Sound recycling from public databases: Another BigData Approach to Sound Collections

Conference Paper · August 2017

DOI: 10.1145/3123514.3123550

2 authors:

Hernán Ordiales
11 PUBLICATIONS **5** CITATIONS

Matías Lennie Bruno
3 PUBLICATIONS **3** CITATIONS

Sound recycling from public databases

Another BigData approach to sound collections

Hernán Ordiales
RedPanal.org
MediaLab CASo
Riobamba 985
Buenos Aires C1116ABC, Argentina
h@ordia.com.ar

Matías Lennie Bruno
RedPanal.org
MediaLab CASo
Riobamba 985
Buenos Aires C1116ABC, Argentina
matias.lennie@casadelbicentenario.gob.ar

ABSTRACT

Discovering new sounds from large databases or Internet is a tedious task. Standard search tools and manual exploration fails to manage the actual amount of information available. This paper presents a new approach to the problem which takes advantage of grown technologies like Big Data and Machine Learning, keeping in mind compositional concepts and focusing on artistic performances. Among several different distributed systems useful for music experimentation, a new workflow is proposed based on analysis techniques from Music Information Retrieval (MIR) combined with massive online databases, dynamic user interfaces, physical controllers and real-time synthesis. Based on Free Software tools and standard communication protocols to classify, cluster and segment sound. The control architecture allows multiple clients request the API services concurrently enabling collaborative work. The resulting system can retrieve well defined or pseudo-aleatory audio samples from the web, mix and transform them in real-time during a live-coding performance, play like another instrument in a band, as a solo artist combined with visual feedback or working alone as automated multimedia installation.

CCS CONCEPTS

• **Applied computing** → **Performing arts**; **Sound and music computing**; • **Information systems** → **Multimedia databases**; *Digital libraries and archives*; *Collaborative search*; RESTful web services; Clustering; • **Computing methodologies** → Classification and regression trees; Markov decision processes;

KEYWORDS

Music Information Retrieval, BigData, Audio discovery, Performance, Experimental, Live coding, Collaborative, NetMusic, Realtime, User Interface, Machine Learning

ACM Reference format:
Hernán Ordiales and Matías Lennie Bruno. 2017. Sound recycling from public databases. In *Proceedings of AM '17, London, United Kingdom, August 23–26, 2017,* 8 pages.
DOI: 10.1145/3123514.3123550

1 INTRODUCTION

Nowadays many musicians use the same tools, both for composition as live performances. From acoustic or electronic instruments to the same computer software for synthesis or real-time processing. Even the same workflows or effects chains. Leading to similar results, as Curtis Roads mentions in the opening of one of his books *"Each tool opens up aesthetic possibilities but also imposes aesthetic constraints"* [25]. On the other hand, in the technical domain (not music related) people are realizing the potential of Big Data and Analytics, leaving as an indirect consequence the easy access to the IT resources that enable those.

In the music domain, that can reflect in new instruments based on the web or local networks, using new sounds discovered from large online databases, both previously classified by Music Information Retrieval (MIR) techniques or by clustering algorithms. Also, the synthesis process could be rethought using already available live-coding tools, some of them, usually reserved only for a small group of highly qualified users. The Internet and the incredible amount of different information also allow the rising of new collaboration processes and composition techniques. Even the live-coding environment can improve by adding dynamic graphic user interfaces to control the whole or part of the process (based on the usage or public participation).

