Audio Engineering Society

Convention Paper 8112

Presented at the 128th Convention
2010 May 22–25 London, UK

Browsing Sound and Music Libraries by Similarity

Stéphane Dupont[1], Christian Frisson[2], Xavier Siebert[1], Damien Tardieu[1]

[1] *Université de Mons, TCTS and MathRo Labs, 31 Boulevard Dolez, B-7000, Mons, Belgium*

[2] *Université catholique de Louvain, TELE Lab, 2 Place du Levant, B-1348 Louvain-la-Neuve, Belgium*

Correspondence should be addressed to Stéphane Dupont (`stephane.dupont@umons.ac.be`)

ABSTRACT

This paper presents a prototype tool for browsing through multimedia libraries using content-based multimedia information retrieval techniques. It is composed of several groups of components for multimedia analysis, data mining, interactive visualization, as well as connection with external hardware controllers. The musical application of this tool uses descriptors of timbre, harmony, as well as rhythm and two different approaches for exploring/browsing content. First, a dynamic data mining allows the user to group sounds into clusters according to those different criteria, whose importance can be weighted interactively. In a second mode, sounds that are similar to a query are returned to the user, and can be used to further proceed with the search. This approach also borrows from multi-criteria optimization concept to return a relevant list of similar sounds.

1. INTRODUCTION

A number of recent technological advances are making it possible to collect, store and use vast amounts of multimedia content in a digital format, such as music, images, videos, or 3D content. Search technologies for such content, initially very limited compared to text, have hence been actively researched, within the field of multimedia information retrieval (MIR). These are making use of automatic analysis and recognition of multimedia, providing the basic tools to organize the content according to similarity, as well as query for similar content from an example, an approach also known as query-by-example.

Music production and sound design are also benefitting from this trend. One of the dominant musical instrument synthesis paradigm is based on very large libraries of high-quality recordings of real instruments, covering their different articulations and expressive modes. Furthermore, the production of some music styles heavily relies on pre-recorded musical phrases, organized into libraries. And of course, personal collections of tunes are also handled digitally. Music information retrieval research is developing innovative approaches to handle such content.

We describe and demonstrate here a multimedia information retrieval tool with an application to browsing audio and music content. An important issue related to multimedia content is that, as opposed to text, it can be interpreted from multiple facets. In the audio/music domain, sound can be characterized according to several dimensions, such as rhythm or timbre, already mentioned above, some of which are themselves characterized by a range of features, for instance temporal and spectral features of timbre. Methods for exploring and visualizing multimedia libraries based on content analysis hence have to consider the fact that the user may be trying to organize or search sounds according to some of the analysis facets only.

Two exploration methods are proposed in this work. In the first one, unsupervised data mining is applied to organize the content according to similarity. In particular, k-means clustering is performed in an interactive fashion, and clusters of similar audio files are created dynamically, according to the facets the user is interested in. Interactive visualizations in the form of maps with similar sounds located close to each other. A second method for exploring the data is also proposed. Borrowing from multi-criteria optimization concepts, we propose to return to the user the Pareto set of similar elements from a user query example. Returned sounds can be used to iterate on the process. The browsing steps performed by the user are subsequently stored and visualized as a tree structure, allowing the user to return to previous search steps.

The user can obviously listen to the library content. The audio engine has been designed to allow the user to scrub along the waveforms, to change the signal pitch, as well as to play rhythmic content in tempo. Besides being able to interact with a keyboard and a mouse, other types of hardware can be also used to control those functionalities.

The paper is organized as follows. Section 2 presents the architecture of the content-based multimedia exploration framework that has been developed in this work. Section 3 gives a brief overview of the content analysis schemes that are used for the application to audio content. Section 4 describes in more detail the two database exploration approaches that have been implemented.

2. MEDIACYCLE

MediaCycle is the name of the general framework we have been developing. It can be seen as both a library allowing multimedia data managment and an application. The architecture has been designed to allow modular development and rapid prototyping of new application concepts. For instance, it has been used for querying a database of laughter [5], for browsing a collection of dance videos for an interactive installation [16] and in the present paper, browsing and querying a sound database.

