ESSENTIA: an Open-Source Library for Sound and Music Analysis

Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera,
Oscar Mayor, Gerard Roma, Justin Salamon, José Zapata, Xavier Serra
Music Technology Group, Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
{name.surname}@upf.edu

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.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*Software libraries*; C.3 [**Special-Purpose and Application-Based Systems**]: [Signal processing systems]

General Terms

Algorithms, Design, Experimentation

MM'13, October 21–25, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2502081.2502229.

Keywords

Open Source, Signal Processing, Audio Analysis, Music Information Retrieval, Sound and music computing

1. INTRODUCTION

There are many problems within the Music Information Research (MIR) 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 and MARSYAS, jAudio, jMIR, Aubio, LibXtract, yaafe,[1] openSMILE [5], Auditory Toolbox, and CLAM Music Annotator (the comparative overview of the majority of the available tools is presented in [9]). 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 [4]. Apart from these software tools, there is the online service Echonest,[3] 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). 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 Matlab-based MIRtoolbox and C++ based MARSYAS provide the richest sets of spectral, time-domain, rhythmic, and tonal features. Tonal features are frequently missing in other existing tools (libXtract, jAudio/jMIR, yaafe) as well as rhythmic ones (openSMILE, libXtract). Even when such features are provided, they are incomplete and their implementation does not rely on the latest state of the art, which is the case for pitch detection and beat tracking. Furthermore no tools provide high-level description of music in terms of semantic categories such as genres or mood, out of the box, apart

[1] `http://sourceforge.net/projects/yaafe`
[2] `http://vamp-plugins.org/download.html`
[3] `http://developer.echonest.com`

from MIRtoolbox that includes emotion detection. 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[4] license and well-suited for both research and industrial applications.[5] 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 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[6] plugin and can be used with the popular Sonic Visualiser software [4] 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 over 6 years incorporating the work of more than 20 researchers and developers through its history. During these years it has been used for in-house research and development by Music Technology Group and a number of partners. Version 2.0 marks the first open-source release of the library, making it available to the wider public of researchers and developers. It 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 and predominant melody detection algorithms. In addition, Essentia can be optionally complemented with Gaia,[7] 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).

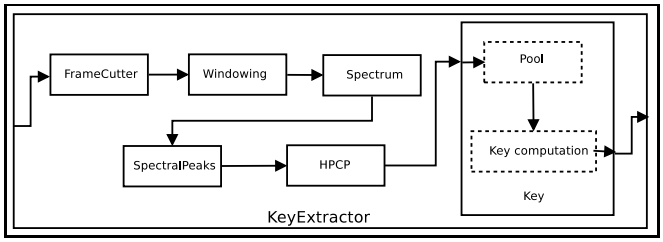

Figure 1: An example of the composite algorithm KeyExtractor combining the algorithms FrameCutter, Windowing, Spectrum, SpectralPeak, HPCP, and the Key algorithm composite itself.

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, 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).

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 be-

[4] `http://gnu.org/licenses/agpl.html`
[5] Commercial licensing is also offered.
[6] `http://vamp-plugins.org`
[7] `http://github.com/MTG/gaia`

ing 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 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 up-to-date reference for all available algorithms.[8] Furthermore, [3] provides more details on the available algorithms and includes references to the related publications. A great part of the algorithms computes a variety of low-level, mid-level, and high-level descriptors 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 reusing the FFmpeg[9]/Libav[10] libraries to provide a very convenient way to load audio files from disk with the support of almost all existing audio formats. A special loader reads the metadata tags (ID3 tags) stored in the given file. 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, downmix, trim the audio to a given start/end time, frame cutting, windowing, resampling, FFT, autocorrelation 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 MIR descriptors

The library contains a variety of algorithms which allow to compute:

