Proc.1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002), Leuven, Belgium, November 15, 2002

FINDING SONGS THAT SOUND THE SAME

Jean-Julien Aucouturier
SONY CSL
6, rue Amyot
75005 Paris. France
+33 01 44 08 05 13
jj@csl.sony.fr

Francois Pachet
SONY CSL
6, rue Amyot
75005 Paris. France
+33 01 44 08 05 16
pachet@csl.sony.fr

ABSTRACT

A crucial dimension of Content-based music management systems is their ability to compute automatically similarities between music titles. We propose a technique that allows users to find music titles that *sound similar* to songs they like. The technique relies on a modelling of the timbral characteristics of a music signal by distributions of Cepstrum coefficients. The resulting models are then compared to yield a similarity measure. The paper describes the algorithm, and proposes an evaluation of the quality of the extracted similarity measure. Additionally, we illustrate the use of this measure in two Electronic Music Distribution applications developed in the context of the European project Cuidado.

1. INTRODUCTION

The exploding field of Electronic Music Distribution (EMD) deals with the dream of making accessible millions of music titles to millions of users. This fantasy has naturally emerged from the recent progress in digitalisation of music and compression technologies and the wide spread use of personal computers connected to the Internet.

However, this EMD dream requires more than compression and network technologies to be achieved. Faced to millions of music titles, end users need, more than ever, powerful content-based management systems to help them navigate in these huge catalogues, much as they need search engine to find web pages in the Internet.

Not only users want to find quickly music titles they already know, but they also – and more importantly – need systems that help them find titles they do not know yet but will probably like

1.1. Computing Music Similarity

Many content-based techniques have been proposed recently to help users navigate in large music catalogues. The most widely used is collaborative filtering. This technique 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 exploit these technologies or variants ([1, 2, 3]) with various degrees of success.

The main drawback of these approaches is that they are essentially content-blind; the music itself is ignored, and only users tastes are considered. The resulting recommendations are therefore at best superficially relevant.

Other content-based management techniques attempt at extracting 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 ([4]), rhythm or melodies ([5]). The resulting descriptors can be used for querying music catalogues by content information rather than by song or artist names, and as such provide a first layer to content-based music access. Query by humming is probably the most spectacular of these approaches ([6]). However, these are limited essentially by the difficulty for non-specialists to identify the right descriptors. Query by humming for instance, is largely dependent of the ability of the user to sing correctly a song. Furthermore, these techniques by construction only help users to find what they actually look for, and therefore address only a small fraction – and the easiest one - of the EMD problem.

In this paper we propose to go further in the direction of content-based extraction by computing automatically music similarities between music titles based on their global *timbral 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). 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" makes much 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 timbral quality of a music title, considered in its globality.

We therefore introduce here a measure of the similarity of the "global timbre" of music titles. For instance,

- 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.

1.2. Related work on Timbre description

There has been a large quantity of work about timbre. However most of them have focussed on monophonic simple sound samples, aiming at *Instrument Recognition* ([7]), i.e. identifying if a note is being played on a trumpet or a clarinet. Here, we are concerned with full polyphonic music and complex instrumental textures, for which we want to extract a global timbre description.

Among related work in this domain, *Automatic Genre Classification* ([8]) 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 ([9]) deals with identifying the title and artist of an arbitrary music signal. This is done by comparing the unlabelled 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 system performs approximate matching of one song to another. It uses Mel Frequency Cepstrum Coefficients, which are modelled with Gaussian Mixture models, and compared to yield a similarity measure.

In the remaining of this paper, we describe the algorithm, evaluate the quality of the measure, and give many examples of songs that are found similar by the system. We also describe two applications of this measure in the context of the European project Cuidado [10].

2. ALGORITHM

In this section, we describe the techniques used to compute the timbral similarity measure between two songs.

2.1. Feature Space

2.1.1. Requirements

We need to extract features from the music signal that we can compare in order to measure timbre similarity. Similar timbres must be represented by close "points" in a multi-dimensional feature space, and, conversely, close points in this space should correspond to similar timbres.

At the same time, since we do not want to take into account the melodic content of the songs, the feature set should be relatively independent of pitch.

