of the frames on which we compute the MFCCs.

As this 6-dim space is too big to explore completely, we make the hypothesis that the influence of SR, DSR, EMD and Window Size are both independent of the influence of N and M. However, it is clear from the start that N and M are linked: there is an optimal balance to be found between high dimensionality and high precision of the modeling (*curse of dimensionality*).

A. influence of SR

To evaluate SR, we fix N, M and DSR to their default values used in [2] (8,3 and 2000 resp.). In Table II, we see that the SR has a positive influence on the precision. This is probably due to the increased bandwidth of the higher definition signals, which enables the algorithm to use higher frequency components than with low SR.

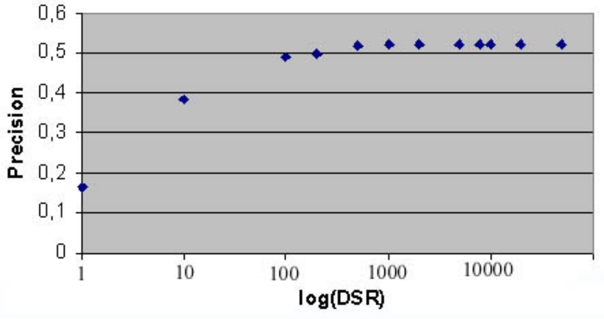

Fig. 2. Influence of the distance sample rate

TABLE III
Distance function

Distance	R-Precision
EMD-KL	0.477
EMD-MA	0.406
DSR=2000	0.488

B. influence of DSR

To evaluate DSR, we fix N = 8, M=3 and SR= 44KHz. In Figure 2, we see that the DSR has a positive influence on the precision when it increases from 1 to 1000, and that further increase has little if any influence. Further tests show that the optimal DSR does not depend on either N or M.

C. influence of EMD

To evaluate the EMD against our sampling scheme using DSR, we fix N = 8, M=3 and SR= 11KHz. We compare EMD with Kullback-Leibler, EMD with Mahalanobis and sampling with DSR=2000. In Table In III, we see that EMD with Mahalanobis distance performs worst, and that EMD with Kullback Leibler and sampling are equivalent (with a slight advantage to sampling). The difference between MA and KL is probably due to the fact that MA takes less account of covariance differences between components (2 gaussian components having same means and different covariance matrices have a zero Mahalanobis distance).

D. influence of N,M

To explore the influence of N and M, we make a complete exploration of the associated 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 boundaries result from preliminary tests (moving N while M=3, and moving M while N=8) showing that both default values N=8 and M=3 are not optimal, and that the optimal (N,M) was well above (10,10). Figure 3 shows the results of the complete exploration of the (N,M) space.

We can see that too many MFCCs ($N \geq 20$) hurt the precision. When N increases, we start to take greater account of the spectrum's fast variations, which are correlated with pitch. This creates unwanted variability in the data, as we

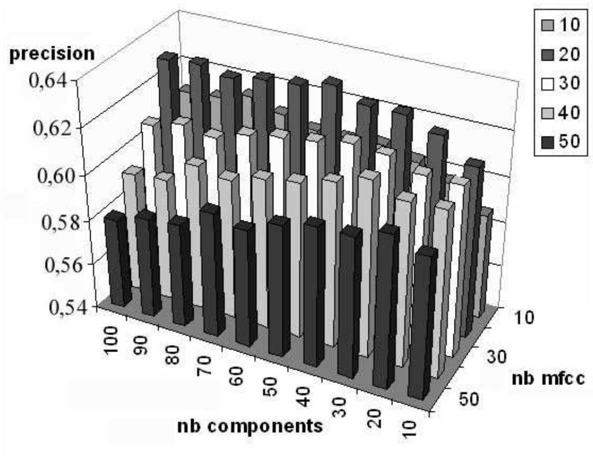

Fig. 3. Influence of the number of MFCCs and the number of components

want similar timbres with different pitch to be matched nevertheless.

We also notice that increasing the number of components at fixed N, and increasing N at fixed M is eventually detrimental to the precision as well. This illustrates the curse of dimensionality mentioned above. The best precision $p = 0.63$ is obtained for 20 MFCCs and 50 components. We can also note that the number of MFCCs is a more critical factor than the number of Gaussian components : $\forall M, N \neq 20, p(N_0 = 20, M) \geq p(N, M)$. This means we can decrease M to values smaller than optimum without much hurting the precision, which is an interesting point as the computational cost of comparing models depends linearly on M.

