Live Repurposing of Sounds: MIR Explorations with Personal and Crowdsourced Databases

Anna Xambó
Centre for Digital Music
Queen Mary University of London, UK
a.xambo@qmul.ac.uk

Gerard Roma
CeReNeM
University of Huddersfield, UK
g.roma@hud.ac.uk

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology, USA
alexander.lerch@gatech.edu

Mathieu Barthet
Centre for Digital Music
Queen Mary University of London, UK
m.barthet@qmul.ac.uk

György Fazekas
Centre for Digital Music
Queen Mary University of London, UK
g.fazekas@qmul.ac.uk

ABSTRACT

The recent increase in the accessibility and size of personal and crowdsourced digital sound collections brought about a valuable resource for music creation. Finding and retrieving relevant sounds in performance leads to challenges that can be approached using music information retrieval (MIR). In this paper, we explore the use of MIR to retrieve and repurpose sounds in musical live coding. We present a live coding system built on SuperCollider enabling the use of audio content from online Creative Commons (CC) sound databases such as Freesound or personal sound databases. The novelty of our approach lies in exploiting high-level MIR methods (e.g., query by pitch or rhythmic cues) using live coding techniques applied to sounds. We demonstrate its potential through the reflection of an illustrative case study and the feedback from four expert users. The users tried the system with either a personal database or a crowdsourced database and reported its potential in facilitating tailorability of the tool to their own creative workflows.

Author Keywords

live coding, MIR, sound samples, Creative Commons

CCS Concepts

•**Information systems** → **Music retrieval;** •**Applied computing** → *Sound and music computing;* Performing arts;

1. INTRODUCTION

The continuous increase of digital storage capacity, cloud computing and the social web are some of the factors that have facilitated an expansion of digital media collections, for either private use, shared use with an online community, or both. The development of Creative Commons (CC) licenses helped with creating a legal environment promoting shareability and repurposing[1] of digital content. Sound and music are not an exception, and large audio online databases are available using CC licenses (e.g., ccMixter,[2] Europeana Sounds,[3] Freesound,[4] Jamendo,[5] Internet Audio Archive).[6] Such databases have become a valuable resource for practitioners and researchers because they facilitate access to large collections of varied sounds, which can be incorporated into their professional workflows. However, the "findability" of sounds for particular purposes using traditional methods (e.g., query by text, directory lists, side annotations) can become a time-consuming and tedious manual labor, and automatic annotation is still not mature

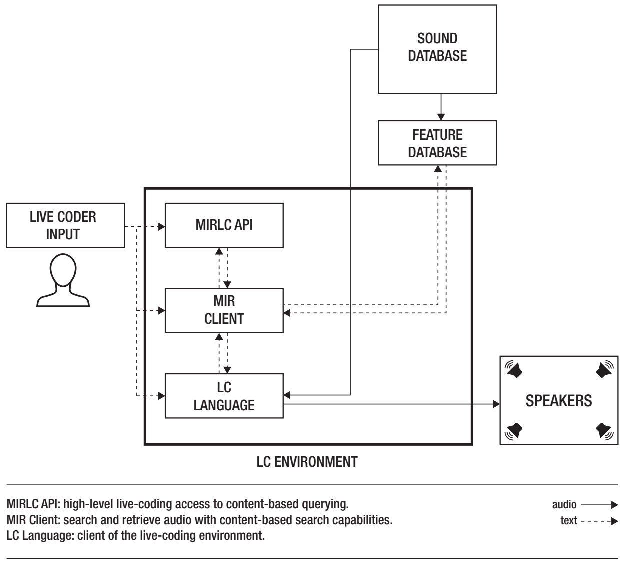

Figure 1: Block diagram of audio repurposing applied to live coding.

[1] *Audio repurposing* here refers to sampling and manipulating audio samples, generally from other artists, through retrieval (e.g., using queries by filter or similarity) from a database. *Live audio repurposing* refers to this practice in real time, suitable for performance.