2.1.2. Mel Frequency Cepstrum Coefficients

As said before, there has been a substantial amount of research on timbre and instrument recognition, in most of which the analyzed acoustic data consist of short monophonic samples of a simple instrument. In this context, it has been demonstrated that a large part of the timbre of instruments was explained by their spectral envelope ([11]). The spectral envelope of a signal is a curve in the frequency-magnitude space that "envelopes" the peaks of its short-time spectrum.

In this paper, we estimate the spectral envelope of the signal using Mel Frequency Cepstrum ([12]). The cepstrum is the inverse Fourier transform of the log-spectrum.

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

We call mel-cepstrum the cepstrum computed after a non-linear frequency warping onto a psychoacoustic frequency scale (the *Mel* scale). The c_n are called Mel Frequency Cepstrum Coefficients (MFCCs).

The low order MFCCs account for the slowly changing spectral envelope, while the higher order ones describe the fast variations of the spectrum. Therefore, to obtain a timbre measure that is independent of pitch, we only use the first few coefficients. In [13], we have measured that the optimum dimension of the set was around 10 coefficients. In this work, we shall use the first 8 coefficients.

2.1.3. Implementation

Each musical signal is cut into 2048 points frames (50ms), and for each frame, we compute the short-time spectrum. We then compute the first 8 MFCCs. In the current implementation, the processing is done in Matlab using raw audio, i.e. .wav files. However, the huge majority of music files available for analysis is compressed using the MPEG audio compression standard, which thus have to be first decompressed into wav files. One interesting possibility for speeding computation is the calculation of the MFCCs directly from the mpeg data. This idea has been proposed by Tzanetakis in [14].

2.2. Modelling

The feature extraction yields a feature vector of dimension 8 for each frame, which is believed to be a good and compact representation of the timbre of the frame. A typical 3-minute song is therefore represented with 3600 feature vectors, i.e. 30,000 coefficients, which then have to be compared with data from other songs. In order to reduce both the quantity and variability of the data to be compared, we model the distribution of each song's MFCCs as a mixture of Gaussian distributions over the space of all MFCCs.

2.2.1. The Gaussian Mixture Model

A Gaussian Mixture Model (GMM) estimates a probability density as the weighted sum of M simpler Gaussian densities, called components or states of the mixture. ([15]):

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

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 c_m is a mixture coefficient (also called state probability).
An equivalent definition is hierarchical sampling: to sample from the density, first draw a state at random (using a distribution over states) and then sample from that component.

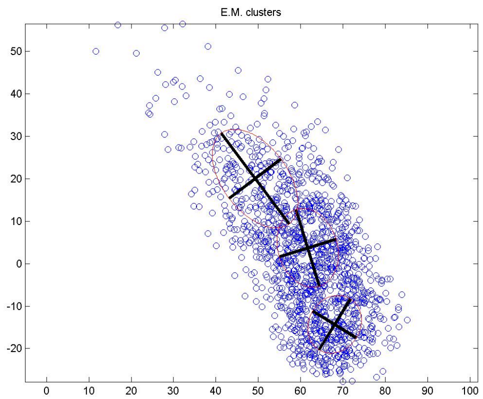

Figure 1: GMM modelling of a distribution of MFCCs

2.2.2. Implementation

We initialise the GMM's parameters by k-mean clustering, and train the model with the classic E-M algorithm ([15]). Figure 1 shows a 2D projection of a typical feature space (which is originally dimension 8). The circles represent MFCCs and the crossed ellipses are the projection of the Gaussian distributions in the trained GMM.

In this work, we use mixtures of M=3 Gaussian distributions, which have proved sufficient to model the MFCC distribution of most songs.

2.3. Distance between models

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. There are 2 ways such a distance can be computed.

2.3.1. Likelihood

One can match one song (A) against the timbre model of another song (B), by computing the "probability of the data given the model" (likelihood), i.e. computing the probability that the MFCCs of song A be generated by the model of B, using the formula given in 2.2.1. This is the most precise and logical way to compute a distance, but it requires to have access to song A's MFCC, which are relatively heavy to compute and to store.

