ESSENTIA: AN AUDIO ANALYSIS LIBRARY FOR MUSIC INFORMATION RETRIEVAL

Dmitry Bogdanov[1], **Nicolas Wack**[2], **Emilia Gómez**[1], **Sankalp Gulati**[1], **Perfecto Herrera**[1]
Oscar Mayor[1], **Gerard Roma**[1], **Justin Salamon**[1], **José Zapata**[1] **and Xavier Serra**[1]
Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[1]{name.surname}@upf.edu
[2]essentia@wackou.otherinbox.com

ABSTRACT

We present Essentia 2.0, an open-source C++ library for audio analysis and audio-based music information retrieval released under the Affero GPL license. It contains an extensive collection of reusable algorithms which implement audio input/output functionality, standard digital signal processing blocks, statistical characterization of data, and a large set of spectral, temporal, tonal and high-level music descriptors. The library is also wrapped in Python and includes a number of predefined executable extractors for the available music descriptors, which facilitates its use for fast prototyping and allows setting up research experiments very rapidly. Furthermore, it includes a Vamp plugin to be used with Sonic Visualiser for visualization purposes. The library is cross-platform and currently supports Linux, Mac OS X, and Windows systems. Essentia is designed with a focus on the robustness of the provided music descriptors and is optimized in terms of the computational cost of the algorithms. The provided functionality, specifically the music descriptors included in-the-box and signal processing algorithms, is easily expandable and allows for both research experiments and development of large-scale industrial applications.

1. INTRODUCTION

There are many problems within the Music Information Research discipline which are based on audio content and require reliable and versatile software tools for automated music analysis. Following the research necessities, different audio analysis tools tailored for MIR have been developed and used by academic researchers within the last fifteen years. These tools include the popular MIRtoolbox [39] and MARSYAS [69], jAudio [48], jMIR [47], Aubio [11], LibXtract [12], yaafe, [1] Auditory Toolbox [60], and CLAM Music Annotator [2]. Furthermore, a number of Vamp plugins [2] were developed by different researchers for computation and visualization of music descriptors using hosts such as Sonic Visualiser software [13], Sonic Annotator, [3] and Audacity. [4] Apart from these software tools, there is the online service Echonest, [5] which provides a collection of audio features for any uploaded track via an API. This service, however, does not disclose the implementation details of the features offered to a user, which might be a significant limitation at least in the context of academic research. The above-mentioned tools provide different limited sets of descriptors, and require different programming languages (Matlab, C++, Java) and software environments (standalone GUI-based or command-line applications, running within a Vamp host) to be extracted. The feature sets vary from tool to tool and one may have to combine different tools in order to obtain the desired expanded combination of features. The comparative overview of the majority of the available tools is presented in [50]. Surely, all the above-mentioned tools are very important for the research community, nevertheless we believe there is a lack of generic robust tools which provide more comprehensive sets of state-of-the-art music descriptors and are optimized for faster computations on large collections.

2. ESSENTIA 2.0

We present Essentia 2.0, an extensive open-source library for audio analysis and audio-based music information retrieval released under the Affero GPL [6] license and well-suited for both research and industrial applications. [7] In its core, Essentia is comprised of a reusable collection of algorithms to extract features from audio. The available algorithms include audio file input/output functionality, standard digital signal processing (DSP) building blocks, filters, generic algorithms for statistical characterization, and spectral, temporal, tonal and high-level music descriptors. The library is written in C++, which provides considerable performance benefits. Importantly, it also includes Python bindings to facilitate the usage of the library for the users

[1] http://sourceforge.net/projects/yaafe

[2] http://vamp-plugins.org/download.html
[3] http://omras2.org/sonicannotator
[4] http://audacity.sourceforge.net
[5] http://developer.echonest.com
[6] http://gnu.org/licenses/agpl.html
[7] Commercial licensing is offered in addition to the Affero GPL.

who are familiar with the matlab/python environment. Using Essentia's dedicated python modules, one can rapidly get familiar with the available algorithms and design research experiments, explore and analyze data on-the-fly. In addition, the majority of the MIR descriptors' algorithms are wrapped into a Vamp [8] plugin and can be used with the popular Sonic Visualiser software [13] for visualization of music descriptors.