The general architecture or MediaCycle is presented in Figure 1, whose different modules are briefly described here.

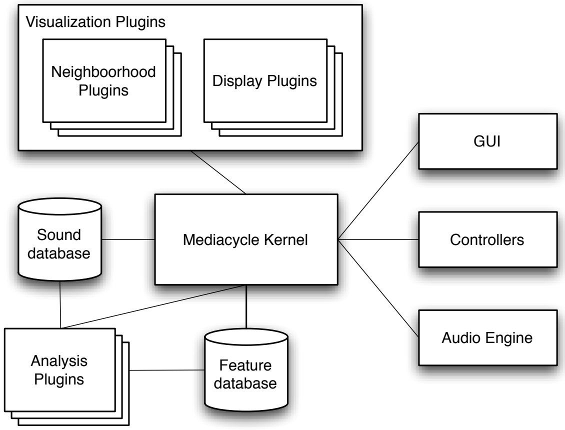

Fig. 1: MediaCycle architecture.

2.1. The Kernel

The main element is the kernel, whose main task is to manage the media database and call the necessary plugins for sound analysis and on screen visualization. Taking advantage of polymorphism the kernel can handle different kind of media types (sounds, images, videos) and decide at run time which functions must be used, for instance for analysis or for playback.

2.2. Analysis Plugins

A plugin architecture has been chosen for media analysis to facilitate the implementation of new feature extractors. Up until now, analysis plugins have been developed for images, dance videos and audio. Audio analysis approaches used in this particular paper are outlined in Section 3.

2.3. Audio Engine

Besides allowing for multiple sounds playback, the main requirements of the audio rendering module were to allow for:

- real-time control of the sound playback, in order to be able to skim through the sound or change its pitch, or to allow for beat synchronous playback, all crucial features in sound/music search,
- positioning of the sources in a 3D virtual space, in order to be able to locate them in relation with the position of their visual representation on the screen, likely making it easier to locate the area of interest, especially when the user is searching for specific sounds while several are playing together [8, 11].

This has been implemented on top of OpenAL [1], making use of streaming sources.

2.4. Renderer

Visualization of the audio (waveform) and of the browsing interface is made with Open Scene Graph (OSG), a scene graph visualization library running on top of OpenGL and allowing the definition of high level objects. These objects can either render geometry or represent 3D transformations, and, when associated as a graph, they define a rendered image. The scene graph is built by adding objects representing each sound. Each of these objects has a parent node that specifies its position, which can change over time, when the user changes the clustering properties for instance. Sounds can be activated by clicking on the associated objects, or in a second mode by hovering over the displayed objects, which can be performed using standard keyboard/mouse controllers. When a sound is activated, the scene graph is enhanced with a waveform display as well as a curser indicating the current playback position. OSG will allow us to extend this work to 3D representations very naturally. More details about the visualizations associated with the two exploration modes introduced earlier are given in Section 4.

2.5. Gestural and Remote Control

Besides the standard keyboard/mouse couple, gestural controllers and remote applications can be connected to the MediaCycle framework through an OpenSoundControl (OSC) server/client, so as to control MediaCycle applications using a dedicated OSC namespace. To map raw events from controllers to events that MediaCycle can recognize as part of the namespace, the PureData dataflow visual programming environment can being used, as described in more details in another paper [9].

2.6. TCP Server

Finally, the MediaCycle kernel has been equiped for TCP support. The functionalities provided by the kernel, as well as its associated visualization and analysis plugins can hence be accessed remotely, as has been done for instance in a prototype experiment in [5], and then for a deployed web site in [16]. In the first case for instance, audio is streamed from an web browser to the kernel module, which analyze the content, and subsequently allows to retrieve similar sounds in the growing database.