2.3.2. Sampling

If we assume that we don't have access to the songs' MFCC when we want to compute the distance, but only to their timbre models, one can also directly match the models. It is easy to compute a distance between two Gaussian distributions (M=1), using for instance the classical Kullback-Leibler distance ([15]):

$$4D_{i,j} = tr(\Gamma_i\Gamma_j^{-1} - \Gamma_j\Gamma_i^{-1}) + (\mu_i - \mu_j)^T(\Gamma_j^{-1} - \Gamma_i^{-1})(\mu_i - \mu_j),$$

given here for 2 multi-dimensional Gaussian distributions, of mean vectors μ_1 and μ_2, and covariance matrices Γ_1 and Γ_2, and where tr(A) is the trace of matrix A, and T is the transposition operator.

However, it is a trickier problem to evaluate a distance between two *sets* of Gaussian Distributions, like in a GMM (M>1). The method we have chosen in this work is to sample from one GMM, and to compute the likelihood of the samples given the other GMM. This corresponds roughly to re-creating a song from its timbre model, and applying the likelihood method defined above to this newly created song and the other song's model.

The precision of this method obviously depends on the number of samples that are generated from the GMM. To fine-tune this "sampling rate", we have conducted a stability analysis. Figure 2 shows the standard deviation of the distance between two songs against the number of samples used in the distance computation. 100 distances are computed for each duplet of songs, and for each sample rate. We also average over 100 different duplets of songs. The curve has an asymptotic behaviour, and suggests that the limit point for good performance is about 1000 samples for a GMM with M=3.

2.3.3. Normalization

Both methods yield distances that are not symmetric:

$$D(i,j) \neq D(j,i).$$

Therefore, we force the symmetry by computing:

$$D_{sym}(i,j) = \frac{1}{2} \times (D(i,j) + D(j,i))$$

Also, the sampling method may yield a non-zero distance from one song to itself (notably when the sampling rate is too low). To obtain a distance between 0 and 1, we therefore normalize the distance to:

$$D_{norm} = \frac{\left\{ D_{sym} - \frac{1}{2} \times (D(i,i) + D(j,j)) \right\}}{\max_{i,j}(D(i,j))}$$

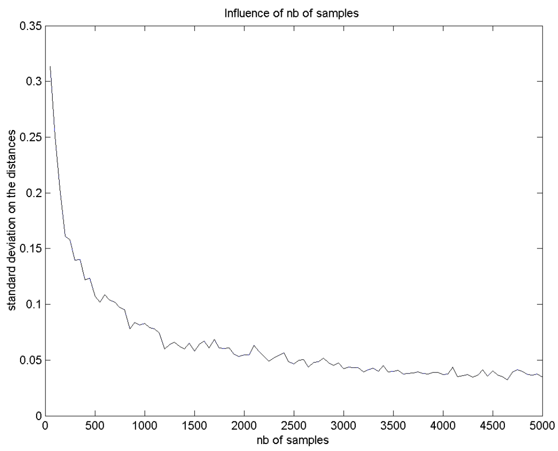

Figure 2: Influence of the sample rate in the "sampled" distance between two GMMs

2.4. Database Integration

2.4.1. Offline learning and fast distance computation

One great advantage of our method is that it is well suited to large musical databases. The most intensive parts of the process are the computation of MFCCs for each song (possibly including the decoding from mp3 to .wav), and the modelling of the MFCC distribution with a GMM, with the iterative EM algorithm. These steps need to be done only once for each song, and can be done offline. The whole process in our current, non-optimized implementation takes about 1 minute per song.

As described in 2.3.2, the MFCCS themselves need not be stored. Only the parameters of the GMM (or "timbre model") of each song are stored in a metadata database. In dimension 8, each Gaussian distribution in the GMM is represented with 17 floating-point numbers (1 mixture coefficient, 8 coefficients for the mean vector, and 8 coefficients for the covariance matrix, which is assumed to be diagonal). These can be easily stored, and quickly accessed in a relational database. In our current implementation, computing 10,000 distances to one song takes about 30 seconds.

2.4.2. Pre-computation

For applications that require even faster distance calculation (see for instance section 4.2), the distances between all songs in the database can be pre-computed and stored in a similarity matrix. This currently takes between a few hours and a few days to process a 10,000-song database, but then the distances can be accessed in a few milliseconds. Specific database issues arise about how to efficiently store and index such very large sparse matrices (order of 100 million entries), which are not dealt with in this paper.