The design of Essentia is focused on robustness, time and memory performance, and ease of extensibility. The algorithms are implemented keeping in mind the use-case of large-scale computations on large music collections. One may develop his own executable utility with a desired processing flow using Essentia as a C++ library. Alternatively a number of executable extractors are included with Essentia, covering a number of common use-cases for researchers, for example, computing all available music descriptors for an audio track, extracting only spectral, rhythmic, or tonal descriptors, computing predominant melody and beat positions, and returning the results in yaml/json data formats.

Essentia has been in development for more than 6 years incorporating the work of more than 20 researchers and developers through its history. The 2.0 version contains the latest refactoring of the library, including performance optimization, simplified development API, and a number of new descriptors such as the state-of-the-art beat tracking [16, 74, 75] and predominant melody detection [56] algorithms. In addition, Essentia can be optionally complemented with Gaia, [9] a library released under the same license, which allows to apply similarity measures and classifications on the results of audio analysis, and generate classification models that Essentia can use to compute high-level description of music. Gaia is a C++ library with python bindings for working with points in high-dimensional spaces, where each point represents a song and each dimension represents an audio feature. It allows to create datasets of points, apply transformations (gaussianization, principal component analysis, relevant component analysis, classification with support vector machines), and compute custom distance functions. The functionality of Gaia can be used to implement search engines relying on item similarity of any kind (not limited to music similarity).

3. OVERALL ARCHITECTURE

The main purpose of Essentia is to serve as a library of signal-processing blocks. As such, it is intended to provide as many algorithms as possible, while trying to be as little intrusive as possible. Each processing block is called an Algorithm, and it has three different types of attributes: inputs, outputs and parameters. Algorithms can be combined into more complex ones, which are also instances of the base Algorithm class and behave in the same way. An example of such a composite algorithm is presented in Figure 1. It shows a composite tonal key/scale extractor based on [28], which combines the algorithms for frame cutting, windowing, spectrum computation, spectral peaks detection, chroma features (HPCP) computation and finally the algorithm for key/scale estimation from the HPCP (itself a composite algorithm).

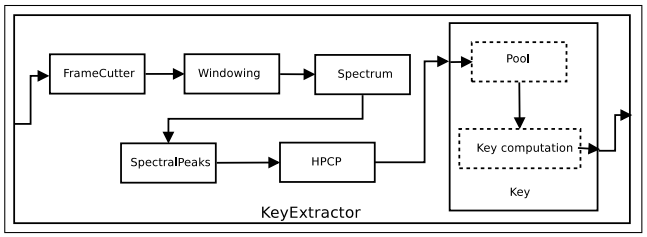

Figure 1. An example of the composite algorithm KeyExtractor combining the algorithms FrameCutter, Windowing, Spectrum, SpectralPeak, HPCP, and the Key algorithm composite itself.

The algorithms can be used in two different modes: *standard* and *streaming*. The standard mode is imperative while the streaming mode is declarative. The standard mode requires to specify the inputs and outputs for each algorithm and call their processing function explicitly. If the user wants to run a network of connected algorithms, he/she will need to manually run each algorithm. The advantage of this mode is that it allows very rapid prototyping (especially when the python bindings are coupled with a scientific environment in python, such as ipython, numpy, and matplotlib).

The streaming mode, on the other hand, allows to define a network of connected algorithms, and then an internal scheduler takes care of passing data between the algorithms inputs and outputs and calling the algorithms in the appropriate order. The scheduler available in Essentia is optimized for analysis tasks, and does not take into account the latency of the network. For real-time applications, one could easily replace this scheduler with another one that favors latency over throughput. The advantage of this mode is that it results in simpler and safer code (as the user only needs to create algorithms and connect them, there is no room for him to make mistakes in the execution order of the algorithms), and in lower memory consumption in general, as the data is streamed through the network instead of being loaded entirely in memory (which is the usual case when working with the standard mode).