E. influence of Windows Size

To evaluate the influence of the window size used to segment the waveforms, we fix N = 20, M=50, SR=44 KHz and DSR = 2000. In Figure 4, we see that the window size has a small positive influence on the precision when it increases from 10 ms to 30ms, but that further increase up to 1 second has a negative effect. This behaviour results from the fact that MFCCs are only meaningful on stationary frames (larger frames may include more transients and variations) and that larger frames means less data available for the training, which decreases the precision of the models.

F. Conclusion

In conclusion, this systematic exploration of the influence of 4 parameters of the original algorithm results in an improvement of the precision of more than 16%, from the original $p = 0.48$ to the optimal $p = 0.63$ for (SR=44kHz, N=20, M=50, DSR=2000).

While this 63% of precision may appear poor, it is important to note that our evaluation criteria necessarily underestimates the quality of the measure, as it doesn't consider relevant matches that occur over different clusters (*false negatives*), e.g. a Beethoven piano sonata is timbrally close to a jazz piano solo). Indeed, the subjective performance reported in [2] was much better than the corresponding $p = 0.48$ evaluated here. The current test is mainly meaningful in a relative way, as it is able to objectively measure an increase or loss of performance due to some parameter change.

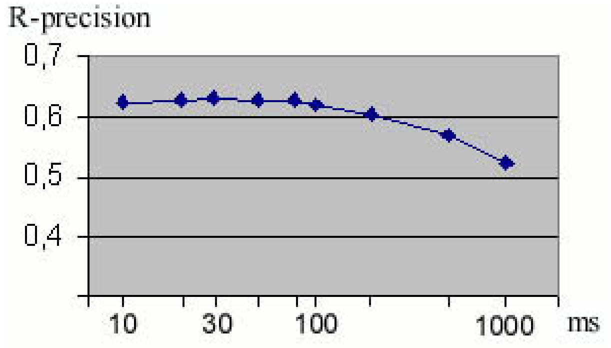

Fig. 4. Influence of the windows size

In the next 2 sections, we examine the influence of a number of algorithmic variants concerning both the front-end and the modeling, and see if they are able to further improve the R-precision.

IV. Evaluating front-end variations

A. Processing commonly used in Speech Recognition

MFCCs are a very common front-end used in the Speech Recognition community (see for instance [27]), and a variety of pre and post-processing has been tried and evaluated for speech applications. Here we examine the influence of 7 common operations :

- ZMeanSource : The DC mean is removed from the source waveform before doing the actual signal analysis. This is used in speech to remove the effects of A-D conversion.
- Pre-emphasis : It is common practice to pre-emphasize the signal by applying the first order difference equation :

$$s'_n = s_n - k s_{n-1} \tag{5}$$

to the samples s_n in each window, with k a preemphasis coefficients, $0 < k < 1$. Pre-emphasis is used in speech to reduce the effects of the glottal pulses and radiation impedance and to focus on the spectral properties of the vocal tract.

- Dither : Certain kind of waveform data can cause numerical problems with certain coding schemes (*finite wordlength effects*). adding a small amount of noise to the signal can solve this. The noise is added to the samples using :

$$s'_n = s_n + q\mathcal{RND} \tag{6}$$

where $\mathcal{RND}$ is a uniformly distributed normalized random value and q is a scaling factor.

- Liftering : Higher order MFCCs are usually quite small, and this results in a wide range of variances from low to high order. this may cause problems in distribution modeling. Therefore it is common practice in speech to rescale the coefficients to have similar magnitude. This is done by filtering in the cepstrum domain (L*i*F*tering*) according to :

$$c'_n = (1 + \frac{L}{2} \sin \frac{\pi n}{L} c_n) \tag{7}$$

where L is a liftering parameter.

- Cepstral mean compensation (CMC) : The effect of adding a transmission channel on the source signal is to multiply the spectrum of the source by a channel transfer function. In the cepstral log domain, this multiplication becomes an addition which can be removed by subtracting the cepstral mean.
- 0'th order coefficient : The 0'th cepstral parameter C_0 can be appended to the c_n. It is correlated with the signal's log energy :

$$E = \log \sum_{n=1}^{N} s_n^2 \tag{8}$$

- Delta and acceleration coefficients : The performance of a speech recognition system can be greatly enhanced by adding time derivatives to the basic static parameters. Delta Coefficients are computed using the following formula :

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2} \tag{9}$$

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.

Table IV shows the results on the test database. We notice that subtracting the cepstral mean and computing delta and acceleration coefficients for large time windows severely degrade the performance. Pre-emphasis and Dither have little effect compared to the original MFCCs. Nevertheless, liftering, normalizing the original signal, appending short-term delta and acceleration coefficients and appending the 0'th coefficient all improves the precision of the measure.

We have tried to combine this operations (this is referred to as "Best 3" and "Best 4" in Table IV), however this does not further improve the precision. We should also consider fine-tuning the number of Gaussian components again considering the increase in dimensionality due to the appending of delta and acceleration coefficients.