3. RESULTS

Experiments were performed in the context of the Cuidado European IST project ([10]). In this project we have setup a database of 17,075 popular music titles, together with metadata extracted automatically through different techniques. Metadata include information about artists, genres, tempo, energy, … and the herein discussed timbre models.

3.1. Examples

Here we give some examples of duplets (or n-plets) of songs that are found similar by our system, i.e. whose timbre models are closely matched one to another. Many more examples can be found on the project web page ([16]).

3.1.1. Same songs

As a benchmark, it is interesting to note that duplicates of a same song (i.e. different mp3 encoding, different radio broadcasting…) are always closely matched. This echoes the work done on music title identification mentioned in the introduction.

3.1.2. Same artist

There are many examples of songs by the same artist that are closely matched by our system (however see 3.2 for a discussion about this).

- Piano pieces: *Franz Schubert Op90- No2 in E flat major* and *Franz Schubert Op90- No4 in A flat major*
- Harpsichord pieces: *Bach - Wohltemperierte Clavier - Fuga II in C minor* and *Bach - Wohltemperierte Clavier - Praeludium IV in C sharp minor*
- Heavy guitar overload: *Therapy - Brainsaw* and *Therapy - Stop it you're killing me*

- Trip Hop: *Portishead - Mysterons (live)* and *Portishead - Sour Times*
- Orchestral Textures: *Wagner - Ride of the Valkyries, Wagner - Solti - Brunheild*

3.1.3. Same Genre

These similar songs have different artists, but show some kind of genre/style similarity (whatever this means, as music genre is a rather ill-defined concept). Here are some typical examples:

- Piano pieces: *Scriabin - Sonate pour Piano no 2, Mozart -Sonate pour Piano KV 533-1* and *Weber - Sonate pour Piano opus 49 no 3*
- Harpsichord pieces: *Bach - Das Wohltemperierte Clavier - Praeludium IV in C sharp minor BWV849* and *Couperin - Gavotte*
- “Power Rock": *Therapy - Brainsaw, Skunk Anansie - Intellectualise My Blackness, Nirvana - Smells Like Teen Spirit.*
- "Acoustic Guitar Folk": *Nick Drake - From the Morning, Spain - Hoped and prayed, Belle & Sebastian - Is It Wicked Not to Care*, and *Smashing Pumpkins - Landslide*
- "Woman Rock Singer": *Leah Andreone - It's OK* and *Meredith Brooks - Bitch*

3.1.4. “Interesting” results

The following similarities found by the system are rather unexpected but much more interesting: the songs have different artists or genres, but also different dates of production, different cultural backgrounds, etc.
These surprising associations constitute the really interesting results, since this kind of similarity cannot be assessed by a non-signal method, contrary to artist and genre similarity.

- Piano music:
  - "Classical” and “Contemporary”: *Rachmaninov - Lugansky - Moment Musical opus 16 no 2, Gyorgy Ligeti - Concerto for Piano and Orchestra.*
  - "Classical” and “Jazz": *Schumann - Horowitz - Kreisleriana, Op 16-5 (sehr langsam)* and *Bill Evans - I loves you Porgy*
- Orchestral textures:
  - "Jazz” and “Classical": *Orchestre Symponique de Montreux - Porgy and Bess* and *Prokofiev - Celibidache - Symphonie no 5-1 opus 100.*
  - “Classical” and “Pop": *Beethoven - Romanze fur Violine und Orchester Nr. 2 F-dur op.50* and *Beatles - Eleanor Rigby*
  - "Classical” and “Musicals": *Beethoven - Romanze fur Violine und Orchester Nr. 2 F-dur op.50* and *Gene Kelly – Singin’ in the rain*
- "Trip Hop" and "Celtic Folk ": *Portishead - Mysterons* and *Alan Stivell - Arvor You*. (same kind of harpy theremin-like ambiance)

These associations provoke an exciting feeling of “discovery”, comparable to the one that one gets when recognizing the origin of a sampled bit in a contemporary song, e.g. Stevie Wonder sampled in a hip-hop tune.

The feeling users have when they gain a sudden insight into previously puzzling phenomena is studied by cognitive scientists under the name of “Aha !”. We believe that our technique is able to create such musical “Aha”. The previous examples, and many more, can be heard on the project’s web page ([16]).