AM '17, London, United Kingdom

978-1-4503-5373-1/17/08. . . $15.00
DOI: 10.1145/3123514.3123550

2 RELATED WORK

Although in the past decade most of the topics covered by this work had many publications, most related research are either on MIR, Big Data or music performances. Little existing work combines these three areas. One from Music Technology Group [26] (MTG) developed an approach, identifying loops in unstructured audio and presents an instrument prototype, but not focusing on it neither the performer experience, just working on Freesound.org web sounds and its public (but close in the open-source sense) API [1]. More focused in Music Information Retrieval or Audio Content Analysis [15], there do exist impressive works oriented to classify or organize large audio databases [2] [23] or with a pedagogical intentions [30] but they haven't applied on professional performing. There's also some work regarding network architectures and collaborative development [17][8] proposing frameworks for algorithmic composition or performing, using digital musical instruments and about exploring musical collections [9] through graphical views, allowing navigation of the visual representations and hear those samples or audio extracts. Regarding sonification using online datasets, there is the ATLAS project which translate CERN data into sound, more precisely mapping as notes and rhythms [13].

This work also explores and proposes adapting user interfaces to control sound in real-time.

3 WHY SOUND RECYCLING?

The actual online amount of recorded audio is impossible to hear and manage in measurable time. Unquestionably, resources availability is evolving faster than musical tools and workflows. The lack of a previous classification and easy access to those sounds through uncomplicated and intuitive interfaces leaves open a world of possibilities and enables collaborative developments. For example, nowadays there are about a billion of Creative Commons [10] (CC) licensed works (of all kinds) available online (see Figure 1). Audio related content beyond music pieces such as samples, live and speech recordings are hosted on online platforms like the Internet Archive, SoundCloud, Freesound.org, Free Music Archive (FMA), Redpanal.org, ccMixter and others. Audio tracks reach 4 million, videos and other multimedia are close to 43 million [21]. Initiatives like Europeana aggregate content from national libraries and institutions. This content had been shared by CC or public domain [11].

In this context, the development of new ways of live collaboration with musical goals and the exploitation of available resources is mandatory to take advantage and achieve new results.

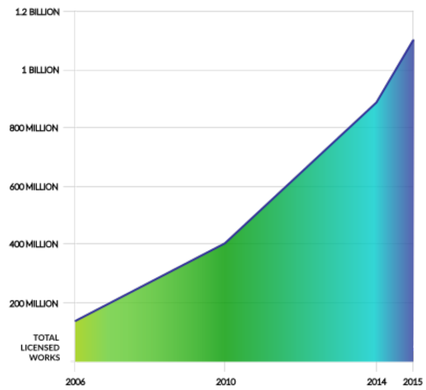

Figure 1: Growth of Creative Commons licensed works

4 DEFINING A WORKFLOW FOR LIVE PERFORMANCES

A series of guidelines were defined to model the whole process.

- Use the availability of technologies from Big Data and Analytics, especially those related with Music Information Retrieval
- Modular design
- Enable collaboration over a private or public network
- Free software components
- Cross-platform (at least Windows, Mac, Linux and ARM OS)
- Live-coding oriented, live performances and experimental music composition
- Include physical controller
- Dynamic user interface
- Use of well-known standards and open protocols (MIDI, OSC)

5 ARCHITECTURE DESIGN

A distributed system, including an online database, a public REST API, controllers and synthesis engine tools is proposed as a solution (see Figure 2). This workflow can be useful for offline work but is also effective for live performances or multimedia installations. A primary goal of a distributed system is to appear to the users of the system as a single computer beyond if it is composed off a collection of independent computers network or not. The end user must see a single coherent entity [28].

In the proposed system, sounds are retrieved from a local database or network or the Internet (Cloud) using and standard REST API. When and how to retrieve them can be triggered from various controllers (digital