[2] http://ccmixter.org

[3] http://www.eusounds.eu

[4] http://freesound.org

[5] http://www.jamendo.com

[6] http://archive.org/details/audio

NIME'18, June 3-6, 2018, Blacksburg, Virginia, USA.

enough [6, 7]. Also, the remix of CC content often brings legal issues if sounds with different licenses are combined.

There are still wide technological gaps preventing the reuse of such content given professional workflows from practitioners and researchers in areas such as music performance, composition, and production. Some recent initiatives, such as AudioCommons [10] and FluCoMa,[7] are driven towards creating ecosystems using open and personal audio content, respectively, and are designed for their use in a broad context of professional and artistic workflows. Some commercial apps already support access to both types of databases, for example Soundly,[8] which is a basic sound effects editor that integrates local and cloud-based sounds, but the way of searching sounds is limited to textual queries.

In this context, we explore the use of live coding combined with MIR techniques as a high-level musical approach to searching and repurposing sounds from both personal and open online sound databases (Figure 1) using either low-level descriptors (e.g., instantaneous and technical characteristics), mid-level descriptors (e.g., tonal and temporal characteristics) and high-level descriptors (e.g., semantic, musical, cultural, and psychological characteristics) [14, 26]. The musical practice of live coding is based on improvisation and generation of code in real time [4, 9, 12, 21], however, the integration of MIR techniques for sound retrieval has been explored little in this field. The motivation for combining both types of databases is that, on the one hand, musicians tend to build their own collections of sounds, but on the other hand, accessing online databases can often be useful if certain sounds are difficult to record or generate. Typically, a performer would have more control over their own collection but perhaps be less surprised and be limited in diversity. Conversely, using a crowdsourced database might provide diverse and rich content, but variable quality and unexpected results. Our approach can inform the design and development of tools for ecosystems such as AudioCommons (e.g., functionalities of an API that aims to be used across the different tools, information about the most used descriptors) and FluCoMa (e.g., exploring state-of-the-art descriptors for local databases) as well as bring a new way of discovering sounds and creating sound compositions in music performance. This paper describes a system that leverages existing technologies while adding two new modules: the MIRLCRep front-end (Section 4.2) to provide a high-level live-coding interaction with content-based queries, and the FluidSound back-end (Section 4.3) to automatically extract audio descriptors on users' machines. The system is based on web services that are accessed from within the SuperCollider [16] language to search and retrieve audio either from the Freesound [1] database or from local databases. We report on a case study and users' feedback on using the system, which indicate its potential as a discovery and performative tool that can be used beyond live coding with an engaging improvisational approach.

2. BACKGROUND IN AUDIO RETRIEVAL

The social web has brought a number of large audio databases (see Section 1). These databases differ in the type of content (e.g., sounds, music), and in how they categorize the sounds (from a top-down categorization such as Europeana to a bottom-up folksonomy such as Freesound). The retrieval of audio content can be based on user-annotated data (e.g., descriptions, tags, titles) and automatically extracted data (e.g., descriptors). A summary of the main techniques for audio retrieval on such databases can be found in [11], where two types of queries are described: (1) *queries by filter*, i.e., retrieval of sounds that match a given interval or threshold filtered by single or combined audio descriptors, typically a range of values; (2) *queries by similarity*, i.e., retrieval of similar sounds from a given example (target). It is worth mentioning that the latter has been widely explored in sound synthesis with techniques such as *musical mosaicing* [28], *concatenative sound synthesis* [24], or *mashup* [26] by retrieving short sounds similar to a reference. A common practice for efficient computation in real time is to pre-analyze the sound database and store a feature representation, which allows the combination of an offline analysis and the real-time use of the audio clips [26]. For the offline analysis, it is common to use a feature extractor program and store the information of each audio file as a single vector with the audio descriptors aggregated [11]. This approach is typically used for low-level descriptors, which in the context of our system makes sense because the structure of the sounds is unknown.

