3. **81** - Real-time bag of words, approximately
   J. R. R. Uijlings, A. W. M. Smeulders, R. J. H. Scha
   http://dl.acm.org/citation.cfm?id=1646405

4. **57** - Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features
   Hideki Nakayama, Tatsuya Harada, Yasuo Kuniyoshi
   http://dl.acm.org/citation.cfm?id=1646419

5. **46** - Multilayer pLSA for multimodal image retrieval
   Rainer Lienhart, Stefan Romberg, Eva Hörster
   http://dl.acm.org/citation.cfm?id=1646408

CIVR 2010

1. **43** - Signature Quadratic Form Distance
   Christian Beecks, Merih Seran Uysal, Thomas Seidl
   http://dl.acm.org/citation.cfm?id=1816105

2. **41** - Feature detector and descriptor evaluation in human action recognition
   Ling Shao, Riccardo Mattivi
   http://dl.acm.org/citation.cfm?id=1816111

3. **38** - Unsupervised multi-feature tag relevance learning for social image retrieval
   Xirong Li, Cees G. M. Snoek, Marcel Worring
   http://dl.acm.org/citation.cfm?id=1816044

4. **29** - Co-reranking by mutual reinforcement for image search
   Ting Yao, Tao Mei, Chong-Wah Ngo
   http://dl.acm.org/citation.cfm?id=1816048

5. Two papers were tied for 5th place in citations:
   - **20** - On the sampling of web images for learning visual concept classifiers
     Shiai Zhu, Gang Wang, Chong-Wah Ngo, Yu-Gang Jiang
     http://dl.acm.org/citation.cfm?id=1816051
   - **20** - Plant species identification using leaf image retrieval
     Carlos Caballero, M. Carmen Aranda
     http://dl.acm.org/citation.cfm?id=1816089

MIR 2008

1. **285** - The MIR flickr retrieval evaluation
   Mark J. Huiskes, Michael S. Lew
   http://dl.acm.org/citation.cfm?id=1460104

2. **203** - Outdoors augmented reality on mobile phone using loxel-based visual feature organization
   Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bismpigiannis, Radek Grzeszczuk, Kari Pulli, Bernd Girod
   http://dl.acm.org/citation.cfm?id=1460165

3. **119** - Learning tag relevance by neighbor voting for social image retrieval
   Xirong Li, Cees G.M. Snoek, Marcel Worring
   http://dl.acm.org/citation.cfm?id=1460126

4. **58** - Spirittagger: a geo-aware tag suggestion tool mined from flickr
   Emily Moxley, Jim Kleban, B. S. Manjunath
   http://dl.acm.org/citation.cfm?id=1460102

5. **42** - Content-based mood classification for photos and music: a generic multi-modal classification framework and evaluation approach
   Peter Dunker, Stefanie Nowak, André Begau, Cornelia Lanz
   http://dl.acm.org/citation.cfm?id=1460114

MIR 2010

1. **82** - New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative
   Mark J. Huiskes, Bart Thomee, Michael S. Lew
   http://dl.acm.org/citation.cfm?id=1743475

2. **78** - How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation
   Stefanie Nowak, Stefan Rüger
   http://dl.acm.org/citation.cfm?id=1743478

3. **45** - Exploring automatic music annotation with “acoustically-objective” tags
   Derek Tingle, Youngmoo E. Kim, Douglas Turnbull
   http://dl.acm.org/citation.cfm?id=1743400

4. **39** - Feature selection for content-based, time-varying musical emotion regression
   Erik M. Schmidt, Douglas Turnbull, Youngmoo E. Kim
   http://dl.acm.org/citation.cfm?id=1743431

5. **34** - ACQUINE: aesthetic quality inference engine – real-time automatic rating of photo aesthetics
   Ritendra Datta, James Z. Wang
   http://dl.acm.org/citation.cfm?id=1743457

ESSENTIA: an open source library for audio analysis

Over the last decade, audio analysis has become a field of active research in academic and engineering worlds. It refers to the extraction of information and meaning from audio signals for analysis, classification, storage, retrieval, and synthesis, among other tasks. Related research topics challange understanding and

modeling of sound and music, and develop methods and technologies that can be used to process audio in order to extract acoustically and musically relevant data and make use of this information. Audio analysis techniques are instrumental in the development of new audio-related products and services, because these techniques allow novel ways of interaction with sound and music.