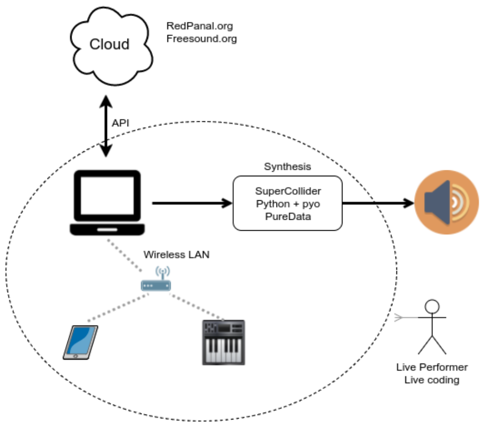

Figure 2: Live performance workflow

or physical) over the local network. And control how to perform the synthesis process too, modifying it in real-time, maybe concurrently (many users) because this architecture allows many clients. Under this architecture, there is the option to be deterministic or stochastic. Stochastic procedures differentiate from a deterministic one since it integrates random choice into the decision making process [25]. Under this scheme, every time a sample is retrieved, there are many candidates (or neighbors in a cluster) and if one of them is chosen using a random function (in a direct or indirect way) the system is defined as a stochastic one. That enables the option of the kind of the decision making to the composer.

6 IMPLEMENTATION OF THE CLOUD INSTRUMENT

All the software, web services, samples and documentation built were named APICultor (standing as the Application Programing Interface of the Culture). All the tools are publicly available online [1]. This work makes extensive use of techniques from Music Information Retrieval or Audio Content Analysis (ACA) [15] to extract audio features or descriptors, in order to apply them with musical purposes. It started as a dedicated software, evolved and became into a framework for live music performances. Built in Python (a well known general purpose language) includes sound pre-processing, MIR analysis using Essentia [4] and LibRosa [19] libraries. In RedPanal.org case, the sounds were previously processed, classified, clustered and segmented using the Essentia [4] and aubio[5] libraries to perform MIR analysis. An offline demo database was also built, including a mock of an HTTP API for testing purposes. Engine tools like SuperCollider [18] or Python plus Pyo [3] module are proposed as a solution, although any software which performs real-time process and can receive MIDI or OSC messages could be used.

[1] Another Big Data approach to sound collections. Available at https://github.com/sonidosmutantes/apicultor.

MIR Descriptors and sound textures

Defining a sound texture is not an easy task, but many agree to be treated as a low-level phenomenon [27]. Simple sound elements are called atoms, and can be classified as a low-level property, and their distribution and arrangement as a high-level one. There is some consensus among people about that a sound texture should exhibit similar characteristics over the time.

Audio features or descriptors extracted from raw input after pertinent analysis can be thought in three levels or data representations. Low-level descriptors like Cepstrum, spectral flux (amount of change of spectral shape), HFC and LFC (high and low frequency content). Mid-level representations such as pitch, onsets or beats. And high-level like music style, artist, mood, etc. Another classification consists in separate into spectral, temporal, tonal or rhythm descriptors, and there is no limitation into building new ones to describe new features [12].

For example, using the Spectral Centroid calculation, which shows the balancing point of the spectrum (in other words its center of gravity). It determines the frequency area around which most of the signal energy concentrates and in the literature is often correlated with the timbre dimension brightness or sharpness [15]. And is defined as the frequency-weighted sum of the power spectrum normalized by its unweighted sum:

$$\text{Centroid}(n) = \frac{\sum_{k=0}^{K/2-1} k|X(k,n)|^2}{\sum_{k=0}^{K/2-1} |X(k,n)|^2}$$

With $X(k,n)$ representing the Short Time Fourier Transform (STFT) of the frame n and k moving along each frequency bin (block length equal to K). Resulting value, can be converted to a parameter in the 0..1 range dividing by $(\frac{K}{2}-1)$ or to frequency value using $Centroid(n) * F_s/N$ with F_s sampling rate and N the number of FFT points used. Tweaking this value could be one of the many approaches to obtain different sounds with diverse textures.

A more elaborated use, consist in define or configure many MIR descriptor values in order to find a sound that matches all of them. Even "x, y" diagrams which compares two descriptors could be useful and interesting to find new sounds. Like in the Figure 3 where guitar sounds are grouped according those descriptors and numbers representing the ID of each sound. Later in section 6 this idea is developed and an user interface with "x, y widgets" or controls is showed (Figure 8).