2.1 Audio Retrieval from Online Databases

Online audio databases offer a a great potential for creative audio repurposing, particularly when web APIs are supported for remote interaction. The Audio Commons Initiative is an EU-funded project aimed at promoting the use of open audio content and at developing technologies with which to support an ecosystem of content repositories, production tools and users. The project is looking for a common metadata specification across the online databases and a unified API specification that includes details of audio content, audio retrieval, and licenses.

2.2 Audio Retrieval from Personal Databases

Personal databases of audio files emerge from common practices such as improvisation or field recording. Music practitioners using computers can nowadays easily end up with large quantities of unlabeled or only partially labeled audio. While a significant amount of research has dealt with analysis of music and applications in music retrieval, current commercial tools for music creation still lack content-based mechanisms for retrieving audio from large personal collections. FluCoMa is an EU-funded project that aims to tackle the issue of usability of state of the art MIR and machine learning technologies for *fluid corpus manipulation*.

3. LIVE AUDIO REPURPOSING

In this section, we explain the approach to audio repurposing using live coding techniques and survey relevant examples from academia and industry. Music live coding is a music improvisation practice that is based on generating code in real time by either writing it directly or using interactive programming [4, 9, 12, 21]. With a few exceptions [18], most of the examples of audio repurposing in live coding use audio clips retrieved by textual queries, as shown in live coding sessions with programming languages for music such as Gibber [19] and Tidal [17], social media remixing tools such as Live Coding Youtube [13], and data-agnostic tools such as DataToMusic [27]. In gibberwocky [20], multiple audio clips are similarly used from a sound palette managed in Ableton Live connected to the live coding environment.

A number of systems have explored content-based audio retrieval in creative domains, such as Floop [23], Freesound Explorer [11], BBCut [8], LoopMashVST,[9] CataRT [25], and APICultor [18], which can inform our application to

[7] http://www.flucoma.org
[8] http://getsoundly.com
[9] http://www.youtube.com/watch?v=SuwVV9zBq5g

live coding. The common trends of these systems are: (1) content-based retrieval is combined often with textual queries which give flexibility and a higher level query closer to the human language (Floop, Freesound Explorer); (2) a combination of low-level and mid-level content-based queries used for real time interation (CataRT); (3) generally, the descriptors available are limited to a small number to avoid the complexity of navigating a high-dimensional space (Floop, Freesound Explorer, BBCut, LoopMashVST); (4) the systems start from filtered subspaces of the database that assure the sound results are within a range with the purpose of having a more controlled subspace (Floop, Freesound Explorer); (5) most of these systems are based on GUIs or hardware to control the MIR parameters (LoopMashVST, CataRT, APICultor).

Next, we present a novel approach to audio repurposing of sounds based on live coding and MIR, which gives the flexibility of choosing different descriptors without the constraints of a graphical user interface, and that works with online crowdsourced and personal sound databases.

4. THE SYSTEM

The general architecture is depicted in Figure 1. The system is based on web services that are accessed from within the SuperCollider language to search and retrieve audio either from remote or local databases. SuperCollider modules are provided as quarks,[10] which is the standard format for packaging and distributing software in this environment. The technologies underlying the Freesound API[11] are used for both the online and the local cases. The content-based search capabilities of this API are based on the Essentia [3] library for audio analysis and the Gaia[12] library for content-based indexing. In the online case, access to the index is provided by the Freesound API, and the Freesound quark.[13] In the local case, the Gaia index is wrapped by a web service running in the user's machine. A corresponding quark named FluidSound is provided.[14] Both components act as MIR client. The MIRLC layer provides high level access to content-based querying provided by both quarks, with a focus on live coding usage.

4.1 Concept

The system aims at providing a musical approach to operate with audio clips in live coding using MIR techniques during live performance. This facilitates the use of large databases. In the case of Freesound, the user can access almost 400,000 sounds either by text search, content search, or a combination of both. When using local databases, the scale is limited by the personal hardware and operating system, but the underlying technologies are the same.