3. AUDIO FEATURES

The software we are developing is designed to handle audio, images and video files. Feature extraction modules are implemented as plugins, as introduced in the architecture overview in the previous section. Several audio analysis algorithms have already been implemented to cover several perceptual properties of sounds and music extracts, including timbre, harmony, and rhythm.

3.1. Timbre description

The features included in the timbre description are mostly inspired by previous work in audio, speech [14] and music [13] analysis, and include: total energy, zero-crossing rate, spectral shape descriptors (spectral centroid, spectral spread, spectral skewness and spectral kurtosis), loudness, sharpness, spread, spectral flatness, spectral crest, spectral slope, spectral decrease, spectral roll-off, spectral variation, as well as mel-frequency cepstral coefficients and their first temporal derivate.

Generally speaking, when describing a signal, we can distinguish features that are computed over the whole signal ("global descriptors") from features that are computed over successive short time frames of the signal ("instantaneous descriptors"). The descriptor cited above are instantaneous descriptors, in our case computed on signal frames of 21 ms length with 75% overlap between frames. The initial number of features is hence proportional to the music excerpt duration, but this information is then summarized into fixed size vectors by computing the mean and variance of each timbre feature, in effect resulting in a fixed number of global descriptors for the whole sound file.

3.2. Harmony description

Harmony descriptors are also included, based on the concept of chroma [10]. It consists in projecting the power spectrum of each analysis frame on a chromatic scale, which decomposes each octave into twelve semitones. The chromas characterizing the harmony and melody of a frame are then the twelve sums of the energetic contributions of each semitone on a range of retained octaves (5 in our case). Here too, the computation of mean and variance of each chroma results in a fixed size global descriptor vector of harmony.

3.3. Rhythm description

Rhythm descriptors are available too for sound extracts that correspond to short music loops. First, the Perceptual Spectral Flux (PSF) [12] is computed from the power spectra of each analysis frame. Rhythmic features[3] are then computed as the average PSF values in the neighbourhood of beats and tatums instants in the musical bar, and in three frequency bands with cutoffs at 100 Hz and 8000 Hz. For a 4/4 rhythmic signature (containing 8 tatums), 24 features are hence obtained per musical bar. When the extract contains several bars, averaging is used to always lead to a fixed size global descriptor vector of rhythm. This method requires the knowledge of the interval between two successive beats. When this value is unknown, it can be estimated thanks to a beat extraction algorithm (f.i. [7]). Ideally, a meter estimation algorithm would also be needed to improve the rhythm features.

4. BROWSING

Two modes for exploration sound libraries are proposed. These are described in the following sections.

4.1. Exploration by Interactive Clustering

In this mode, we want to provide the user with interactive visualizations organized according to groups of similar sounds.

Visualization [18], as well as the representation of large multimedia databases has been a subject of research. As stated in [17], 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 sounds or music excerpts is used, the ideal is to build a representation in which the similarity distances are preserved. When the whole collection is to be displayed, the idea is to associate to an object characterized by a feature vector 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 are often considered: dimensionality reduction (using PCA or Linear Discriminant Analysis [4]), graphs and multi-dimensional scaling [17], and clustering.

Clustering has been used in this work. It aims at providing grouping of similar sounds and relies on the k-means algorithm using the euclidian distance. The groupings must take into account the different selected features (Section 3). But as stated before, the approach needs to remain dynamic, allowing the user to emphasize one or several of those different facets. User interface sliders are provided to adjust the weights of these facets, or even turn them off. The number of clusters can also be chosen by the user and, to preserve near real-time interaction, the clustering is interrupted after a fixed number of iterations. Clusters are then represented in a flower-like scheme, where the center is the currently selected sound and other sounds are gathered into clusters around the selected one. The 2D radial distance between a sound and the selected centered element is proportional to the distance between them, while the distance between the sound and its cluster center is used to compute the 2D angular coordinate to locate it in the cluster. A screenshot of the resulting visualization is given in Figure 2. Starting from the initial view, the interface allows the user to enter any given cluster, whose elements can further be clustered according to emphasized facets, resulting in a dynamic hierarchical directory structure.

