Towards An Educational Music Processor for Folk and Popular Musics

Anh-Thu G. Phan, M.A., Ed.M. Music Education
Teachers College
Columbia University
New York, NY USA
agp2132@tc.columbia.edu

Center for Vietnamese Philosophy, Culture & Society
Temple University
Philadelphia, PA USA
anh.thu.phan@temple.edu

Thanh-Nhan Ngo, Ph.D. Linguistics
Courant Computer Science, Medical Language Processing
New York University
New York, NY USA
nhan@cs.nyu.edu

Center for Vietnamese Philosophy, Culture & Society
Temple University
Philadelphia, PA USA
nhan@temple.edu

***Abstract*—This paper describes an educational musical processor that takes spectrographic data of a sonic object and turns them into a series of meaningful layers associated with different musical knowledge representations, in such a way that it can be understood, reproduced, played, compared, and taught by everyone across cultures, regardless of their musical backgrounds. Any music audio file can be used as input. Within the scope of this paper, the authors focus on processing of musical audio files onto common graphic platform of physical sound properties, in Hertz, Decibels, and milliseconds, so that culturally dependent musical units such as notes, beats, measures, phrases, chords, and sections can be viewed in separate layers. Syntactic techniques, such as frequency of occurrences, and adjacency are applied to musical units, such as pitches and musical chords. They are key pitches in context and key chords in context. The results are then mapped onto circles of fifths which reveal distinct patterns of each song, each section of one song, of each artist, each genre, and each culture. Semi-automatic generation of layers of annotations on top of the spectrogram helps teachers to quickly discover and/or compare distinctive features of a song, while preparing lessons. Learners of all levels can choose the most prominent patterns of the song to learn. This can also advance methods for preservation for further studies of sonic objects in the future.**

Keywords—music processor, music representation, audio files, frequency, pitch, melody, chord, harmonic progression, musical form, circle of fifths, frequency of occurrence, distinctive feature, key words in context, key pitches in context, key chords in context.

I. Introduction

For years, social sciences and humanities rarely involve the study of music. With the advent of artificial intelligence, especially an extensive understanding of natural language processing, and the massive use of digitized sonic objects, it is possible to imagine a platform to present, preserve, and analyze musical and linguistic elements using string grammar, as in [1]. Music and songs can be thought of as linear strings of pitches or syllables from performers to listeners. Music and linguistic units, such as pitches and syllables, are inherently not well-tempered. Notating human music in western notation has stripped away many essential music features, as in [2]. Adding these once-lost features back to their context (adjacent pitches and syllables) reveals culturally dependent characteristics of sonic artefacts, as in [3]. In this paper, natural and well-tempered pitches are presented in different layers or superimposed on each other to help further analysis and comparison, as in [4] and [5]. Linearity, adjacency, regularity, structure, and unit oriented properties of music from pitches, phrases, sections are similar enough, to that of natural language processing (NLP), as in [6]. This NLP processor helps sketch major components of this educational music processor (EMP).

This paper is largely divided into two parts, one part using technologies to convert musical objects into physical data, measured by Hertz (Hz), Decibels (dBs), and milliseconds (ms). From these units, a layer of musical clefs and layers of interpretive manual insertion of beats, measures, phrases, sections, lyrics, and chords, are overlaid laid. The second part is dedicated to describing an EMP via a test case. It displays frequencies of occurrences of pitches and chords, and strings of adjacent pitches and adjacent key chords. The results expose the unprecedented characteristics of the test song.

II. Obtaining Data

Data of this paper can be divided into two categories: phenomenal and conventional (or knowledge). Phenomenal data are physical signals such as time, frequency, and intensity. Conventional data are interpretive symbols and knowledge that have been used in music and languages in different cultures. Examples of conventional data in this paper are musical beats, pitches, phrases, sections, quarter notes, measures, treble and bass clefs, and chords, which exist only in culturally learned contexts.

III. Technologies Used

Top-of-the-art technologies are tested and chosen based on three criteria: user-friendliness, cost, and standardization.

978-1-5386-7983-8/18/$31.00 ©2018 IEEE

A. Sound Analysis Software

1) *Sonic Visualiser,* is a free software [7] that allows visualized representations of audio input such as soundwaves and spectrograms. It allows manually or plug-in overlaid annotations on top of one another. For examples: *Chordino Chord Estimation,* generates an SVL output[1] containing frames (defined in IV.A. below) associated with the chord labels [8]. *Melodia Melody Extraction* generates an SVL output containing frames associated with the melodic pitches [9]. *Queen Mary Note Onset Detector/Percussive Onset Detector* generates an SVL output containing frames associated with an onset of a percussive sound [10]. *BBC Intensity* generates an SVL output containing frames associated with an intensity value [11].

