EGT: Enriched Guitar Transcription

Loïc Reboursière and Stéphane Dupont

Laboratoire de Théorie des Circuits et Traitement du Signal (TCTS),
Faculté Polytechnique de Mons (FPMs), Belgique
{loic.reboursiere,stephane.dupont}@umons.ac.be

Abstract. EGT (Enriched Guitar Transcription) is a real-time and automatic guitar playing transcription software. Unlike most of the automatic score transcription software, not only the *note on*, *note off* events and pitch tracking are performed, but all the main guitar playing techniques are detected as well, providing a more complete transcription of the playing. These detections are made possible thanks to the use of an hexaphonic pickup (one pickup per string) enabling a string-by-string analysis. These transcriptions can then be used in many different contexts and / or embedded in different tools in order to obtain high-level information on the instrumentalist playing. This paper will demonstrate two use cases: a complete and realtime tablature writer and a 3D neck model controlled by the detected guitar playing events.

Keywords: Guitar playing techniques, hexaphony, music information retrieval, automatic score transcription, augmented guitar, guitar controller.

1 Introduction

From instrument-controlled synthesizer (may they be[1]analog or [2]digital), to automatic score writing, the detection of instrumentalist's playing and gestures have been the focus of many studies. These range from Music information Retrieval domain to augmented instruments area in which the instrumentalist playing is analyzed, characterized and used for different purposes. Regarding the guitar, research also range from MIR [2], [1]to augmented instruments [5], [3] and elements of playing at different scale have been studied, [4], [9], [6] .

In [8], we presented algorithms to detect each of the major guitar playing techniques. EGT (i.e, Enriched Guitar Transcription), integrates all these algorithms into a single software, performing real-time hexaphonic analysis of the string signals. The following playing techniques can be detected: hammer-on, pull-off, bend, harmonic, slide, palm muting. In addition to these, *note on* and *note off* events are reported, as well as the pitch of the played note. Besides, the plucking point (i.e, the position where the string is plucked) is although calculated.

[1] http://www.joness.com/gr300/GR-500.html
[2] http://windsynth.net/basics.html

M. Mancas et al. (Eds.): INTETAIN 2013, LNICST 124, pp. 163–168, 2013.

The detection software has been wrapped within a VST audio plugin. This standard has been chosen due to its integration in most audio software. The detected guitar playing events can be outputted in both OSC and MIDI formats.

This plugin doesn't only compute all the guitar events detection but provides other useful functionalities, which will be fully describe in section 2. In section 3, we describe two uses of this plugin: a complete and real-time tablature writer and a 3D guitar neck representation which reacts accordingly to the various detected guitar playing events.

2 Sofware Implementation

2.1 VST Choice and Output Formats

The VST audio plugin format was chosen because of its popularity and its availability on most music software, from Digital Audio Workstation, i.e ProTools, to any blank page real-time music software making, i.e Pure Data. As previously mentioned, all detected events can be fully outputted through OSC and the MIDI standard can be used as well to drive existing MIDI synthesizer. These communication options then make the plugin open to the majority of audio software. It needs to be pointed out that the use of MIDI standard entails a reduction of the set of techniques that are transcribed, as it does not provide a way to describe techniques such as palm muting, harmonics, etc. Some flexibility can be offered through the provision of MIDI control change messages though. When using OSC, the richness of all detected guitar playing events is kept however.

2.2 Software Elements

Setting Up. The software is made up of several elements organized in 4 tabs: General, Detector, Region and Behaviour. Several general options can be adjusted to set up the detection system: an **audio device** (and then sampling frequency and block size) as well as an external **MIDI device** can be specified. **Input gains** can be set up individually for each string but an automatic gain system can compute each gain after the instrumentalist has played it's loudest. The **tuner** is calibrated by defining the original tuning of the guitar. Eight different **OSC senders** can be defined to transmit the detected events through the network.

Predefined setups for the **detection parameters** will be available regarding the type of guitar and/or playing style and/or pickup used. A learn function will be available for each playing techniques if predefined settings don't fit the user.

Time Discretization. One of the key element when transcribing what an instrumentalist is playing is the notion of timing and duration of the notes. As digital notes and amplitudes, digital rhythms and durations need to be quantized. Present in most of the music software, MIDI transport is the norm used for discrete time representation. It has to be noticed here, that as a VST plugin, our detection software is meant to be incorporated into existing audio software, using, as a matter of fact, its host MIDI transport function.However not all audio software use MIDI transport and it seemed relevant to also include our own time quantization module. In the future, the software will allow the user to choose between an internal generation of discrete time and the one from its host.

For the moment, timing information is output as float values with *noteOn* and *noteOff* OSC messages. A subdivision parameter is accessible under the General tab in order to define the smallest quantization unit for notes duration (e.g, quarter, eight, sixteenth, etc.).