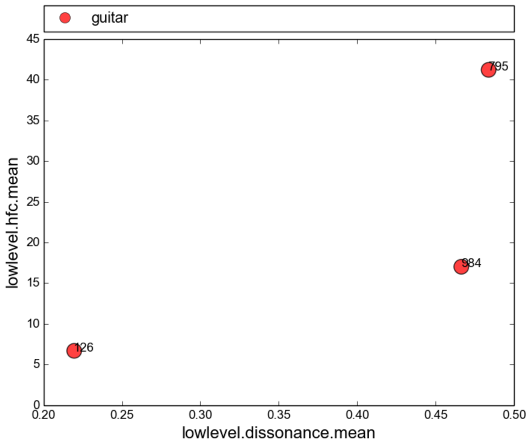

Figure 3: Mean dissonance value vs. high hrequency content mean

Early experiments

State machine (Markov-process). One of the first experiments include an automated music machine. A Markov model [22] was used, and for each state a sound with different MIR descriptors was defined. Each "MIR state" in the meaning of a series of values from each descriptor (for example mean values for Inharmonicity or low frequency content) plus the probability of transition associated with each state, see Figure 4. This schema could be thought as a stochastic composition using web sounds as the source. Useful for multimedia installations.

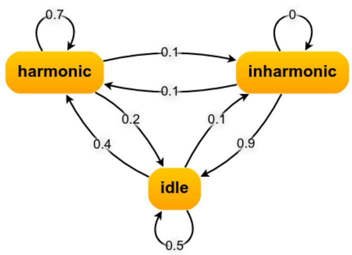

Figure 4: Simple MIR sound state machine

As Figure 5 shows is very simple to "compose" or define transitions in plain text. With another JSON file defining MIR states, giving values for each proposed descriptor, all the needed as "score" for the system is fulfilled.

```
{ "statesArray": [
{ "id": 0, "text": "harmonic" },
{ "id": 1, "text": "inharmonic" },
{ "id": 2, "text": "idle" }
],
"linkArray": [
{ "from": 0, "to": 0, "text": "0.7" },
{ "from": 0, "to": 1, "text": "0.1" },
{ "from": 1, "to": 0, "text": "0.1" },
{ "from": 1, "to": 1, "text": "0" },
{ "from": 1, "to": 2, "text": "0.9" },
{ "from": 2, "to": 1, "text": "0.1" },
{ "from": 0, "to": 2, "text": "0.2" },
{ "from": 2, "to": 0, "text": "0.4" },
{ "from": 2, "to": 2, "text": "0.5" }
],
"statesArray": [
{ "id": 0, "mir": [{
"content": "hang",
"sfx.duration": "* TO 3",
"sfx.inharmonicity.mean": "0.1"}]},
{ "id": 1, "mir": [{
"sfx.duration": "7.2",
"lowlevel.hfc.mean": "* TO 0.0005",
"lowlevel.spectral_complexity.mean": "1"}]
    }]
}
```

Figure 5: JSON definition of the state machine

Clustering. Identifying clusters using a k-means algorithm [16] which allows predefining the number of groups wanted, so them can be associated to each of the states from Section 6. A concept, idea or sound texture can be settled for each cluster and guarantee similar sound characteristics, choosing the correct MIR descriptors and values for each one. This allows automated performances evolving in time, always different but with the same underlying structure.

This algorithm, minimizes the neighbors distance using:

$$arg\,min \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2$$

Where $x1, x2, \ldots, xn$ are each observation or descriptor value, and μ its mean. The value of i represents each of the k clusters required.

Visualization is also useful, for example, clustering sounds from a test dataset using Inharmonicity as a value, each radio circle represent the amount of neighbors for each cluster (the number is the ID of the sound), see Figure 6.

On the other hand, its possible to find outliers and discard unwanted samples by using a hierarchical clustering algorithm.

