Toward Automatic Music Audio Summary Generation from Signal Analysis

Geoffroy Peeters
IRCAM
Analysis/Synthesis Team
1, pl. Igor Stravinsky
F-75004 Paris - France
peeters@ircam.fr

Amaury La Burthe
IRCAM
Analysis/Synthesis Team
1, pl. Igor Stravinsky
F-75004 Paris - France
laburthe@ircam.fr

Xavier Rodet
IRCAM
Analysis/Synthesis Team
1, pl. Igor Stravinsky
F-75004 Paris - France
rod@ircam.fr

ABSTRACT

This paper deals with the automatic generation of music audio summaries from signal analysis without the use of any other information. The strategy employed here is to consider the audio signal as a succession of "states" (at various scales) corresponding to the structure (at various scales) of a piece of music. This is, of course, only applicable to certain kinds of musical genres based on some kind of repetition.

From the audio signal, we first derive dynamic features representing the time evolution of the energy content in various frequency bands. These features constitute our observations from which we derive a representation of the music in terms of "states". Since human segmentation and grouping performs better upon subsequent hearings, this "natural" approach is followed here. The first pass of the proposed algorithm uses segmentation in order to create "templates". The second pass uses these templates in order to propose a structure of the music using unsupervised learning methods (K-means and hidden Markov model).

The audio summary is finally constructed by choosing a representative example of each state. Further refinements of the summary audio signal construction, uses overlap-add, and a tempo detection/ beat alignment in order to improve the audio quality of the created summary.

1. INTRODUCTION

Music summary generation is a recent topic of interest driven by both commercial needs (browsing of online music catalogues), documentation (browsing over archives) as well as music information retrieval (understanding musical structures). As a significant factor resulting from this interest, the recent MPEG-7 standard (Multimedia Content Description Interface) [10], proposes a set of meta-data in order to store multimedia summaries: the Summary Description Scheme (DS). This Summary DS provides a complete set of tools allowing the storage of either sequential or hierarchical summaries.

However, while the storage of audio summaries has been normalized, few techniques exist allowing their automatic generation. This is in contrast with video and text where numerous methods and approaches exist for the automatic summary generation. Most of them assess that the summary can be parameterized at three levels [8]:

The type of the source (in the case of music: the musical genre) to be summarized. In this study, we are addressing music audio summary without any prior knowledge of the "music". Hence, we will only use the audio signal itself and information which can be extracted from it.

The goal of the summary The goal is not a priori determined. A documentalist and a composer for example do not require the same information. We therefore need to get the "music" structure, to be able to select which type of information we want for the summary. It is important to note that the "perfect" summary does not exist since it at least depends directly on the type of information sought.

The output format It consists mainly of an audio excerpt. Additional information can also be provided as is the case in the realm of video where many techniques [1, 5, 13] propose additional information, by means of pictures, drawings, visual summary, etc ... The same is feasible in audio by highlighting, for example, parts of the signal or its similarity matrix [7] in order to locate the audio excerpt in the piece of music.

2. AUTOMATIC AUDIO SUMMARY GENERATION

Various strategies can be envisioned in order to create an audio summary: time-compressed signal, transient parts signal (highly informative), steady parts signal (highly representative), symbolic representation (score, midi file, etc ...). Our method is based on deriving musical structures directly from signal analysis without going into symbolic representations (pitch, chords, score, ...). The structures are then used in order to create an audio summary by choosing either transient or steady parts of the music. The choice of this method is based on robustness and generality (despite it is restricted to certain kind of musical genre based on repetition) of the method.

2.1 State of the art

Few studies exist concerning the Automatic Music Audio Summary Generation from signal analysis. The existing ones can be divided into two types of approache.

2.1.1 "Sequences" approach

Most of them start from Foote's works on **similarity matrix**. Foote showed in [7] that a similarity matrix applied to well-chosen features allows a visual representation of the structural information of a piece of music. The signal's features used in his study are the Mel Frequency Cepstral Coefficients (MFCC) which are very popular in the ASR community. The similarity $s(t_1, t_2)$ of the feature vectors at time t_1 and t_2 can be defined in several ways: Euclidean, cosine, Kullback-Leibler distance, ... The similarity of the feature vectors over the whole piece of music is defined as a similarity matrix $\underline{S} = [s(t_i, t_j)]$ $i, j = 1, ..., I$. Since the distance is symmetric, the similarity matrix is also symmetric. If a specific segment of music ranging from times t_1 to t_2 is repeated later in the music from t_3 to t_4, the succession of feature vectors between $[t_1, t_2]$ is supposed to be identical (close to) the ones between $[t_3, t_4]$. This is represented visually by a lower (upper) **diagonal** in the similarity matrix. An example of a similarity matrix estimated on a popular music song (Moby "Natural Blues") is represented in Figure 1 [top]. The first 100 s of the music are represented. In this figure, we see the repetition of the sequence $t = [0 : 18]$ at $t = [18 : 36]$, the same is true for $t = [53 : 62]$ which is repeated at $t = [62 : 71]$. Most of works on Automatic Music Audio Summary Generation starts