Even though most of the algorithms are available for both the standard and streaming mode, the code that implements them is not duplicated as either the streaming version of an algorithm is deduced/wrapped from its standard implementation, or vice versa.

4. ALGORITHMS

In this section we briefly review the most important algorithms included in the library. The interested reader is referred to the documentation for a complete reference for all available algorithms. [10] A great part of them computes a variety of low-level, mid-level, and high-level descrip-

[8] http://vamp-plugins.org

[9] http://github.com/MTG/gaia

[10] http://essentia.upf.edu

tors useful for MIR. In addition there are tools for working with audio input/output and processing, and gathering data statistics.

4.1 Audio input/output and filtering

Essentia has a variety of audio loaders to provide a very convenient way to load audio files from disk. These loaders output the stream of stereo or mono samples using the FFmpeg [11] /Libav [12] libraries. Almost all existing audio formats are supported, and additionally audio can be loaded from video files or even Flash files. It is possible to downmix a file to mono, resample it, trim the audio to a given start/end time, normalize the resulting samples using a given ReplayGain [13] value, and apply an Equal-loudness filter [14] to the resulting audio. A special loader reads the metadata tags stored in the given file (e.g., ID3 tags). Essentia can also write audio files to any format supported by FFmpeg/Libav, and may add beeps to the audio according to the given (onset) time positions.

In addition, Essentia provides algorithms for basic processing of audio streams: it allows to apply replay gain, frame cutting, windowing, resampling, FFT, auto-correlation computation, etc. A variety of audio filters are implemented in Essentia, including the generic IIR filtering, first and second order low/band/high/all-pass filtering and band rejection, DC component removal, moving average filter, and the filter approximating an equal-loudness curve.

4.2 Spectral descriptors

A variety of algorithms for computation of low-level spectral descriptors is included. In particular, they compute:

- the energy of the given frequency band or the set of bands of a spectrum, the Bark band energies [51] of a spectrum, the Mel band energies and the Mel-frequency cepstral coefficients of a spectrum using the MFCC-FB40 algorithm [21];
- the ERB band energies of a spectrum [49] and the Gammatone feature cepstral coefficients [58] similar to MFCCs;
- the Linear Predictive Coding coefficients and the associated reflection coefficients [43];
- the spectral flux [17, 68];
- the high-frequency content (HFC) measure [33, 45], the roll-off frequency of a spectrum [51], and the spectral contrast feature [1];
- the spectral peaks and the spectral complexity [41], the inharmonicity and dissonance measures [51, 52], and the Strong Peak of a spectrum [23];
- the stereo panorama distribution [30];
- spectral whitening based on spectrum envelope estimation in [54].

[11] http://ffmpeg.org
[12] http://libav.org
[13] http://wiki.hydrogenaudio.org/index.php?title=ReplayGain_1.0_specification
[14] http://replaygain.hydrogenaudio.org/proposal/equal_loudness.html

4.3 Time-domain descriptors

A number of algorithms for computation of time-domain descriptors are included. They compute the total duration and the duration of the perceptually meaningful part of the signal above a certain energy level [51], Zero-crossing rate [51], and loudness estimations based on the Stevens' power law [65], the LARM model [59], the Equivalent sound level (Leq) [64], and the Vickers' model [70].

4.4 Tonal descriptors

A number of algorithms for computation of tonal descriptors are included. In particular, they provide:

- the pitch salience function of a signal, the estimation of the fundamental frequency of the predominant melody by the MELODIA algorithm [56], and the pitch estimation by the YinFFT method [11];
- the Harmonic Pitch-Class Profile (HPCP) of a spectrum (also called chroma features) [28], the tuning frequency [27], the key and the scale of a song [28];
- the sequence of chords present in a song [28], chords histogram, chords change rate, key and scale of the most frequent chord;
- the tristimulus [51] and the ratio of a signal's odd to even harmonic energy [44] computed based on harmonic peaks;
- the tonic frequency of the lead artist in Indian art music [55];
- a set of descriptors derived from high-resolution (10 cents) HPCP features related to tuning system or scale and used for comparative analysis of recordings from Western and non-Western traditions [29].