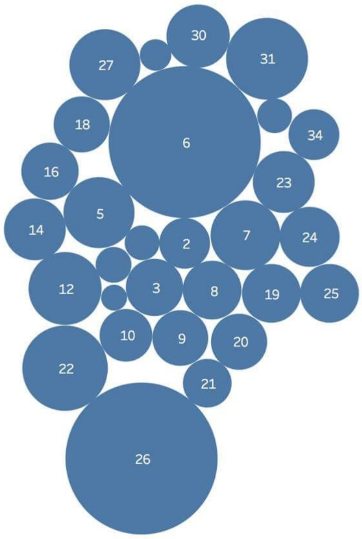

Figure 6: Inharmonicity clusters

Prototype

Our first prototype instrument was based on the workflow described in Figure 2 and keeping in mind that the audience will be developers and experimental musicians with some algorithmic knowledge. To test the system, Free Libre Sounds were used as a source, licensed under Creative Commons licenses [10] from RedPanal.org. Some short public live performances were produced to demonstrate and test all the technologies involved.

All the development was influenced by agile methodologies and software engineering techniques in general. Including, iterative and incremental approach [7], automatic testing, version control and branching strategies. That derived in creating a developing community with collaborators on different places all over the world. The system was contrasted periodically with real users during live-coding and live performance sessions. Keeping in mind all the guidelines (defined in Section 4) during the process and testing it consistently in many OS like Linux (Ubuntu from 15.04 to 16.10), Mac OS (10.11 El Capitan and Sierra), and Windows 10 operating systems. The synthesis module was also tested in a Raspberry Pi architecture (using Raspbian OS).

API calls are based on different MIR descriptors like tempo, rhythm, mood, frequency content and others. Working with frame values (time series of feature vectors) or statistical ones like mean or variance when needed.

Effects chain

Freeze effect plays a central part in the proposed workflow because allows managing previously unknown sounds, achieving infinite sustain and privileging (or not) harmonic content. Following the chain of Figure 7 unknown segmented samples retrieved from a database can fit and be played during a live performance. Basic parameters are the width of the freeze and start position. Other real-time effects like pitch shifter, modulation, and panning (and others) help to elaborate the sound modeling.

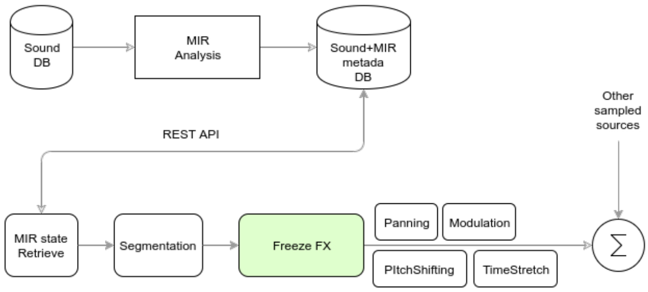

Figure 7: Freeze FX chain

Supercollider [18] synth based on a standard library freeze effect is enough to obtain decent results.

With Pyo (the Python DSP toolbox [3]) a tool oriented to music composers who wants to build tools, is easy to implement much of the real-time spectral processing like the freeze effect using Phase Vocoder [6] or granular synthesis.

An attractive option to freeze audio signals in time domain consists in using the audio extrapolation method [14]. Using an autoregressive processes (AR) where each time domain signal is modeled by

$$x(n) = \sum a_k x(n-k)$$

And $a_1, a_2, ..., a_p$ are the AR coefficients (that need to be identified by another algorithm).

UI Design

Basic approach is to use currently available commercial controllers (capable of send MIDI or OSC messages) to control MIR parameters, sample change, effects configuration, etc. Pads, knobs, sliders, xy controls, touch surfaces (tablets or mobile phones) and keyboards are useful, but not enough to explore the new dynamics and possibilities enabled by the proposed workflow.

Dynamic UI design using microservice provided by OpenStageControl [2] (sends OSC and MIDI) enables

[2] A libre desktop OSC bi-directional control surface application. Available at http://osc.ammd.net.