We should note here that the finding that including c0 slightly improves the performance is at odds to some of the results reported in [7]. In any case, the overall influence (either positive here or negative elsewhere) of this variant is small (a few percent). We further discuss these results in the concluding section.

B. Texture windows

The previous experiment shows that adding some short-term account of the MFCC statistics (i.e. delta or acceleration coefficients) has a positive (although limited) influence on the R-precision. In this paragraph, we investigate the modelling of the long-term statistics of the feature vectors.

TABLE IV
Influence of Pre/Post Processing

Variant	R-Precision
Acceleration $\Theta = 50$	0.179
Delta $\Theta = 50$	0.522
Cepstral Mean Compensation	0.525
Delta $\Theta = 10$	0.60
Acceleration $\Theta = 10$	0.610
Delta $\Theta = 2$	0.624
Delta $\Theta = 5$	0.625
Acceleration $\Theta = 5$	0.625
Pre Emphasis $k = 0.97$	0.628
Acceleration $\Theta = 1$	0.628
Original MFCC	0.629
Dither $q = 5\%$	0.629
Lifter $L = 22$	0.630
Delta $\Theta = 1$	0.631
ZMeanSource	0.631
Acceleration $\Theta = 2$	0.631
0'th coefficient	0.652
Best 3	0.605
Best 4	0.609

It has been shown that, for modeling music, using a larger scale texture window and computing the means and variances of the features over that window results in significant improvements in classification. Tzanetakis in [30] reports a convincing 15% precision improvement on a genre classification task when using accumulations of up to about 40 frames (1 second). This technique has the advantage of capturing the long-term nature of sound textures, while still assuring that the features be computed on small stationary windows (as proved necessary in section III)

We report here the evaluation results using such texture windows, for a texture window size w_t growing from 0 to 100 frames by steps of 10. $w_t = 0$ corresponds to using directly the MFCCs without any averaging, like in section III. For $w_t \geq 0$, we compute the mean and average of the MFCCs on running texture windows overlapping by $w_t - 1$ frames. For an initial signal of n frames of N coefficients each, this results in $n - w_t + 1$ frames of $2N$ coefficients : N means and N variances. We then model the resulting feature set with a M-component GMM. For the experiment, we use the best parameters obtained from section III, i.e. $N = 20$ and $M = 50$. Figure 5 shows the influence of w_t on the R-precision. It appears that using texture windows has no significant influence on the R-precision of our similarity task, contrary to the classification task reported by Tzanetakis : the maximum increase of R-precision is 0.4% for $w_t = 20$, and the maximum loss is 0.4% for $w_t = 10$.

Several directions could be further explored to try to adapt Tzanetakis' suggestion of texture windows. First, the computation of N-dimensional means and variances

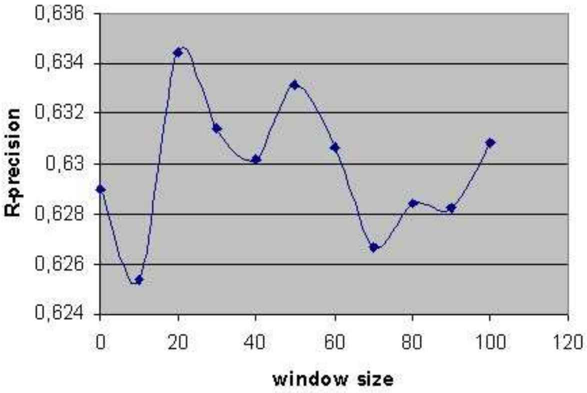

Fig. 5. Influence of the texture window size

doubles the dimension of the feature space, hence the optimal number of GMM components M should be adapted accordingly. Second, the use of one single mean (and variance) vector for each window may create a "smearing" of very dissimilar frames into a non-meaningful average. It is likely that using a small size GMM for each texture window would increase the precision of the modelling. However, this raises a number of additional issues which were not studied here, among which :

- Which is the optimal number of gaussians, for each frame, and then for the global model ?
- Should the gaussian centres be tracked between neighboring frames ?

Finally, in the single-component case, the mean of the frame-based means (with no overlap) of a signal $\{a_i\}$ is trivially equal to the global mean :

$$\frac{1}{n}\sum_{i=0}^{i=n-1}\frac{1}{m}\sum_{j=im+1}^{j=(i+1)m}a_j=\frac{1}{nm}\sum_{i=0}^{i=nm-1}a_i \qquad (10)$$

Although the extension of this behaviour in the case of multi-component GMMs cannot be written explicitely (as it results from a learning algorithm), this suggests that the real influence of this processing remains unclear. The extra information captured by texture windows may be more appropriately provided by an explicit segmentation pre-processing, or time-sensitive machine learning techniques like hidden Markov models, as we investigate in section V.