2) *MuseScore (offline)* is a free standalone open-source software for music notation [12]. The companion *MuseScore (online),* is an online platform allowing users to share and edit sheet music, and connect to other websites, or through Youtube videos [13]. *MuseScore* offline and *MuseScore* online versions are used *in tandem* by the EMP to create a panel of western musical notation.

3) *AnthemScore* is a standalone which suggests a transcription of an audio file (MP3, WAV, etc.) into sheet music using western notations [14]. It suggests pitches based dominant frequencies' amplitudes of the song's spectrogram.

B. The educational musical processor (EMP)

1) The EMP is coded in PHP 7.2 with SVG 1.1[2], follows the international, world wide web, archival, librarian and multilingual standards, Unicode and ISO/IEC 10646 [15], world wide web standards are HTML [16], XML [17], URI and MusicXML.[3] The EMP also adheres to the Open Archive Initiative/Object Reuse and Exchange (OAI/ORE) [18] with Dublin Core Metadata Initiative (DCMI) [19].

2) The EMP runs about 6,000 lines of codes on a Dell T1650 workstation, with Ubuntu 18.04.1.

IV. A Test Case: Despacito by Luis Fonsi 2017

The video version of *Despacito* by Luis Fonsi ft. Daddy Yankee (hereafter, simply *Despacito*)[4] was chosen to guide the design of our folk and popular music processor. The song is the most viewed YouTube video.

A. Frame

Frame is a common feature that pairs all data collected. Each frame is 1/44,100 second or 10/441 millisecond. 44,100 Hz is one of the standard digital sample rates.

B. Spectrographic data plotted as the background layer

Despacito is 4 minutes and 42 seconds long. *AnthemScore* reads *Despacito* and generates a spreadsheet of spectrographic data. Data can either be obtained from every 1 ms to every 10 ms, in time, and from every 5 cents to every 25 cents, in pitch. The authors chose to test two sets of data: (a) a fine set of 1 ms by 25 cents in a spreadsheet of 351 pitch columns by 281,522 one-ms rows; and (b) a crude set of 10 ms by 25 cents in a spreadsheet of 351 pitch columns by 28,153 ten-ms rows. Each cell contains an intensity value in dBs. The former spreadsheet is 774 MB and the latter, 94 MB, in size.

The EMP reading of the fine spreadsheet took 130 ms, with a pitch range from 26.7171 to 4,308.66 Hz, and intensity range from 0 to 17,872 dBs. The crude spreadsheet took 15.5 ms to read, with the intensity range from 0 to 4,246.1 dBs.

C. Musical grand staff and piano keyboard as SVG Layer 1

The 351 pitch values is set on the vertical axis of *Despacito* spectrographic graph, *Layer 0* (Fig. 1), ranging from C^0 (0 *cent* at 26.7171 Hz) to B^8 (10,700 *cents* at 4,366.08 Hz). The pixel colors, from deep blue, lowest, to white, strongest, represent the intensity in dBs of the song. This allows a treble and a bass clef, *Layer 1*, with a piano 88 keys from A^0 (900 *cents*) to C^8 (9,600 *cents*) to be drawn on top of *Layer 0*. The staff lines run along the horizontal time axis.

D. Layer 2: Labelling of Beats, Measures, Chords, Phrases and Sections on Despacito spectrogram

These data are generated by *Sonic Visualiser* in frames with manual inputs, in SVL format. They are superimposed on *Layer 0* and *Layer 1* to show how human perception paired with the physical data.

On top of each panel of spectrogram, a bar graph of dB values at each frame in the song is drawn in red with white dots underfoot representing the percussive onsets (Fig. 2).

The beats are shown in yellow arrows inside red transparent measure boxes and the chords, in cyan, at their onset time positions (Fig. 3).

At the bottom of *Layer 0*, the phrase boxes in orange contain their lyrics in black, while the lowest light brown bars represent sections in the song (Fig. 4).

Manual markings of *Despacito* result in 83 phrases, 113 lyric expressions, 22 chords, and 26 sections (representing 11 music forms labelled from A to L),.

