TENOR 2016

INTERNATIONAL CONFERENCE ON TECHNOLOGIES FOR MUSIC NOTATION & REPRESENTATION

ANGLIA RUSKIN UNIVERSITY

CAMBRIDGE, UK

Second International Conference on Technologies for Music Notation and Representation

TENOR 2016

27-29 May, 2016

Anglia Ruskin University, Cambridge UK

Proceedings published by :
Anglia Ruskin University, Cambridge, UK

ISBN : 978-0-9931461-1-4

Editors :
Richard Hoadley, Dominique Fober, Chris Nash

Credits :
Nicolas Taffin (Logo design) Nicolas Viel (Layout design)

Anglia Ruskin University
Cambridge Chelmsford Peterborough

GRAME
CENTRE NATIONAL DE CRÉATION MUSICALE

UNIVERSITY OF CAMBRIDGE
Faculty of Music

Centre for Music & Science

TENOR 2016

The Second International Conference on Technologies for Music Notation and Representation

The second International Conference on Technologies for Music Notation and Representation seeks to extend and consolidate the scope and depth of material presented at the inaugural conference held in Paris in May 2015. We have included three initial workshops providing demonstrations and practical, creative and hands-on experience of a number of intriguing notation-based technologies, two music session during which composers and technologists seek to illustrate approaches which are then performed and a single more formal concert.

Posters and papers focus on new forms of representation as well as music notation, transcription, sonic visualisation, and musical representation which are frequently associated with the fields of musical analysis, ethnology, and acoustics. The papers explore many recent forms of notation and representation in all domains of music. This year we have extended the conference call to include cognition and ontological issues of performance practice arising from the use of traditional and/or graphical notation systems in live electronics.

The workshops are taking place at the Centre for Music and Science, Faculty of Music, University of Cambridge. The scholarly conference, posters, papers, music sessions and concert are taking place at the Department of Music and Performing Arts, Anglia Ruskin University, Cambridge.

TENOR 2016 Organising Committee

Organising committee

Torsten Anders (University of Bedfordshire)
Nicholas Brown (ARU)
Ian Cross (Cambridge University)
Sarah Faber (ARU)
Dominique Fober (Grame, Lyon)
Tom Hall (ARU)
Richard Hoadley (ARU)
Krisztian Hofstadter (ARU)
Paul Jackson (ARU)
Mariana Lopez (ARU)
Thor Magnusson (University of Sussex)
Chris Nash (University of the West of England)
Roderick Watkins (ARU)

Steering committee

The Steering Committee is responsible for guiding future directions with regards to the TENOR conference. Its members currently include :

Jean Bresson, Ircam-CNRS UMR STMS
Pierre Couprie, IReMus Université Paris-Sorbonne
Dominique Fober, GRAME
Yann Geslin, INA-GRM
Richard Hoadley, Anglia Ruskin University
Mike Solomon, Ensemble 101

Scientific committee

Hasnizam Abdul Wahid, Andrea Agostini, Torsten Anders, Trevor Bača, Karim Barkati, Sandeep Bhagwati, Andrew Blackburn, Alan Blackwell, Alain Bonardi, Bruno Bossis, Jean Bresson, Nicholas Brown, Michael Clarke, Pierre Couprie, Cécile Davy-Rigaux, Frédéric Dufeu, Emile Ellberger, Simon Emmerson, Sarah Faber, Dominique Fober, Jason Freeman, Ichiro Fujinaga, Jérémie Garcia, Yann Geslin, Daniele Ghisi, Jean-Louis Giavitto, Gérald Guillot, Georg Hajdu, Tom Hall, Keith Hamel, Richard Hoadley, Cat Hope, Adrian Hull, Florent Jacquemard, Mieko Kanno, David Kim-Boyle, Leigh Landy, Mariana Lopez, Thor Magnusson, Mikhail Malt, Peter Manning, Peter McCulloch, Chris Nash, Douglas Nunn, Jean Penny, François Picard, Paul Rhys, Philippe Rigaux, Eleanor Selfridge-Field, Ryan Ross Smith, Mike Solomon, Alice Tacaille, Matthew Thibeault

Keynote

Making concrete the ineffable: from music to mathematical models

Music as conceptualised in the mind and communicated in performance is more than that which is captured in conventional notation. It may be more than that which can be captured in notation, at least in human-readable form. Just how much of music can we make concrete through notation, or represent in graphs or mathematical models? And, of the multitude of possibilities, how do we select for the most relevant and crucial things to represent? Suppose that, in addition, beyond representation we wish to reveal why—why did the performer or composer choose this over that? Why is this passage surprising?—thus veering towards questions of music cognition. How will the objective influence that which we devise to notate? We shall examine some of these issues through a series of experiments that aim to make tangible the ineffable nature of practicing and performing music. In particular, we consider the intentions that shape performances and the deeper music structures that guide them. The decisions and entities that musicians grapple with then provide impetus for the music representations.