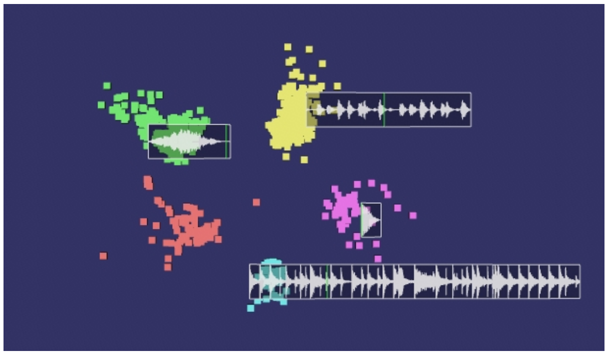

Fig. 2: Interactive visualization of sound library using clustering.

4.2. Similarity Browsing

In this mode, we want to provide the user with an approach allowing to retrieve similar sounds from a query example, and possibly iterate over this process.

To choose the sounds to return to the user, we need a definition of similarity that takes into account the different selected features 3. The usual measures of similarity (for instance on images [15]) are based on euclidean distance or cosine distance with equal and fixed weights. The underlying assumption is that all features are equally important for the perceived similarity, thereby discarding content that is very similar for only a subset of features. However, in this application, it is important to be able to identify sounds that are very similar for only some criteria. We therefore introduce another measure of similarity based on the concept of "Pareto ranking" borrowed from multicriteria optimization [6]. In short, a solution c of a multicriteria optimization problem is said to be "non dominated" if there is no solution c' at least as good on all criteria and strictly better for at least one criteria. The set of non-dominated solutions is called the *Pareto front* and the solutions belonging to the front have a *Pareto rank* of 1. If we remove all the solutions of the Pareto front from the collection, the new set of non-dominated solutions have a rank of 2 and so on. In terms of similarity, a solution belonging to the Pareto front is the closest to the target for a given set of feature weights. Fig. 3 shows an example of Pareto front. In the top left figure the data are represented in the features space. In the top right figure, the same data are plotted in the feature distance space, that we call criteria space. The bottom figures show a comparison between the closest point according to the Pareto front on the left and to the euclidean distance on the right. The important point is that euclidean distance completely ignores the two clusters that are very close for only one feature, whereas Pareto ranking does not. In this work we are using a mixture of Pareto rank and of euclidean distance to retrieve sounds that are close for all the features and sounds that are close considering a subset of features. Formally, out of the collection $\mathcal{C}$ we select the sounds c whose rank relative to the target is smaller than $\bar{R}$, where $\bar{R}$ is defined as follows:

$$\bar{R} = \min R \ s.t. \ |\{c \in \mathcal{C} | rank(c) \leq R\}| \geq M$$

In this subset the M closest sounds according to the euclidean distance with equal weights are selected.

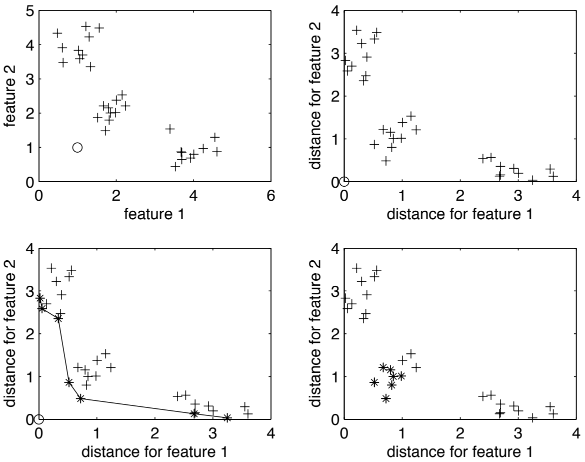

Fig. 3: Example of Pareto rank selection. Target is a circle, selected items are stars. Top-left: items in the feature space. Top-right: items in the criteria space. Bottom left: selection using Pareto rank. Bottom-right: selection using euclidean distance.