Figure 8: MIR State UI Prototype

UI prototyping and live edition from a browser. Indeed allows concurrent access and widget synchronization, allowing collaborative uses. MIR descriptors values are controlled with knobs and sliders, before sending them to the synthesis process (see Figure 8). Its also possible to retrieve neighbors of the cluster defined by x, y controls which, for example, allows to define clusters quickly using SpectralCentroid and Dissonance or HFC (High Frequency Content) and Tempo or every desired combination.

7 SEARCHING FOR NEW COMPOSING TECHNIQUES

Some lines of work have been explored and incorporated to aforementioned APICultor framework which includes linking or matching sound processes with climate ones (using online datasets and freeze effect). The use of complementary no-input techniques [29] and geographic and demographic info of a determinate place to link with folk music.

The architecture (Section 5) is open to deterministic or stochastic compositions, only changing the nature of the algorithms involved in the retrieving method and sample selection. Either choosing which one of the group or when to change the sample sound.

Manipulation of the samples through freeze effect (see Figure 7) allows establishing textural concept of music instead of the typical approach based on defined melodies or motives.

On the other hand, most efforts were centered in constructing efficient models with intrinsic value, reflecting creative ideas and giving meaning to the process. Having a proper care in the calculation of the aforementioned MIR descriptors, testing and contrasting them experimentally.

Slightly explored fields (not used yet in a live performance) includes Big Data sonification and allowing participation from the public through mobile devices and local area networks.

8 RESULTS

The new software APICultor was achieved successfully enabling live-performances since its beginnings. Sounds from the Cloud (RedPanal.org and Freesound.org) or local databases were retrieved without problems using the same REST API. These calls were based on MIR descriptors previously selected such tempo, rhythm, mood, frequency content (and others) allowing the composer surface its criteria to sample select. Using live-coding tools and convenient real-time effects like pitch transposing (shifting) and freeze effect (to achieve an harmonically sustained sound) to process those samples during live performances. MIDI controllers and OSC devices like mobile phones and tablets were used to trigger those sounds in real-time. A couple of interdisciplinary performances combined with other musicians and conventional instruments took place in Buenos Aires during 2016. Including "La Casa del Bicentenario" and "Noche de los museos".Sound recycling emerged as a side work and useful tool for electronic musicians. And in context with the contemporary world, the Internet and the amount of sound data public available on the web.

Also, a soundwork named "Dialectic in suspense" was composed and performed using the proposed workflow and tools. The central theme treats conflicting relation between nature and human contradictory development. Natural spaces and ambient sounds mixed with human residual pollution, are combined with real-time audio and data processing that shows both human and nature strategies to overcome the critical anthropocentric presence.

No specific operating system is required, all the tools involved were tested and worked well on all the platforms (Windows, Mac, Linux x86 and ARM). All the guidelines in section 4 were accomplished (in more or less detail).

The final instrument architecture, including the User Interface to Live Coding, can be seen in Figure 9 and samples can be heard at RedPanal website [3].

9 CONCLUSIONS AND FUTURE WORK

New workflows were tested with success, and during the process, new related research paths were discovered like dynamic UI and online public participation through setting MIR features. APICultor fulfilled its goals and

[3] Sound recycling of the sounds from the RedPanal free culture community. Available at http://redpanal.org/p/reciclado-de-samples.

