Music performance by discovering community loops

Gerard Roma
Music Technology Group
Universitat Pompeu Fabra
gerard.roma@upf.edu

Xavier Serra
Music Technology Group
Universitat Pompeu Fabra
xavier.serra@upf.edu

ABSTRACT

Technologies for discovering sounds in large databases can help breaking the boundary between exploration and music performance. In this paper, we present a system for exploring loops from Freesound. Sound files are grouped by their most common repetition periods, so that they can be played in sync. A graph layout algorithm is used to organize sounds in a two-dimensional plane so that loops with similar timbre are spatially close. The result is a system that can be used as a musical instrument: since sounds will always play in sync, the user can freely explore the variety of sounds uploaded by the Freesound community, while continuously producing a rhythmic music stream.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*systems*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods*

General Terms

Algorithms, Design, Human Factors

Keywords

Audio discovery, web audio, music loops

1. INTRODUCTION

Music loops are very popular for music creation in many genres, especially repetitive electronic music. During the 1990s a market emerged for CDs of audio loops that were used in hardware samplers and computer software. Currently, online stores targeting DJs such as Beatport[1] offer catalogs of loop packs. Community-based sites like Looperman[2] or Freesound[3] also contain thousands of loops. However, the interfaces for searching loops in such sites are typically designed for offline use, in the sense that they are detached from the music creation experience. In contrast, software aimed at music production and performance offers sophisticated technologies for syncing and transforming samples, including loops. In this article, we propose a more musical method for exploring online loop databases using web audio.

2. RELATED WORK

Despite the adoption of electronic and dance music, and its reliance on loop samples, literature on indexing and retrieval of loops is relatively scarce. With respect to identification of music loops, there has been work on the isolation of loops in polyphonic music [9]. Here, frequently repeated patterns in a music files are identified through the autocorrelation of chroma features. Detection of loops in polyphonic music, however, is a different problem from indexing databases of (already cut) loop files. The former is more related to music structure and to an interpretation of what constitutes a "relevant" loop in the context of repetitive music, although from a practical point of view, an infinite number of loops could be extracted from the same track. In this paper, we depart from sound clips which may or may not have been cut as loops. Our problem, then, is identify loopable material in the context of unstructured and possibly unlabeled data. This opens the possibility of finding rhythmic patterns in all kinds of sounds, including environmental audio.

Some works have explored indexing and retrieval. Audiocycle [2] is a prototype that implemented analysis of musical loops, restricted to 4/4 rhythm meters and specific instrument classes. Although the dataset size is not explained, scaling to large databases is reported as future work. Other projects have focused specifically on drum loops [5], which affords a transcription approach. Most projects describe some kind of interface, either by using information visualization techniques to map the sounds to a 2D plane [2] [9] or using a Query-by-Example(QbE) approach [7] [6]. In general, research on indexing and retrieval of music loops lacks public datasets and evaluation metrics that facilitate comparison of different approaches. Instead of focusing on specific instrument sounds or drum loops, our system tries to find loopable sounds in a wide variety of audio. This

[1]http://www.beatport.com/

[2]http://looperman.com

[3]http://freesound.org

supports a wider range of music styles, depending on the available audio material.

Another trend related to our work is the discovery of sound clips using Self-Organizing Maps (SOM). For example in [10] the SOM was used for visualizing drum sample collections. An application to music is described in [11]. In [1] a system for browsing sound effects using SOMs is described. Our system also allows the visualization of the sound collection in a 2D plot, but the organization is done in a web client using a force-directed graph layout.

3. BEAT SPECTRUM ANALYSIS

While extensive MIR research has covered rhythmic analyisis of polyphonic music, not all algorithms are appropriate for analyzing loop samples. Many beat tracking and tempo induction algorithms produced by the MIR community focus mainly on percussive sounds, which is appropriate for popular polyphonic music but not for many music loops. Also, some algorithms require longer durations than the one or two bars typically spanned by loop samples. In this paper, we explore an alternative approach based on the Foote's Beat Spectrum [4]. The Beat Spectrum can be seen as a rather "classic" method for rhythm analysis. In the context of unstructured databases like Freesound, the interest of this method lies in its generality, as it does not focus on specific instrument sounds. While we follow the classic implementation using Mel Frequency Cepstral Coefficients (MFCC), the same approach can be base on any frame-level feature, such as Harmonic Pitch Class Profiles (HPCP). In order to obtain the beat spectrum, we first compute the self-similarity matrix of the sequence of MFCCs using cosine distance. From this matrix, the beat spectrum can be obtained by summing all the diagonals. Note that diagonals represent pairs of time points in the sequence of descriptors that share the same time lag, so a peak in the beat spectrum represents a typical repetition period in the underlying feature (in this case, timbre).