from this similarity matrix using either MFCC parameterization [3], pith or chromagram [4] features. They then try to detect the lower (upper) diagonals in the matrix using various algorithms, and to find the most representative or the longest diagonals.

2.1.2 "States" approach

A study from Compaq [9] also uses this MFCC parameterization in order to create "key-phrases". In this study, the search is not for lower (upper) diagonal (succession of events) but for states (collection of similar and contiguous states). The song is first divided into fixed length segments which are then grouped according to a cross-entropy measure. The longest example of the most frequent episode constitutes the "key-phrase" used for the summary. Another method proposed by [9], close to the method proposed by [2], is based on the direct use of a hidden Markov model applied to the MFCC. While temporal and contiguity notions are present in this last method, poor results are reported by the authors.

2.1.3 Conclusion

One of the key points of all these works stands in the use of static features (MFCC, pitch, chromagram) as signal observation. **'Static"** features represent the signal around a given time, but does not model any temporal evolution. This implies, when looking for repeated patterns in the music, the necessity to find identical evolution of the features (through the search of "diagonals" in the similarity matrix), or the necessity to averages features over a period of time in order to get states.

3. EXTRACTION OF INFORMATION FROM THE SIGNAL

The choice of signal features used for similarity matrix or summary generation plays an essential role in the obtained result. In our approach, the features used are **"dynamic"**, i.e. they model directly the temporal evolution of the spectral shape over a fixed time duration. The choice of the duration on which the modeling is performed, determines the kind of information that we will be able to derive from signal analysis.

This is illustrated on Figure 1 for the same popular music song (Moby "Natural Blues") as before. On Figure 1 [middle], a short duration modeling is performed which allows deriving sequence repetition through upper (lower) diagonals. Compared to the results obtained using MFCC parameterization (Figure 1 [top]), we see that the melody sequence $t = [0 : 18]$ is in fact repeated not only at $t = [18 : 36]$ but also at $t = [36 : 54]$, $t = [71 : 89]$, ... This was not visible using the MFCC because at time $t = 36$ the arrangement of the music changes which masks the repetition of the initial melody sequence. Note that the features sample rate used here is only 4 Hz (compared to 100 Hz for the MFCC). On Figure 1 [bottom], a long duration modeling is used in order to derive the structure of the music such as introduction/verse/chorus/... In this case, the whole music (250 s) is represented. Note that the features sample rate used here is only 1 Hz.

In Figure 2, we show another example of the use of dynamic features on the title "Smells like teen spirit" from artist Nirvana. The [top] panel shows the similarity matrix obtained using MFCC features. The [middle] panel shows the same using dynamic features with a short duration modeling. We see the repetition of the guitar part (at $t = 25$ and $t = 30$), the repetition of the verse melody (at $t = 34$ and $t = 42$), the bridge, then the repetion of the chorus melody (at $t = 67$, $t = 74$, $t = 82$) and finally the break at $t = 91$. The [bottom] panel, illustrates the use of a long duration modeling for structure representation.

Several advantages come from the use of dynamic features: 1) for an appropriate choice of the modeling's time duration, the search for repeated patterns in the music can be far easier, 2) the amount of data (and therefore also the size of the similarity matrix) can be