As shown in Figure 1, our system accesses a database with the sounds, which can be either local or online, through a content-based descriptor index. The MIRLCRep module or MIRLC API is the user interface layer on top of the MIR client, the latter implemented by the Freesound and FluidSound quarks. This module deals with the low-level textual and content-based queries to the feature database, converting the returned values in JSON format to SuperCollider Dictionary objects. Once the requested sounds are identified in the sound database, the live coding environment allows loading them and managing the resulting buffers.

[10] `http://github.com/supercollider-quarks`
[11] `http://freesound.org/docs/api`
[12] `http://github.com/MTG/gaia`
[13] `http://github.com/g-roma/Freesound.sc`
[14] `http://github.com/flucoma/FluidSound`

4.2 MIRLCRep Front-End

MIRLCRep[15] is designed for repurposing audio samples from Freesound using and expanding the Freesound quark for SuperCollider by providing a more human-like queries and real-time performance capabilities. The module is inspired by Freesound Radio [22], where sounds could be retrieved and organized in parallel and sequential structures. In this module, sounds are loaded in user-defined groups and played in loop. Sounds of each group can be triggered either simultaneously or sequentially. The benefit of this approach is to easily load groups of related sounds and operate them with a higher level of control than operating single sounds. Compared to the previous examples (see Section 3), this prototype provides high-level functions (e.g., playback controls) for operating on a group of sounds, as well as both content-based and text-based queries. Each group is instantiated in a variable and methods can be applied (e.g., `id(1234)`, `random(2)`, `a.content(1, 'dur', 1, 'key', 'A')`, `similar`). In contrast to previous live audio repurposing projects, our approach provides tailorability by promoting the exploration of MIR parameters and the creation of subspaces of sounds through code, as opposed to a GUI. Textual feedback of the processes launched and the returned queries is provided, as well as methods for inspection (e.g., `whatpitch`, `whatbpm`).

While the MIR client deals with basic management of remote queries and parsing of the results, MIRLCRep adds higher-level operations for live music performance: asynchronous management of multiple sounds by a single query; simplified queries by content, similarity, tag, filter, and sound id; simplified random queries; a new architecture of groups of sounds with functions for playing them in sequence and in parallel, with asynchronous management; a new set of functions to control the playback of a group of sounds (e.g., play, stop, mute single sounds, solo single sounds); retrieval of sounds by content and similarity (making sure that there are no sounds repeated in the same group); retrieval of sounds by random queries avoiding nonexistent results (if a random picked sound does not exist in the database, it will keep searching randomly until finding an existing sound); and a customizable text file that prints the sounds used in a given session by title and name of the contributor of the sound to the database.

4.3 FluidSound Back-End

While the Freesound quark has been available for several years, a similar module is introduced here for local databases. Since all the code for Freesound as well as the Essentia and Gaia libraries is available under a free software license, a server and client analogous to Freesound component for content-based audio indexing and retrieval can be set-up locally, with some modifications. We call this the FluidSound back-end. The system starts a web server on a personal computer, which wraps a Gaia index. An analysis script using the Essentia library scans a personal audio database and adds the descriptors to the index. The server provides a simple API for content-based retrieval using range and similarity queries. Instead of Freesound URLs, the server provides local paths to audio files. Also unlike Freesound, the FluidSound provides only the metadata available in the audio files, so no text descriptions or tags are provided. The SuperCollider client interacts with the local web server and loads the files from the personal database.

[15] `http://github.com/axambo/MIRLC`

5. THE STUDY

5.1 Research question

Our aim is to identify the challenges and opportunities of live coding using MIR techniques by gaining the performer's insight into the system and how descriptors are used in live coding. In particular, we are interested in understanding the differences between using a personal database vs. an online crowdsourced database when live coding using MIR techniques from a live coder's perspective.

5.2 Methodology

In alignment with NIME evaluation guidelines [2], we conducted a study from a performer's perspective to qualitatively assess our system based on a practice-based approach [5]. The objectives were to: (1) describe the design decisions from the first author's experience as an autoethnography [15] during the development process (see Section 6); (2) get feedback from practitioners and describe interesting behaviors that emerged from the use of this system to improve the system's capabilities beyond own practice (see Section 7).