Elaine Chew

Elaine Chew is Professor of Digital Media at Queen Mary University of London, where she is affiliated with the Centre for Digital Music and co-Leads its Cognition, Creativity & Expression research theme. A classically trained pianist and operations researcher, she uses mathematical and computational models and scientific visualisation to represent, analyse, and explain aspects of musicianship, including musical prosody and structure, cognition, and interaction. She also works with composers to create a nd premiere new compositions, and designs and performs in concert-conversations that probe the nature of music making and listening.

Keynote

Notation as hybrid technology

The imagining, design and construction of musical objects beyond the scale of vernacular form or language is always technical. On a fundamental level it is also technological; the creation of music is an iterative, distributed process of inscription through technologies. Notation – broadly considered – is the common element of these technologies. In this talk I approach questions of notation in contemporary music in the light of such a view of historical practices; metaphors range from measurement and control through language to format and symbolic context for action. I argue that the common practice notation of modern Western art music has particular properties which are essential to the strength of that tradition, that characterise it as a contribution to human culture, but which also present challenges to its evolution. Notation can be viewed as the surface trace of an unconstrained model – the graphical centre of a network of technologies affording actions that may be physical or conceptual. It remains liminal; we consider the interaction of material and virtual elements through the very material on which notation makes its marks. This network must be considered in a cultural context which itself is technologically informed: a discourse of informal concepts and operations. Reflection on our own informal discourse is crucial in formulating approaches to notation in our contemporary hybrid musical practices.

Jonathan Impett

Jonathan Impett's professional and research activities cover many aspects of contemporary musical practice, as trumpet player, composer and theorist. In the field of historical performance, he is a long-standing member of both The Orchestra of the Eighteenth Century and The Amsterdam Baroque Orchestra. He is also a member of the experimental chamber ensemble Apartment House. As a soloist he has given premieres of works by composers including Scelsi, Berio, Harvey and Finnissy. He directed the live electronic chamber ensemble Metanoia, and was awarded a Prix Ars Electronica for his development of the metatrumpet. His compositions have been broadcast throughout Europe; a new CD will be released by Attacca in 2015. As an improviser he has played with musicians as divers as Paul Dunmall and Amit Chaudhuri.

Work in the space between composition and improvisation has led to continuous research in the areas of interactive systems and interfaces. The current 'active sound space' project uses ALife populations of wave models to create interactive works combining aspects of composition and sound art. A monograph on the music of Luigi Nono will be published by Ashgate in 2016, and Jonathan is currently working on a project considering the nature of the contemporary musical object, 'The work without content'.
Having been Head of Music at the University of East Anglia, he is now Associate Professor at Middlesex University, London, and Director of Research at the Orpheus Instituut, Ghent – a major centre for artistic research.

TENOR 2016

REAL-TIME CORPUS-BASED CONCATENATIVE SYNTHESIS FOR SYMBOLIC NOTATION

Daniele Ghisi
STMS Lab
(IRCAM, CNRS, UPMC)
danieleghisi@bachproject.net

Carlos Agon
STMS Lab
(IRCAM, CNRS, UPMC)
carlos.agon@ircam.fr

ABSTRACT

We introduce a collection of modules designed to segment, analyze, display and sequence symbolic scores in real-time. This mechanism, inspired from *CataRT*'s corpus-based concatenative synthesis, is implemented as a part of the *dada* library for Max, currently under development.

1. INTRODUCTION

Corpus-based concatenative synthesis is a largely known technique, providing mechanisms for real-time sequencing of grains (extracted from a large corpus of segmented and descriptors-analyzed sounds), according to their proximity in some descriptors space. Among the existing tools dealing with such technique, the *CataRT* modules [1] are probably the most widely used: taking advantage of the features in the FTM library [2], they allow segmentation and analysis of the original sound files, as well as the exploration of the generated corpus via an interactive two-dimensional display, inside the Max environment.