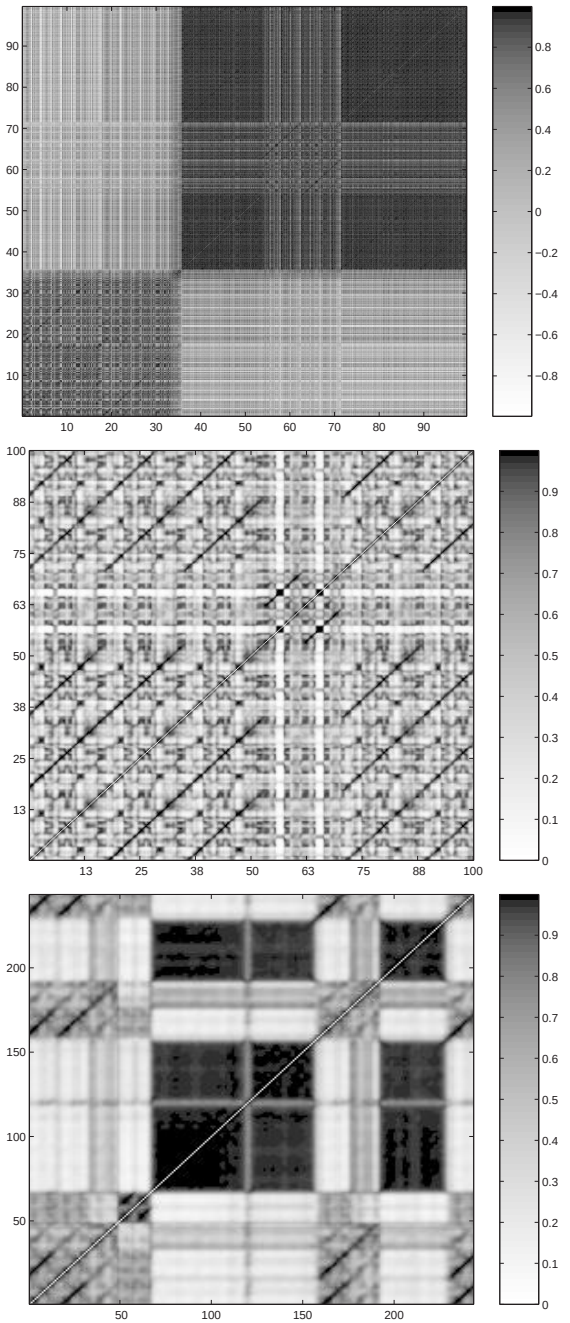

Figure 1: Similarity matrix computed using [top] MFCC features, [middle] Dynamic features with short duration modeling, [bottom] Dynamic features with long duration modeling, on title "Natural Blues" from artist Moby

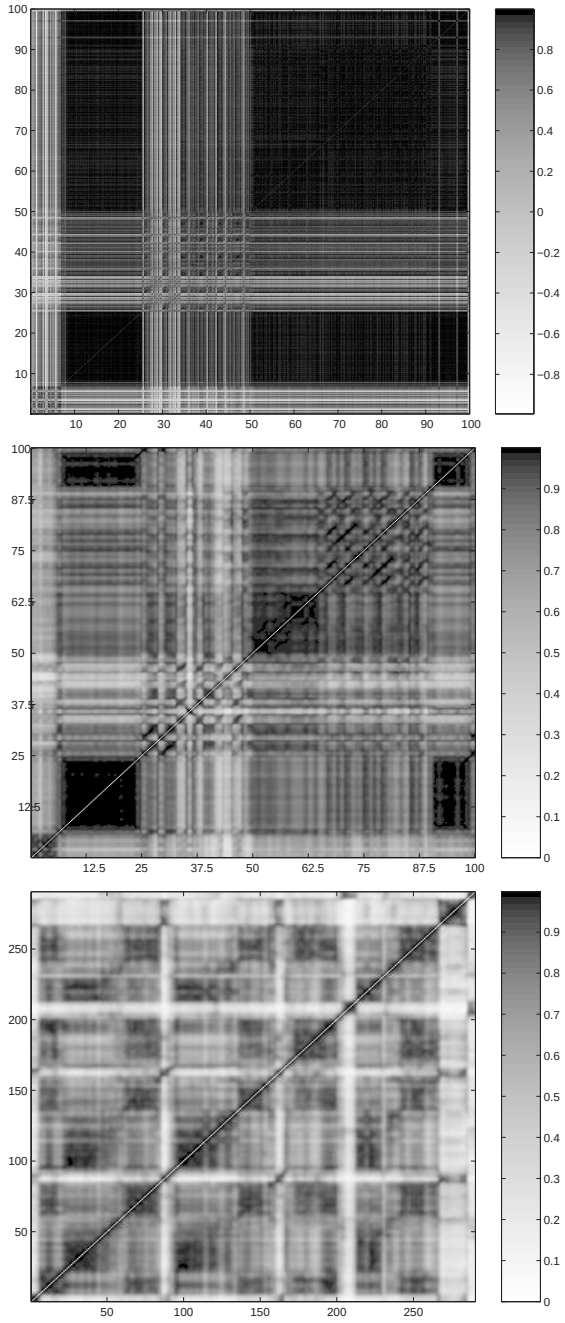

Figure 2: Similarity matrix computed using [top] MFCC features, [middle] Dynamic features with short duration modeling, [bottom] Dynamic features with long duration modeling, on title "Smells like teen spirit" from artist Nirvana

greatly reduced: for a 4 minute long music, the size of the similarity matrix is around 24000*24000 in the case of the MFCC, it can be only 240*240 in the case of the "dynamic" features.

