AUDIOCYCLE: A SIMILARITY-BASED VISUALIZATION OF MUSICAL LIBRARIES

J. Urbain[1]*, T. Dubuisson*[1]*, S. Dupont*[1]*, C. Frisson*[2]*, R. Sebbe*[1] *and N. d'Alessandro*[1]

[1]TCTS Lab, Faculté Polytechnique de Mons, Boulevard Dolez 31, 7000 Mons, Belgium
[2]TELE Lab, Université Catholique de Louvain, Place du Levant 2, 1348 Louvain-la-Neuve, Belgium

ABSTRACT

This paper presents AudioCycle, a prototype application for browsing through music loop libraries. AudioCycle provides the user with a graphical view where the audio extracts are visualized and organized according to their similarity in terms of musical properties, such as timbre, harmony, and rhythm. The user is able to navigate in this visual representation and listen to individual audio extracts. AudioCycle draws from a range of technologies, including audio analysis from music information retrieval research, 3D visualization, spatial auditory rendering, audio time-scaling and pitch modification. The proposed approach extends on previously described music and audio browsers. Possible extension to multimedia libraries are also suggested.

***Index Terms*—** Browsing, music similarities, visualization, 3D rendering

1. INTRODUCTION

Music production is more and more relying on pre-recorded material. One of the dominant musical instruments synthesis paradigm is based on very large libraries of high-quality recordings of real instruments, covering their different articulations and expressive modes [1]. Furthermore, the production of some music styles heavily relies on pre-recorded musical phrases, organized into libraries. This is most apparent in styles like hip-hop or remixing, where these phrases, often referred to as "loops" (for instance drum loops), are indeed looped, sequenced, and mixed together to form full audio tracks [2] and compositions.

However, the tools available today for browsing through large musical libraries hinders the creative process. Loops can be searched off-line through rigid file system hierarchies, or through the use of symbolic descriptors stored as meta-data in the library. With the growing availability of multimedia content, there is a still larger demand for more flexible and efficient tools to access content and search for data. Information indexing and retrieval can rely on automatic technologies to describe contents on one hand and, on the other hand, allow formulating queries and structure the media database to help the user to navigate through it. The work presented here envisions a convenient and fast way of exploring large audio libraries, relying on similarity analysis and musically relevant audio analysis and transformation, involving rhythm, harmony and timbre [3, 4]. Loops can be played in a beat-synchronous fashion, relying on a phase vocoder algorithm. AudioCycle differs from previously published approaches. Compared to SoundTorch [5] for instance, it provides a range of audio analysis approaches (not only timbral) and enables to weight their importance. On the usability and rendering side, AudioCycle enables to play synchronously any combination of loops, even if they are very distant on the similarity map.

The paper is organized to describe the different blocks of the AudioCycle architecture, schematized in Figure 1. First, an *Audio Analysis (2)* is performed on a set of loops (*1*) loaded by the user. Features representative of the musical properties of rhythm, harmony, and timbre are extracted. This is presented in Section 2. Section 3 is interested by *Visualization (4)* techniques that can help users to navigate and search inside large digital media libraries. These techniques rely on the previously created Features Database (*3*). User's interactions influence the subsequent *3D Visual Rendering (5)* - also addressed in Section 3 - and *3D Audio Rendering (6)*, which spatializes, synchronizes and plays the selected loops - presented in Section 4. Suggestions for future activities are finally proposed in Section 5.

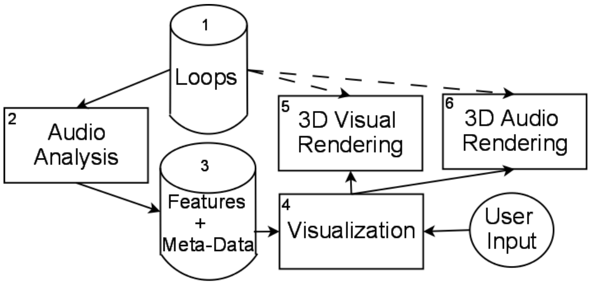

Fig. 1. AudioCycle Architecture

2. MUSIC ANALYSIS

To evaluate similarities between music chunks, a set of relevant mathematical descriptors is needed. They will be introduced hereunder. Then, a method for computing an important rhythmic notion, the tempo (expressed in Beats Per Minute - BPM), will be described. To conclude this Section, results of experiments assessing the relevance of the chosen timbral features are presented.

