A System for Dynamic Playlist Generation Driven by Multimodal Control Signals and Descriptors

Luca Chiarandini [#1], Massimiliano Zanoni [*2], Augusto Sarti [*3]

[#] *Yahoo! Research Barcelona*
Barcelona, Catalunya (Spain)
[1] chiarluc@yahoo-inc.com

[*] *Dipartimento di Elettronica e Informazione, Politecnico di Milano*
Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
[2] zanoni@elet.polimi.it
[3] sarti@elet.polimi.it

***Abstract*—This work describes a general approach to multimedia playlist generation and description and an application of the approach to music information retrieval. The example of system that we implemented updates a musical playlist on the fly based on prior information (musical preferences); current descriptors of the song that is being played; and fine-grained and semantically rich descriptors (descriptors of user's gestures, of environment conditions, etc.). The system incorporates a learning system that infers the user's preferences. Subjective tests have been conducted on usability and quality of the recommendation system.**

I. Introduction

In past decade, our way of searching, listening to and producing multimedia content has dramatically changed. Personal and shared collections of content have grown exponentially, and so has the complexity of tasks related to organizing such collections and searching through them. This has motivated a proliferation of solutions for multimedia information retrieval (MIR) and, more specifically, music recommendation.

The goal of multimedia playlist generation systems is to rank content according to some metrics. We are interested in defining metrics that account for the user's preferences; for intentional control signals (generated, for example, by specific or dedicated controllers); as well as (possibly unintentional) gestures (summarized by signal-like or symbol-like descriptors); and/or other types of unintentional stimuli coming, for example, from environmental events. Suitable multimodal stimuli, however, required modeling and mapping onto the feature-based representation of the multimedia content, where playlist generation takes place.

Most of today's systems are based on global descriptors that apply to the whole multimedia excerpt. Two are the most widespread *recommendation* paradigms: *Context-based systems* and *Content-based systems*. Context-based systems use manually edited labels (tags) that drive content selection based on similarity metrics. More effective solutions are offered by content-based models, which use low-level or high-level descriptors that are extracted from the signal. In both cases, however, playlist generation is usually based on descriptors that are valid for the whole excerpt, without considering local mood or genre changes that the individual excerpts go through.

In this work we propose a *dynamic* content-based multimedia playlist generator based on local (short-term) descriptors of the multimedia excerpts, which allow us to accommodate or account for transitions and changes that occur during the multimedia excerpts. This is achieved by considering the pieces as a sequential collection of fine grains, here referred to as called *cells* (each described by a set of local features). This choice makes the playlist generation system able to swiftly react to changes that occur *during* the rendering of the excerpt, whether embedded in the excerpt or coming from the user (listener) or from the environment. Every time an internal mood change or an external event takes place, the system can decide to issue a playlist update that better fits the prior knowledge (user's preferences) and the current mood/status of the stream that is being rendered. In doing so, the system can also specify which cell could best accommodate the next excerpt switch (keeping the conditions of the renderer into account).

This approach is particularly suitable for multimedia content that exhibits significant temporal evolution (trans-genre excerpts, mood transitions), or users that tend to favor mood swings, but could be easily integrated with more traditional playlist generation platforms. The fact that the system capitalizes on local descriptions, however, introduces a new level of control. The system that we propose, in fact, is also sensitive to inherent mood or genre transitions of running excerpt, which makes the system *self-aware* or *rendering-aware*. Thanks to these characteristics, the system offers a wide range of potential applications, including internet-radio stream generation (sensitive to online user-feedback); sound stream generation for clubs, driven by descriptors of audience behavior; music rendering for sports training based on biological signals (e.g. heart rate, step rate, etc.); and more.