Preset. The preset system of EGT is handling 2 elements:

- a complete back-up of the system: all modified or created elements (input gains, detectors, behaviours, etc.) are saved.
- combination of those elements (preset) can as well be defined and managed (added, deleted or modified).

A XML file format was used to serialize the preset data, and the open source TinyXML[3] library has been implemented to enable this functionality.

Region and Behaviour. A six strings real-time playing techniques detection system can be CPU consuming and may be too much information to work with, depending on the situation. The two remaining tabs, Regions and Behaviours, enable the user to define, respectively, **where** on the fretboard he wants the detected events to be available and **what** he wants to detect. The idea behind these two tabs is to filter the flow of detection to what was really useful for the user. In both of our application cases however, all detections have been used.

The region concept was already prototyped in [7] and was expanded in this project as two types of regions can be defined: picking regions and fretboard regions. Picking regions are linked to plucking point detection. Three picking regions are defined by default (bridge, soundhole and neck) but others can be graphically created and spread all over the neck. A fretboard region is simply defined in a global sense as a **group of notes** which can be further characterized as chord, arpeggio or free depending on the time between each note of the group.

Figure 1 shows an example of behaviour and the region on which the behaviour is applied. To define a behaviour, the filtered playing techniques need to be selected (**what**) as well as the picking and fretboard regions (**where**) previously defined. MIDI and OSC outputs are available for the filtered events. Finally, a VST audio effect plugin can be chosen to be activated each time the behaviour is detected.

The combination of the region and the behaviour tabs can be used in many different contexts. A remote control zone, e.g, could be defined on specific notes (especially on hard to reach high end notes of the lower strings, but not only) in order to change presets or effects depending on the use case. Another example could be the one of different places on the fretboard linked to specific sounds or effects: the first five frets on all the strings could be setup with an overdrive effect and notes between the 12th and 15th fret on the three high strings could

[3] `http://www.grinninglizard.com/tinyxml/`

be setup with a reverb effect which amount could be controlled by the amount of bend. One last example could be chord recognition: as chords can be defined as a group of notes played under a defined amount of time, a specific chord is to be detected if the notes are played in the right order (upward or downward).

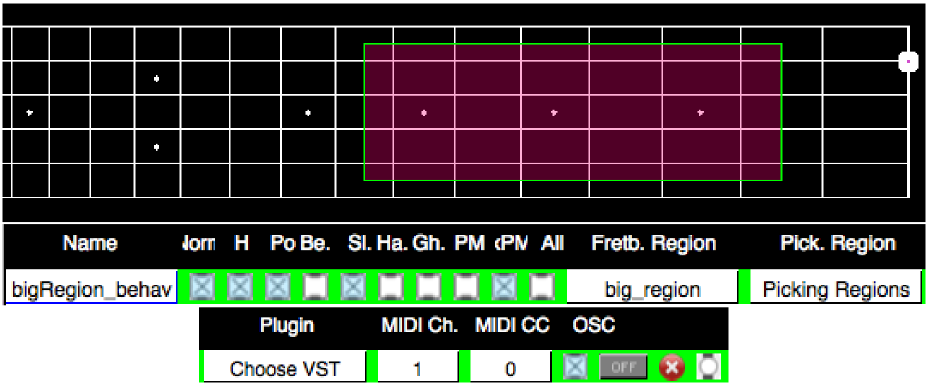

Fig. 1. This behaviour filters normal notes as well as hammer-on, pull-off and slide techniques performed in the fretboard region showed on the top part of this graphic

3 Use Cases

Two use cases are demonstrated here: a real-time and complete guitar score transcription in[4]TuxGuitar, an open-source tablature software and a real-time manipulation of a 3D representation of a guitar fretboard done in[5]openFrameworks.

3.1 TuxGuitar Input: Real-Time Score Writing

Tablature is the most common type of guitar score available to guitarists due to its easy readability. Indeed, on tablature, notes are represented by a number of fret on a specific string. All the guitar's specific techniques have their own representation as well. We decided to build a plugin for TuxGuitar tablature software which would receive OSC messages from EGT and display them on the tablature editor. This opensource software was chosen among others (i.e, KGuitar and DGuitar) because of its continuity in terms of development as well as its rich plugin API for which previous code examples were available. [6]MusicXML format has been investigated as well. Although it is implemented in several notation software i.e Finale, Sybellius, or specific guitar tablature

4 http://tuxguitar.herac.com.ar/
5 http://www.openframeworks.cc/
6 http://www.musicxml.com/

ones, i.e Guitar Pro, the implementation quality appeared uneven, and we hence decided to postpone the integration. The Guitar Pro file format may also be consider in future development.

The current TuxGuitar plugin implementation gives a first basic framework whose accuracy result tends to depend on the complexity of the played track as well as on the quality of the interpretation. It has to be noticed that no precise study has yet been performed on the accuracy of the system. From the first tryouts though, a stronger and global robustness step is needed regarding the time management and events detection. TuxGuitar software may as well be a source of mistakes regarding the display as the real-time entry plugin we built is more a hack than a native API function.