6. CASE STUDY: OWN INSIGHTS

This case study is about the experience of developing the tool as a practitioner, reported by the first author. It covers the progressive discoveries while developing, rehearsing and performing with the tool MIRLCRep over the past 16 months and is centered on the use of the online crowdsourced database Freesound. Throughout this process, rehearsals with the tool have informed its development as well as the choice of descriptors suitable for live coding. For this reason, oftentimes the line between rehearsal and development is fuzzy. Although the tool can be tailored to different music styles, the first author has been using it influenced by her musical taste, close to experimental electronic music. The final intention was to build a tool suitable for performance based on live repurposing of sounds. The use of crowdsourced sounds has informed the next stage of the system of using it also with personal databases (see Section 7).

During a first round of development, which ended with a test-bed performance (Figure 2), the tool was used to query looping sounds from Freesound using tags and similarity.[16] The theme of the concert was noise music, thus the rehearsals and performance used sound samples related to the theme. In the rehearsals, the most used pattern by the live coder was to explore the online sound database by using tags related to the concert's theme (e.g., `noise`, `hammer`, `cell-phone`, `digits`, `saw`) and then retrieve other related sounds by similarity. Getting sounds by content was less used at this stage because the parameters were too long to type (e.g., `.lowlevel.spectral_complexity.mean:` or `tonal.key_key:`). Overall, the drawback of this approach was that there appeared unwanted sounds (e.g., a chord of a guitar), thus the serendipity became a disadvantage due to the uncontrolled random results. At the same time, the advantage was to know in advance what subspaces of the database were interesting without knowing a large heterogeneous database such as Freesound. The use of the similarity queries was found to give musically consistent results but with a certain level of unpredictability, which agrees with the literature on the difficulty of defining music similarity [14]. As areas of improvement, the search and retrieval processes could have been made more visible to both the audience and the live coder, in alignment with

Figure 2: The first author live coding with MIRLCRep at Noiselets 2017, Freedonia, Barcelona, Spain (photograph by Helena Coll).

transparency expected in the practice of live coding.[17] For example, showing more permanently textual feedback of the sounds that are running and whether the groups are played in sequence or in parallel.

In a second round of development, the way of retrieving by content was simplified for live coding (e.g., shorter commands for sound queries and content-based filters). The commands related to query by content were the most frequently used by the performer.[18] Target sounds were retrieved by mid-level descriptors (e.g., bpm, pitch), and similar sounds were found by filtering results according to some metrics (e.g., 120 bpm, pitch with high confidence measure). When rehearsing with the tool, the performer found preferred combinations of parameters and values (e.g., applying a low confidence measure to be able to create contrast). Finding the suitable combinations described above was an important part of the rehearsals. With this approach, there was more control over subspaces of the crowdsourced database without *a priori* knowing the sounds, while still retrieving new sounds with each rehearsal.

Here, live coding became a combination of high-level textual queries (e.g., tags) with content-based queries (e.g., pitch, bpm, key, or scale). MIR techniques worked well for music improvisation in front of the challenge of dealing with an unknown online crowdsourced database. The combination of metadata with audio content analysis provided flexibility and variation to the performance and was helpful to produce a coherent sound palette. It was possible to control short vs. long sounds, pitched vs. unpitched sounds, and rhythmic vs. tonal sounds, yet there was a certain level of unpredictability. The use of high-level tags (e.g., `happy`, `sad`, `angry`, `excited`, `techno`, `drumbeat`, `dancehall`) worked as a workaround to the use of high-level content-based descriptors (e.g., mood descriptors, genre descriptors).