C. Spectral Contrast

In [15], the authors propose a simple extension of the MFCC algorithm to better account for music signals. Their observation is that the MFCC computation averages the spectrum in each sub-band, and thus reflects the average spectral characteristics. However, very different spectra can have the same average spectral characteristics. Notably, it is important to also keep track of the relative spectral distribution of peaks (related to harmonic components) and valleys (related to noise). Therefore, they extend the MFCC algorithm to not only compute the average spectrum in each band (or rather the spectral peak), but also a correlate of the variance, the *spectral contrast* (namely the amplitude between the spectral peaks and valleys in each subband). This modifies the algorithm to output 2 coefficients (instead of one) for each Mel subband. Additionally, in the algorithm published in [15], the authors replace the Mel filterbank traditionally used in MFCC analysis by an octave-scale filterbank (C_0-C_1, C_1-C_2, etc.), which is assumed to be more suitable for music. They also decorrelate the spectral contrast coefficients using the optimal Karhunen-Loeve transform.

We have implemented and evaluated two variants of Spectral Contrast here. For convenience, both variants use the MFCC Mel filterbank instead of the authors' Octave filters, and use the MFCC's Discrete Cosine Transform to approximate the K-L Transform. This has the advantage of being data independent, and thus better adapted to the implementation of a similarity task, where one wish to be able to assess the similarity between any duplet of song without first having to consider the whole available corpus (as opposed to the authors' supervised classification task, where the KL can be trained on the total data to be classified). Moreover, it has already been reported that the DCT was a satisfying approximation of the K-L transform in the case of music signals ([21]. In the first implementation (SC1), the $2N_{chan}$ coefficients (where N_{chan} is the number of subbands in the filterbank) are all appended in one block, and reduced to N cepstrum coefficients using the dct. In the second implementation, both streams of data (the N_{chan} peaks and the N_{chan} Spectral Contrast) are decorrelated separately with the DCT, resulting in $2N$ cepstral coefficients, as if we used e.g. delta coefficients.

Also, following the intuition of [15], we investigate whether removing the percussive frames in the original signal would improve the MFCC modeling of the music signals. As a pre-processing, we do a first pass on the signal to compute its frame-based Spectral Flatness ([16]), with the following formula :

$$SFM_{db}=10\log_{10}\frac{G_m}{A_m} \qquad (11)$$

where G_m is the geometrical mean and A_m the arithmetical mean of the magnitudes of the spectrum on each window. Spectral Flatness is notably used in Speech to segment voiced and unvoiced signals. Here, we discard frames with a high spectral flatness (using the 3σ criteria) before computing traditional MFCCs on the remaining frames. This is way to bypass the limitations of MFCCs stressed in [15] (poor modeling of the noisy frames), without providing any cure for it, as does Spectral Contrast.

We see that all three front-ends perform about 1% better than standard MFCCs, and that the $2N$ implementation performs best. For further improvement, Spectral Contrast could be combined with traditional Pre/Post Processing as seen above.

V. Dynamic modeling with hidden Markov models

In section IV, we have shown that appending delta and acceleration coefficients to the original MFCCs improves

TABLE V
Influence of Spectral Contrast

Implementation	R-Precision
SC1	0.640
SC2	0.656
SFN	0.636
standard MFCC	0.629

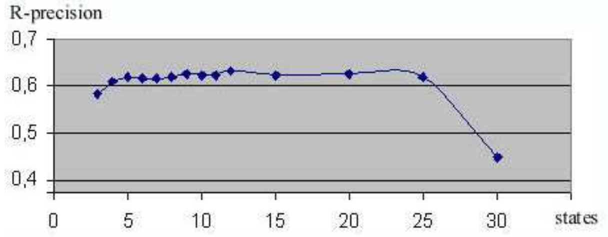

Fig. 6. Influence of the number of states in HMM modelling

the precision of the measure. This suggests that the short-term dynamics of the data is also important.

Short-term dynamical behavior in timbre may describe e.g. the way steady-state textures follow noisy transient parts. These dynamics are obviously important to compare timbres, as can be shown e.g. by listening to reverted guitar sounds used in some contemporary rock songs which bear no perceptual similarity to normal guitar sounds (same static content, different dynamics). Longer-term dynamics describe how instrumental textures follow each other, and also account for the musical structure of the piece (chorus/ verse, etc.). As can be seen in section IV, taking account of these longer-term dynamics (e.g. by using very large delta coefficients) is detrimental to the similarity measure, as different pieces with same "sound" can be pretty different in terms of musical structure.