CataRT is oriented to real-time interaction on audio data, essentially omitting any symbolic representation of events. Although some work has been done recently to link symbolic notation with *CataRT*, such as [3], none of these works, to the best of our knowledge, is meant to fully bring the ideas of concatenative synthesis into the symbolic domain.

In this article we describe a corpus-based concatenative system designed and implemented in order to bring into Max the ability to segment, analyze and explore symbolic scores, in a similar fashion than *CataRT* does with sounds. This system will be distributed as part of the *dada* library (currently under development [1]), which will contain a set of non-standard two-dimensional interfaces dealing with symbolic musical content. The *dada* library, in turn, is based upon the *bach* library, which provides Max with a set of tools for the graphical representation of musical notation, and for the manipulation of musical scores through a variety of approaches ranging from GUI interaction to constraint programming, and sequencing. The *bach* library is oriented to real-time interaction, and is meant to interoperate easily with other processes or devices controlled by Max, such as DSP tools, MIDI instruments, or generic hardware systems [4, 5]. A number of high-level modules based on *bach*, solving typical algorithmic and computer-aided composition problems, have also been collected to form the *cage* library [7].

The system we describe in this article naturally extends the concept of score granulation (introduced in [6] and then later implemented in the *cage.granulate* module [7]), allowing a finer control on the concatenation of grains, according to some relationships between the grain features extracted during the analysis process. Moreover, the feature extraction is heavily based on the *bach* lambda loop visual programming pattern [5], hence making analysis fully customizable.

2. OVERVIEW AND MODULES

The system relies on three different modules: *dada.segment*, performing segmentation and feature extraction, *dada.base*, implementing the actual database engine, and *dada.catart*, a two-dimensional graphic interface capable of organizing and interacting with the extracted grains.

2.1 Segmentation

The *dada.segment* module performs the segmentation of the original scores, contained either in a *bach.roll* (as unmeasured musical data) or *bach.score* (as classically notated musical data), in one of the following manners:

- Via markers: each marker in the original *bach.roll* is considered as a cutting point at which the score is sliced. All the slices (grains) are then collected.

- Via equations: a single value (in milliseconds for *bach.roll*, or as a fraction of the bar or beat duration, for *bach.score*) or more generally an equation can be used to establish the size of each grain. In *bach.roll* this equation can take as variable the grain onset, and is especially useful when segmentation needs to

[1] A 0.0.1 alpha version of *dada* is publicly available at the address http://data.bachproject.net/download.php?file=dada_0.0.1.zip), requiring Max 6.1.7 or higher (http://cycling74.com), bach 0.7.8.5 or higher (http://data.bachproject.net/download.php?file=bach_0.7.8.5.zip), and cage 0.3.5 or higher (http://data.bachproject.net/download.php?file=cage_0.3.5.zip). This release is actually a very crude Macintosh-only release, including the very first modules of *dada*. Among such modules are all the tools used within the scope of this paper. A modified version of the two examples proposed in section 3 can be accessed via *dada.catart*'s help file.

be performed roughly independently from the musical content itself. In *bach.score*, voices are pre-segmented into chunks of measures (according to a pattern established via the ‘presegment’ attribute), and each chunk is in turn segmented into grains whose duration is determined by the aforementioned equation - possibly having as variables the measure number, the measure division (beat), and the measure overall symbolic duration (see for instance fig. 1).

- Via label families: differently from sound files, scores easily allow non-vertical segmentations, where only a portion of the musical content happening in a given time span is accounted for (see fig. 2). If labels are assigned to notes or chords in the original score, a grain is created for each label, containing all the elements carrying such label.

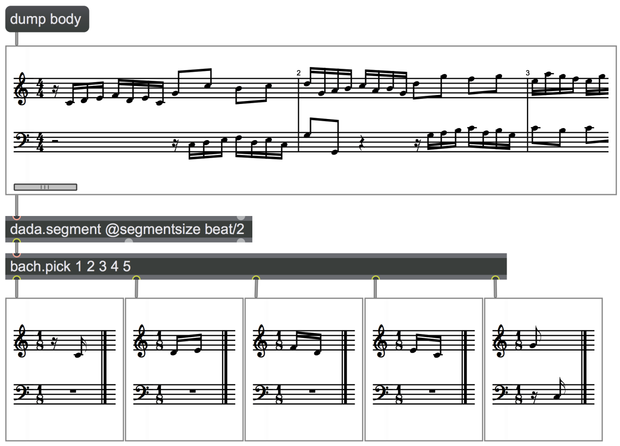


Figure 1. Segmentation of a *bach.score* into grains having length equal to half of the beat (i.e. an eighth note). The first five grains are displayed in the bottom part of the patch.