2.1. Feature Extraction

Following the approach of music analysis proposed in [6], features providing information about the timbre, the harmony and the rhythm of music excerpts are extracted from frames whose length is 30ms and whose hopsize is 10ms.

Timbre is defined as the quality of tone distinctive of a particular singing voice or musical instrument [7]. It is linked to the relative intensities of a sound harmonics, independently from its pitch and intensity. It has been shown [4, 3] that timbre can be efficiently captured by the Mel-Frequency Cepstral Coefficients (MFCCs). To characterize the timbre in our application, we use a filterbank of 20 filters covering the frequency range between 0Hz and the Nyquist frequency (22050Hz for the music excerpts we used), and keep only the first 12 MFCCs (+ the energy).

Harmony can be defined as the combination of simultaneous musical notes in a chord [7]. In addition, the melody is defined as a rhythmic succession of single tones organized as an aesthetic whole [7]. A widely used method to capture these properties is to project the power spectrum of each frame on a chromatic scale, which decomposes each octave into 12 "semitones". The "chromas" characterizing the harmony and melody of a frame are then the 12 sums of the energetic contributions of each semitone on all the octaves [6].

Rhythm is defined as the aspect of music comprising all the elements (as accent, meter, and tempo) that relate to forward movement [7]. From a signal processing point of view, it is closely related to the notion of periodicity [6]. It is common to compute the Perceptual Spectral Flux (PSF) [4, 3], which can be seen as an integration of the local derivatives of the spectrogram frequency bins, to capture the rhythm information. Unlike [3] and [6] approaches, calculating local autocorrelation functions on overlapping frames of the PSF, we have chosen to save as rhythmic features the PSF values in the neighbourhood of the beats and tatums positions for one bar. It has also been decided to split the contributions of the PSF in 3 frequency bands: low frequencies (0 to 100 Hz), medium frequencies (100 to 8000 Hz) and high frequencies (8000 Hz to the Nyquist frequency). A 4/4 rhythmic signature is assumed. Thus, to capture the beat and tatum information of one 4/4 bar, features around 8 particular instants must be computed. This method requires the knowledge of the interval between two successive beats. When this value is unknown, it can be estimated thanks to an algorithm that will be presented in Section 2.2.

The targeted application must be able to deal with thousands of musical extracts, fastly compare and display them (see Section 3). This objective has two implications on the feature sets. First, the feature vectors must be of reasonable length, otherwise computation times would dramatically increase. Second, to ease comparisons, characteristic vectors of each musical excerpt should have a constant length. Regarding the rhythm, the method aforementioned already provides a constant number of 24 features per music excerpt. However, as the number of timbre and chroma features is proportional to the music excerpt duration, this information is summarized into fixed size vectors by storing only the mean and variance of each timbre and harmony feature.

2.2. Beat Estimation

The notion of "beat" is often expressed by musicians and listeners through the act of foot tapping along to the music. Beat tracking systems form the basis of applications like automatic transcription, or evaluation of music similarity.

In literature, the PSF is commonly used to extract the BPM value. The method proposed in [8] estimates it by computing the autocorrelation of the PSF and by favoring autocorrelation values located in a range defined by the user. It is performed by weighting the autocorrelation function by a Gaussian window centered around the most likely BPM value and whose spread is set by the user. Once the largest peak is identified in the weighted autocorrelation, a second potential value of BPM is given as the largest value of autocorrelation at 0.33, 0.5, 2 or 3 times the first value of BPM. This system was tested on the database of the MIREX 2006 contest [9] and achieved 77% of agreement with the manual BPM. Among other beat tracking systems, one can cite the method presented in [10], where the PSF is computed from a reassigned spectrogram or [11], in which the structure of a musical excerpt is estimated by computing a PSF from different frequency bands in the spectrogram and by estimating chord-change possibilities and drum patterns at the local maxima of the PSF. Because Ellis' method [8] provides good performance and because the whole source code is provided on his website, it has been chosen as the beat period estimation in our system.

2.3. Instruments Classification

In order to determine to what extent the chosen timbre features are indeed representative of timbre perception and to define which set of features is the most appropriate for the similarity estimation and visualization in AudioCycle, their performance for classifying instruments has been assessed.

The features are extracted from musical loops of the ZeroG ProPack ACID database [12]. We have manually tagged the files using 7 instruments classes: Brass, Drums, Vocals, Percussion, Electric Bass, Acoustic Guitar and Electric Guitar. Only 4380 files for which the class is clear have been used.