To explicitly model this short-term dynamical behavior of the data, we try replacing the GMMs by hidden Markov models (HMMs, see [26]). A HMM is a set of GMMs (also called states) which are linked with a transition matrix which indicates the probability of going from state to another in a Markovian process. During the training of the HMM, done with the Baum-Welsh algorithm, we simultaneously learn the state distributions and the markovian process between states.

To compare HMMs with one another, we adapt the Monte Carlo method used for GMMs : we sample from each model a large number N_S of sequences of size N_F, and compute the log likelihood of each of these sequences given the other models, using equation 3. The probabilities $\mathcal{P}(\mathcal{S}_i^{\mathcal{A}}/\mathcal{B})$ are computed by Viterbi decoding.

Previous experiments with HMMs by the authors ([3]) have shown that models generalize across the songs, and tend to learn short-term transitions rather than long-term structure. This suggests that HMMs may be a good way to add some dynamical modeling to the current algorithm. In figure 6, we report experiments using a single HMM per song, with a varying number of states. The output distribution of each state is a 4-component GMM (the number of component is fixed). To compare the models, we use $N_S = 200$ and $N_F = 100$.

From figure 6, we see that HMM modeling performs no better than static GMM modeling. The maximum R-precision of 0.632 is obtained for 12 states. Interestingly, the precision achieved with this dynamic model with 4*12=48 gaussian components is comparable to the one obtained with a static GMM with 50 states. This suggests that although dynamics are a useful factor to model the timbre of individual monophonic instrument samples (see for instance [13]), it is not a useful addition to model polyphonic mixtures like the ones we are dealing with here. Probably, the dynamics modeled here by the HMMs are not meaningful, since they are a mix from all the individual sources, which are not synchronised.

VI. Conclusions

A. Best Results

The systematic evaluation conducted here gives the following conclusions :

- by fine-tuning the original algorithm's parameters, we are able to increase the precision by more than 15% (absolute), to a maximum of 63%.
- the best number of MFCCs and GMM Components is 20 and 50 respectively.
- among common speech processing front-ends, delta coefficients and 0th order MFCCs increase the precision by an unsubstantial extra 2% (absolute), to a maximum of 65,2%.
- dynamic modeling with hidden Markov models do not increase the precision any further.

Once again, we can argue that the R-precision value, measured using a simple ground truth based on artists is necessarily underestimating the actual precision of the measure. Moreover, the precision-recall curve of the best measure (using 20MFCCs + 0th order coefficient + 50 GMMs) in Figure 7 shows that the precision decreases linearly with the recall rate (with a slope of about -5% per 0.1% increase of recall). This suggests that the measure gets all the more so useful and convincing as the size of the database (i.e. the size of the set of relevant items to each query) grows.

We should also emphasize that such an evaluation qualifies more as a measure of relative performance ("is this variant useful ?") rather than as an absolute measure. It is a well-known fact that precision measures depend critically on the test corpus and on the actual implementation of the evaluation process. Moreover, we do not claim that these results generalize to any other class of music similarity/classification/identification problems.

B. "Everything performs the same"

The experiments reported here show that, except a few critical parameters (sample rate, number of MFCCs), the actual choice of parameters and algorithms used to implement the similarity measure make little difference if any.

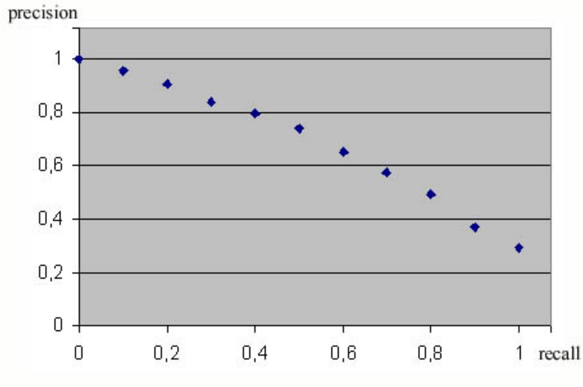

Fig. 7. Precision vs Recall graph for the best measure

We notice no substantial improvement by examining the very many variants investigated here : Complex dynamic modelling performs the same as static modeling. Complex front-ends, like spectral contrast, performs the same as basic MFCCs. Complex distance measures, such as EMD or ALA as reported in [7], performs the same as Monte Carlo, or even simpler centroid distances as also reported in [7]. This behaviour has been mentioned before in the published partial comparisons between existing distance measures : Baumann ([5]) compares [22],[2] and [4] and observes that "the different approaches reach similar performance". Pampalk in [25] remarks that the cluster organization of [22] and [2] are similar. Berenzweig et al. in [7] also conclude that the "different training techniques for GMMs (Kmeans or EM)" and "MFCC or anchor space feature achieve comparable results".