In the following, we will concentrate on the use of dynamic features for structural representation. Since the information derived from signal analysis is supposed to allow the best differentiation of the various structures of a piece of music, signal features have been selected from a wide set of features by training the system on a large hand-labeled database of various musical genres. The features selected are the ones which maximize the mutual information between 1) feature values and 2) manually entered structures (supervised learning).

The selected signal features, which are also used for a music **"fingerprint"** application which we have developed [14], represent the variation of the signal energy in different frequency bands. For this, the audio signal $x(t)$ is passed through a bank of N Mel filters. The evolution of each output signal $x_n(t)$ of the $n \in N$ filters is then analyzed by Short Time Fourier Transform (STFT), noted $X_{n,t}(\omega)$. The window size L used for this STFT analysis of $x_n(t)$ determines the kind of structure (short term or long term) that we will be able to derive from signal analysis. Only the coefficients (n, ω) which maximize the Mutual Information are kept. The feature extraction process is represented in Figure 3. These features constitute the observations from which we derive a state representation of the music.

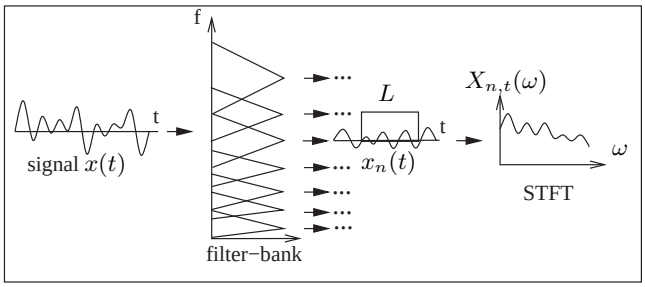

Figure 3: Features extraction from signal. From left to right: signal, filter bank, output signal of each filter, STFT of the output signals

4. REPRESENTATION BY STATES: A MULTI-PASS APPROACH

The summary we consider here is based on the representation of the musical piece as a succession of states (possibly at different temporal scales) so that each state represents a (somehow) similar information found in different parts of the piece. The information is constituted here by the dynamic features (possibly at different temporal scale L) derived from signal analysis.

The states we are looking for are of course specific for each piece of music. Therefore no supervised learning is possible. We therefore employ unsupervised learning algorithms to find out the states as classes.

Several drawbacks of unsupervised learning algorithms must be considered:

- usually a previous knowledge of the **number of classes** is required for these algorithms
- these algorithms depends on a good **initialization of the classes**
- most of the time, these algorithms do not take into account contiguity (spatial or temporal) of the observations.

A new trend in video summary is the **"multi-pass" approach** [15]. As for video, human segmentation and grouping performs better when listening (watching in video) to something for the second time [6]. A similar approach is followed here.

- The first listening allows the detection of variations in the music without knowing if a specific part will be repeated later. In our algorithm the first pass performs a signal segmentation which allows the definition of a set of templates (classes) of the music [see part 4.1].
- The second listening allows one to find the structure of the piece by using the previously mentally created templates. In our algorithm the second pass uses the templates (classes) in order to define the music structure [see part 4.2]. The second pass operates in three stage: 1) the templates are compared in order to reduce redundancies [see part 4.2.1], 2) the reduced set of templates is used as initialization for a K-means algorithm (knowing the number of states and having a good initialization) [see part 4.2.2], 3) the output states of the K-means algorithm are used for the initialization of a hidden Markov model learning [see part 4.2.3]. Finally, the optimal representation of the piece as a HMM state sequence is obtained by application of the Viterbi algorithm.

This multi-pass approach allows solving most of the unsupervised algorithm's problems. The global flowchart is depicted into Figure 4.