II. Related Works

Multimedia recommendation and playlist generation techniques have been extensively studied and developed in the literature. The main paradigms that seem to be consolidated the best are *Collaborative Filtering* and *Content-Based In-*

978-1-4577-1434-4/11/$26.00 ©2011 IEEE

formation Retrieval. Collaborative filtering consists of analyzing data on large numbers of users to infer similarities between preferences and selections. This process relies on text descriptors (tags) of the content, which are manually generated. Most of the commercial products available today fall into this category. *Apple iTunes Genius*, for example, is a playlist generator and a content recommendation system based on collaborative filtering of a very large corpus of data coming from the libraries of the users. In this case playlist generation is based on a context-based approach that uses meta-data descriptors (tags) of the libraries ([1]). The main problem of collaborative filtering is the need of manual labeling, which is complete at best, but often affected by errors and format misalignments.

Content-Based Information Retrieval systems are based on extraction of relevant descriptors from the audio content, sometimes combined with external parameters: time of the day ([2]), information derived from the background noise ([3]) or from the step frequency of a running person ([4]). The approach presented in this paper refers to this category. An example of Content-Based Information Retrieval system is *SIMAC* ([5]), which is able to create a playlist that ranks the music selection based on audio features that are weighted in compliance with the inferred musical preferences of the user.

Examples of popular music recommendation systems that are based on a both Collaborative Filtering and Content-Based Information Retrieval are *Last.fm*[1], *The Echonest*[2], etc.

An interesting approach to playlist generation was proposed by Shan et al. [6], who considered emotional features as similarity parameters. In this work we integrate mood features a controllable feature, but we could readily accommodate mood analysis systems for producing this controlling feature. An example of gesture-based mood analysis (specifically developed for expressive gestures in artistic performance) is proposed in [7].

Several approaches have been developed for assessing the user's preferences. In [8] the authors compare several techniques (Collaborative Filtering, Gaussian Mixture Models-GMM, etc.) and evaluate them on a sample dataset. In this study collaborative filtering seems to produce the best results in comparison with GMM based on Mel Frequency Cepstral Coefficients (MFCC). Nonetheless, we adopted a GMM-based method based on multiple features (including MFCC) because we are interested in a *local* playlist generation method. This choice, in fact, is more suitable for on-the-fly (short-term) closed-loop analysis.

III. Approach Overview

Two are the basic key aspects where Multimedia Playlist Generation systems tend to differ from each other: the model used to describe contents with the relative retrieving policies; and the adopted human-machine interaction model. The system that we propose is based on local descriptors as well as multimodal control signals and features according to an approach described in this Section. In the next Section we will describe an implementation instance of this approach.

As shown in fig. 1 three are the main blocks that the system is composed of: *Controller*, *Playlist generator* and *Renderer*.

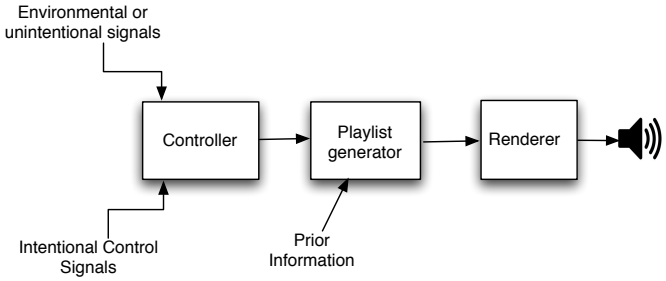

Fig. 1. Playlist generation approach.

A. Playlist-generator

The core of the system is the Playlist Generator. Its tasks are the generation of a proposals given a content description model (e.g. features), the retrieving systems (e.g. similarity functions used) and the ranking policies. In order to achieve fast reactiveness to stimuli, which is one of the main features of the approach that we propose, a model based on local descriptors is here adopted. Given the content database, segmentation is performed by first identifying key temporal points in the piece (*anchors*). The homogeneous portion of contents between two consecutive anchors is a *grain* or *cell*. From this view a multimedia content is considered as a consecutive collection of cells.