2.2 Analysis

Grain analysis is performed during the segmentation process. On one side, *dada.segment* is capable of adding some straightforward metadata to the segmented grains, such as their duration, onset, index, label (if segmentation is carried out via label families) and notation object type (either ‘roll’ for *bach.roll* or ’score’ for *bach.score*); in case the grain comes from a *bach.score*, tempo, beat phase (the beat on which the grain starts), symbolic duration and bar number can also be added.

On the other hand, *dada.segment* allows the definition of custom features via a lambda loop mechanism (see [3, 5]): grains to be analyzed are output one by one from the rightmost (lambda) outlet, preceded by the custom feature name; the user should provide a patching algorithm to extract the requested feature, and then plug the result back into *dada.segment*’s rightmost (lambda) inlet. Feature names, defined in an attribute, are hence empty skeletons which will be ‘filled’ by the analysis implementation, via patching. This programming pattern is widely used throughout the *bach* library (one can easily compare the described mechanism, for instance, with *bach.constraints*’s way of implementing custom constraints [5]), and allows users to implement virtually any type of analysis on the incoming data. Nevertheless, some ready-to-use abstractions are provided (see fig. 3) for standard features such as centroid, spread, loudness or item counting.

Analyzed features are collected for each grain, and output as metadata from the middle outlet of *dada.segment*.

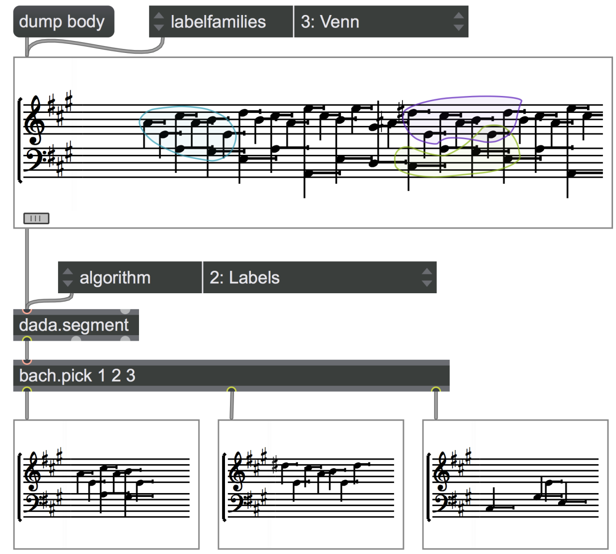


Figure 2. Segmentation of a *bach.roll* according to label families. Labeled items are automatically enclosed in colored contours in the original *bach.roll*. Notice how families can overlap (in the example above, one note is labeled twice, and hence assigned to two families at the same time). The first three grains (corresponding to the first three label families) are displayed in the bottom part of the patch.

2.3 Database

Once the score grains have been produced and analyzed, they are stored in a SQLite database, whose engine is implemented by the *dada.base* object. Hence, data coming from *dada.segment* are properly formatted and fed to *dada.base*, on which standard SQLite queries can be performed (see figure 3). Databases can be saved to disk and loaded from disk.

2.4 Interface

Finally, the *dada.catart* object provides a two-dimensional graphic interface for the database content. Its name is an explicit acknowledgment to the piece of software which inspired it. Grains are by default represented by small circles in a two dimensional plane. Two feature can be assigned to the horizontal and vertical axis respectively; two more features can be mapped on the color and size of the circles. Finally, one additional integer valued feature can be mapped on the grain shape (circle, triangle, square, pentagon, and so forth), adding up to a total number of five