Figure 1 shows an example similarity matrix and the corresponding beat spectrum. From this base representation we can accomplish two important tasks in dealing with loops: we can identify them from other non-loopable samples in unstructured data, and we can index them according to their most common periods for rhythmic music creation.

4. IDENTIFYING LOOPS IN UNSTRUCTURED AUDIO

Since the decision about what is and what is not a loop can easily become complicated, we propose a pragmatic heuristic to identify loops in unstructured data: we assume that loops will have some rhythmic content, and that they have been devised so that the duration of the sound has a harmonic relation with the main pulse of this rhythm. This is in practice what defines loops in electronic music creation, since it will create the rhythm sensation when repeating the sound. Thus, we analyze the largest N peaks in the beat spectrum and for each one we compute the error with respect to the closest time position that corresponds to an integer divisor of the total duration. If a peak is found below a given error threshold T, the sound is labelled as "loop". Both parameters (the number of peaks and the threshold) can be used to adjust the sensitivity and so that we are more or less strict in the selection of loops. A slightly better detection can be

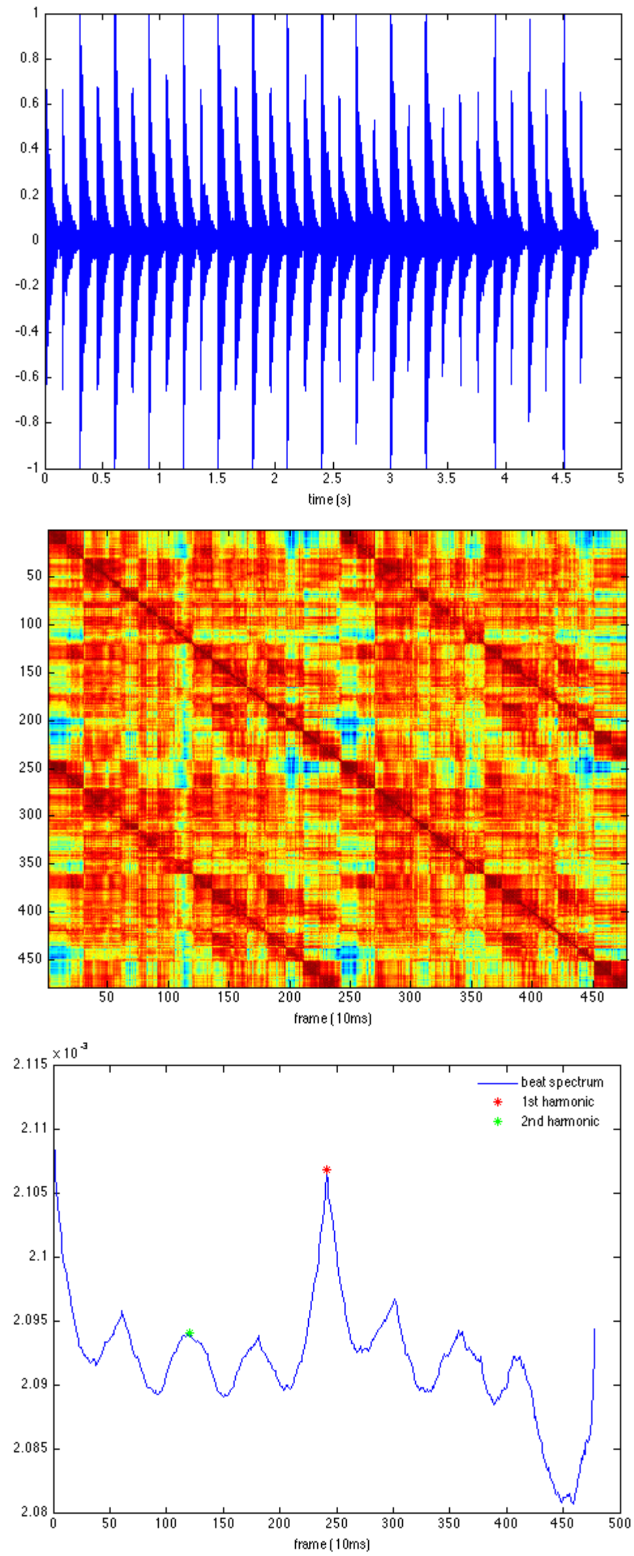

Figure 1: Waveform, MFCC similarity matrix and beat spectrum from a rhythmic loop

achieved by training an SVM classifier with some features extracted from the beat spectrum: in addition to the pres-