For each feature, a single value is used for describing the cell, computed as follows. A feature is defined as $f : T \mapsto D$ where T is the time Domain and D is an arbitrary co-domain. Given $H_f(t)$ the function that return the value of the generic feature f at time t, and given the k-th cell C_k where $[t_k, t_k + \Delta t]$ is its time range, the value $F_f(C_k)$ of the feature f describing the whole segment is:

$$F_f(C_k) = \frac{1}{\Delta t} \int_{t_k}^{t_k+\Delta t} H_f(\tau) d\tau \quad (1)$$

With this approach the system turns out to be highly sensitive to external events and control signal, and to internal changes (self-aware). In fact, every time a new cell begins, a new playlist can be generated in order to best fit the characteristics of the new cell. This way the playlist dynamically changes to follow the evolution of the content. As shown in Fig. 1, the playlist generator takes into account also the history of the system (prior information) in order to capture the preferences of the users.

B. Controller

The controller is responsible for collecting external stimuli and pass them to the playlist generator. The high reactiveness of the system opens the way to a large variety of control

[1] http://www.last.fm/
[2] http://the.echonest.com/

signals. These stimuli can be intentional (e.g. controller-based or gesture-driven), or they can be unintentional behavior-driven or environmental events. Examples of the sort could be descriptors extracted from a blob-detection on video streams on a dance floor, or the user's heartbeat. The control block provides an interpretation of control signals and maps them onto the feature space that describes the content, in order to affect the playlist generation process. This can be achieved through machine learning methodologies.

C. Renderer

The renderer manages the playing policies and it is strictly application dependent. One of its main task is, for instance, to allow users to switch between global or local playing method. With the global methods, the system tends to play an entire song before to make a transition to the new one. With the local one, instead, the rendering is cell-dependent and each grain is considered as an independent excerpt of content.

The Renderer is also responsible for implementing between-cell transition methodologies: cross-fading, BPM (beats per minutes) alignment, etc.

D. Application Example

A possible application scenario is that of a music stream generation for a club-like environment. The system generates and updates a playlist based on intentional controls on the part of the DJ, who interacts with the GUI (graphic user interface) through gestures to control the BPM or other parameters, and based on other environmental gestures. The application, trough the tracking of blob activities on the dance floor and using classification system, is able to extract mood information of the audience and the playlist can be conditioned by the extracted mood. The system also takes into account the past history and the renderer can be chosen in order to automatically perform tempo and loudness cross-fading between songs or cells. An extension to video application, synchronized with audio, can be included.

IV. An Application to music playlist generation

As a possible instance of the proposed dynamic playlist generation system, a dynamic music playlist generation has been implemented. The application we propose follows the general schema described in Section III. Hence, with reference to fig. 1, the application is composed by a *playlist generator*, a *controller* and a *renderer*. Although, for computational reason, the feature extraction phase is not performed in real-time but as a preprocessing step. For that reason an *analyzer* is added.

A. Analyzer

During a preprocessing phase, the database of musical content is analyzed in order to perform grain segmentation and feature extraction. For each grain the set of features is extracted and saved in a XML structured file.

Segmentation: the definition of the anchors is performed through texture analysis based on peak detection over the spectrum *novelty* function [9], which determines the temporal locations of significant texture changes. The *novelty* curve has been obtained by the convolution of the similarity matrix, resulted by the correlation between all pairs of frames of the spectrogram of the signal, along with the main diagonal and a *Chequerboard Gaussian filter*. A Chequerboard Gaussian kernel is a point to point multiplication between the two-dimensional Gaussian function and the following function

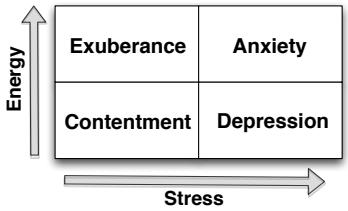

Fig. 2. Discrete Mood bi-dimensional plane

$$f(x,y) = \begin{cases} +1 & \text{if } sign(x) = sign(y) \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

Feature Extraction: in order to give a description of each cell, a set of highly discriminant features of intuitive interpretation is defined