C. *Existence of a glass ceiling*

The experiments reported here also suggest that the precision achievable by variations on the same classical pattern recognition scheme adopted by most contributions so far (including ours) be bounded. Figure 8 shows the increase in R-precision achieved by the experiments in this paper, over a theoretical parameter space λ (which abstracts together all the parameters and algorithm variants investigated here). The curve shows an asymptotic behaviour at around 65% (although this actual value depends on our specific implementation, ground truth and test database).

Obviously, this paper does not cover all possible variants of the same pattern recognition scheme. Notably, one should also evaluate other low-level descriptors (LLDs) than MFCCs, such as the one used in MPEG7 ([14]), and feature selection algorithms such as discriminant analysis. Similarly, newer methods of pattern recognition such as support vector machines have proved interesting for music classification tasks ([20], [23]) and could be adapted and tested for similarity tasks. However, the set of features used here, as well as the investigation of dynamics through delta coefficients and HMMS is likely to capture most of the aspects covered by other LLDs. This suggests that the "glass ceiling" revealed in Figure 8 may also apply for further implementations of the same kind.

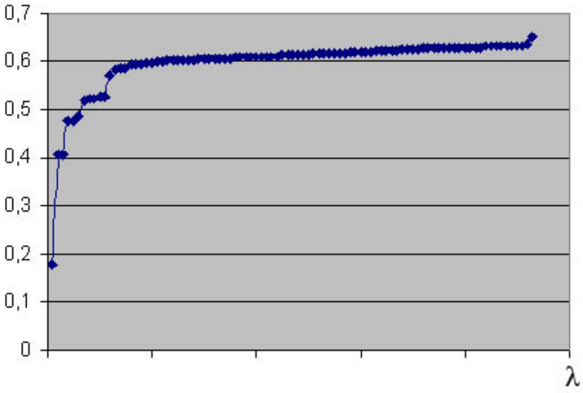

Fig. 8. Increase in R-precision over the whole parameter space used in this paper

D. *False Positives are* very *bad matches*

Even if the R-precision reported here does not account for a number of *false negatives* (songs of different clusters that actually sound the same), the manual examination of the best similarity measure shows that there also remain some *false positives.* Even worse, these bad matches are not "questionably less similar songs", but usually are *very* bad matches, which objectively have nothing to do with the seed song.

We show here a typical result of a 10-nearest neighbors query on the song "HENDRIX, Jimi - I Don't Live Today" using the best set of parameters found above :

```
1. HENDRIX, Jimi - I Don't Live Today
2. HENDRIX, Jimi - Manic Depression
3. MOORE, Gary - Cold Day in Hell
4. HENDRIX, Jimi - Love or Confusion
5. MITCHELL, Joni - Dom Juan's Reckless
 Daughter
6. CLASH, The - Give Them Enough Rope
7. CLASH, The - Stay Free
8. MARDI GRAS BB - Bye Bye Babylon
9. HENDRIX, Jimi - Hey Joe
10. HENDRIX, Jimi - Are You Experienced
```

All songs by Hendrix, Moore and the Clash sound very similar, consisting in the same style of rock electric guitar, with a strong drum and bass part, and strong, male vocals. However, the song by Joni Mitchell ranked in 5th position is a calm folk song with an acoustic guitar and a female singer, while the 8th item is a big band jazzy tune. Similar bad matches are sometimes reported in the literature, e.g. in [25] "a 10-second sequence of Bolero by Ravel (Classical) is mapped together with London Calling by The Clash (Punk Rock)", but most of the times, the very poor quality of these matches is hidden out by the averaging of the reported results.

Interestingly, in our test database, a small number of songs seems to occur frequently as false positives. Table VI ranks songs in the test database according to the number of times they occur in the first 10 nearest neighbors over all queries (N_{10}) divided by the size of their cluster of similar songs ($card(C_S)$). It appears that there are a small number of very frequent songs, which can be called "hubs". For instance, the first song, `MITCHELL, Joni - Don Juan's Reckless Daughter` occurs more than 6 times more than it should, i.e. is very close to 1 song out of 6 in the database (57 out of 350). Among all its occurrences, many are likely to be false positives. This suggests that the 35% remaining errors are not uniformly distributed over the whole database, but are rather due to a very small number of hubs (less than 10%) which are close to all other songs. These hubs are especially intriguing as they usually stand out of their clusters, i.e. other songs of the same cluster as a hub are not usually hubs themselves.

A further study should be done with a larger test database, to see if this is only a boundary effect due to our small, specific database or a more general property of the measure. However, this suggests ways to improve the precision of the measure by boosting([29]), where alternative features or modeling algorithms are used to specifically deal with the hubs.