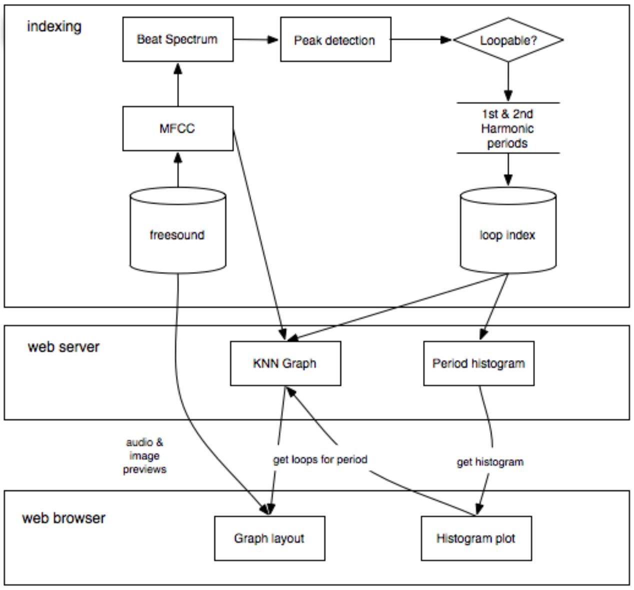

Figure 2: Block diagram of the system

ence of a harmonic peak, the value of the peak (the spectrum is normalized by the peak at zero lag), and the entropy of the beat spectrum[8](since loops contain rhythmic pulses, the spectrum will be peaky, while environmental sounds with no clear pulse will tend to be flat). These measures retain the advantage of not being directly related with timbre or other musical features that could be used to distinguish between "typical" loop sounds (e.g. with timbres associated to drum or synthetic sounds) and environmental sounds, so that classification is purely based in rhythmic aspects of the sound, and any loopable sound (at least with regularities in the MFCC sequence) can be identified, regardless of its source. For sounds that are identified as loopable, we extract the peak with minimum error among the highest N peaks, and seek a second harmonic of the duration defined by the first peak. The presence of the second harmonic ca be used as a stricter criterion for detecting loops. In our current system, the presence of the first harmonic is used as a criterion and both harmonics are used to index loops. This can be used as a relaxed version of tempo indexing, since for in many cases, for music creation, octave errors (i.e. sounds with double or half tempo) are not necessarily a problem. The harmonic repetition period acts then as a synchronization pulse, as most sounds that will be played together will contain typically 2 or 3 periods of the same length.

5. INFORMATION VISUALIZATION

We implemented a web-based prototype in order to test these ideas, using loops from Freesound. The index, computed as described above, currently contins about 30.000 loops. We use several information visualization techniques in order to present loops in a web environment. The graphical interface is based on D3.js[4]. First, we compute a histogram of all the harmonic repetitive periods in the database. Since the beat spectrum is computed from frame-level descriptors, harmonic periods are measured in frames, and the histogram