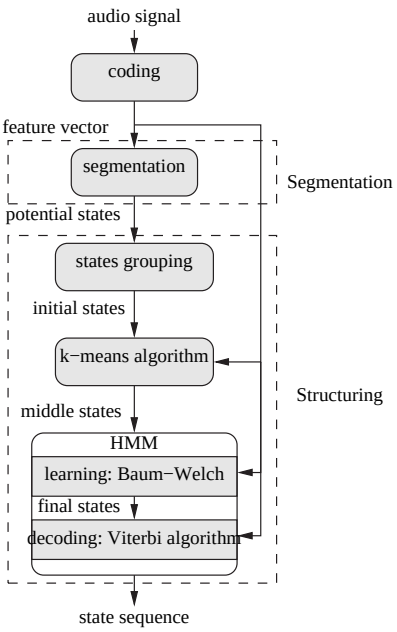

Figure 4: States representation flowchart

4.1 First pass: segmentation

From the signal analysis of part 3, the piece of music is represented by a set of feature vectors $\underline{f}(t)$ computed at regular time instants.

The upper and lower diagonals of the similarity matrix $\underline{\underline{S}}$ of $\underline{f}(t)$ (see Figure 5 [top]) represent the frame to frame similarity of the features vector. Therefore it is used to detect large and fast changes in the signal content and segment it accordingly (see Figure 5 [middle]).

A high threshold (similarity ≤ 0.99) is used for the segmentation in order to reduce the "slow variation" effect. The signal inside each segment is thus supposed to vary little or to vary very slowly. We use the values of $\underline{f}(t)$ inside each segment to define "potential" states $\underline{s}_k$. A "potential" state $\underline{s}_k$ is defined as the mean value of the features vectors $\underline{f}(t)$ over the duration of the segment k (see Figure 5 bottom panel).

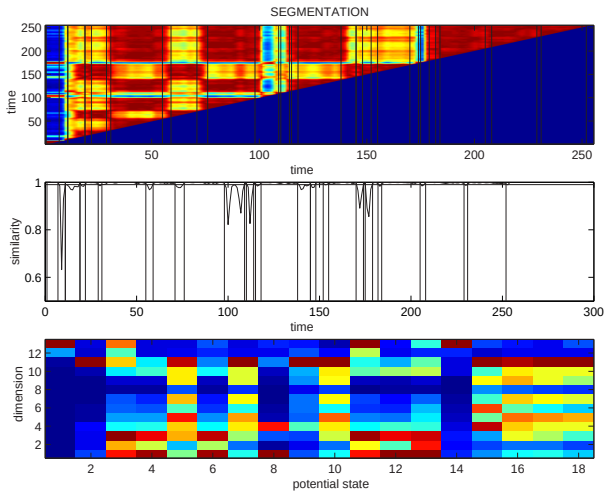

Figure 5: Feature vectors segmentation and "potential" states creation [top:] similarity matrix of signal features vectors [middle:] segmentation based on frame to frame similarity [bottom:] "potential" states found by the segmentation algorithm

4.2 Second pass: structuring

The second pass operates in three steps:

4.2.1 Grouping or "potential" state reduction

The potential states found in [4.1] constitute templates. A simple idea in order to structure the music would be to compute the similarity between them and derive from this the structure (similarity between values should mean repetition of the segment over the music).

However, we should insist on the fact that the segments were defined as the period of time between boundaries defined as large and fast variations of the signal. Since the "potential" states $\underline{s}_k$ are defined as the mean value over the segments, if the signal vary slowly inside a segment, the potential states may not be representative of the segment's content. Therefore no direct comparison is possible.

Instead of that, the "potential" states have been computed in order to facilitate the initialization of the unsupervised learning algorithm since it provides 1) an estimation of the number of states and 2) a "better than random" initialization of it. Before doing that, we need to group nearly identical (similarity ≥ 0.99) "potential" states. After grouping, the number of states is now K and are called "initial" states. This grouping process is illustrated in Figure 6.

4.2.2 K-means algorithm

K-means is an un-supervised classification algorithm which allows at the same time to estimate class parameters [1] and to assign each observation $\underline{f}(t)$ to a class. The K-means algorithm operates in an iterative way by maximizing at each iteration the ratio of the between-class inertia to the total inertia. It is a sub-optimal algorithm since it strongly depends on a good initialization. The inputs of the algorithm are 1) the number of classes, given in our case by the segmentation/grouping step and 2) states initialization, also given by the segmentation/grouping step.

K-means algorithm used:
Let us note K the number of required classes.

[1] In usual K-means algorithm, a class is defined by its gravity centre.

