Querying Freesound with a microphone

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

On the web, searching for sounds is usually limited to text queries. This requires adding textual descriptions to each audio file, which is indexed effectively as a text document. Recent developments in browser technologies allow developers to access the audio input or microphone of the computer, enabling Query by Example (QbE) applications.
We present a demonstration system that allows users to make queries on Freesound.org by recording audio in the browser. A basic prototype is available online.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*systems*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*web-based services*

General Terms

Algorithms, Design, Human Factors

Keywords

Query by example, web audio, audio retrieval

1. INTRODUCTION

Text is the fundamental medium of the web, but it is not necessarily the best way to deal with sound files. Researchers on audio indexing and retrieval have worked on alternative interaction paradigms for a long time. Query by Example (QbE) is generally used in music and audio retrieval for describing systems that use suitable examples (often audio) as queries for retrieving similar documents. For music with recognizable melodies, Query by Humming (QbH) is a popular approach, since most people recognize and remember songs by their predominant melody. However, due to the traditionally limited capabilities of web technologies, these paradigms have started appearing in online tools relatively recently. Technologies based on audio fingerprinting such as Shazam are now widely used in the mobile arena. Companies like midomi[1] have implemented QbH using Adobe Flash technology. Audio input can now also be used for text queries using google's speech recognition. Some research has also tackled the problem of querying sound effects and general audio. This problem is perhaps more complex, since producing a query that is similar to any arbitrary sound may not always be possible, and the relevant audio features will be different for different kinds of sounds. However, with current web standards it is possible to start experimenting with this idea using online databases. Since its inception, Freesound[2] has become a very popular site for sharing sounds. The only restrictions are that complete musical pieces are not allowed ("sound, not song", since the site is not thought as a music hosting application), and that they must be licensed using one of three Creative Commons licenses (CC-0, CC-BY or CC-BY-NC). In addition, the moderation stage also ensures that a sufficient text description is provided. Based on these minimal principles, web users have created a large collection currently holding more than 200.000 files including environmental sounds, musical instruments, loops, speech and any kind of sound one may imagine. In this paper, we describe a system for querying the database by recording sounds in a web browser.

2. RELATED WORK

The concept of QbE has been used in many research projects related with sound effects and environmental sound. For example, the project in [7] analyzed the case of environmental audio queries recorded on location. The idea of using vocal queries has also been extensively studied, including experiments with loops, percussive sounds and environmental sounds. A comprehensive review can be found in [4]. The idea of querying sound effects using onomatopoeias has also been explored [6]. However, onomatopoeias can be considered as iconic representations (discrete, language-dependent classes) so they are significantly different from QbE. The idea of querying Freesound with vocal imitations was recently explored for specific sound classes [2]. Outside research lab experiments, to the best of our knowledge, there is no system readily available with the scale of the Freesound database that can be used online in an interactive web setting.

[1] http://www.midomi.com/
[2] http://www.freesound.org/

3. ESSENTIA, GAIA AND FREESOUND SIMILARITY SEARCH

The Freesound website has traditionally offered an implementation of QbE in the form of "similarity search": for any sound, the user can query the database for sounds that are similar according to its acoustic properties. This feature is powered by two technologies developed over the past few years at the Music Technology Group: *Essentia*[3] [3] and *Gaia*[4]. The first is a library for audio analysis that features an extensive collection of descriptors. The second is an optimized engine for nearest neighbor search. When a sound is uploaded to Freesound, it is automatically analyzed using an *Essentia* extractor program that collects a large quantity of descriptors. These can be summarized by the main *Essentia* namespaces: *lowlevel* includes mainly low level descriptors computed from the magnitude spectrum, *tonal* includes descriptors extracted from chroma features (HPCP), *rhythm* contains descriptors extracted from onset detection, beat and tempo tracking algorithms, and *sfx* several descriptors aimed at sound effects, describing some properties of the spectrum and the temporal evolution of pitch and amplitude. Once the sound has been moderated, the analysis files of descriptor statistics generated by the extractor are indexed by a *Gaia* instance, a large in-memory database that contains the vector space defined by file-level statistics of all descriptors. The server allows the definition of different *views* corresponding to different sets of descriptors. A simple web server based on twisted-python[5] wraps *Gaia* as a web service that responds to similarity and descriptor queries. In addition to QbE, the server can be queried using descriptor ranges (e.g sounds with an average loudness comprised between two given values). Both functionalities are available to developers via a REST API[6], along with combinations of text and content-based queries (text queries are supported by a separate server running Apache Solr[7]). Through the website, only similarity search is supported via the mentioned "similar sounds" feature. For a more detailed description of the Freesound architecture, see [1]. With the release of *Essentia* and *Gaia* as open source software, new possibilities are open to new developers, since the same *Essentia* extractor that runs in Freesound can be used by an any developer to analyze any arbitrary sound and search for similar sounds in Freesound. For example, this could be useful for music creation applications using Corpus-based Sound Synthesis techniques [5]. The system described in this article is a demonstration of this functionality.