7. USERS' FEEDBACK

7.1 Study Design

Next, we tested the tool with four experts (P1–P4) with more than 10 years of experience as music practitioners and qualitatively analyzed the results. We were especially interested in identifying differences between using crowdsourced vs. personal databases. From discussing the autoethnography experience between the authors, we

[16] The live performance's track is available online at `http://carpal-tunnel.bandcamp.com/track/n02-petermann`

[17] `http://toplap.org/wiki/ManifestoDraft`

[18] A video demo is available online at `http://vimeo.com/249968326`

identified a set of criteria that we wanted to evaluate: (1) predictability vs. serendipity, looking at the positives and negatives of having (or not) control over the selection of sound samples; (2) learnability, looking at how easy it is to use the system in a single session to see to what extent it is easy to use; (3) the size of the database, looking at the benefits and limitations of working with a specific online or personal sound database; (4) strategies, looking at the most successful combinations of descriptors and values given a particular online or personal sound database.

We asked the participants to try the tool in a single session at their home as a first step toward observing how other practitioners use the system. Two participants used their own personal database (P1 with 600 sounds, P2 with 2,000 sounds) and two participants used Freesound as a crowdsourced database (P3, P4). Participants had 30 minutes to learn the tool, and 10 minutes for recording a musical performance. They were asked to record the 10-minute session, which was sent to us together with the automatic list of sounds used, and fill in an online post-questionnaire with questions based on the criteria of evaluation. There were several 5-point Likert item questions (from strongly disagree to strongly agree) about predictability vs. serendipity ("Predictability is more important than serendipity when retrieving sounds from the database", "Serendipity is more important than predictability when retrieving sounds from the database", "Predictability is as important as serendipity when retrieving sounds from the database"); questions about the learnability of the tool ("One session of an hour is enough to feel confident with the system", "I found the system unnecessarily complex", "I thought the system was easy to use"); the size of the database ("I felt there were too many sounds in the database", "I felt there were not enough sounds in the database"). Finally, we surveyed, as open questions: the strategies used from a check-box list and additional comments about interesting queries, how to integrate this tool in their music creative workflow, and how to improve MIRLCRep. Here we focus on describing the actual queries the users tried, and getting their feedback regarding how they would attempt to use the system.

7.2 Findings

In general, the content-based queries were preferred. The most used combinations include "get random sounds" (P1–P3), "get sounds by content" (P1–P3), "get similar sounds" (P1–P3), "get sounds by tag" (P3, P4), and "get sounds by ID" (P1). The users of personal databases (P1, P2) agreed that their favorite combination was getting similar sounds, whilst the users of Freesound preferred "content with filter" (P3) and "sound by tag" (P4). Their favorite combinations were "similar sounds" (P1, P2); "sounds by content with filter" (P3); "sound by tag" (P4). Similarity was preferred by the users of personal databases: "querying by some of the more esoteric features and quickly getting a batch of similar things, and playing in sequence" (P1); "I like them all—both the curated (simplified high-level) and the low-level access. I especially liked multiple query of the same, to get 'variations' (like 'similar')" (P2). A user of Freesound preferred doing queries targeted to a specific task: "I tried to create four soundscapes; garden, street, city, airport" (P4), whilst the other user reported the need of more time to explore spectral feature queries (P3), which indicates that the tool requires a certain amount of time for exploration and discovery. The users reported how this tool could fit in their music creative workflow: The two users less interested in live coding envisioned to use the tool "both for exploring freeform corpora (like I have been today), and combining more beat driven stuff" (P1) as well as opening "a huge potential for sound design ideas for now (more than live coding)" (P2). The practitioners did adapt the tool or manifested interest in doing it to their workflows (e.g., "I made some adjustments to the class" (P3); "I am seriously considering using this library, it allows randomness and chance which are essential components of my aesthetic, it is really fun" (P3); "will be fun to combine with patterns in SC, no doubt" (P1); "I could integrate this with Tidal so I could query sounds live" (P4)).