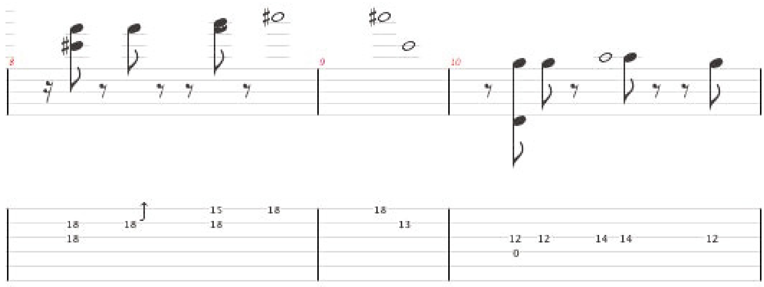

Fig. 2. Some of the first measures of Shine On You Crazy Diamond by Pink Floyd. The detected guitar playing events have been performed by EGT.

3.2 OpenFrameWorks: Control of a Guitar Neck 3D Representation

For this use case, we wanted to have an aesthetic representation of a guitar neck which would react to EGT's detected events. An OSC receiver has been implemented in this OpenFrameWorks program in order to interpret EGT's messages and make them interact with the 3D model. The model is based on an hexagonal tube, representing the fretboard, inside which hexagonal planes represents frets. A plucked note is represented by a burst on the string whose amplitude depends on guitar's notes amplitude. When a note is plucked the tube turns on itself so that the string corresponding edge appears in the foreground. The different playing techniques have their own graphical representations. Examples of this interaction can be found[7]online.

4 Conclusion and Perspectives

EGT is a guitar transcription software upon which higher-level tools can be developed. It detects the following techniques: hammer-on, pull-off, harmonic, slide, bend, palm muting, as well as note start and end timings, pitch and plucking point. Additionally, a concept of regions and behaviors has been designed in

[7] https://vimeo.com/34504237

order to filter a set of guitar events happening on a specific place of the guitar fretboard. The software has been wrapped into a VST plugin and the detected events can be sent through OSC or in a reduced version through MIDI. Two use-cases have been presented: a real-time tablature writer and a 3D guitar neck representation animated using parameters from the detected events.

This tool appears to be a first good base for guitar's playing transcription. However, a precise user study with different playing styles needs to be done to make the detection system more robust and adapted to the instrumentalist's playing. EGT is also a first step towards higher-level concepts: one can, e.g, easily imagine a complete musical phrase record system to help studying and analyzing the playing at a macro structure level. More globally, these detections open up to a substantial study on mappings, to see how they can be used to interact with other media (e.g, sound or video synthesis parameters, light control, etc.). A lot of strategies can now be investigate to make the control of guitar effects evolve to a more refine and instrumentalist linked control.

Acknowledgments. Numediart is a long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N°716631).

References

1. Barbancho, A.M., Klapuri, A., Tardon, L.J., Barbancho, I.: Automatic transcription of guitar chords and fingering from audio. Trans. Audio, Speech and Lang. Proc. 20(3), 915–921 (2012)
2. Fiss, X., Kwasinski, A.: Automatic real-time electric guitar audio transcription. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, Prague Congress Center, Prague, Czech Republic, May 22-27, pp. 373–376. IEEE (2011)
3. Graham, R.: A live performance system in pure data: Pitch contour as figurative gesture. In: Proc.of Pure Data Convention, Bauhaus-Universität, Weimar, Germany (August 2011)
4. Guaus, E., Ozaslan, T., Palacios, E., Ll Arcos, J.: A left hand gesture caption system for guitar based on capacitive sensors. In: Proc. of NIME (2010)
5. Lähdeoja, O.: An approach to instrument augmentation: the electric guitar. In: Proceedings of the 2008 Conference on New Interfaces for Musical Expression, NIME 2008 (2008)
6. Penttinen, H., Välimäki, V.: Time-domain approach to estimating the plucking point of guitar tones obtained with an under-saddle pickup. Applied Acoustics 65(12), 1207–1220 (2004)
7. Reboursière, L., Frisson, C., Lähdeoja, O., Mills III, J.A., Picard, C., Todoroff, T.: Multimodal guitar: A toolbox for augmented guitar performances. In: Proc. of NIME (2010)
8. Reboursière, L., Lähdeoja, O., Drugman, T., Dupont, S., Picard, C., Riche, N.: Left and right-hand guitar playing techniques detection. In: Proc. of NIME (2012)
9. Traube, C., Depalle, P.: Extraction of the excitation point location on a string using weighted least-square estimation of comb filter delay. In: Proceedings of the Conference on Digital Audio Effects, DAFx (2003)