- *low-level spectral descriptors* (the Bark/Mel/ERB band energies of a spectrum, the Mel-frequency and Gammatone feature cepstral coefficients cepstral coefficients, the spectral complexity, the spectral contrast feature, the inharmonicity and dissonance measures, etc);
- *time-domain descriptors* (the duration of the perceptually meaningful part of the signal above a certain energy level, Zero-crossing rate, different loudness estimation models, etc);
- *pitch and tonal descriptors* (the pitch salience function of a signal and the fundamental frequency of the predominant melody, the Harmonic Pitch-Class Profile of a spectrum also called chroma features, the tuning frequency, the key and the scale of a song, chords-related descriptors, etc);
- *rhythm descriptors* (the beats positions using the beat tracker based on the complex spectral difference feature and the multifeature beat tracker, the BPM histogram of a song, the novelty curve for the audio signal, and the BPM distribution and tempogram based on it, a number of onset detection functions, the rhythm transform, etc.);
- *SFX descriptors* characterizing signal envelope of short sounds;
- other *mid- and high-level descriptors* (the "danceability", the intensity, a number of classifier models including musical genre, ballroom music classification, moods, western/non-western music, live/studio recording, perceptual speed, male/female singer, dark/bright timbre, and speech/music classification, etc).

Table 1 summarizes the categories of MIR descriptors available in popular music analysis tools in comparison with Essentia. The latter provides the largest overall amount of features, according to a rough estimation, and includes a large number of high-level classifier-based features (genres, moods, etc.) in the box without a necessity of having the associated ground truth music collections. In contrast, MARSYAS and MIRToolbox (except for a single emotion-based classifier) lack pre-trained classifier models.

4.3 Extractors

A few useful extractors for commonly used MIR features have been written as algorithms. 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 [13], and in particular for mood classification, semantic autotagging [12], music similarity and recommendation [2, 1], visualization and interaction with music [1, 7], sound indexing [10], musical instruments detection [6], cover detection [11], and acoustic analysis of stimuli for neuroimaging studies [8]. 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, music similarity, autotagging, and beat detection, and they have usually ranked among the best ones.

Essentia and Gaia have been used extensively in a number of research projects, including the CANTATA EU (music videos recommendation system), SALERO EU (search engine for sound effects), PHAROS EU (music search and recommendation), Buscamedia, PROSEMUS, DRIMS and CompMusic (automated low-level and high-level semantic description of music), and SIEMPRE EU (audio analysis for multimodal databases of music performances). Furthermore, previous versions of Essentia have been exploited for industrial applications including the non-commercial sound service Freesound[11] (large-scale indexing and content-based

[8] `http://essentia.upf.edu`
[9] `http://ffmpeg.org`
[10] `http://libav.org`
[11] `http://freesound.org`

Table 1: MIR descriptor categories available in the popular music analysis tools.

Tool	Environment	Spectral	Time-domain	Pitch	Tonality	Rhythm	High-level	Auditory
libXtract	C/Python/Java	+	+	−	+	−	−	−
jAudio/jMIR	Java	+	+	−	−	+	−	−
MARSYAS	C++/Ruby/Python/Java	+	+	+	−	+	+	+
MIRToolbox	Matlab	+	+	+	+	+	+	+
Aubio	C/C++/Python	+	−	+	−	+	−	−
yaafe	C++/Python/Matlab	+	+	−	−	+	−	−
openSMILE	C++	+	+	+	+	−	−	−
Essentia	C++/Python	+	+	+	+	+	+	−

search of sound recordings), industrial products by BMAT[12] and Stromatolite[13] (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 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.[14] It contains the complete documentation for the project and installation instructions for Debian/Ubuntu, Mac OS X and Windows. The source code is available at the Github repository.[15]

7. ACKNOWLEDGMENTS

The work on Essentia has been partially funded by the PHAROS (EU-IP, IST-2006-045035), CANTATA (ITEA 05010, FIT-350300-2006-33), Buscamedia (CEN-20091026), 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] 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. *Inf. Process. & Management*, 49(1):13–33, 2013.

[2] 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.

[3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. ESSENTIA: an audio analysis library for music information retrieval. In *Int. Soc. for Music Inf. Retrieval Conf. (ISMIR'13)*, 2013.

[4] 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.

[5] F. Eyben, M. Wöllmer, and B. Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *ACM Int. Conf. on Multimedia (MM'10)*, page 1459–1462, 2010.

[6] 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.

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

[8] 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.

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

[10] 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 Process.*, 2010.

[11] 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 Process.*, 16(6):1138–1151, 2008.

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

[13] 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 Inf. Retrieval Evaluation Exchange (MIREX'10)*, 2010.

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

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

[14] `http://essentia.upf.edu`

[15] `http://github.com/MTG/essentia`