4.5 Rhythm descriptors

A number of the algorithms are related to a rhythmic representation of the audio signal. They include:

- the beat tracker based on the complex spectral difference feature [16], the multifeature beat tracker [74, 75] (which combines 5 different beat trackers taking into account the maximum mutual agreement between them), and the beat tracker based on the BeatIt algorithm [22];
- an extractor of the locations of large tempo changes from a list of beat ticks;
- the statistics of the BPM histogram of a song;
- the novelty curve for the audio signal, and the BPM distribution and tempogram based in it [25];
- onset detection functions for the audio signal including HFC [33, 45], Complex-Domain spectral difference [4] and its simplified version [10], spectral flux and Mel-bands based spectral flux [18], overall energy flux [38], spectral difference measured by the modified information gain, and the beat emphasis function [15], and the list of onsets in the audio signal given a list of detection functions [10];
- rhythm transform [26] (kind of the FFT of the MFCC representation);
- beat loudness (the loudness of the signal on windows centered around the beat locations).

4.6 SFX descriptors

A number of the algorithms are intended to be used with short sounds instead of full-length music tracks. They return the logarithm of the attack time for the sound, a measure of whether the maximum/minimum value of a sound envelope is located towards its beginning or end, the pitch salience, the normalized position of the temporal centroid of a signal envelope, the Strong Decay [23], the estimation of flatness of a signal envelope and descriptors based on its derivative.

4.7 Other high-level descriptors

In addition to the low-level descriptors, Essentia also contains the following mid- and high-level descriptors:

- the "danceability" of a song based on the Detrended Fluctuation Analysis [67], the intensity of the input audio signal (relaxed, moderate, or aggressive), and the dynamic complexity related to the dynamic range and the amount of fluctuation in loudness [66];
- the fade-ins/fade-outs present in a song and audio segmentation using the Bayesian Information Criterion [24];
- the principal component analysis and Gaia transformations to a set of descriptors, in particular, classification using the models pre-trained in Gaia. Currently, the following classifier models are available: musical genre (4 different databases), ballroom music classification, moods (happy, sad, aggressive, relaxed, acoustic, electronic, party), western/non-western music, live/studio recording, perceptual speed (slow, medium, fast), gender (male/female singer), dark/bright timbre classification, and speech/music. The approximate accuracies of these models are presented in [5].

4.8 Statistics

The algorithms for computing statistics over an array of values, or some kind of aggregation, are also provided. These algorithms allow to compute:

- the mean, geometric mean, power mean, median of an array, and all its moments up to the 5th-order, its energy and the root mean square (RMS);
- flatness, crest and decrease of an array, typically used to characterize the spectrum [51];
- variance, skewness, kurtosis of a probability distribution, and a single Gaussian estimate for the given list of arrays (returns the mean array, its covariance and inverse covariance matrices).

4.9 Extractors

As Essentia algorithms can themselves be composed of multiple algorithms, a few useful extractors have been written as algorithms. Given an audio track, they compute the loudness, the tuning frequency, all tonal information (key, scale, chords sequence, chords histogram, etc), the BPM and beat positions of a music track as well as other rhythm-related features, a variety of low-level features with/without an applied equal-loudness filter, and the overall track description with the majority of the available low-level, mid-level and high-level features.

In addition, a number of executable extractors are included with the library as examples of its application. These standalone examples can be used straight away for batch computation of descriptors with no need to dive into the API of the library. They include a number of specific extractors (predominant melody, beat tracking, key, MFCCs, etc.) as well as generic extractors returning the majority of the available descriptors in yaml/json formats. For industrial applications, one may need to decide on the descriptors and type of processing required, and implement his own extractor.