E. Layers for collections of pitch intensity of each beat, measure, chord, phrase and section.

Specifically, *Layer 3* shows dBs by measures, *Layer 4*, dBs by beats, *Layer 5*, dBs by chords, *Layer 6*, dBs by phrases and *Layer 7*, dBs by sections. Fig. 5 (left) shows *Layer 5*.

F. Layer 8, music sheet of Despacito according to AnthemScore

This is an estimate by AnthemScore. The notes are displayed over Layer 0, Layer 1 and Layer 2 to pair the estimates with the spectrographic data and other manual data.

[1] SVL is an XML Sonic Visualiser data exchange format.

[2] At https://www.w3.org/TR/2011/REC-SVG11-20110816/.

[3] MusicXML 3.0, in XML format for exchanging digital sheet music, see https://www.w3.org/2017/12/musicxml31/.

[4] At https://www.youtube.com/watch?v=kJQP7kiw5Fk.

The sums of all pitch dB values within a beat, a measure, a chord, a phrase and a section, reveal the strongest pitches within, evidence of played notes, marked on the vertical axis. These layers are intended to reveal also differences in the manual perception against the raw spectrographic data.

G. Layer 9, AnthemScore estimation is displayed through MuseScore applications.

This layer is the most interactive, by *AnthemScore*. Viewers can play portions of *Despacito,* section by section. They can also play back the sheet music, section by section.

Layer 0 to 9, takes about 235 seconds to produce by the EMP, viewed at http://mlp.cs.nyu.edu/vietmusic/au2spec.php.

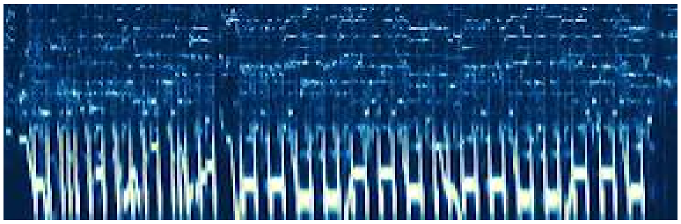


Fig. 1. A segment of spectrogram of *Despacito's* recording.

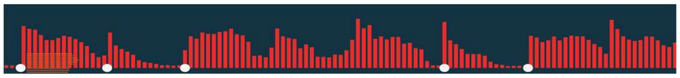


Fig. 2. An example of *Despacito* intensity and percussive onsets.

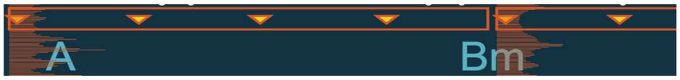


Fig. 3. An example of *Despacito* markings: down arrows for beats, rectangular boxes for measure, cyan letters for chord symbols.

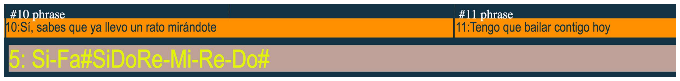
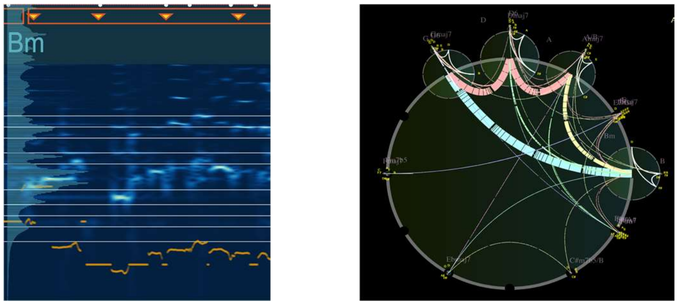

#10 phrase
10:Sí, sabes que ya llevo un rato mirándote
#11 phrase
11:Tengo que bailar contigo hoy
5: Si-Fa#SiDoRe-Mi-Re-Do#

Fig. 4. An example of *Despacito* markings: orange rectangular boxes for lyrics and light brown rectangular boxes for sections.