Essentia is an open-source C++ library for audio analysis and audio-based music information retrieval released under the **Affero GPLv3 license** (also available under proprietary license upon request). 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 that can be computed from audio. In addition, Essentia can be complemented with **Gaia**, a C++ library with python bindings which allows searching in a descriptor space using different similarity measures and classifying the results of audio analysis (same license terms apply). Gaia can be used to generate classification models that Essentia can use to compute high-level description of music.

Essentia is not a framework, but rather a collection of algorithms wrapped in a library. It doesn't enforce common high-level logic for descriptor computation (so you aren't locked into a certain way of doing things). It rather focuses on the robustness, performance and optimality of the provided algorithms, as well as ease of use. The flow of the analysis is decided and implemented by the user, while Essentia is taking care of the implementation details of the algorithms being used. A number of examples are provided with the library, however they should not be considered as the only correct way of doing things.

The library includes **Python bindings** as well as 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. The extractors cover 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. Furthermore, it includes a **Vamp plugin** to be used for visualization of music descriptors using hosts such as Sonic Visualiser.

The library is **cross-platform** and 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 out-of-the-box and signal processing algorithms, is **easily expandable** and allows for both research experiments and development of large-scale industrial applications.

Essentia has been in development for more than 7 years incorporating the work of more than 20 researchers and developers through its history. The 2.0 version marked the first release to be publicly available as free software released under AGPLv3.

Algorithms

Essentia currently features the following algorithms (among others):

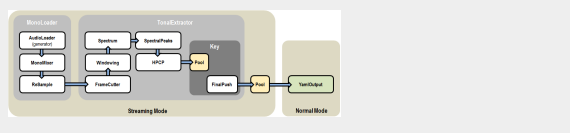
- **Audio file input/output:** ability to read and write nearly all audio file formats (wav, mp3, ogg, flac, etc.)
- **Standard signal processing blocks:** FFT, DCT, frame cutter, windowing, envelope, smoothing
- **Filters (FIR & IIR):** low/high/band pass, band reject, DC removal, equal loudness
- **Statistical descriptors:** median, mean, variance, power means, raw and central moments, spread, kurtosis, skewness, flatness
- **Time-domain descriptors:** duration, loudness, LARM, Leq, Vickers' loudness, zero-crossing-rate, log attack time and other signal envelope descriptors
- **Spectral descriptors:** Bark/Mel/ERB bands, MFCC, GFCC, LPC, spectral peaks, complexity, rolloff, contrast, HFC, inharmonicity and dissonance
- **Tonal descriptors:** Pitch salience function, predominant melody and pitch, HPCP (chroma) related features, chords, key and scale, tuning frequency
- **Rhythm descriptors:** beat detection, BPM, onset detection, rhythm transform, beat loudness
- **Other high-level descriptors:** danceability, dynamic complexity, audio segmentation, semantic annotations based on SVM classifiers

The complete list of algorithms is available online in the official documentation.

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 the figure below. 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).

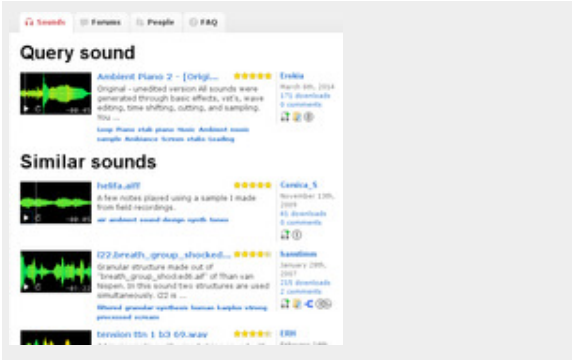


The algorithms can be used in two different modes: standard and streaming. The **standard mode** is imperative while the streaming mode is declarative. The standard mode requires to specifying the inputs and outputs for each algorithm and calling their processing function explicitly. If the user wants 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.

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, semantic autotagging, music similarity and recommendation, visualization and interaction with music, sound indexing, musical instruments detection, cover detection, beat detection, and acoustic analysis of stimuli for neuroimaging studies. Essentia and Gaia have been used extensively in a number of research projects and industrial applications. As an example, both libraries are employed for large-scale indexing and content-based search of sound recordings within **Freesound**, a popular repository of Creative Commons licensed audio samples. In particular, Freesound uses audio based similarity to recommend sounds similar to user queries. Dunya is a web-based software application using Essentia that lets users interact with an audio music collection through the use of musical concepts that are derived from a specific musical culture, in this case Carnatic music.