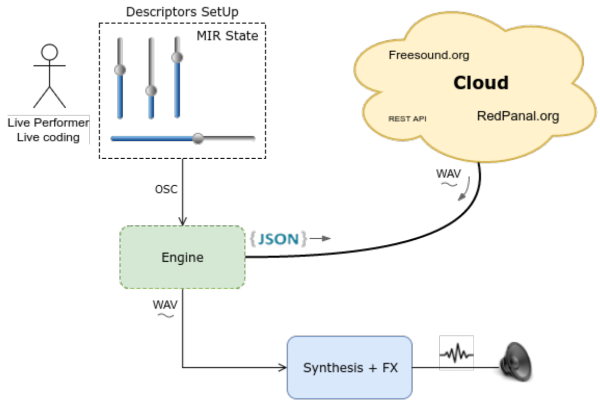

Figure 9: Cloud Instrument

became a framework, although because of its experimental nature, is still in development. The architecture based on Big Data, networks, controllers and a synthesis process proved to be useful, original and interesting to many musicians. On the other hand, , the experimental prototype performed live with commitment but without notice, because without a previous explanation nobody of the audience can perceive the nature of the instrument since by definition a distributed system should appear as a single one. Was observed that this system could be easily integrated with a real-time visualization engine, taking MIR descriptors or final sound as the input source. Even an exciting option is to take information from a previously modeled 3D environments (i.e. games or buildings) and add to the process in real-time, having feedback and sound modeling at the same time [24].

In the remaining work, there is the option of enhancing the sample recommendation process during the live performance (even the choice of UI advice based on past use) and provide the system as a Docker [20] image, to have dependencies into lightweight container package and avoid any dependencies trouble.

ACKNOWLEDGMENTS

Thanks to the support of the MediaLab/CASo (Centro de Arte Sonoro) from "La Casa del Bicentenario" Ciudad Autonoma de Buenos Aires, Argentina. RedPanal and APICultor community.

REFERENCES

[1] Vincent Akkermans, Frederic Font, Jordi Funollet, Bram De Jong, Gerard Roma, Stelios Togias, and Xavier Serra. [n. d.]. FREESOUND 2: AN IMPROVED PLATFORM FOR SHARING AUDIO CLIPS. ([n. d.]). http://mtg.upf.edu/system/files/publications/freesound

[2] Paolo Annesi, Roberto Basili, Raffaele Gitto, Alessandro Moschitti, and Riccardo Petitti. 2007. Audio Feature Engineering for Automatic Music Genre Classification. In *RIAO2007*. http://art.uniroma2.it/publications/docs/2007

[3] Olivier Belanger and Olivier. 2016. Pyo, the Python DSP toolbox. In *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*. ACM Press, New York, New York, USA, 1214–1217. https://doi.org/10.1145/2964284.2973804

[4] D Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, P Herrera, O Mayor, G Roma, J Salamon, J Zapata, and Xavier Serra. 2013. ESSENTIA: An audio analysis library for music information retrieval. *Proceedings of the International Conference on Music Information Retrieval (ISMIR)* (2013), 493–498.

[5] Paul M Brossier. 2006. The Aubio Library at MIREX 2006. In *Proc. International Society for Music Information Retrieval Conference*.

[6] Jean-Franç Ois Charles. 2008. A Tutorial on Spectral Sound Processing Using Max/MSP and Jitter. (2008). http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2008.32.3.87

[7] Alistair Cockburn. 2008. Using Both Incremental and Iterative Development. *CrossTalk: The Journal of Defense Software Engineering* 21, 5 (2008), 27–30. http://alistair.cockburn.us/Using+both+incremental+and+iterative+development

[8] Diego Dorado and Matías Zabaljáuregui. 2016. Opensemble: A framework for collaborative algorithmic music. (2016).

[9] Stephane Dupont, Thomas Dubuisson, Jerome Urbain, Raphael Sebbe, Nicolas D'Alessandro, and Christian Frisson. 2009. AudioCycle: Browsing musical loop libraries. In *7th International Workshop on Content-Based Multimedia Indexing, CBMI 2009*. 73–80. https://doi.org/10.1109/CBMI.2009.19

[10] Terry Flew. 2005. Creative Commons and the Creative Industries. *Media and Arts Law Review* 10, 4 (2005), 257–264.