3.2. Objective Evaluation

The objective evaluation of the “quality” of our timbral measure is problematic. In the framework of Cuidado, each song is associated with textual metadata, and we could imagine comparing the timbre similarity against a textual similarity of artist or genre. However, this approach is not relevant for two reasons:

3.2.1. Poor correlation with artist or genre

As illustrated in the preceding section, two songs of the same artist or same genre do not necessarily have close timbres.
Fir instance:

- two songs by The Beatles: "*Helter Skelter*" (heavy overloaded guitars), and "*Lucy in the Sky*" (tremolo organ)
- two jazz pieces: "*Ascension*" by John Coltrane (free jazz saxophone), and "*My Funny Valentine*" sung by Chet Baker, etc.

We have conducted a quantitative study of the correlation between timbre and artist/genre similarity in the Cuidado database. This study shows that such examples are not exceptions, but rather are as numerous as examples of the opposite case. The correlation depends on the artist or the genre: some artists/genres are more “coherent” than others, e.g. pre-war blues guitarists are more “homogeneous” than *The Beatles.* Consequently, it is hard to base an objective evaluation on these criteria.

3.2.2. Wrong criteria for interestingness

Moreover, we have shown in 3.1.4 that the really interesting results are precisely the ones that are not correlated with textual metadata such as artist or genre. With such an objective evaluation, the distance that yields the most interesting results would be marked very poorly. In [17], the authors comment further on this and propose a measure of the "interestingness" of the results by comparing a priori and a posteriori similarities between songs. For instance, duplets of songs which have a very low a priori similarity (e.g. songs of very different genres) and yet a very high timbral "a posteriori" similarity are evaluated as very interesting.

3.3. Subjective Evaluation

Given the difficulty of an objective evaluation of the quality of our timbre distance, we have conducted a limited subjective evaluation. Early experiments done in our group on the subject of musical descriptors have shown that deciding whether two songs are "similar" can be uncertain, as it is an ill-defined concept. In particular, it is difficult to evaluate similarity based on one attribute (here *timbre* similarity), because our judgment is simultaneously influenced by other attributes (same tempo, same artist, totally different genre…).

To avoid asking users the "absolute" question whether two songs are similar, we have set up a "relative" test: users are presented a target song S, and two test songs A and B, and have to decide which test song A or B is the closest to S. We then compare this ordering with the $D(S,A)$ and $D(S,B)$. The average result of the test on 10 users is that about 80% of the songs are well ordered by our system. We are now considering larger scale user-tests in the context of Cuidado.

4. APPLICATIONS

The European project Cuidado (*Content-based Unified Interfaces and Descriptors for Audio and Music Databases available Online*) tackles the problems of information overload and the inability to quickly browse audio or search for similarities among sounds. One of its pilot applications, the Music Browser, is a client-server application for Electronic Music Distribution back offices and Internet music portals. Our timbre matching technology has been integrated into the Browser, and we describe here two of its applications: nearest neighbor search, and automatic playlist generation.

4.1. Nearest Neighbor Search

Nearest neighbor search may be seen as an answer to the following problem: "I like this song. Find me other songs that sound the same". The user selects one song "he likes" in a list of songs (e.g. out of the 17,075 songs in the cuidado database), and the system finds out the n closest songs according to the timbre distance. The query can be further filtered by asking only for songs by the same/different artist, or same/different genre.

The system scours the whole database, and therefore often comes up with interesting suggestions: unknown artists, surprising "aha!". Figure 3 shows a screenshot of this application. The query was "*Therapy- Brainsaw*", and the result lists contains songs of many genres, which all contain some kind of "metal-style" electric guitar: Punk Rock (*The Clash*), Metal *(Metallica*, *Therapy*), Hard Rock (*Aerosmith*), Pop (*Pat Benatar*, *The Beatles*), Blues (*Johnny Winter*), Funk (*FFF*), etc.