5. APPLICATIONS

Essentia has served in a large number of research activities conducted at Music Technology Group since 2006. It has been used for music classification [6, 71, 72], and in particular for mood classification [40, 41], semantic autotagging [61, 73], music similarity and recommendation [5, 6, 9, 14], visualization and interaction with music [5, 6, 34, 42, 63], sound indexing [31, 32, 53], detection of musical instruments in polyphonies [19], cover detection [57], instrument solo detection [20], and acoustic analysis of stimuli for neuroimaging studies [37]. Currently, it is actively used within the CompMusic [15] research project [35, 36]. The systems based on Essentia/Gaia have been enrolled in the the Music Information Retrieval Evaluation eXchange (MIREX) campaigns for the tasks of music classification [71,72], music similarity [7,8], autotagging [62], and beat detection [3], and they have usually ranked among the best ones.

Essentia and Gaia have been used extensively in a number of research projects, [16] including the CANTATA EU (recommendation system for music videos), SALERO EU (search engine for sound effects), PHAROS EU (music search and recommendation), Buscamedia, PROSEMUS and DRIMS (automated low-level and high-level semantic description of music), and SIEMPRE EU [46] (audio analysis for multimodal databases of music performances).

Furthermore, previous versions of Essentia have been exploited for industrial applications: it has been used in the non-commercial sound service Freesound [17] (large-scale content-based search of sound recordings), industrial products by BMAT [18] and Stromatolite [19] (music recommendation), Yamaha's BODiBEAT (automatic playlist generation for runners), and Steinberg's LoopMash (audio-based sample matching for music production).

In addition to the off-line audio analysis we expect our library to be potential for real-time applications. However

[15] `http://compmusic.upf.edu`

[16] Detailed information about the research projects can be found online: `http://mtg.upf.edu/research/projects`

[17] `http://freesound.org`

[18] `http://bmat.com`

[19] `http://stromatolite.com`

not all of the present algorithms can be used for such applications due to their computational complexity.

6. CONCLUSIONS

We have presented a cross-platform open-source library for audio analysis and audio-based music information research and development, Essentia 2.0. The library is versatile and may suit the needs of both researchers within MIR community and the industry. In our future work we will focus on expanding the library and the community of users, We plan to add new music descriptors, in particular, adding new semantic categories to the set of high-level classifier-based descriptors, and update the library for real-time applications. All active Essentia users are encouraged to contribute to the library.

The detailed information about Essentia is located at the official web page. [20] It contains the complete documentation for the project including the installation instructions. The source code is available at the official Github repository. [21]

7. ACKNOWLEDGMENTS

The work on Essentia has been partially funded by the PHAROS (EU-IP, IST-2006-045035), Buscamedia (CEN-20091026), CANTATA (ITEA 05010, FIT-350300-2006-33), SIGMUS (TIN2012-36650), CompMusic (ERC 267583), and TECNIO (TECCIT12-1-0003) projects. We acknowledge all the developers who took part in working on this project and Alba Rosado for her help in coordination.

8. REFERENCES

[1] V. Akkermans, J. Serrà, and P. Herrera. Shape-based spectral contrast descriptor. In *Sound and Music Computing Conf. (SMC'09)*, page 143–148, 2009.

[2] X. Amatriain, J. Massaguer, D. Garcia, and I. Mosquera. The clam annotator: A cross-platform audio descriptors editing tool. In *Int. Conf. on Music Information Retrieval (ISMIR'05)*, volume 5, 2005.

[3] E. Aylon and N. Wack. Beat detection using plp. In *Music Information Retrieval Evaluation Exchange (MIREX'10)*, 2010.

[4] J. P. Bello, C. Duxbury, M. Davies, and M. Sandler. On the use of phase and energy for musical onset detection in the complex domain. *Signal Processing Letters, IEEE*, 11(6):553–556, 2004.

[5] D. Bogdanov. *From music similarity to music recommendation: Computational approaches based on audio and metadata analysis*. PhD thesis, UPF, Barcelona, Spain, 2013. In press.

[6] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, and P. Herrera. Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing & Management*, 49(1):13–33, Jan. 2013.

[7] D. Bogdanov, J. Serrà, N. Wack, and P. Herrera. Hybrid similarity measures for music recommendation. In *Music Information Retrieval Evaluation Exchange (MIREX'09)*, 2009.