[4]http://d3js.org/

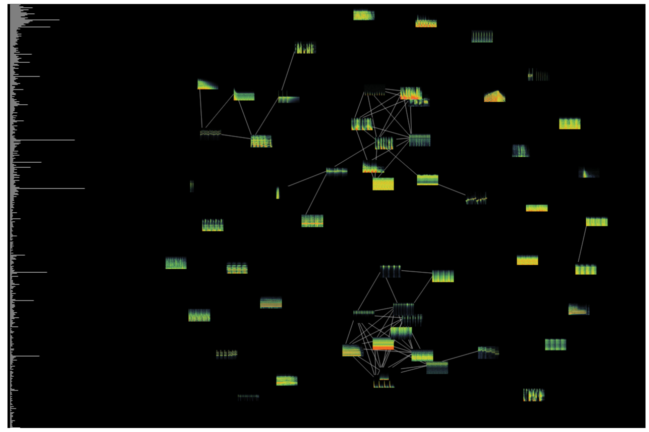

Figure 3: Screenshot of the prototype

is discrete. Peaks in this histogram correspond to frequent periods related with typical tempo values (e.g. 120BPM). This is a consequence of the social nature of the database. A histogram represented using D3 is directly used as an interaction device, so that selecting one bar from the histogram will load all sounds that share the same harmonic period. When queried for sounds with a given period, the server creates a k-nearest neighbors graph of the sounds according to their timbre similarity, as computed by cosine distance over the average of MFCC features. The value of k will control whether the graph is more clustered (or even disconnected) or more connected. This parameter can be set as a fraction of the number of nodes (in our case, this fraction is set empirically to 0.02. This graph is returned to the client, where a force-directed layout is used (again using D3) to display all the sounds in the page. Since the graph is built according to timbre similarity, the layout will tend to group similar sounds together, which helps in the exploration process. Finally, sounds are represented via the small spectrogram thumbnails available for all sounds in Freesound. This allows the user to quickly identify sounds with a strong pulse, as they will contain vertical lines. Sounds with melodies are also easily identified. Figure 3 shows a screenshot of the current state of the prototype.

6. SCHEDULING AND PLAYBACK

Loops can be started by clicking around the graph. Mp3 previews from Freesound are decoded and played using standard Web Audio API functions. The selected harmonic period is used for scheduling the start of each sound after it has been decoded. Since all sounds have durations that are (with some error) multiple of this period, their duration is adjusted to the pulse, and sounds are scheduled rather than actually looped. This ensures a consistent rhythm. For scheduling, a combination of $setInterval()$ and the $currentTime$ of the audio context is used, as described in [12].

7. EARLY EXPERIMENTS

The prototype is still in early development stages. However, reactions in initial demonstrations have been very positive. The prototype allows maintaining a certain creative flow, and can be used as a musical instrument, as well as an exploration of the sounds uploaded by the Freesound community. The main limitation is that some sounds are indexed according to harmonic periods that do not necessarily start at the beginning of the file. Foote proposed that the "phase" of the beat spectrum can be extracted as a novelty curve[3], which can be seen as an onset function. We plan to explore this feature for solving phase issues. Apart from this, there are some obvious improvements that could make the prototype more useful for DJs, such as using a separate pair of outputs for pre-listening, having a separate tray with currently playing sounds, and adding some filters. The current version of the prototype can be accessed at `http://labs.freesound.org/floop/`.

8. REFERENCES

[1] E. Brazil, M. Fernstroem, G. Tzanetakis, and P. Cook. Enhancing sonic browsing using audio information retrieval. Kyoto, Japan, 2002. Advanced Telecommunications Research Institute (ATR), Kyoto, Japan, Advanced Telecommunications Research Institute (ATR), Kyoto, Japan.

[2] S. Dupont, T. Dubuisson, J. Urbain, R. Sebbe, N. d'Alessandro, and C. Frisson. Audiocycle: Browsing musical loop libraries. *Content-Based Multimedia Indexing, International Workshop on*, 0:73–80, 2009.

[3] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000.

[4] J. Foote and S. Uchihashi. The beat spectrum: a new approach to rhythm analysis. *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, pages 881–884, 2001.

[5] O. Gillet and G. Richard. Automatic Transcription of Drum Loops. In *Proceedings of the 2004 IEEE Conference on Acoustics, Speech and Signal Processings (ICASSP'04)*, May 2004.

[6] O. Gillet and G. Richard. Drum loops retrieval from spoken queries. *J. Intell. Inf. Syst.*, 24(2):159–177, May 2005.

[7] A. Kapur. Query-by-beat-boxing: Music retrieval for the dj. In *Proceedings of the International Conference on Music Information Retrieval*, 2004.

[8] O. Lartillot, T. Eerola, P. Toiviainen, and J. Fornari. Multi-feature modeling of pulse clarity: Design, validation and optimization. In *ISMIR*, pages 521–526. Citeseer, 2008.

[9] B. Ong and S. Streich. Music loop extraction from digital audio signals. In *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME)*, pages 681–4, Hannover, Germany, 2008.

[10] E. Pampalk, P. Hlavac, and P. Herrera. Hierarchical organization and visualization of drum sample libraries. 2004.

[11] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. *Proceedings of the tenth ACM international conference on Multimedia MULTIMEDIA 02*, page 570, 2002.

[12] C. Wilson. A tale of two clocks - scheduling web audio with precision http://www.html5rocks.com/en/tutorials/audio/scheduling/, 2013.