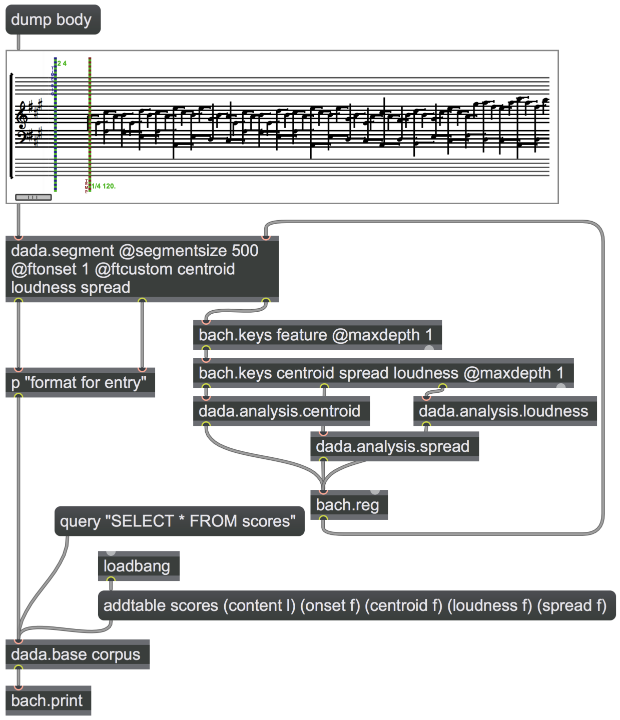


Figure 3. When the patch opens, a table named 'scores' is created in the database named 'corpus', collecting all the grains. This table has five columns: the content of the grain (a *bach* Lisp-like linked list), the onset the grain had in the original score, its centroid, loudness and spread (all floating point numbers). When the 'dump body' message is clicked, the score contained in the *bach.roll* is segmented and analyzed by centroid, loudness and spread (respectively computed via the *dada.analysis.centroid*, *dada.analysis.spread* and *dada.analysis.loudness* modules inside the lambda loop). The database is then filled, and standard SQLite queries can be performed on it.

features being displayed at once (see fig. 4). The database elements can be sieved by setting a *where* attribute, implementing a standard SQLite 'WHERE' clause. The vast majority of the display features can be customized, such as colors, text fonts, zoom and so on.

Each grain is associated with a 'content' field, which is output either on mouse hovering or on mouse clicking. The content is usually assigned to the *bach* Lisp-like linked list representing the score [5]. The sequencing can also be beat-synchronous, provided that a tempo and a beat phase fields are assigned: in this case the content of each grain is not output as soon as the grain is clicked upon (or mouse hovered), and its sequencing is postponed in order for it to align with the following beat, according to the current tempo (obtained from the previously played grains).

In combination with standard patching techniques, these features also allow the real-time display, sequencing and recording of grains (see section 3 for an example).

A *knn* message allows to retrieve the k-th nearest samples for any given (x, y) position. A system of messages inspired by turtle-graphics is also implemented, in order to be able to move programmatically across the grains;

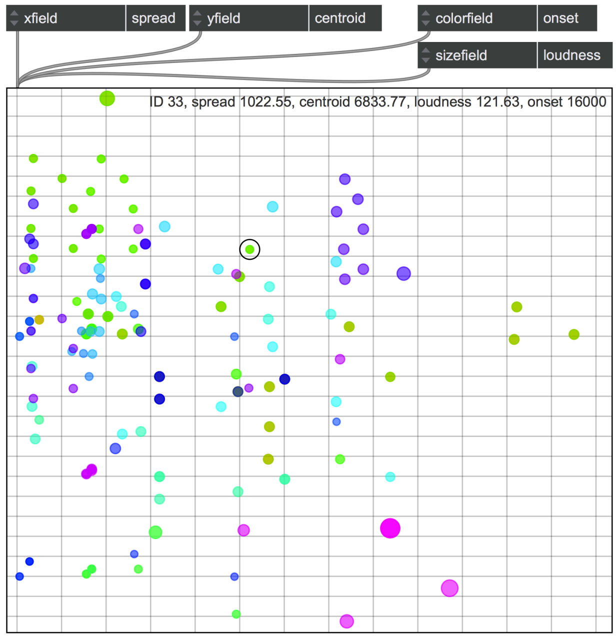


Figure 4. The *dada.catart* object displaying the database built in figure 3. Each element of the database (grain) is represented by a circle. On the horizontal axis grains are sorted according to the spread, while on the vertical axis grains are organized according to their centroid. The colors scale is mapped on the grain onsets, while the circle size represents the grain loudness.

namely a 'turtle' grain can be assigned via the *setturtle* message (setting the turtle on the nearest grain with respect to a given (x, y) position), and then the turtle can range across the grains via the *turtle* message, moving it of some $(\Delta x, \Delta y)$ and then choosing the nearest grain with respect to the new position (disregarding the original grain itself). The turtle is always identified in a *dada.catart* by an hexagon (see fig. 7 for an example).