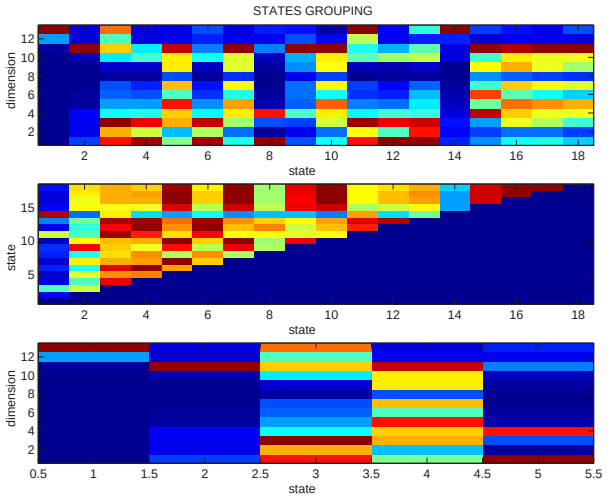

Figure 6: "Potential" states grouping [top:] potential states [middle:] similarity matrix of potential states features vectors [bottom:] "initial" states features vectors

1. Initialization: each class is defined by a "potential" state $\underline{s}_k$
2. Loop: assign the observation $\underline{f}(t)$ to the closest class (according to an Euclidean, cosine or Kullback-Leibler distance),
3. Loop: update the definition of each class by taking the mean value of the observation $\underline{f}(t)$ belonging to each class
4. loop to point 2.

We note $\underline{s}'_k$ the states definition obtained at the end of the algorithm and call them "middle" states.

4.2.3 *Introducing time constraints: hidden Markov model*

Music has a specific nature, it is not just a set of events but a specific temporal succession of events. So far, this specific nature has not been taken into account since the K-means algorithm just associates observations $\underline{f}(t)$ to states $\underline{s}'_k$ without taking into account their temporal ordering. Several refinement of the K-means algorithm have been proposed in order to take contiguity (spatial or temporal) constraints into account. But we found more appropriate to formulate this constraint using a Markov Model approach. Since we only observe $\underline{f}(t)$ and not directly the states of the network, we are in the case of a hidden Markov model (HMM) [11].

Hidden Markov model formulation: A state k produces observations $\underline{f}(t)$ represented by a state observation probability $p(\underline{f}|k)$. The state observation probability $p(\underline{f}|k)$ is chosen as a gaussian pdf $g(\mu_k, \sigma_k)$. A state k is connected to other states j by state transition probabilities $p(k, j)$.

Since no priori training on a labeled database is possible we are in the case of ergodic HMM.

The resulting model is represented in Figure 7.

Training: The learning of the HMM model is initialized using the K-means "middle" states $\underline{s}'_k$. The Baum-Welch algorithm is used in order to train the model. The outputs of the training are the state observation probabilities, the state transition probabilities and the initial state distribution.

Decoding: The state sequence corresponding to the piece of music is obtained by decoding using Viterbi algorithm given the hidden Markov model and the signal feature vectors $\underline{f}(t)$.

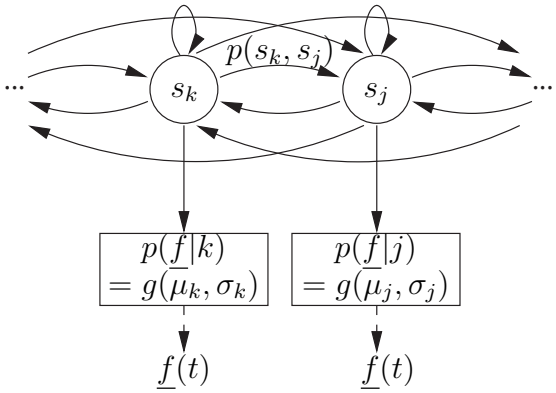

Figure 7: Hidden Markov model

4.2.4 *Results:*

The result of both the K-means and the HMM algorithm is a set of states s_k, their definition in terms of features vectors and an association of each signal features vector $\underline{f}(t)$ to a specific state k.

In Figure 8, we compare the results obtained by the K-means algorithm [middle] and the K-means + HMM algorithm [bottom]. For the K-means, the initialization was done using the "initial" states. For the HMM, the initialization was done using the "middle" states. In the K-means results, the quick state-jumps between states 1, 2 and 5 are explained by the fact that these states are close to each other. These state-jumps do not appear in the HMM results since these jumps have been penalized by state transition probabilities, giving therefore a smoothest track.

The final result using the proposed method is illustrated in Figure 9. The white line represents the state belonging of each observations along time. The observations are represented in background in a spectrogram way.