[8] D. Bogdanov, J. Serrà, N. Wack, and P. Herrera. Hybrid music similarity measure. In *Music Information Retrieval Evaluation Exchange (MIREX'10)*, 2010.

[9] D. Bogdanov, J. Serrà, N. Wack, P. Herrera, and X. Serra. Unifying low-level and high-level music similarity measures. *IEEE Trans. on Multimedia*, 13(4):687–701, 2011.

[10] P. Brossier, J. P. Bello, and M. D. Plumbley. Fast labelling of notes in music signals. In *Int. Symp. on Music Information Retrieval (ISMIR'04)*, page 331–336, 2004.

[11] P. M. Brossier. *Automatic Annotation of Musical Audio for Interactive Applications*. PhD thesis, QMUL, London, UK, 2007.

[12] J. Bullock and U. Conservatoire. Libxtract: A lightweight library for audio feature extraction. In *Int. Computer Music Conf. (ICMC'07)*, volume 9, 2007.

[13] C. Cannam, C. Landone, and M. Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *ACM Int. Conf. on Multimedia (MM'05)*, page 1467–1468, 2010.

[14] O. Celma, P. Cano, and P. Herrera. Search sounds an audio crawler focused on weblogs. In *7th Int. Conf. on Music Information Retrieval (ISMIR)*, 2006.

[15] M. E. P. Davies, M. Plumbley, and D. Eck. Towards a musical beat emphasis function. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09*, pages 61–64, 2009.

[16] N. Degara, E. A. Rua, A. Pena, S. Torres-Guijarro, M. E. Davies, and M. D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Trans. on Audio, Speech, and Language Processing*, 20(1):290–301, 2012.

[17] S. Dixon. Onset detection revisited. In *Int. Conf. on Digital Audio Effects (DAFx'06)*, volume 120, page 133–137, 2006.

[18] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[19] F. Fuhrmann and P. Herrera. Quantifying the relevance of locally extracted information for musical instrument recognition from entire pieces of music. In *Int. Society for Music Information Retrieval Conf. (ISMIR'11)*, 2011.

[20] F. Fuhrmann, P. Herrera, and X. Serra. Detecting solo phrases in music using spectral and pitch-related descriptors. *Journal of New Music Research*, 38(4):343–356, 2009.

[21] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Int. Conf. on Speach and Computer (SPECOM'05)*, volume 1, page 191–194, 2005.

[22] F. Gouyon. *A computational approach to rhythm description: Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. PhD thesis, UPF, Barcelona, Spain, 2005.

[23] F. Gouyon and P. Herrera. Exploration of techniques for automatic labeling of audio drum tracks instruments. In *MOSART: Workshop on Current Directions in Computer Music*, 2001.

[24] G. Gravier, M. Betser, and M. Ben. Audio segmentation toolkit, release 1.2. Technical report, 2010. Available online: https://gforge.inria.fr/frs/download.php/25187/audioseg-1.2.pdf.

[25] P. Grosche and M. Müller. A mid-level representation for capturing dominant tempo and pulse information in music recordings. In *Int. Society for Music Information Retrieval Conf. (ISMIR'09)*, page 189–194, 2009.

[26] E. Guaus and P. Herrera. The rhythm transform: towards a generic rhythm description. In *Int. Computer Music Conf. (ICMC'05)*, 2005.

[27] E. Gómez. Key estimation from polyphonic audio. In *Music Information Retrieval Evaluation Exchange (MIREX'05)*, 2005.

[28] E. Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304, 2006.

[29] E. Gómez and P. Herrera. Comparative analysis of music recordings from western and non-western traditions by automatic tonal feature extraction. *Empirical Musicology Review*, 3(3):140–156, 2008.

[30] E. Gómez, P. Herrera, P. Cano, J. Janer, J. Serrà, J. Bonada, S. El-Hajj, T. Aussenac, and G. Holmberg. Music similarity systems and methods using descriptors, 2009. WIPO Patent No. 2009001202.