3. EXAMPLES

3.1 An interactive tonal centroid palette

As a first example, in the patch displayed in figure 6 we segment (in grains of 1 second each) and then analyze the first eight Lieder from Schubert's *Winterreise*. During the analysis process we take advantage of the tonal centroid transform proposed by Harte, Sandler and Gasser in [8], and implemented in the *cage* library (see [7]). The horizontal axis displays the phase of the tonal centroid with respect to the plane of fifths, while the vertical axis displays the phase referred to the plane of minor thirds (both range from -180 to 180 degrees). The analysis subpatch computing the phase of the projection of the tonal centroid on the plane of fifths is displayed in fig. 5 (the one for minor thirds is analogous). Both colors and shapes are mapped on the Lieder number.

We can use this representation as a sort of 'interactive tonal centroid palette': each vertical line refers to a note in the circle of fifths, each horizontal line refers to an augmented chord in the circle of minor thirds. If we focus especially on the horizontal axis, we notice for instance that red circles (belonging to the first Lied, *Gute Nacht*, in D minor) are mostly scattered around the vertical line referring to the D, or that orange triangles (belonging to the second Lied, *Die Wetterfahne*, in A minor) are mostly scattered in the region around A.

A record mechanism is implemented, and the recorded data is collected in the score displayed at the bottom of the image. The score can then be saved, quantized or exported, taking advantage of the features of the *bach* library [5].

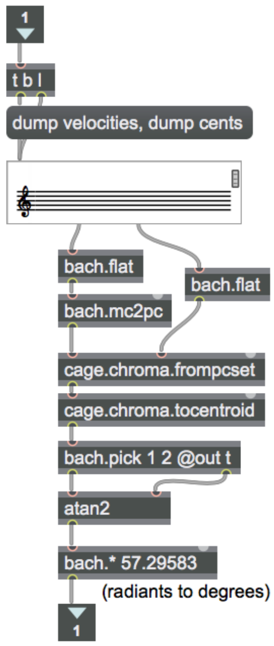


Figure 5. The subpatch computing the phase of the projection of the tonal centroid of a *bach.roll* grain on the plain of fifths. All the pitches, converted to pitch classes, are weighted with their own velocities and gathered in a chroma vector, whose tonal centroid is computed via the algorithm proposed in [8]. The first two components of the tonal centroid (referring to the plane of fifths) are picked, and the angle formed by the vector is computed.

3.2 Rearranging beats

As a second example, consider figure 7, where we have segmented the first Bach invention (BWV 772) by beat. On horizontal axis we display the measure number, on vertical axis we display the position of the beginning of the grain inside the measure (phase). We can then send *turtle* messages in order to navigate through the grains, so that we can read the complete score as it was (patch mechanism at top left corner of the image), or only read the last beats of each measure from last to first measure (top-middle part of the image), or even move in random walks among the beats (top-right part of the image).

4. CONCLUSION AND FUTURE WORK

We have presented a system operating on symbolic musical content, directly inspired by the *CataRT* modules, and implemented as part of the *dada* library for Max, currently under development. This system naturally extends the symbolic granulation engine implemented in *cage.granulate*, allowing to organize score grains according to custom analysis features.

This system can be improved in a certain number of ways. For one thing, the number of predefined analysis modules should be increased, by bridging into the symbolic domain important audio descriptors such as roughness, inharmonicity, and so on. The relationships between audio and symbolic descriptors could be in itself a topic for further investigations. Moreover, the *dada.segment* module is currently able to segment based on given markers, equations or labels; however it is not able, by design, to infer such markers or labels. One of the interesting topics of future research might hence be to integrate inside the process a system for semi-automatic segmentation of scores, and a module for pattern retrieval. Also, the label-based extraction currently works only for *bach.roll*, and a *bach.score* version of such an algorithm should be also implemented.

5. REFERENCES

[1] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with catart," in *IN PROC. OF THE INT. CONF. ON DIGITAL AUDIO EFFECTS (DAFX-06*, 2006, pp. 279–282.

[2] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, "FTM - Complex Data Structures for Max," in *Proceedings of the International Computer Music Conference*, 2005.

[3] A. Einbond, C. Trapani, A. Agostini, D. Ghisi, and D. Schwarz, "Fine-tuned control of concatenative synthesis with catart using the bach library for max," in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[4] A. Agostini and D. Ghisi, "Real-time computer-aided composition with *bach*," *Contemporary Music Review*, no. 32 (1), pp. 41–48, 2013.

[5] ——, "A max library for musical notation and computer-aided composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015/10/03 2015. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00296

[6] ——, "*bach*: an environment for computer-aided composition in Max," in *Proceedings of the International Computer Music Conference (ICMC 2012)*, Ljubljana, Slovenia, 2012, pp. 373–378.