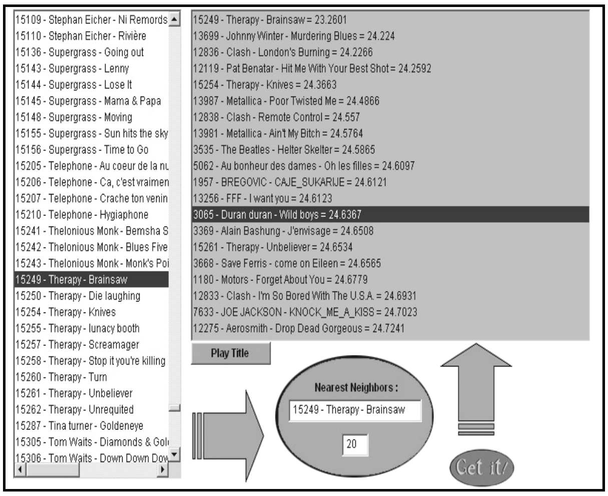

Figure 3: Screenshot of the nearest neighbor application

4.2. Playlist Generation

Extending on the notion of neighboring search (A $\leftrightarrow$ B), we can use our similarity measure to build a continuous path of songs (A $\leftrightarrow$ B $\leftrightarrow$ C $\leftrightarrow$ …). This is useful to build automatically customized radio programs, thereby extending the system of [2] with real content- based analysis.
Furthermore, we can combine timbral continuity with other constraints on the playlist as we have proposed in [18].
For instance, a music playlist can be generated from the following constraints:

- AllDifferent: the playlist should contain 12 different titles,
- Global duration: the playlist should not last more than 76 minutes,
- Cardinality: the playlist should contain at least 60% of "rock" titles,
- Progression: the sequence should contain titles with increasing tempo,
- Distribution: two titles by the same artists should be separated by at least 3 titles, etc.

The Cuidado Music Browser is able to generate such playlists automatically, using a fast algorithm based on adaptive search, and described in [19]. We have now extended the constraint library with three new constraints holding on timbre:

- Timbre Continuity: the playlist should be timbrally homogeneous, and shouldn't contain abrupt changes of textures.
- Timbre Cardinality: the playlist should contain 60 % of pieces that sound like "The Beatles - Yesterday".
- Timbre Distribution: pieces with the same timbre should be as separated as possible ("so I don't get bored"), etc.

We give here an example of a 10-title playlist with the following constraints:

1- Timbre continuity throughout the sequence
2- Genre Cardinality: 30% Rock, 30% Folk, 30%Pop
3- Genre Distribution: the titles of the same genre should be as separated as possible

One solution found by the system is the following playlist:

- `Arlo Guthrie – City Of New Orleans - Genre = Folk/Rock`
- `Belle & Sebastien - The boy done wrong again - Genre = Rock/Alternatif`
- `Ben Harper – Pleasure & Pain - Genre = Pop/Blues`
- `Joni Mitchell - Borderline - Genre = Folk/Pop`
- `Badly Drawn Boy - Camping Next to Water - Genre = Rock/Alternatif`
- `Rolling Stones – You Can't always get what you want - Genre = Pop/Blues`
- `Nick Drake - One of these things first - Genre = Folk/Pop`
- `Radiohead - Motion Picture Soundtrack - Genre = Rock/Brit`
- `The Beatles - Mother Nature's Son - Genre = Pop/Brit`
- `Tracy Chapman - Talkin' about a Revolution - Genre = Rock/Folk`

It is easy to check that the genre cardinality is correct (3 "folk", 3 "pop", 4 "rock"), and the genre distribution constraint is also well satisfied.
One can see that the system has also managed to maintain the timbre continuity by selecting the right subgenres ("Folk/Rock" and "Rock/Folk"), and picking songs which mainly consist of acoustic guitar + voice (Nick Drake, Ben Harper, Tracy Chapman, etc.).

Figure 4 shows a screenshot of the playlist generation system.