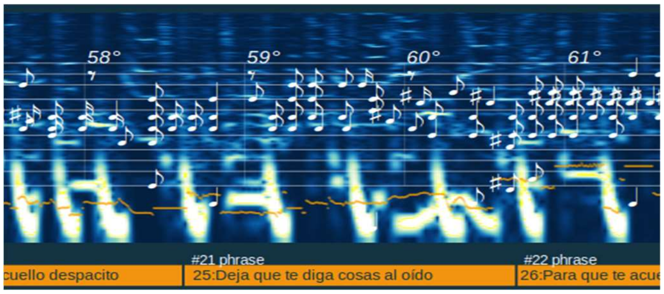


Fig. 5. Left, an example of Layer 5: sum of dBs by Chords B minor over *Despacito* spectrogram image. Right, the chord procession kcic(2) with their frequencies of occurrences in thickness of over the circle of fifths.

Fig. 6. An example of music notation on top of the spectrogram

I. Pitch and chord frequencies of occurrences, and key chords in context kcic(n), and their visual representations on the Circles of Fifths.

AnthemScore estimates that *Despacito* has a range of 47 pitches, from B^0 to $C\#^7$: D^4 occurs 508 times, B^3, 353 times, $F\#^4$, 241 times, A^3, 201 times, E^4, 173 times, and so on. They are drawn on the music clefs, as well as on a circle of fifths with their frequencies as radii. This shows that even though the song has all 12 pitch classes, they strongly lean on 4 central pitches (highest frequencies of occurrence), while others are just decorative. Further analysis confirms this observation.

If the song is estimated section by section, it turns out that *AnthemScore* now produces a range of 60 pitches, from B^0 to D^7. The same 4 notes are still central to the song: D^4, occurs 612 times, B^3, 433 times, $F\#^4$, 292 times, and A^3, 243 times, and so on. Their class positions are more prominent in the circle of fifths. The same analysis on frequencies of occurrences of pitches done between sections shows the tendencies of section movements towards each other, serving as clues to identification of musical forms.

Here the concept of key pitches in context kpic(n) and key chords in context kcic(n), $1 < n < m$, where m is the total number of pitches or chords in a section, are applied. kpic(n) and kcic(n) are modeled after *key words in context*, kwic(n), that helps show regular pitch and chord strings in the song.

A sample of kcic(2), $n = 2$, is shown in Fig. 6 (right) on the circle of fifths with their frequencies of occurrences. It is immediately obvious that *Despacito* has 4 prominent chords, D major, G major, A major and B minor. Chord progressions B minor to G major appears 22 times, G major-D major (18 times), D major-A major (18 times), and finally A major-B minor (10 times). While B minor-G major is most frequent, there is no G major-B minor progression.

Twenty-six song sections are displayed with frequencies of pitches on the music clefs and their proper positions, as well as their pitch classes on the circle of fifths, together with their music forms. Changes in the circles of fifths visualize the class content shifts in forms of song progression by sections.

H. Lyrics in phrase bars

In this final panel, for now, another way of data presentation can be viewed in a familiar manner, in a phrase by phrase layout, containing lyrics with chords associated with their proper lyric syllables. This presentation is a shorthand way for viewers to sing along without being bothered with music clefs and music pitch notes, measures or beats.

Items J. to L. currently take about 20 seconds by the EMP, can be viewed at http://mlp.cs.nyu.edu/vietmusic/au2ana.php.

I. Current Issues

At the moment, the music processor adopts temporarily the use of the best spectrogram analysis by current leading software. This decision creates discrepancies in data units, and requires data unification into *milliseconds* and *cents*. There are other practical issues such as the best approaches for data presentation in layers, as the EMP for *Despacito* currently transfers a massive 330MB of graphic data to browsers.

The EMP begins to modestly scale up to over 4 minutes with 281,522x351 spectrogram of 99 million data points. The first spectrographic and all manual onsets of 9 layers took too long, 235 seconds of processing time. However, they are intended to be processed each separately according to the preference of viewers. The second analysis now takes 20 seconds to produce all 5 subparts, which can also be presented separately based on the viewer preference.

V. Educational Implications

The main goal of this paper is to work toward a model of music teaching and learning that would be more generally accessible, while not sacrificing the contents and quality of musical experiences.

The use of digital technologies to facilitate teachers and students to develop musical skills and knowledge, most efficiently in the following ways. First, the EMP provides a graphical analogue of music that bypasses the need for specialized musical notation and symbols. Graphical representation is especially suited for teaching musics whose tonalities do not often fit well with the western chromatically-based tonal system. In this sense, these graphs help describing the music inputs faithfully with visual aids. After that, the EMP proceeds to calculate prevailing rhythmic and melodic patterns of a songs by constantly comparing units of data based on the theory of string grammar and deep learning. These patterns serve as basic blocks for improvisation and stylistic recognition. Finally, it allows any teachers to analyze and device lesson plans on any songs suggested by the students. The lessons are automatically tailored to match the student's ability. Teachers are then able to have more time focusing on accommodating other peculiar student needs, encouraging individual interpretations, and providing contextual knowledge.