E. Need for another approach ?

The limitations observed in this paper, namely a glass ceiling at about 65% R-precision, and the existence of very bad "hubs", suggest that the usual route to timbre similarity may not be the optimal one. The problem of the actual *perception* of timbre is not addressed by current methods. More precisely, modelling the long-term statistical distribution (accounting for time or not - HMMs or GMMs) of the individual "atoms" or "grains" of sound (frames of spectral envelopes), and comparing their global shape constitutes a strong, hidden assumption on the underlying cognitive process. While it is clear that the perception of timbre results from an integration of some sort (indivivual frames cannot be labelled independently, and may "come" from very different textures), other important aspects of timbre perception are not covered by this approach.

First, all frames are not of equal importance, and these weights does not merely result of their long-term frequencies(i.e. the corresponding component's prior probability π_m) . Some timbres (i.e. here sets of frames) are more *salient* than others : for instance, the first thing than one may notice while listening to a Phil Collins song is his voice, independently of the instrumental background (guitar, synthesizer, etc...). This saliency may depend on the context or the knowledge of the listener and is obviously involved in the assessment of similarity.

Second, cognitive evidence show that human subjects tend not to assess similarity by testing the significance of the hypothesis "this sounds like X", but rather by comparing two competing models "this sounds like X" and "this doesn't sounds like X"([19]). This also suggests that comparing the mean of the log likelihoods of all frames may not be the most realistic approach.

These two aspects, and/or others to be investigated, may explain the paradoxes observed with the current model, notably the hubs. We believe that substantial improvement of the existing measures may not result from further variations on the usual model, but rather come from a deeper understanding of the cognitive processes underlying the perception of complex polyphonic timbres and the assessment of their similarity.

VII. Acknowledgments

The work reported in this paper uses two open-source libraries, TORCH ([9]) and HTK ([34]). We also thank Anthony Beurivé for helping with the optimization of feature extraction code.

References

[1] J.-J. Aucouturier and F. Pachet. Representing musical genre: A state of art. *Journal of New Music Research (JNMR)*, 32(1), 2003.

[2] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use ? In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris (France)*, October 2002.

[3] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden markov models. In *Proceedings of the 110th Convention of the Audio Engineering Society, Amsterdam (The Netherlands)*, May 2001.

[4] S. Baumann. Music similarity analysis in a p2p environment. In *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services, London (UK)*, April 2003.

[5] S. Baumann and T. Pohle. A comparison of music similarity measures for a p2p application. In *Proceedings of the Sixth International Conference on Digital Audio Effects DAFX, London (UK)*, September 2003.

[6] A. Berenzweig, D. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *Proceedings of the AES 22nd International Conference on Virtual, Synthetic, and Entertainment Audio, Espoo (Finland).*, June 2002.

[7] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR 2003), Baltimore, Maryland (USA)*, October 2003.

[8] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford Press, 1995.

[9] R. Collobert, S. Bengio, and J. Marithoz. Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP, 2002.

[10] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Istanbul (Turkey)*, June 2000.

[11] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), New York City, NY (USA)*, July 2000.

[12] J. T. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, 1997. pages 138-147.

[13] J. Herre, E. Allamanche, and C. Ertel. How similar do songs sound? towards modeling human perception of musical similarity. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), Mohonk, NY (USA)*, October 2003.

[14] P. Herrera, X. Serra, and G. Peeters. Audio descriptors and descriptors schemes in the context of mpeg-7. In *Proceedings of the International Computer Music Conference (ICMC), Beijing (China)*, October 1999.

[15] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai. Music type classification by spectral contrast feature. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Lausanne (Switzerland)*, August 2002.

[16] J. Johnston. Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 6(2):314–323, February 1988.

[17] Y. E. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris (France)*, October October 2002.

[18] V. Kulesh, I. Sethi, and P. V. Indexing and retrieval of music via gaussian mixture models. In *Proceedings of the 3rd International Workshop on Content-Based Multimedia Indexing (CBMI), Rennes (France)*, 2003.

[19] M. D. Lee, B. Pincombre, S. Dennis, and P. Bruza. A psychological approach to text document classification. In *Proceedings of the Defence Human Factors Special Interest Group Meeting*, October October 2000.

[20] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) New Paltz, NY (USA)*, October 2003.

[21] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the 1st International Symposium of Music Information Retrieval (ISMIR), Plymouth, Massachusetts (USA)*, October 2000.

[22] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *in proceedings IEEE International Conference on Multimedia and Expo (ICME), Tokyo (Japan)*, August 2001.

[23] N. C. Maddage, C. Xu, and Y. Wang. A svm-based classification approach to musical audio. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR), Baltimore, Maryland (USA)*, October October 2003.