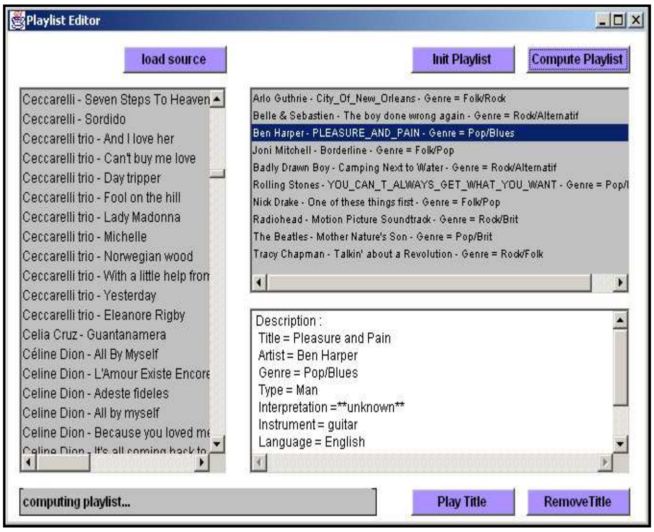

Figure 4. Screenshot of the playlist generation system with constraints on timbre continuity

These examples show that our technique does produce relevant and interesting music similarities, as the reader can assess himself. These similarities are clearly unreachable with Collaborative Filtering techniques, because they are based on an analysis of the actual musical content, rather than on an a posteriori analysis of user profiles.

5. CONCLUSION

In this paper, we have presented a measure of the timbre similarity of polyphonic music pieces, based on the extraction of cepstral coefficients, and on their modelling with Gaussian mixture models. We have discussed the integration of these techniques in some applications in the European project Cuidado [10], notably for automatic playlist generation. The results show that the distance is perceptually relevant, and yields interesting, non-trivial musical similarities. A precise comparison with Collaborative Filtering techniques is under study, however it is already clear that these two approaches are complementary. The applications made possible by this technique can be seen as the first instances of a real content-based EMD system.

6. REFERENCES

[1] French, J. C. and Hauver, D. B. Flycasting: On the Fly Broadcasting, in Proc. WedelMusic Conference, Firenze, Italy,2001

[2] Pestoni, F. et al. Karc: Radio Research, in Proc. WedelMusic Conference, Firenze, Italy,2001

[3] Pachet, F. et al. Musical Datamining for EMD, in Proc. WedelMusic Conference, Firenze, Italy,2001

[4] Scheirer, E. Music-Listening Systems, PhD Thesis, MIT Cambridge, MA USA 2000

[5] Klapuri, A. et al. Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals, in Proc. DAFX-00, Verona, Italy, 2000.

[6] Ghias, A. et al. Query By Humming -- Large Musical Information Retrieval in An Audio Database, Proc. of the Third ACM International Conference on Multimedia, San Francisco, CA, 1995

[7] Herrera-Boyer P., et al. Towards Instrument Segmentation for Music Content Description: a Critical Review of Instrument Classification Techniques, in proc. ISMIR 2000.

[8] Tzanetakis, G. Automatic Musical Genre Classification of Audio Signals, in proc. ISMIR 2001.

[9] Allamanche, E. Content-based Identification of Audio Material Using MPEG-7 Low Level Description, in proc. ISMIR 2001

[10] Pachet, F. Metadata for music and sounds: The Cuidado Project, in proc. of the CBMI Workshop, University of Brescia, September 2001.

[11] Schwarz, D. & Rodet, X. Spectral Estimation and Representation for Sound Analysis-Synthesis. Proceedings ICMC 1999

[12] Rabiner, L.R. & Juang, B.H., Fundamentals of speech recognition. Prentice-Hall 1993

[13] Aucouturier, J.J. and Sandler, M. Segmentation of Musical Signals Using Hidden Markov Models, in Proc. 110th Convention of the AES, Amsterdam 2001

[14] Tzanetakis, G. and Cook, P. Sound Analysis using Mpeg Compressed Audio. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Istanbul, 2000

[15] Bishop, C. Neural Networks for pattern recognition, Oxford University Press

[16] Project Web Page: www.csl.sony.fr/~jj/Timbre/timbre.html

[17] Aucouturier, J.-J. & Pachet, F. Music Similarity Measures: What's The Use ?. In proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR) 2002 Paris, France.

[18] Pachet, F. et al. A Combinatorial approach to content-based music selection, in proc. IEEE International Conference on Multimedia Computing and Systems, Firenze (It), 1999

[19] Aucouturier, J-J. & Pachet, F. Scaling up Playlist Generation Systems, submitted to IEEE International Conference on Multimedia and Expo, Lausanne 2002.