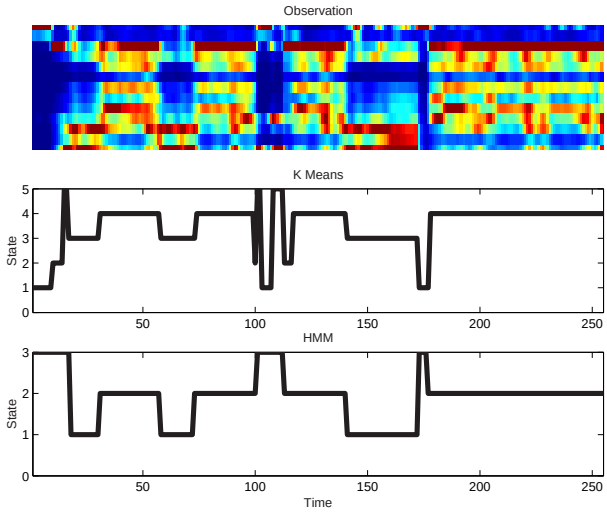

Figure 8: Unsupervised classification on title "Head over Feet" from artist Alanis Morisette [top:] signal features vectors along time [middle:] state number along time found using K-Means algorithm [bottom:] state along time found using hidden Markov model result of initialization by the K-Means Algorithm

5. AUDIO SUMMARY CONSTRUCTION

So far, from the signal analysis we have derived features vectors used to assign, through unsupervised learning, a class number to

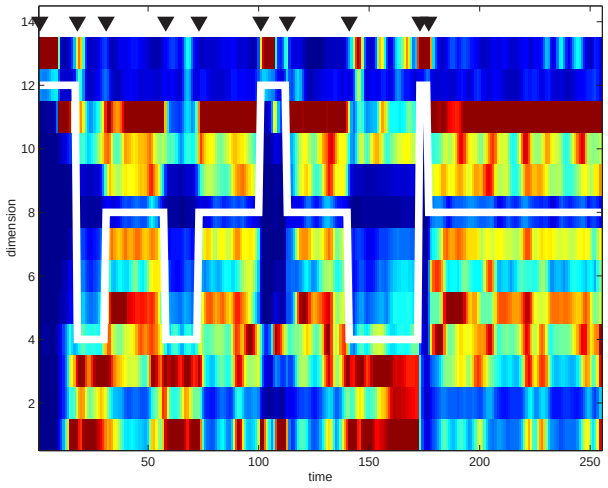

Figure 9: Results of un-supervised classification using the proposed algorithm on title "Head over Feet" from artist Alanis Morisette

each time frame. Let us take as example the following structure: AA B A B C AA B. The generation of the audio summary from this state representation can be done in several ways:

- providing audio example of class transitions (A → B, B → A, B → C, C → A)
- providing an unique audio example of each of the states (A, B, C)
- reproducing the class successions by providing an audio example for each class apparition (A, B, A, B, C, A, B)
- providing only an audio example of the most important class (in terms of global time extend or in term of number of occurrences of the class) (A)
- etc ...

This choice relies of course on user preferences but also on time constraints on the audio summary duration.

In each case, the audio summary is generated by taking short fragments of the state's signal. For the summary construction, it is obvious that "coherent" or "intelligent" reconstruction is essential. **Information continuity** will help listeners to get a good feeling and a good idea of a music when hearing its summary.
Overlap-add: The quality of the audio signal can be further improved by applying an overlap-add technique of the audio fragment.
Tempo/Beat: For highly structured music, beat synchronized reconstruction allows improving largely the quality of the audio summary. This can be done 1) by choosing the size of the fragments as integer multiple of 4 or 3 bars, 2) by synchronizing the fragments according to the beat position in the signal. In order to do that, we have used the tempo detection and beat alignment proposed by [12].

The flowchart of the audio summary construction of our algorithm is represented on Figure 10.