- *Harmony*: identified by a *key*, a *mode* and the *keyClarity*, is defined as the confidence of the detected harmony ([10] and [11]). It is extracted from the analysis of the *Wrapped Chromagram* that represents the energy distribution along the pitches or pitch classes, as states in [12]. The method used to determine the harmony feature is based on computing the key strength (key clarity), which is the maximum value in the cross-correlation graph performed on the obtained Chromagram and the Chromagram of all possible key candidates.
- *RMS*: computed by taking the root average of the square of the amplitude.
- *Brightness*: defined as the power of the high-frequency bands with respect to the lower-frequency ones [13][12]. Given $X(\omega)$ the power spectrum the Brightness feature is defined as:

$$b = \frac{\int_0^t X(\omega)}{\int_t^\infty X(\omega)} \quad (3)$$

  where t is a threshold, here set at $1500Hz$.
- *Tempo*: summarizes the main beat frequency of the musical piece (in BPM) and is estimated by detecting the periodicities in the onset curve [12]
- *Mood*: is a high-level feature describing the emotional content of a musical piece. [14]. In the system, we used a bi-dimensional four discrete states representation of the mood as shown in Figure 2. The vertical axis (*Energy*) is related to the strength of the signal detected through the intensity features. The horizontal axis (*Stress*) indicates whether the emotion is positivity and it is calculated using

timbric features. A set of three Support Vector Machine (SVM) is used to classify the audio data in the plane; first of all, the items are first classified to determine the Energy and then, for each class, to determine the Stress. The features are extracted in 32ms-long frames. According to the work by Lartillot et al. [12] intensity feature is *RMS* and timbric features are: *Spectral Centroid, Spectral roll-off, Spectral Flux, Inharmonicity, MFCC*. The SVMs have been trained using (Contentment, Exuberance, Anxiety, Depression) a set of 20, 5 seconds-long music excerpts for each of the four classes, according to the following characteristics:

- *Contentment*: quiet music with a positive emotional content
- *Exuberance*: loud music with a positive emotional content
- *Anxiety*: loud music with a negative emotional content
- *Depression*: quiet music with a negative emotional content

In order to extract the features values, a 250ms-long frame has been considered for Harmony, RMS and Brightness and 16s-long frame for Tempo. An Hamming windowing technique considering 50% overlap has been apply in both cases.

B. Playlist Generator

The playlist of ranked proposals is generated in real-time based on features described above. Musical cells that best fit feature set are chosen and ranked according to a similarity function and added to the playlist. Given the cells C_k and C_p, a similarity function $c(C_k, C_p)$ is introduced in order to compute a distance measure based on feature values. Given that the co-domains D of the feature functions are not always comparable, specific similarity metrics for features is needed: $c_f(F_f(C_k), F_f(C_p)) : DxD \mapsto [0, 1]$.

- *harmony metric*: is defined as the composition of two functions: the *key similarity function* and the *key clarity similarity function*. The *key similarity function* captures the harmonic distance between two keys and it is based on perceptual distance between chords as defined in [15]. The *key clarity* gives a measure of the amount of inharmonic noise present in the musical segment: a lower *key clarity* value describes a higher presence of noise. If this value is very high, a strong harmonic component is perceived by the listener. On the contrary, when this value is low, the segment does not present a well-defined harmony. As a result, as shown in 3, when two segments have both a high key clarity, the overall harmony similarity function should take much in account the key similarity measure. However, when two segments have both a low key clarity, the actual value of the key is not important since the value has a low confidence.
  In order to obtain the behavior of the function shown in fig. 3 the *harmony metric* computed on two musical cells is defined as:

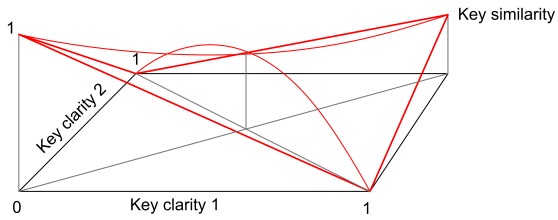

Fig. 3. The qualitative graph of harmony similarity

$$c_{harm} = d_1 + (1 - d_1) \cdot d_s \tag{4}$$

where

$$d_1 = (1 - F_{KC}(C_k)) \cdot (1 - F_{KC}(C_p)) \tag{5}$$
$$d_2 = c_K \cdot (1 - |F_{KC}(C_k) - F_{KC}C_(p)|)(1 - F_{KC}(C_p)) \tag{6}$$

where F_{KC} is the *keyclarity* value and c_K is the *key similarity function*.

