# *THE EEAR*: BUILDING A REAL-TIME MIR-BASED INSTRUMENT FROM A HACK

**Cárthach Ó Nuanáin, Martin Hermant, Ángel Faraldo, Daniel Gómez**
Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
firstname.lastname@upf.edu

## ABSTRACT

In this demo we present *The Eear*, a set of musical creation tools that deal intelligently with chord and key qualities in musical signals. Using ubiquitous mobile devices such as smart phones and tablets, feature extraction is performed in real-time and sent to an audio plugin instrument on the desktop side for key aware sample selection and playback. We report on the progress of this instrument from its early incarnation as a hack presented at Sónar 2015 in Barcelona.

## 1. INTRODUCTION

The initial idea for the system was put together in a 24-hour period at the Music Hack Day at Sónar 2015. Our hack was very much dictated by the materials we had on hand, namely, an acoustic guitar, an AKAI style drum pad, several laptops and some cheap earphones (eventually used as "close contact" microphones for the guitar!). We were thinking about how to build a tool that enables laptop musicians to interact and jam with with traditional musicians while using some of our recent experiments and research into harmony detection, real-time concatenative synthesis and real-time MIR as part of the GiantSteps project [1].

Offline, we analysed and segmented a large portion of the Beatport 2 minute preview excerpts into 2-bar phrases and annotated the filenames according to key and BPM. Online, we attached a the earphones to the acoustic guitar and fed the input to a Pd patch running essentiaRT [2] - a real-time subset running within Max and Pd environments. The extracted chord information from the guitar was then used to assign audio clips, matching by key, to each pad on the hardware interface. In the next section we describe how we re-implemented this more formally following the hack.

[1] http://www.giantsteps-project.eu/
[2] http://mtg.upf.edu/technologies/EssentiaRT



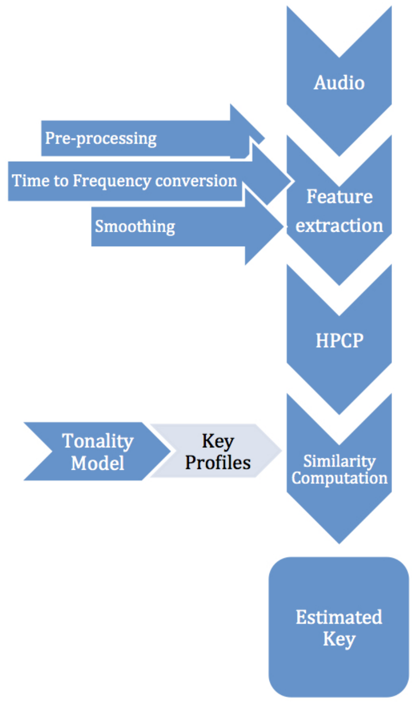


**Figure 1**. Key Extraction Block Diagram

## 2. IMPROVED SYSTEM

### 2.1 Key Extractor

It is important to distinguish between the role of the Key extractor as it works in offline or in real-time scenarios. In the offline batch extraction mode it captures an estimation of the overall *key* of the piece of music based on the whole track. In real-time or online modes, depending on the size of the captured audio, the key extractor will relay the localised *chord* rather than the overall key that is occurring over longer timespans.

The operation of the key feature extraction follows closely that described in [2] using functionality implemented in the Essentia library [1]. Figure 2 shows the block diagram of the algorithm and in the next section we will describe how we adapted it for real-time scenarios with mobile devices.

### 2.2 Mobile

The mobile application uses a version of the Essentia library compiled and optimised for ARM processors running Android and iOS. The application itself is written using