Examples

Essentia can be easily used via its python bindings. Below is a quick illustration of Essentia's possibilities for example on detecting beat positions of music track and its predominant melody in a few lines of python code using the standard mode:

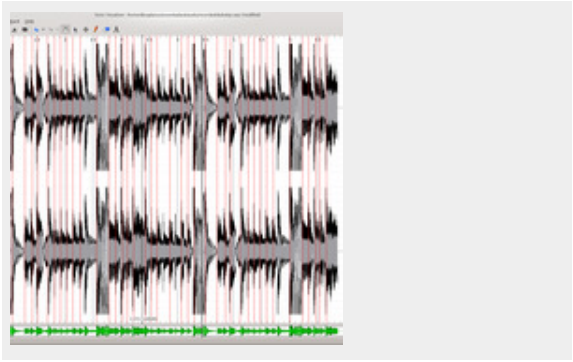
```
from essentia.standard import *;
audio = MonoLoader(filename =
'audio.mp3')(); beats, bconfidence =
```

```
BeatTrackerMultiFeature()(audio); print beats; audio = EqualLoudness()(audio); melody, mconfidence = PredominantMelody(guessUnvoiced=True, frameSize=2048, hopSize=128)(audio); print melody
```

Another python example for computation of MFCC features using the streaming mode:

```
from essentia.streaming import * loader = MonoLoader(filename = 'audio.mp3') frameCutter = FrameCutter(frameSize = 1024, hopSize = 512) w = Windowing(type = 'hann') spectrum = Spectrum() mfcc = MFCC() pool = essentia.Pool() # connect all algorithms into a network loader.audio >> frameCutter.signal frameCutter.frame >> w.frame >> spectrum.frame spectrum.spectrum >> mfcc.spectrum mfcc.mfcc >> (pool, 'mfcc') mfcc.bands >> (pool, 'mfcc_bands') # compute network essentia.run(loader) print pool['mfcc'] print pool['mfcc_bands']
```

Vamp plugin provided with Essentia allows to use many of its algorithms via the graphical interface of Sonic Visualiser. In this example, positions of onsets are computed for a music piece (marked in red):



An interested reader is referred to the documention online for more example applications built on top of Essentia.

Getting Essentia

The detailed information about Essentia is available online on the official web page: http://essentia.upf.edu. It contains the complete documentation for the project, compilation instructions for Debian/Ubuntu, Mac OS X and Windows, as well as precompiled packages. The source code is available at the official Github repository: http://github.com/MTG/essentia. In our current work we are focused on expanding the library and the community of users, and all active Essentia users are encouraged to contribute to the library.

References

[1] Serra, X., Magas, M., Benetos, E., Chudy, M., Dixon, S., Flexer, A., Gómez, E., Gouyon, F., Herrera, P., Jordà, S., Paytuvi, O, Peeters, G., Schlüter, J., Vinet, H., and Widmer, G., Roadmap for Music Information ReSearch, G. Peeters, Ed., 2013. [Online].

[2] Bogdanov, D., Wack N., Gómez E., Gulati S., Herrera P., Mayor O., Roma, G., Salamon, J., Zapata, J., Serra, X. (2013). ESSENTIA: an Audio Analysis Library for Music Information Retrieval. International Society for Music Information Retrieval Conference(ISMIR'13). 493-498.

[3] Bogdanov, D., Wack N., Gómez E., Gulati S., Herrera P., Mayor O., Roma, G., Salamon, J., Zapata, J., Serra, X. (2013). ESSENTIA: an Open-Source Library for Sound and Music Analysis. ACM International Conference on Multimedia (MM'13).

SIGMM Award for Outstanding PhD Thesis in Multimedia Computing, Communications and Applications

Award Description

This award will be presented at most once per year to a researcher whose PhD thesis has the potential of very high impact in multimedia computing, communication and applications, or gives direct evidence of such impact. A selection committee will evaluate contributions towards advances in multimedia including multimedia processing, multimedia systems, multimedia network services, multimedia applications and interfaces. The award will recognize members of the SIGMM community and their research contributions in their PhD theses as well as the potential of impact of their PhD theses in multimedia area. The selection committee will focus on candidates' contributions as judged by innovative ideas and potential impact resulting from their PhD work.

The award includes a US$500 honorarium, an award certificate of recognition, and an invitation for the recipient to receive the award at a current year's SIGMM-sponsored conference, the ACM International Conference on Multimedia (ACM Multimedia). A public