- *RMS metric (c_R)* and *brightness metric (c_B)*: are defined the euclidean distance

$$c_R = 1 - |F_R(C_k) - F_R(C_p)| \tag{7}$$
$$c_B = 1 - |F_B(C_k) - F_B(C_p)| \tag{8}$$

where F_R is the *RMS* value and F_B the *brightness* value.

- *tempo metric*: is computed as a normalized Gaussian function centered in one of the two BMP values and with an appropriate variance. It returns high values when the two tempos are near and reasonably decreasing values when they move away.
- *mood metric*: based on subjective distance, it has been defined as in [14]. It is summarized in Table I.

	Exuberance	Anxious	Contentment	Depression
Exuberance	1.00	0.40	0.33	0.10
Anxious	0.40	1.0	0.10	0.33
Contentment	0.33	0.10	1.0	0.40
Depression	0.10	0.33	0.40	1.0

TABLE I
MOOD SIMILARITY TABLE

In order to obtain an overall similarity value that takes into account all metrics, a weighted average is computed:

$$c(C_k, C_p) = \frac{\sum_f w_f \cdot c_f(F_f(C_k), F_f(c_p))}{\sum_h w_h} \tag{9}$$

where w_h is the weight of feature h.
During the real-time phase, the playlist is dynamically regenerated each time user preferences, expressed on feature values, vary, each time a new contextual conditions occurs or each time a new segment starts playing. The resulting playlist is a ranked list of pairs $\{S_i, [start_i, end_i]\}$, where S_i is a song and $[start_i, end_i]$ is an interval inside S_i that satisfy the user constraints. The ranking process performed to

order retrieved proposals it is performed as the composition of the similarity measure define in equation 9 and the play history. The system in fact embodies a learning based system to capture the user tastes. Each time the user performs a choice the chosen music cell is added to the history. A Gaussian Mixture Model (GMM) is periodically trained over it and the resulting GMM model is then used.

C. Controller

Due to the highly dynamic interaction with the system the human-interaction mechanism turns out to be a very important component. In order to control the playlist generator with both intentional and unintentional control signal, some interaction systems have been implemented. Control signals are mapped over RMS, brightness, tempo and mood features as defined in section IV-A. Harmony is only used as a strong constraint in playlist generation: brutal harmony changes are not encouraged. Here implemented controller:

- *Screen interface*: the user interacts with the software through a traditional screen GUI (Figure 4).

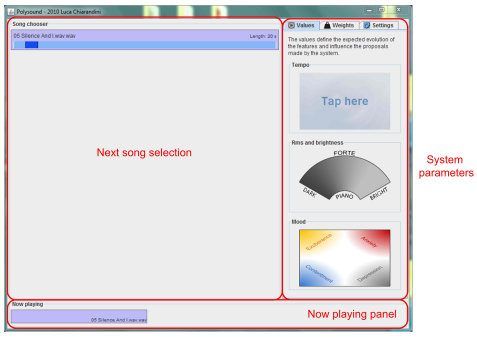

Fig. 4. Screen interface

  Implemented in Java, it is used to edit the real-time parameters in a precise way. In addition, it allows users to choose the next song to be played among a list of proposal made by the system.
- *Tapping interface* A special input device has been developed to allow the user to specify a musical tempo value by sending a set of impulses to the system (tapping on a membrane, pressing a button, etc.) which gives an stimation of the BPM value based on the impulse frequency.
- *Tangible interface*: similarly to what happens in the ReacTable ([16]), the system can be controlled through a set of tangible objects on a tabletop whose position is detected by a video device (web-cam); the user can control the system by moving the object on the table or turning them. In addition to that, the interface is equipped with a finger tracking system.
- *Wii Remote interface*: the software can be controlled using the Wii remote, a Bluetooth wireless controller used in the Nintendo Wii Console. The remote is equipped with an accelerometer and a set of buttons; the accelerometer is used to detect the frequency in which the user moves the controller and set the tempo features, the button are used for both selecting the next audio item and setting the RMS and brightness features. In order to use the system as personal trainer for running, a real application has been tested using the Wii remote to capture the running step frequency and mapping it to tempo feature. The Wii remote was held by the runner.