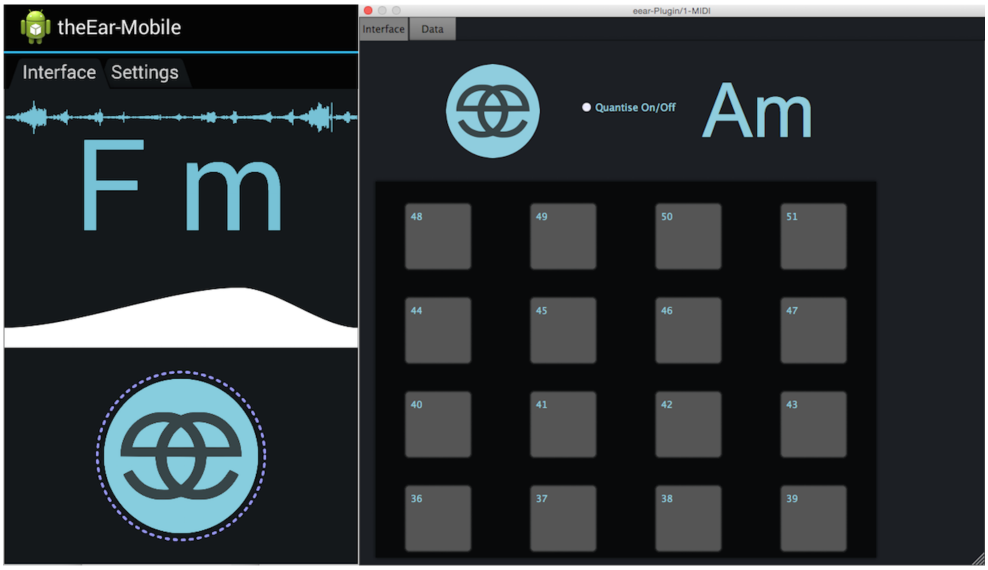


**Figure 2**. Left: Mobile Interface, Right: Plugin Interface

the JUCE framework [3] framework, a cross-platform C++ library especially suited for working with audio.

The application takes in audio from the device's microphone at framerate of 512 samples and stored in a ringbuffer. When the application is ready to analyse the key, the contents of the ringbuffer is fed through the algorithm network. Before the FFT stage, the signal is sliced into frame sizes of 4096 samples with 50% overlapping hop size and multiplied through a Blackman–Harris window.

Three other scalar features are extracted to further facilitate similarity searches between the live input and the dataset search. These include the Spectral Flatness - to give an indication of the harmonicity or inharmonicity of the signal -, its Spectral Centroid and its RMS level.

### 2.3 Desktop

On the desktop side we outline two musical scenarios where real-time key detection may be creatively useful and describe our implementations in both cases. Firstly we envisaged a musician with a traditional piano roll keyboard performing with live with other musicians. It can be a struggle to join in if you are not an experienced player or the other musicians are wind players and are playing in awkward keys. We therefore propose a simple Max for Live device that can use the real-time key information received over OSC to filter out notes that don't fit, replacing those notes by duplicating the correct ones that surround them.

Our other application acts as an intelligent sampler, using the Akai MPC pad format that was made popular during the hip-hop era but experiencing a resurgence today. It takes the form of a VST plugin (see the figure) with MIDI input and audio output. Before performing, the user analyses a folder of audio samples, extracting key, RMS, spectral details as outlined in the mobile section but with the addition of BPM. We supply already the *GiantSteps Key Dataset* [3], comprising 2-minute excerpts of 604 tracks in contemporary electronic dance music styles like techno and house.

The user can then set the desired key manually or automatically using the key information from the mobile sent over OSC. The key and accompanying features are then used to query corresponding audio in the supplied dataset. Matching audio samples are segmented into shorter user-defined loops and assigned to pads on the sampler. Thus with a broad collection of sounds in different keys, a sampling focussed laptop musician can also perform with real musicians, confident that they are consistent in tonality.

**Acknowledgements**

This research has been partially supported by the EU-funded MusicBricks project (H2020-ICT-2014-1 Grant agreement nr 644871) [4] .

## 3. REFERENCES

[1] D Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, P Herrera, O Mayor, G Roma, J Salamon, J Zapata, and Xavier Serra. ESSENTIA: an Audio Analysis Library for Music Information Retrieval. *International Society for Music Information Retrieval Conference (ISMIR 2013)*, pages 493–498, 2013.

[2] Emilia Gómez. Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, 18(3):294–304, 2006.

[3] Knees Peter, Faraldo Ángel, Herrera Perfecto, Vogl Richard, Böck Sebastian, Hörschläger Florian, and Le Goff Mickael. Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015.

[3] http://www.juce.com

[4] http://musictechfest.net/musicbricks/