Acknowledgment

The authors thank the *Linguistic String Project*'s Medical Language Processing of the Computer Science Department, New York University Courant Institute of Mathematical Sciences, and the *Center for Vietnamese Philosophy, Culture & Society* of Temple University's Henry Luce Grant No. 270380 / *Expanding Opportunities for Research on Vietnam*, as well as Ambassador Nguyễn Phương Nga and the Cultural Section of Permanent Mission of the Socialist Republic of Vietnam to the United Nations who donated a top rank Dell workstation toward the EMP project.

References

[1] Z. S. Harris, String Analysis of Sentence Structure. The Hague: Mouton & Co., 1962.

[2] P. Vĩnh, "Hệ thống Điệu và Hơi trong nghệ thuật ca - đàn Huế," [A system of scales and airs in the arts of songs and music of Ca Hue], Chim Việt Cành Nam, 2014, retrieved at http://chimvie3.free.fr/54/vinhphuc_cahue.htm.

[3] T. N. Ngô and G. A. T Phan, "A Contribution to Teaching Vietnamese Music: Key Pitches in Context and Pitch Contour Graph," Journal of Social Sciences and Humanities, vol. 3, no. 5, pp. 573-585, Hanoi: Vietnam, 2017.

[4] G. A. T. Phan and T. N. Ngô, "Capturing the Music: A case study of lý con sáo three regional "Songs of the Starling"," presented at the 46th Annual Mid-Atlantic Region Association for Asian Studies Conference: Mobility, Technology and the Environment, Drexel University, Philadelphia, PA, 2017.

[5] G. A. T. Phan and T. N. Ngô, "Initial thoughts on analyzing sonic objects to aid multicultural education," presented at the 2017 Asian American Education Conference, Teachers College, Columbia University, New York City, NY, 2017.

[6] N. Sager and T. N. Ngô, "The computability of strings, transformations, and sublanguage," The legacy of Zellig Harris: Language and information into the 21st Century, ed. by Bruce E Nevin and Stephen Johnson. John Benjamins Publishing Co. Amsterdam/Philadelphia. Volume 2:79-120, 2002.

[7] Center for Digital Music of Queen Mary, University of London, Sonic Visualiser, version 3.1.1, 2018, retrieved at https://www.sonicvisualiser.org/.

[8] M. Mauch and S. Dixon, "Approximate Note Transcription for the Improved Identification of Difficult Chords," proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010), 2010.

[9] J. Salamon and E. Gómez, "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics", IEEE Transactions on Audio, Speech and Language Processing, 20(6):1759-1770, Aug. 2012.

[10] D. Barry, D. Fitzgerald, B. Lawlor, and E. Coyle, "Drum Source Separation using Percussive Feature Detection and Spectral Modulation," in ISSC 2005, 2005.

[11] C. Baume and B. Y. Raimond, Broadcasting Corporation, Intensity Vamp plugin, 2013, retrieved at https://github.com/bbc/bbc-vamp-plugins/releases.

[12] MuseScore BVBA, 2018, retrieved at https://musescore.org/en.

[13] MuseScore BVBA, 2018, retrieved at https://musescore.com/dashboard.

[14] Lunaverus, AnthemScore, 2018, retrieved at https://www.lunaverus.com/.

[15] The Unicode Standard, version 11.0, 2018/06/05, retrieved at https://www.unicode.org/versions/Unicode11.0.0/.

[16] World Wide Web, HyperText Markup Language (HTML) 5.2, 2017, retrieved at https://www.w3.org/TR/2017/REC-html52-20171214/.

[17] World Wide Web, Extensible Markup Language (XML) 1.0 (Fifth Edition), 2013, retrieved at https://www.w3.org/TR/xml/.

[18] Open Archive Initiative - Object Reuse and Exchange (OAI/ORE), ORE User Guide - Primer, 2008, retrieved at https://www.openarchives.org/ore/1.0/primer.

[19] Dublin Core Metadata Initiative, DCMI Specifications, 2005, retrieved at http://dublincore.org/specifications/.