[7] A. Agostini, E. Daubresse, and D. Ghisi, "*cage*: a High-Level Library for Real-Time Computer-Aided Composition," in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[8] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *In Proceedings of Audio and Music Computing for Multimedia Workshop*, 2006.

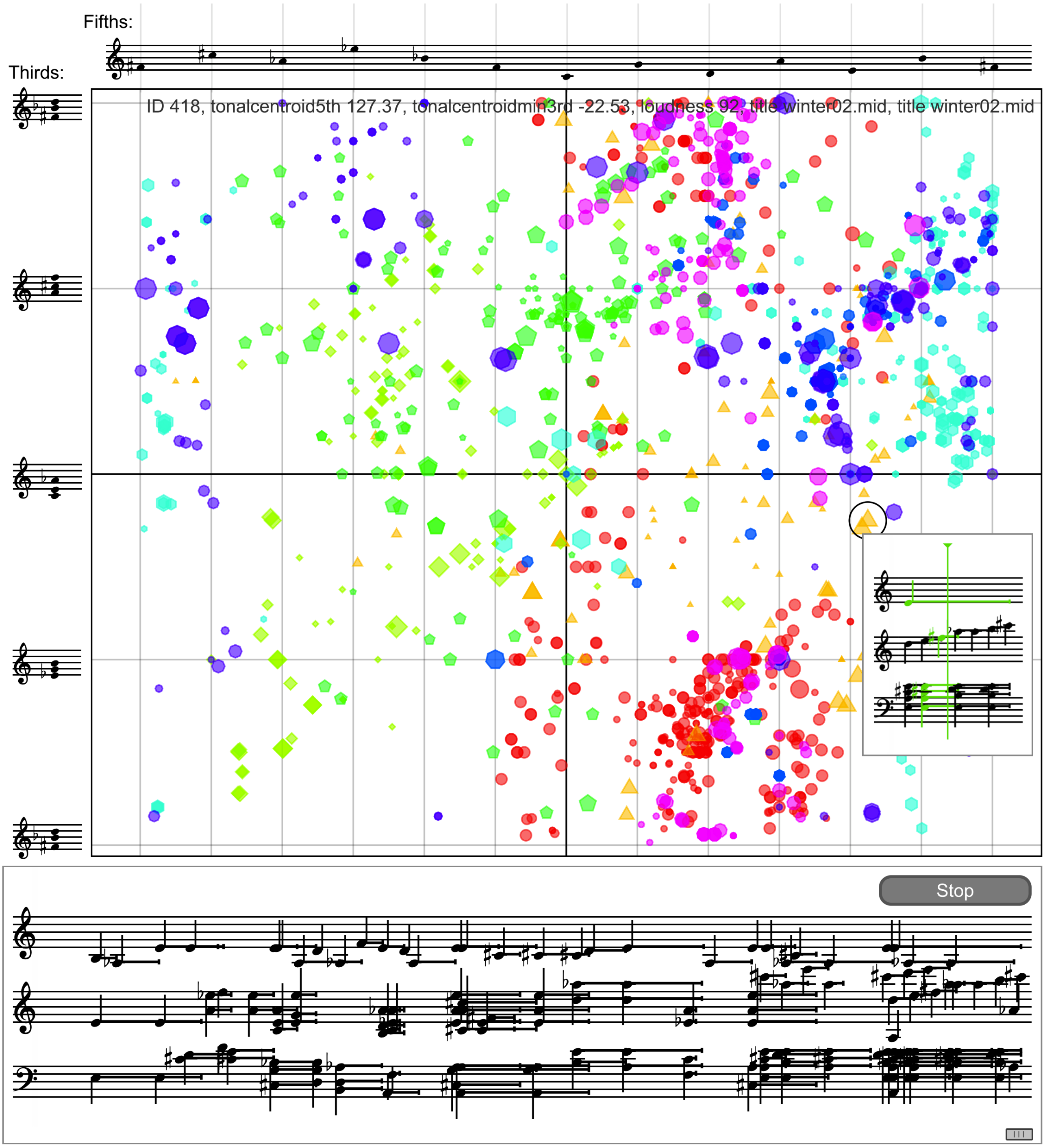


Figure 6. A patch displaying the database build from the first eight Lieder of Schubert's *Winterreise*, organized by tonal centroids (the phase of the projection on the plane of fifths is on horizontal axis, the phase of the projection on the plane of minor thirds is on the vertical axis). Both colors and shapes identify the Lieder number (1 being the circle, 2 being the triangle, 3 being the square, and so on). When the recording mechanism is turned on, grains can be played via mouse hovering, and the bottommost *bach.roll* contains the recorded result.