Improvements were also suggested, such as "easier navigation of what's there in terms of distribution of features" (P1); "getting single shot playback, of indexed retrieval (let's say I ask for 10 similar sounds I can play them, remove entries, and continue a sort of interactive query)" (P2); "linking up with the pattern library would enable escaping from the looping sample layers paradigm which is ubiquitous and all too easy in electronic music" (P3); or "be integrated with something higher level like Tidal so the sounds can be transformed once imported" (P4). Two users (P1, P4) indicated that some functions failed occasionally (e.g., mute, stop) and the metaphor of group size linked to arrays was confusing at first (P4). Multiple query parameters at once was also desired: "tag and scale and key and tempo" (P4). Note that this is possible by directly passing long queries from Freesound quark into MIRLCRep. However this was unclear for this user, partly because it was barely documented in the context of this study.

8. DISCUSSION

From the users' feedback, it is clear that with a one-hour session it is difficult to understand the system and to relate it to the database and to the artist's aim. Users reported that understanding abstract information retrieval concepts such as content-based filters contributed to the difficulty. In both databases, the use of musical mid-level and high-level descriptors applied to sounds that have generally unknown structures (as opposed to music) gave a level of unpredictability and experimentation in the musical process that can be interesting for the practitioner. The experimentation of the limitations of the tool and subverting high-level descriptors was sought by the users, which is an expected behavior from expert practitioners. In the case of crowdsourced sounds, querying was perceived as a non-linear process, where sounds are retrieved organically following their own downloading times. Another observed difference between the two types of databases is the importance of segmentation when analyzing the sounds in the local databases. The possibility of segmenting sounds in advance, which is only possible in the local setting, adds flexibility and more precision of retrieving sounds by MIR features, which the user typically has little control when working with crowdsourced databases. By listening to the sessions sent by the participants, we found that the musical outcome from personal databases had a more homogeneous musical result, probably because of using sounds from one author only and their preference to use similarity queries. Thus, the combination of both types of databases is promising to add more contrast and surprise from the personal databases standpoint, and more homogeneity and consistency from the crowdsourced databases standpoint. These findings inform both AudioCommons and FluCoMa ecosystems in terms of the most popular descriptors used and common workflow practices around live performance when using MIR techniques.

9. CONCLUSIONS

This paper focused on MIR and live coding with audio samples and contributed with the comparison between using an open online and personal sound databases framed within the ecosystems AudioCommons and FluCoMa. Future work includes a better integration of the available descriptors into the live coding session; documenting useful combination of queries; mixing online and personal databases; conducting case studies and participatory design sessions with other practitioners; and exploring its use in pedagogical contexts, such as workshops. This research can inform real-time interactive systems for performance, education, and interactive art, as well as live coding and MIR research.

10. ACKNOWLEDGMENTS

We thank the participants of the study and the reviewers of this paper. AudioCommons is funded by the European Commission through the Horizon 2020 programme, research and innovation grant 688382. FluCoMa is funded by the European Commission through the Horizon 2020 programme, research and innovation grant 725899. The icon from Figure 1 is "Female" by Wouter Buning from the Noun Project.

11. REFERENCES

[1] V. Akkermans, F. Font Corbera, J. Funollet, B. De Jong, G. Roma Trepat, S. Togias, and X. Serra. Freesound 2: An Improved Platform for Sharing Audio Clips. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*. International Society for Music Information Retrieval (ISMIR), 2011.

[2] J. Barbosa, J. Malloch, M. M. Wanderley, and S. Huot. What Does "Evaluation" Mean for the NIME Community? In *Proceedings of the 15th International Conference on New Interfaces for Musical Expression*, pages 156–161, 2015.

[3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra. Essentia: An Audio Analysis Library for Music Information Retrieval. In *Proceedings of the International Society for Music Information Retrieval 2013*, pages 493–498, 2013.

[4] A. R. Brown. Code Jamming. *M/C Journal*, 9(6), 2006.

[5] L. Candy. Practice Based Research: A Guide. *CCS Report*, 1:1–19, 2006.

[6] P. Cano and M. Koppenberger. Automatic Sound Annotation. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing*, pages 391–400, 2004.