[24] F. Pachet, A. LaBurthe, A. Zils, and J.-J. Aucouturier. Popular music access: The sony music browser. *Journal of the American Society for Information (JASIS), Special Issue on Music Information Retrieval*, 2004.

[25] E. Pampalk, S. Dixon, and G. Widmer. On the evaluation of perceptual similarity measures for music. In *Proceedings of the Sixth International Conference on Digital Audio Effects DAFX, London (UK)*, September 2003.

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

[27] L. Rabiner and B. Juang. *Fundamentals of speech recognition.* Prentice-Hall, 1993.

[28] Y. Rubner, C. Tomasi, and L. Guibas. The earth movers distance as a metric for image retrieval. Technical report, Stanford University, 1998.

[29] R. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI), Stockholm (Sweden)*, August 1999.

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

[31] E. Voorhes and D. Harman. Overview of the eighth text retrieval conference. In *Proceedings of the Eighth Text Retrieval Conference (TREC), Gaithersburg, Maryland (USA)*, November 1999. http://trec.nist.gov.

[32] M. Welsh, N. Borisov, J. Hill, R. von Behren, and A. Woo. Querying large collections of music for similarity. Technical Report Technical Report UCB/CSD00 -1096, U.C. Berkeley Computer Science Division, 1999.

[33] B. Whitman and P. Smaragdis. Combining musical and cultural features for intelligent style detection. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris (France)*, October October 2002.

[34] S. Young. The htk hidden markov model toolkit. Technical Report TR152, Cambridge University Engineering Department CUED, 1993.

TABLE I
Composition of the test database

Artist	Description	Size
ALL SAINTS	Dance Pop	9
APHEX TWIN	Techno	4
BEATLES	British Pop	8
BEETHOVEN	Classical Romantic	5
BRYAN ADAMS	Pop Rock	8
FRANCIS CABREL	French Pop	7
CAT POWER	Indie Rock	5
CHARLIE PATTON	Delta Blues	10
THE CLASH	Punk Rock	21
VARIOUS ARTISTS	West Coast Jazz	14
DD BRIDGEWATER	Jazz Singer Trio	12
BOB DYLAN	Folk	13
ELTON JOHN	Piano Pop	5
FREHEL	French Prewar Singer	8
GARY MOORE	Blues Rock	9
GILBERTO GIL	Brazilian Pop	15
JIMI HENDRIX	Rock	7
JOAO GILBERTO	Jazz Bossa	8
JONI MITCHELL	Folk Jazz	9
KIMMO POHJONEN	World Accordion	5
MARDI GRAS BB	Big Band Blues	7
MILFORD GRAVES	Jazz Drum Solo	4
VARIOUS	“Musette” Accordion	12
PAT METHENY	Guitar Fusion	6
VARIOUS ARTISTS	Jazz Piano	15
PUBLIC ENEMY	Hardcore Rap	8
QUINCY JONES	Latin Jazz	9
RASTA BIGOUD	Reggae	7
RAY CHARLES	Jazz Singer	8
RHODA SCOTT	Organ Jazz	10
ROBERT JOHNSON	Delta Blues	14
RUN DMC	Hardcore Rap	11
FRANK SINATRA	Jazz Crooner	13
SUGAR RAY	Funk Metal	13
TAKE 6	Acapella Gospel	10
TRIO ESPERANCA	Acapella Brasilian	12
VOCAL SAMPLING	Acapella Cuban	13

TABLE VI
15 Most Frequent False Positives

Song	N_{10}	$card(C_S)$	$\frac{N_{10}}{card(C_S)}$
MITCHELL, Joni - Don Juan's Reckless Daughter	57	9	6.33
RASTA BIGOUD - Tchatche est bonne	30	7	4.23
MOORE, Gary - Separate Ways	35	9	3.88
PUBLIC ENEMY - Cold Lampin With Flavor	27	8	3.37
GILBERTO, Joao - Tin tin por tin tin	25	8	3.12
CABREL, Francis - La cabane du pcheur	22	7	3.14
MOORE, Gary - Cold Day In Hell	27	9	3.0
CABREL, Francis - Je t'aimais	20	7	2.86
MOORE, Gary - The Blues Is Alright	25	9	2.77
MARDI GRAS BIG BAND - Funkin'Up Your Mardi Gras	19	7	2.71
RASTA BIGOUD - Kana Diskan	18	7	2.57
BRIDGEWATER, DD - What Is This Thing Called Love	30	12	2.5
Frehel - A la derive	20	8	2.5
ADAMS, Bryan - She's Only Happy When She's Dancin'	20	8	2.5
MITCHELL, Joni - Talk To Me	22	9	2.44