Before classifying the feature vectors, they have to be manipulated in order to assess if their dimension can be reduced. For this purpose, the Generalized Fisher criterion and Principal Component Analysis (PCA) have been tested [13].

Three ways of reducing the dimension of feature vectors have been investigated: applying the Fisher criterion alone, using PCA alone and applying Fisher criterion followed by PCA. These different approaches have been applied on the means and variances of the timbre features. For each configuration, a Multi-Layer Perceptron was trained with 60% of the objects, validated with another 10% and tested with the remaining 30%. The operation was repeated 100 times and the objects were always randomly selected. Classification performance for timbre information reaches 93% of accuracy when applying a PCA to the 12 means of the timbral features. This supports our decision to summarize the timbre information (see Section 2.1) and confirms that these vectors still contain enough information to distinguish instruments with a very high accuracy, which is the aim of timbre characterization. Similar tests are under way to assess the opportuneness of summarizing the chroma information the same way.

3. VISUALIZATION

3.1. Visualization Techniques

The representation of large multimedia databases has been a subject of research in various cases of media content. As stated in [14], the visualization has to meet three requirements: overview (faithful overview of the distribution of excerpts in the collection), structure preservation (the relations between music excerpts should be preserved in the projection in the visualization space) and visibility (visually understanding of the content of each music excerpt). Thus, if the similarity between music excerpts is used, the ideal is to build a representation in which the similarity distances are preserved. This is ensured by our choice of the node-link visualization paradigm, great for emphazising two Gestalt laws (similarity and connectedness) [15]. When displaying the whole collection of music excerpts is desired, the idea is to associate to an object x_i a vector y_i aiming at transposing the similarity between x_i and its neighbours in the 2D representation. Three kinds of approaches have been considered in this study: dimensionality reduction (using PCA or Linear Discriminant Analysis [16]), graphs and multi-dimensional scaling [14], and clustering. This latter is described hereafter since it is an original approach and it has been fully implemented in our system.

The clustering used relies on a K-Means algorithm and aims at grouping loops characterized by their extracted feature vectors. The Euclidian distance has been chosen. The emphasis of the clustering can be changed by the user by scaling differently the feature space dimensions related to timbre, harmony, and rhythm. The number of clusters is chosen by the user and, to preserve near real-time interaction, the clustering is stopped after a limited number of iterations. Clusters can be represented in a flower-like scheme, where the center is the currently selected loop and other loops are gathered into clusters around the selected one. The 2D radial distance between a loop and the selected centered element is proportional to the distance between them (in a feature space with scaling factors complementary to the one used for clustering) , while the distance between the loop and the cluster center (in the scaled space used for clustering) is used to compute the 2D angular coordinate to locate it in the cluster. This scheme allows to explore organization schemes where the radial and angular coordinates are associated to complementary musical properties, e.g. timbre and rhythm.

3.2. 3D Imaging, OSG and OpenGL

Visualization of the audio loops is made with Open Scene Graph (OSG) [17], a scene graph visualization library running on top of OpenGL and permitting the definition of high level objects. The scene graph is built by adding cubes representing each loop and linked to an OpenAL source each (see Section 4). Each of these cubes has a parent node that specifies its position, which can change over time, when the user changes the clustering properties for instance. When a loop is activated, the scene graph is enhanced with a waveform display as well as a curser indicating the current playback position. Loops can be activated by clicking on the associated cube. This event is then processed, the waveform is displayed and the associated OpenAL source is activated. A screenshot of the Audiocycle view is shown in Fig. 2.