4. PROTOTYPE

We have implemented an experimental search system based on recent functionalities incorporated in the HTML standard for audio capture. The prototype is intended as a testbed while we implement this functionality in Freesound. It can be currently accessed at `http://labs.freesound.org/fsmic/`. The user is presented a basic bootstrap[8] layout with a record button. Upon startup, the page calls *navigator.getUsermedia()*[9], which prompts the user to allow the use of the microphone. The user can then capture audio via the computer microphone (or anything attached to the audio input) which is handled by recorder.js[10]. We also connect the incoming stream to a web audio API *AnalyserNode* in order to provide an oscilloscope visualization that gives some feedback about the audio being recorded. This is accomplished by connecting the stream to an AnalyserNode. Recorder.js returns a Blob that is then packed using a FormData object[11] and uploaded as multipart/form-data to the server through an AJAX request. The server is a very simple twisted-python script. When the audio is received, the server spawns a thread that extracts some descriptors using *Essentia*'s python bindings. The descriptors are then packed into a content-based query for the Freesound API, using the python API client[12]. The API returns a list of the most similar sounds to the one that the user recorded. The recording is discarded, and the list is rendered and returned to the client.

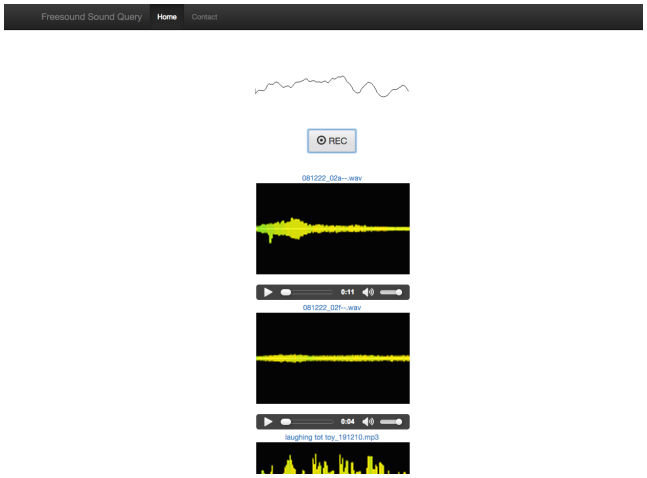

Figure 1: Screenshot of the current protoype

5. INITIAL EXPERIMENTS

While no formal evaluation of the system has been conducted, it is clear that the main issue is the selection of descriptors. The current implementation uses Mel Frequency Cepstral Coefficients (MFCC), computed over windows of 46ms with 50% overlap. To represent a given audio clip, we compute mean and variance of 13 frame-level MFCCs and of their derivative, which gives a vector of 52 descriptors that can be considered a generic timbre representation. The selection of features must cope with the fact that no assumptions can be made about the incoming sound with the exception of some probability that queries are performed by vocal imitations. In general, the usefulness of the system is limited by the number of available sounds similar to a given target. At the same time, the interface supports serendipitous discovery. Performing some vocal imitations

[3]https://github.com/MTG/essentia
[4]https://github.com/MTG/gaia
[5]http://twistedmatrix.com/
[6]http://freesound.org/docs/api/
[7]http://lucene.apache.org/solr/
[8]http://getbootstrap.com
[9]http://w3c.github.io/mediacapture-main/getusermedia.html
[10]https://github.com/mattdiamond/Recorderjs
[11]http://www.w3.org/TR/2010/WD-XMLHttpRequest2-20100907/
[12]https://github.com/g-roma/freesound-python

makes it easy to retrieve characteristic sounds. Beyond the limitations of the human voice, using musical instruments or other objects provides very interesting results. We plan to add some option for faceted analysis, for example using more musical descriptors for instrument sounds. For monophonic pitched sounds, the *YinFFT* algorithm implemented in *Essentia* provides a strength value that we have found useful for identifying pitched sounds in Freesound. Also it is easy to identify stable tones by using the statistics of the temporal evolution of pitch. In order to add this feature, an evaluation of the user query (potentially discarding non-pitched queries) would be useful. A search for rhythmic content could also be implemented by filtering the database with the "loop" tag. In this case, several rhythm descriptors can be used, such as tempo, tempo estimation statistics (BPM histogram peaks) and onset rate. We are now in the process of integrating this functionality in the site, however, the brevity of the code makes it a good example of the potential of the Freesound API for developers interested in QbE or corpus-based synthesis applications. The source code is available at https://github.com/g-roma/fsmic.

6. REFERENCES

[1] V. Akkermans, F. Font, J. Funollet, B. De Jong, G. Roma, S. Togias, and X. Serra. Freesound 2: An improved platform for sharing audio clips. In *International Society for Music Information Retrieval Conference (ISMIR 2011), Late-breaking Demo Session*, Miami, Florida, USA, 2011.

[2] D. S. Blancas and J. Janer. Sound retrieval from voice imitation queries in collaborative databases. In *AES 53rd Conference on Semantic Audio*, London, UK, 2014. Audio Engineering Society, Audio Engineering Society.

[3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: an open source library for audio analysis. *ACM SIGMM Records*, 6, 2014.

[4] G. Lemaitre, A. Dessein, P. Susini, and K. Aura. Vocal imitations and the identification of sound events. *Ecological Psychology*, 23(4):267–307, 2011.

[5] D. Schwarz. *Data-Driven Concatenative Sound Synthesis*. PhD thesis, 2004.

[6] S. Sundaram and S. Narayanan. Classification of sound clips by two schemes: using onomatopoeia and semantic labels. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1341–1344. IEEE, 2008.

[7] J. Xue, G. Wichern, H. Thornburg, and A. Spanias. Fast query by example of environmental sounds via robust and efficient cluster-based indexing. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5–8, March 2008.