Starting from the initial query, the interface also allows the user to iterate, using the returned sounds as queries for further search. When analysing audio content, the dimensionality of the feature space is large. A straightforward application of this approach hence lead to a large set of returned sound, as similarity according to a single feature may lead the sound to be ranked high (curse of dimensionality). We are currently exploring feature grouping and dimensionality reduction as strategies prior to ranking, as well as clustering strategies on the ranked sounds, in order to achieve a more relevant selection of similar sounds.

4.3. Browsing Steps

The browsing process can be represented by a tree structure, which can serve various purposes. In the first browsing mode, each node added to the tree corresponds to a browsing step. The resulting tree is mostly used to keep track of the user's actions and to go back to a previous step. In the second browsing mode, each sound file queried by the user is represented by a node, starting from the root node. Potential similar elements, in the Pareto sense, are represented by children of the given node. The visualization makes use of a node-link diagram and we opted for a simple tree layout with straight lines representing the links, implementing the algorithm described in [2].

5. CONCLUSION

A novel approach and prototype implementation of a sound library browser has been presented. It allows to display the sound files organized on a map according to timbre, harmony, rhythm, or a combination of these, as well as search for sounds that are similar to a query example. This has been developed as a general framework designed to support the exploration of similar concepts for multimedia content browsing, for instance with application to dance videos [16].

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] OpenAL. http://www.openal.org/.

[2] Christoph Buchheim, Michael Jünger, and Sebastian Leipert. Improving walker's algorithm to run in linear time. In *Proc. of the 10th International Symposium on Graph Drawing*, 2002.

[3] L. Couvreur, F. Bettens, T. Drugman, T. Dubuisson, S. Dupont, C. Frisson, M. Jottrand, and M. Mancas. Audio thumbnailing. In Thierry Dutoit and Benoît Macq, editors, *QPSR of the numediart research program*, volume 1, pages 67–85, 2008. http://www.numediart.org.

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

[5] Stéphane Dupont, Thomas Dubuisson, John Anderson Mills III, Alexis Moinet, Xavier Siebert, Damien Tardieu, and Jérôme Urbain. Laughtercycle. In Thierry Dutoit and Benoît Macq, editors, *QPSR of the numediart research program*, volume 2, pages 23–31. numediart, 6 2009.

[6] M. Ehrgott. *Multicriteria optimization.* Springer Verlag, 2005.

[7] D. Ellis. Beat tracking with dynamic programming. In *Proceedings of MIREX 2006*, 2006.

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

[9] Christian Frisson, Stéphane Dupont, Xavier Siebert, Damien Tardieu, Thierry Dutoit, and Benoît Macq. Devicecycle: rapid and reusable prototyping of gestural interfaces, applied to audio browsing by similarity. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, 2010.

[10] C.A. Harte and M.B. Sandler. Automatic chord identification using a quantised chromagram. In *118th Audio Engineering Society's Convention*, 2005.

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

[12] T. Jehan. *Creating Music by Listening.* PhD thesis, September 2005.

[13] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification). in the CUIDADO project. Paris, IRCAM, 2004.

[14] L.R. Rabiner and B.H. Juang. *Fundamentals of speech recognition.* Prentice hall, 1993.

[15] A. Rorissa, P. Clough, and T. Deselaers. Exploring the relationship between feature and perceptual visual spaces. *Journal of the American Society for Information Science and Technology*, 59(5):770–784, 2008.

[16] Damien Tardieu, Ricardo Chessini, Julien Dubois, Stéphane Dupont, Sullivan Hidot, Barbara Mazzarino, Alexis Moinet, Xavier Siebert, Giovanna Varni, and Alessandra Visentin. Video navigation tool: Application to browsing a database of dancers' performances. In Thierry Dutoit and Benoît Macq, editors, *QPSR of the numediart research program*, volume 2, pages 85–90. numediart Research Program on Digital Art Technologies, 9 2009.

[17] A. Veroust-Blondet. Vitalas - state of the art on advanced visualisation methods. Technical report, 2007.

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