[11] Frederic Font, T. Brookes, G. Fazekas, M. Guerber, A. La Burthe, D. Plans, M. Plumbey, M. Shaashua, W. Wang, and Xavier Serra. 2016. Audio Commons: bringing Creative Commons audio content to the creative industries. *61st AES Conference on Audio for Games* (2016). http://mtg.upf.edu/node/3423

[12] Geoffroy Peeters. 2014. *A large set of audio features for sound description (similarity and classification) in the CUIDADO project.* Technical Report. http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters

[13] Ewan Hill, Juliana Cherston, Steven Goldfarb, and Joseph A Paradiso. 2016. ATLAS data sonification: a new interface for musical expression and public interaction ATLAS data sonification: a new interface for musical expression. (2016). http://pos.sissa.it/

[14] I Kauppinen and K Roth. 2002. Audio signal extrapolation Theory and applications. *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx-02)* 1 (2002), 105–110. http://www.unibw-hamburg.de/EWEB/ANT/dafx2002/papers/DAFX02

[15] Alexander Lerch. 2012. *An introduction to audio content analysis: Applications in signal processing and music informatics.* https://doi.org/10.1002/9781118393550 arXiv:arXiv:1011.1669v3

[16] David J. C. MacKay. 2004. Information theory, inference, and learning algorithms. Cambridge University Press, Chapter 20, 628. http://www.inference.phy.cam.ac.uk/mackay/itila/book.html

[17] Joseph Malloch, Stephen Sinclair, and Marcelo M. Wanderley. 2008. A network-based framework for collaborative development and performance of digital musical instruments. In *Proceedings of the International Conference On Sound and Music Computing*. Vol. 4969. Springer Berlin Heidelberg, Berlin, Heidelberg, 401–425. https://doi.org/10.1007/978-3-540-85035-9_28

[18] James McCartney. 1996. SuperCollider: a new real time synthesis language. In *Proceedings of the 1996 International Computer Music Conference*. 257–258. http://www.audiosynth.com/icmc96paper.html

[19] Brian Mcfee, Colin Raffel, Dawen Liang, Daniel P W Ellis, Matt Mcvicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and Music Signal Analysis in Python. *PROC. OF THE 14th PYTHON IN SCIENCE CONF* (2015). http://dawenl.github.io/publications/McFee15-librosa.pdf

[20] Dirk Merkel. 2014. Docker: lightweight Linux containers for consistent development and deployment. (2014), 2 pages. https://doi.org/10.1097/01.NND.0000320699.47006.a3

[21] Ryan Merkley. 2015. State of the Commons 2015. *Creative Commons Newsletter* (2015). https://stateof.creativecommons.org/2015/

[22] F. Richard Moore. 1991. *Elements of Computer Music*. Vol. 1. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 110 pages. https://doi.org/10.2307/1513134

[23] Nikolaos Nikolaou. 2011. Music Emotion Classification. (2011).

[24] Natanael Olaiz, Pau Arumi, Toni Mateos, and David Garcia. 2009. 3D-audio with CLAM and Blender's Game Engine. *Linux Audio Conference* (2009). http://lac.linuxaudio.org/2009/cdm/Thursday/05

[25] Curtis Roads. 2015. *Composing Electronic Music: A New Aesthetic*. Oxford University Press.

[26] Gerard Roma and Xavier Serra. 2015. Music performance by discovering community loops. (2015). http://mtg.upf.edu/node/3177https://www.youtube.com/watch?v=uDC3M

[27] Nicolas Saint-Arnaud and Kris Popat. 1998. Analysis and Synthesis of Sound Textures. (1998). http://alumni.media.mit.edu/

[28] Andrew S. Tanenbaum and Maarten van. Steen. 2007. *Distributed systems : principles and paradigms*. Pearson Prentice Hall. 686 pages. https://books.google.com.ar/books/about/Distributed

[29] Dominic Thibault. 2015. fXfD, an Digital Approach to the No-Input Practice. *ICMC* (2015). https://quod.lib.umich.edu/cgi/p/pod/dod-idx/fxfd-a-digital-approach-to-the-no-input-practice.pdf?c=icmc; idno=bbp2372.2015.053

[30] Anna Xambó, Alexander Lerch, and Jason Freeman. 2016. Learning to code through MIR. *Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference* (2016), 2–4.