D. Renderer

Every time a new playlist needs to be generated, the renderer visits the XML database and extracts all songs containing the cell that best fit new parameters. The ranked list is returned to the interface to be properly visualized.

Rhythmic features are known to be very important in creating a smooth transition [17] therefore, once a new segment is selected to be played, a transition between the current playing and the new one is performed. A *time-scale* transformation to gradually adjust the tempo of the transition between the two items is performed. Moreover it synchronizes the beats using a peak detection in the correlation function between the two signals items.

V. Evaluation

In order to assess the performance of the system we tested it in various working conditions using different types of stimuli, with satisfying results. The system proved to be responsive and quite flexible, as it could be easily modified to fit different application scenarios. As a matter of fact, after implementing the *Virtual DJ* application, purposing it for a treadmill speed-control application driven by heartbeat and running pace was rather straightforward.

The Virtual DJ scenario proved to be the most interesting one, as it enabled us to accommodate multimodal controls at different levels of abstraction. On this we run some subjective evaluation tests involving 80 people (80% of which were 15 to 30 years old). After a short briefing on its usage, each user tested the system for about 10 minutes and filled out a questionnaire. Learning process required between 3 and 6 minutes to be proficient of the system showing a good ease of use of the application. To evaluate independence from rendering, which is not the main goal of the work, we preliminary did a qualitative test to make sure transitions were seamless handled by the renderer. The most listeners deemed the renderer of high quality. We proceeded with the subsequent evaluation for the playlist generation through the quality of the ranked recommendation. The users generally found the system to update the playlist in a correct fashion with a good consensus: more than 98% said medium or better; more than 71% said good at least and more than 7% said excellent. Good consensus was also reached on evaluations of responsiveness. In general there seems to be a preference from users towards cell-wise analysis here proposed, rather than global (song-wise) analysis, even if, as shown in fig. 6 they are skeptical in using the application for personal usage. They consider it a good Virtual DJ for club and disco. This is probably due to the still high number of parameters.

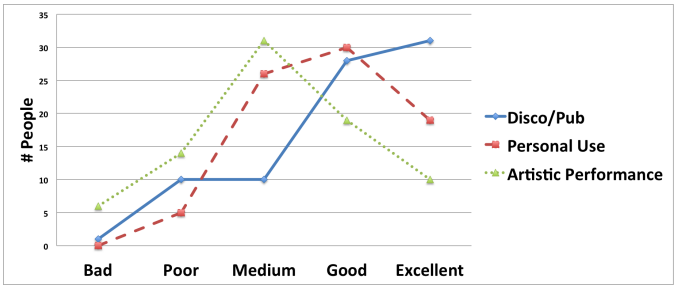

Fig. 5. Application fields

VI. Conclusion and future developments

In this paper we proposed a general approach for dynamic content-based real-time multimedia playlist generation driven by multimodal control signals and features, and showed an implementation of this approach for the application scenario of the Virtual DJ. In this implementation playlists were generated on the fly on a fine-grained time basis to promote changes in the audio excerpt possibly half-way through, where needed. This local approach proved to correctly adapt to external events and mood swings in a very reactive fashion, and to correctly account for inherent changes in the mood of the running excerpt. The proposed system proved able to learn from the choices of the user and offered a variety of interaction mechanism, depending on the nature of the control. The evaluation of the results of a questionnaire proposed to a number of users shows that the system proved user-friendly and intuitive, as well as effective.