[31] M. Haro, J. Serrà, P. Herrera, and A. Corral. Zipf's law in short-time timbral codings of speech, music, and environmental sound signals. *PLoS ONE*, 7(3):e33993, 2012.

[32] J. Janer, M. Haro, G. Roma, T. Fujishima, and N. Kojima. Sound object classification for symbolic audio mosaicing: A proof-of-concept. In *Sound and Music Computing Conf. (SMC'09)*, pages 297–302, 2009.

[20] http://essentia.upf.edu

[21] http://github.com/MTG/essentia

[33] K. Jensen and T. H. Andersen. Beat estimation on the beat. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, page 87–90, 2003.

[34] C. F. Julià and S. Jordà. SongExplorer: a tabletop application for exploring large collections of songs. In *Int. Society for Music Information Retrieval Conf. (ISMIR'09)*, 2009.

[35] G. K. Koduri, S. Gulati, P. Rao, and X. Serra. Raga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4):337–350, 2012.

[36] G. K. Koduri, J. Serrà, and X. Serra. Characterization of intonation in carnatic music by parametrizing pitch histograms. In *Int. Society for Music Information Retrieval Conf. (ISMIR'12)*, pages 199–204, 2012.

[37] S. Koelsch, S. Skouras, T. Fritz, P. Herrera, C. Bonhage, M. Kuessner, and A. M. Jacobs. Neural correlates of music-evoked fear and joy: The roles of auditory cortex and superficial amygdala. *Neuroimage*. In press.

[38] J. Laroche. Efficient tempo and beat tracking in audio recordings. *Journal of the Audio Engineering Society*, 51(4):226–233, 2003.

[39] O. Lartillot, P. Toiviainen, and T. Eerola. A matlab toolbox for music information retrieval. In C. Preisach, P. D. H. Burkhardt, P. D. L. Schmidt-Thieme, and P. D. R. Decker, editors, *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–268. Springer Berlin Heidelberg, 2008.

[40] C. Laurier. *Automatic Classification of Musical Mood by Content-Based Analysis*. PhD thesis, UPF, Barcelona, Spain, 2011.

[41] C. Laurier, O. Meyers, J. Serrà, M. Blech, P. Herrera, and X. Serra. Indexing music by mood: design and integration of an automatic content-based annotator. *Multimedia Tools and Applications*, 48(1):161–184, 2009.

[42] C. Laurier, M. Sordo, and P. Herrera. Mood cloud 2.0: Music mood browsing based on social networks. In *Int. Society for Music Information Retrieval Conf. (ISMIR'09)*, 2009.

[43] J. Makhoul. Spectral analysis of speech by linear prediction. *IEEE Trans. on Audio and Electroacoustics*, 21(3):140–148, 1973.

[44] K. D. Martin and Y. E. Kim. Musical instrument identification: A pattern-recognition approach. *The Journal of the Acoustical Society of America*, 104(3):1768–1768, 1998.

[45] P. Masri and A. Bateman. Improved modelling of attack transients in music analysis-resynthesis. In *Int. Computer Music Conf. (ICMC'96)*, page 100–103, 1996.

[46] O. Mayor, J. Llop, and E. Maestre. RepoVizz: a multimodal on-line database and browsing tool for music performance research. In *Int. Society for Music Information Retrieval Conf. (ISMIR'11)*, 2011.

[47] C. McKay and I. Fujinaga. jMIR: tools for automatic music classification. In *Int. Computer Music Conf. (ICMC'09)*, page 65–68, 2009.

[48] C. McKay, I. Fujinaga, and P. Depalle. jAudio: a feature extraction library. In *Int. Conf. on Music Information Retrieval (ISMIR'05)*, page 600–3, 2005.

[49] B. C. Moore and B. R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *The Journal of the Acoustical Society of America*, 74(3):750–753, 1983.

[50] K. R. Page, B. Fields, D. De Roure, T. Crawford, and J. S. Downie. Reuse, remix, repeat: the workflows of MIR. In *Int. Society for Music Information Retrieval Conf. (ISMIR'12)*, 2012.