[7] K. Choi, G. Fazekas, and M. Sandler. Automatic Tagging Using Deep Convolutional Neural Networks. *arXiv preprint arXiv:1606.00298*, 2016.

[8] N. Collins. The BBCut Library. In *Proceedings of the International Computer Music Conference*, 2002.

[9] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live Coding Techniques for Laptop Performance. *Organised Sound*, 8(3):321–330, 2007.

[10] F. Font, T. Brookes, G. Fazekas, M. Guerber, A. L. Burthe, D. Plans, M. D. Plumbley, M. Shaashua, W. Wang, and X. Serra. Audio Commons: Bringing Creative Commons Audio Content to the Creative Industries. In *Proceedings of the 61st AES International Conference: Audio for Games*. Audio Engineering Society, 2016.

[11] F. Font, G. Roma, and X. Serra. Sound Sharing and Retrieval. In T. Virtanen, M. D. Plumbley, and D. Ellis, editors, *Computational Analysis of Sound Scenes and Events*, pages 279–301. Springer International Publishing, Cham, Switzerland, 2017.

[12] J. Freeman and A. V. Troyer. Collaborative Textual Improvisation in a Laptop Ensemble. *Computer Music Journal*, 35(2):8–21, 2011.

[13] S. W. Lee, H. Ave, A. Arbor, H. Ave, A. Arbor, and G. Essl. Live Coding YouTube: Organizing Streaming Media for an Audiovisual Performance. In *Proceedings of New Interfaces for Musical Expression 2017*, pages 261–266, 2017.

[14] A. Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. Wiley-IEEE Press, Hoboken, NJ, 2012.

[15] T. Magnusson. Confessions of a Live Coder. In *Proceedings of the International Computer Music Conference 2011*, pages 609–616, 2011.

[16] J. McCartney. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal*, 26(4):61–68, 2002.

[17] A. McLean and G. Wiggins. Tidal – Pattern Language for the Live Coding of Music. In *Proceedings of the 7th Sound and Music Computing Conference*, 2010.

[18] H. Ordiales and M. L. Bruno. Sound Recycling from Public Databases: Another BigData Approach to Sound Collections. In *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*, pages 48:1–48:8, 2017.

[19] C. Roberts and J. Kuchera-Morin. Gibber: Live coding audio in the browser. In *Proceedings of the International Computer Music Conference 2012*, 2012.

[20] C. Roberts and G. Wakefield. gibberwocky: New Live-Coding Instruments for Musical Performance. In *Proceedings of New Interfaces for Musical Expression 2017*, pages 121–126, 2017.

[21] J. Rohrhuber, A. de Campo, R. Wieser, J.-K. van Kampen, E. Ho, and H. Hölzl. Purloined Letters and Distributed Persons. In *Music in the Global Village Conference*, 2007.

[22] G. Roma, P. Herrera, and X. Serra. Freesound Radio: Supporting Music Creation by Exploration of a Sound Database. In *Workshop on Computational Creativity Support (CHI2009)*, 2009.

[23] G. Roma and X. Serra. Music Performance by Discovering Community Loops. In *Proceedings of the 1st Web Audio Conference*, 2015.

[24] D. Schwarz. Corpus-Based Concatenative Synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104, 2007.

[25] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton. Real-Time Corpus-Based Concatenative Synthesis with CataRT. In *Proceedings of the 9th International Conference on Digital Audio Effects*, pages 279–282, 2006.

[26] X. Serra, M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. Gómez, F. Gouyon, P. Herrera, S. Jorda, and Others. Roadmap for Music Information Research, 2013.

[27] T. Tsuchiya, J. Freeman, and L. W. Lerner. Data-Driven Live Coding with DataToMusic API. In *Proceedings of the Web Audio Conference*, Atlanta, GA, 2016.

[28] A. Zils and F. Pachet. Musical Mosaicing. In *Proceedings of the COST G-6 Conference on Digital Audio Effects*, pages 39–44, 2001.