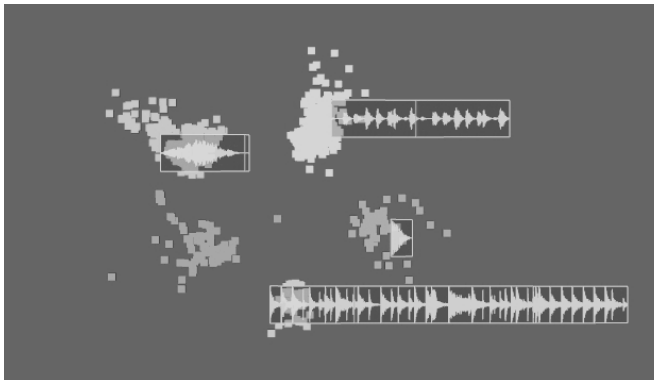

Fig. 2. AudioCycle OSG View

4. AUDIO RENDERING

For a suited audio rendering, the loops are first placed in space in relation with the position of their visual representation on the screen. This tight coupling between the audio and the image is expected to make it easier to locate the area of interest [18, 5], when the user is looking for specific sounds while several loops are playing together. This has been implemented using OpenAL [19] (where streaming sources are tuned to play the selected loops by the user) with a specific extension providing Head Related Transfer Functions based-rendering (in order to enhance the realism in the sound scene). Moreover, as we are dealing with music content, it makes sense to provide the application with facilities to allow beat synchronous playback (through audio time-scaling) of loops that have originally not been played at the same tempo, and key normalization (through pitch shifting) for loops that have originally not been played in the same keys. This is expected to reduce the cognitive load when several loops are playing together, hence easing the search for specific sounds. Eventually, this may also provide the basis for extending AudioCycle to a performance tool, where live composition and performance is made possible through beat and key matching.

5. FUTURE WORK

First, it is planned to redesign some aspects of the visualization [15], by e.g. respecting the symmetry of the 3D node-link diagram paradigm and condensing the mapping of the waveforms around each loop representation [15]. Then, in addition to the usual keyboard/mouse couple support, we are investigating new methods for gestural control of multiple parameters of the AudioCycle application (in terms of navigation, visualisation, etc...), such as a bimanual interaction using a “3D mouse” for navigation, and a “jog wheel” for scrubbing through waveforms, recentering the user’s attention on the task to achieve. Also, in order to achieve the fastest possible audio mining, we initiated a user-centered usability approach, featuring context enquiries with experts (sound designers [1], DJ’s, etc.) and usability tests with software/hardware prototypes, leading to the definition and support of different user and hardware profiles. Finally, this work is part of an ongoing series of projects intending to extend our approach to multimedia content with scalability to very large archives.

6. CONCLUSION

A novel approach and prototype implementation of a music loop library browser has been presented. It allows to display the music extracts organized on a map according to timbre, harmony, rhythm, or a combination of these. Spatial sound is also used to enhance the experience when playing several extracts at the same time. Features relevant to music production have also been integrated, including the possibility to play the loops in a beat-synchronous fashion, as well as alter their pitch. Some avenues for extending and improving the concept and technologies involved are also proposed.

7. ACKNOWLEDGMENTS

This work has been supported by the numediart research project, funded by Région Wallonne, Belgium (grant N°716631).

8. REFERENCES

[1] Martin Russ, *Sound Synthesis and Sampling*, Music Technology. Focal Press, 3 edition, 2008.

[2] Sony Creative Software, “Acid,” http://www.sonycreativesoftware.com.

[3] K. Jensen, “Mutliple scale music segmentation using rhythm, timbre and harmony,” *EURASIP Journal on Advances in Signal Processing 2007*, vol. 7, 2007.

[4] T. Jehan, *Creating Music by Listening*, Ph.D. thesis, September 2005.

[5] S. Heise, M. Hlatky, and J. Loviscach, “Soundtorch: Quick browsing in large audio collections,” in *125th AES Convention*, San Francisco, CA, Oct 2008.

[6] L. Couvreur, F. Bettens, T. Drugman, T. Dubuisson, S. Dupont, C. Frisson, M. Jottrand, and M. Mancas, “Audio thumbnailing,” in *QPSR of the numediart research program*, Thierry Dutoit and Benoît Macq, Eds., 2008, vol. 1, pp. 67–85, http://www.numediart.org.

[7] “Merriam-webster dictionary online,” Consulted on January 14, 2009, http://www.merriam-webster.com/.

[8] D. Ellis, “Beat tracking with dynamic programming,” in *Proceedings of MIREX 2006*, 2006.

[9] “Music information retrieval evaluation eXchange,” http://www.music-ir.org.

[10] G. Peeters, “Time variable tempo detection and beat marking,” in *Proceedings of ICMC 2005*, 2005.

[11] M. Goto, “An audio-based real-time beat tracking system for music with or without drums-sounds,” *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[12] “Zerog,” http://www.zero-g.co.uk/.

[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press - Second Edition, 1990.

[14] A. Veroust-Blondet, “Vitalas - state of the art on advanced visualisation methods,” Tech. Rep., 2007.

[15] Colin Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann, 2 edition, 2004.

[16] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[17] “Openscenegraph,” http://www.openscenegraph.org/.

[18] M. Fernström and E. Brazil, “Sonic browsing: An auditory tool for multimedia asset management,” in *ICAD2001*, Helsinki, Finland, 2001.

[19] “Openal,” http://www.openal.org/.