[51] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO Project Report*, 2004. Available online: http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/.

[52] R. Plomp and W. J. M. Levelt. Tonal consonance and critical bandwidth. *The Journal of the Acoustical Society of America*, 38(4):548–560, 1965.

[53] G. Roma, J. Janer, S. Kersten, M. Schirosa, P. Herrera, and X. Serra. Ecological acoustics perspective for content-based retrieval of environmental sounds. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010.

[54] A. Röbel and X. Rodet. Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *Int. Conf. on Digital Audio Effects (DAFx'05)*, 2005.

[55] J. Salamon, S. Gulati, and X. Serra. A multipitch approach to tonic identification in indian classical music. In *Int. Society for Music Information Retrieval Conf. (ISMIR'12)*, 2012.

[56] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.

[57] J. Serrà, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Trans. on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.

[58] Y. Shao, Z. Jin, D. Wang, and S. Srinivasan. An auditory-based feature for robust speech recognition. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'09)*, pages 4625–4628, 2009.

[59] E. Skovenborg and S. H. Nielsen. Evaluation of different loudness models with music and speech material. In *The 117th AES Convention*, 2004.

[60] M. Slaney. Auditory toolbox. *Interval Research Corporation, Technical Report*, 10, 1998. Available online: http://www.tka4.org/materials/lib/Articles-Books/Speech%20Recognition/AuditoryToolboxTechReport.pdf.

[61] M. Sordo. *Semantic Annotation of Music Collections: A Computational Approach*. PhD thesis, UPF, Barcelona, Spain, 2012.

[62] M. Sordo, O. Celma, and D. Bogdanov. MIREX 2011: Audio tag classification using weighted-vote nearest neighbor classification. In *Music Information Retrieval Evaluation Exchange (MIREX'11)*, 2011.

[63] M. Sordo, G. K. Koduri, S. Şentürk, S. Gulati, and X. Serra. A musically aware system for browsing and interacting with audio music collections. In *The 2nd CompMusic Workshop*, 2012.

[64] G. A. Soulodre. Evaluation of objective loudness meters. In *The 116th AES Convention*, 2004.

[65] S. S. Stevens. *Psychophysics*. Transaction Publishers, 1975.

[66] S. Streich. *Music complexity: a multi-faceted description of audio content*. PhD thesis, UPF, Barcelona, Spain, 2007.

[67] S. Streich and P. Herrera. Detrended fluctuation analysis of music signals: Danceability estimation and further semantic characterization. In *The 118th AES Convention*, 2005.

[68] G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'99)*, page 103–106, 1999.

[69] G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Organised sound*, 4(3):169–175, 2000.

[70] E. Vickers. Automatic long-term loudness and dynamics matching. In *The 111th AES Convention*, 2001.

[71] N. Wack, E. Guaus, C. Laurier, R. Marxer, D. Bogdanov, J. Serrà, and P. Herrera. Music type groupers (MTG): generic music classification algorithms. In *Music Information Retrieval Evaluation Exchange (MIREX'09)*, 2009.

[72] N. Wack, C. Laurier, O. Meyers, R. Marxer, D. Bogdanov, J. Serra, E. Gomez, and P. Herrera. Music classification using high-level models. In *Music Information Retrieval Evaluation Exchange (MIREX'10)*, 2010.

[73] Y. Yang, D. Bogdanov, P. Herrera, and M. Sordo. Music retagging using label propagation and robust principal component analysis. In *Int. World Wide Web Conf. (WWW'12). Int. Workshop on Advances in Music Information Research (AdMIRe'12)*, 2012.

[74] J. Zapata, M. Davies, and E. Gómez. MIREX 2012: Multi feature beat tracker (ZDG1 and ZDG2). In *Music Information Retrieval Evaluation Exchange (MIREX'12)*, 2012.

[75] J. R. Zapata, A. Holzapfel, M. E. Davies, J. L. Oliveira, and F. Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *Int. Society for Music Information Retrieval Conf. (ISMIR'12)*, 2012.