6. CONCLUSION

Music audio summary is a recent topic of interest in the multimedia realm. In this paper, we investigated a multi-pass approach for the automatic generation of **sequential** summaries. We introduced dynamic features which seems to allow deriving powerfull information from the signal for both -detection of sequence repetion in the music (lower/upper diagonals in a similarity matix)

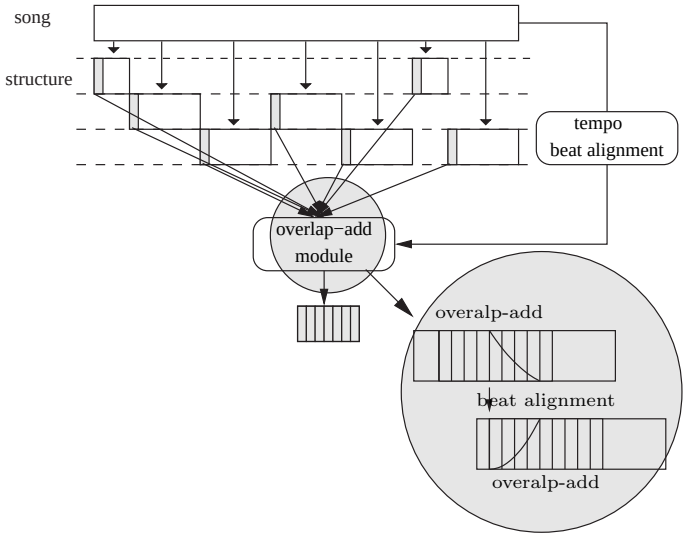

Figure 10: Audio summary construction from class structure representation; details of fragments alignment and overlap-add based on tempo detection/ beat alignment

and -representation of the music in terms of "states". We only investigated the latter here. The representation in terms of "states" is obtained by means of segmentation and unsupervised learning methods (K-means and hidden Markov model). The states are then used for the construction of an audio summary which can be further refined using an overlap-add technique and a tempo detection/ beat alignment algorithm.

Examples of music audio summaries produced with this approach will be given during the presentation of this paper.

Perspectives: toward hierarchical summaries

As for text or video, once we have a clear and fine picture of the music structure we can extrapolate any type of summary we want. In this perspective, further works will concentrate on the development of **hierarchical** summaries. Depending on the type of information wished, the user should be able to select some kind of "level" in a tree structure representing the piece of music. Of course tree-like representation may be arguable, and an efficient way to do it has to be found. Further works will also concentrate on the improvement of the **audio quality** of the output results. When combining different elements from different "states" of the music a global and perceptive coherence must be ensured.

Acknowledgment

Part of this work was conducted in the context of the European I.S.T. project CUIDADO [14] `http://www.cuidado.mu`.

7. REFERENCES

[1] P. Aigrain, P. Joly, and Al. Representation-based user interface for the audiovisual library of year 2000. In *IST-SPIE95 Multimedia computing and networking*, pages 35–45, 1995.

[2] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden markov models. In *AES 110th Convention*, 2001.

[3] J.-J. Aucouturier and M. Sandler. Finding repeating patterns in acoustic musical signals: applications for audio thumbnailing. In *AES 22nd International Conference*, 2002.

[4] R. Birmingham, W. Dannenberg, G. Wakefield, and al. Musart: Music retrieval via aural queries. In *ISMIR*, Bloomington, Indiana, USA, 2001.

[5] S. Butler and A. Parkes. Filmic spacetime diagrams for video structure representation. *Image Communication*, Special issue on Image and Video Semantics: Processing, Analysis, Application, 1995.

[6] I. Deliege. A perceptual approach to contemporary musical forms. In N. Osborne, editor, *Music and the cognitive sciences*, volume 4, pages 213–230. Harwood Academic publishers, 1990.

[7] J. Foote. Visualizing music and audio using self-similarity. In *ACM Multimedia*, pages 77–84, Orlando, Florida, USA, 1999.

[8] K. S. Jones. What might be a summary ? In K. Womser-Hacker and K. and, editors, *Information Retrieval 93: Von der Modellierung zur Anwendung*, pages 9–26. University Konstanz, Konstanz, DE, 1993.

[9] B. Logan and S. Chu. Music summarization using key phrases. In *ICASSP*, Istanbul, Turkey, 2000.

[10] MPEG-7. Information technology - multimedia content description interface - part 5: Multimedia description scheme, 2002.

[11] L. Rabiner. A tutorial on hidden markov model and selected applications in speech. *Proccedings of the IEEE*, 77(2):257–285, 1989.

[12] E. Scheirer. Tempo and beat analysis of acoustic musical signals. *JASA*, 103(1):588–601, 1998.

[13] H. Ueda, T. Miyatake, and S. Yoshizawa. Impact: An interactive natural-motion-picture dedicated multimedia authoring system. In *ACM SIGCHI*, New Orleans, USA, 1991.

[14] H. Vinet, P. Herrera, and F. Pachet. The cuidado project. In *ISMIR*, Paris, France, 2002.

[15] H. Zhang, A. Kankanhalli, and S. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia System*, 1(1):10–28, 1993.