Many are the possible uses of this approach. Examples are

- *Automatic DJ systems*: automatic generation of a musical stream that adapts to the reactions of the listeners, to the changes of mood of the running songs and to other environmental conditions, or that account for possible feedbacks in a broadcast internet radio scenario
- *Gesture-controlled recommendation system*: we can envision using devices such as tangible interfaces (e.g. a ReacTable) or webcams to extract descriptors of the user's gestures and infer his/her emotional status.
- *Music-driven sports trainer*: many smartphones are today equipped with accelerometers or other motion sensors. This can be used for detecting the step frequency while running and adapt the music tempo to this value. We can envision biological sensors that can easily extract information on the heartbeat or other bio-parameter and use such parameters to affect the training.

Future developments concern the implementation of multimodal playlist generation using both audio and video features as well as audio-video rendering. Further work concerns gesture analysis for feature extraction and control, as well as usability studies.

References

[1] L. Barrington, R. Oda, and G. Lanckriet, "Smarter than genius? human evaluation of music recommender systems," in *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, Kobe, Japan, October 2009.

[2] P. Herrera, Z. Resa, and M. Sordo, "Rocking around the clock eight days a week: an exploration of temporal patterns of music listening," in *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010*, Barcelona, Spain, September 2010.

[3] H. Lu, N. Lane, T. Choudhury, and A. Campbell, "Soundsense: Scalable sound sensing for people-centric applications on mobile phones," in *7th international conference on Mobile systems, applications, and services (MobiSys '09)*, Krakw, Poland, June 2009.

[4] N. Masahiro, H. Takaesu, H. Demachi, M. Oono, and H. Saito, "Development of an automatic music selection system based on runner's step frequency," in *9th International Society for Music Information Retrieval Conference (ISMIR 2008)*, Philadelphia, Pennsylvania USA, September 2008.

[5] P. Herrera, J. Bello, G. Widmer, M. Sandler, O. Celma, F. Vignoli, E. Pampalk, P. Cano, S. Pauws, and X. Serra, "Simac: Semantic interaction with music audio contents," in *2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, London, Great Britain, 2005.

[6] M. Shan, F. Kuo, and S. Lee, "Emotion-based music recommendation by affinity discovery from film music," *Expert Systems with Applications: An International Journal archive*, vol. 36, no. 4, May, 2009.

[7] A. Camurria, G. D. Polib, A. Fribergc, M. Lemand, and G. Volpea, "The mega project: Analysis and synthesis of multisensory expressive gesture in performing art applications," *Journal of New Music Research*, vol. 34, no. 1, pp. 5 – 21, 2005.

[8] D. Bogdanov, M. Haro, F. Fuhrmann, E. Gomez, and P. Herrera, "Content-based music recommendation based on user preference examples," in *The 4th ACM Conference on Recommender Systems. Workshop on Music Recommendation and Discovery (Womrad 2010)*, Barcelona, Spain, September 2010.

[9] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of International Conference on multimedia and Expo (ICME)*, New York City, NY, USA, Augusttember 2000.

[10] Krumhansl, *Cognitive foundations of musical pitch*. Oxford UP, 1990.

[11] E. Gomez, "Tonal description of music audio signal," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[12] O. Lartillot and P. Toiviainen, "A matlab toolbox for musical feature extraction from audio," in *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, 2007.

[13] P. N. Juslin, "Cue utilization in communication of emotion in music performance: relating performance to perceptions," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 26, no. 6, pp. 1797–1813, 2000.

[14] D. Liu, L. Lu, and H. Zhang, "Automatic mood detection from acoustic music data," in *Proc. of the International Symposium on Music Information Retrieval (ISMIR 2003)*, Baltimore, Maryland (USA), 2003.

[15] H. Purwins, "Profiles of pitch classes circularity of relative pitch and key experiments, models, computational music analysis, and perspectives," Ph.D. dissertation, Technischen Universitt Berlin, Berlin, Germany, 2005.

[16] S. Jord, M. Kaltenbrunner, G. Geinger, and R. Bencina, "The reactable," in *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005, pp. 579–582.

[17] H. Lin, Y. Yin, M. Tien, and J. Wu, "Music paste: Concatenating music clips based on chroma and rhythm features," in *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, Kobe, Japan, October 2009.