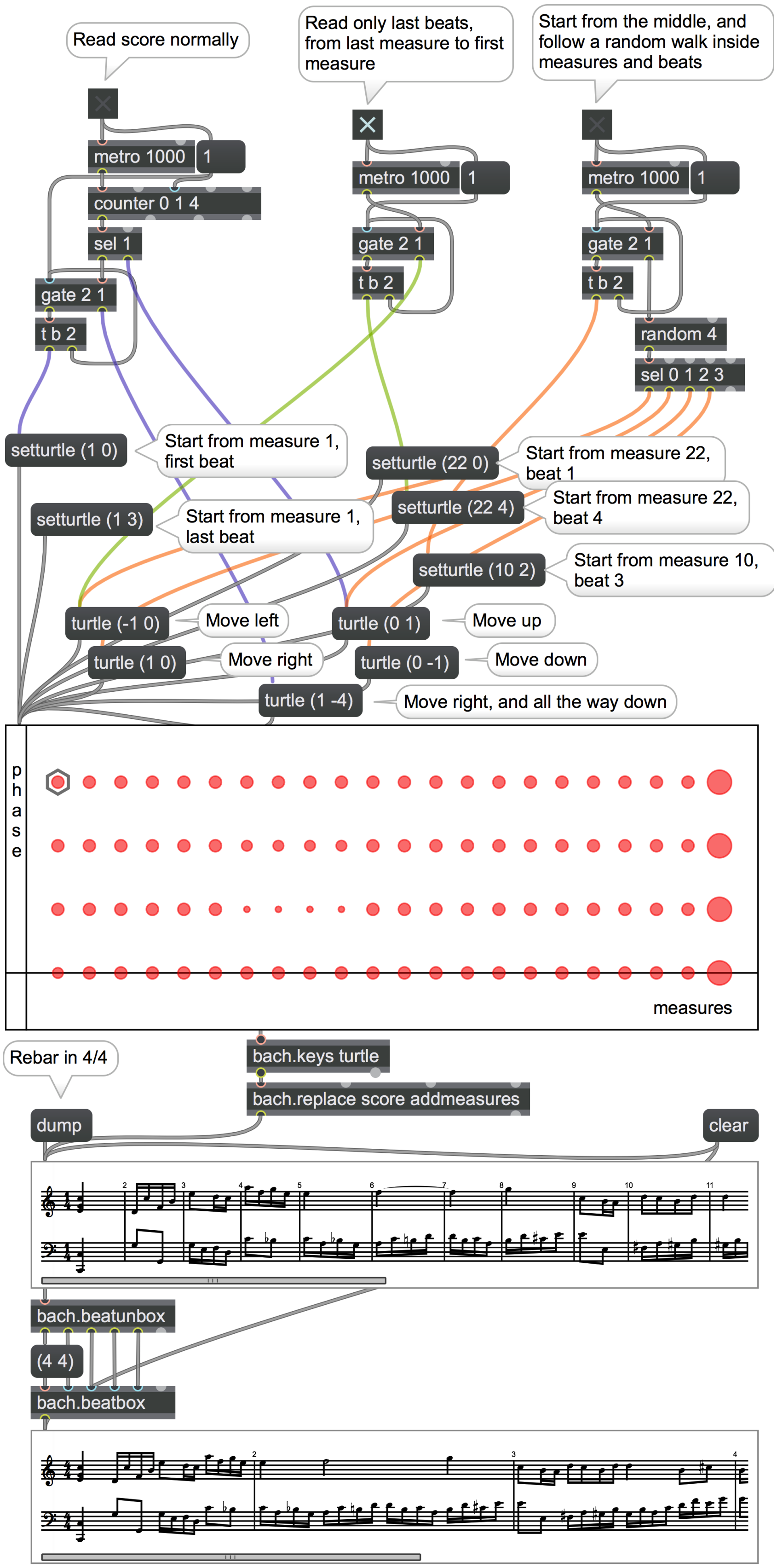


Figure 7. An example showing the manipulation of the first Bach invention (BWV 772), segmented by beat, and rearranged so to play and record the last beats of each measure (starting from last measure, and ending with first one). Notice how ties are preserved during the segmentation process (e.g. between measure 6 and 7) of the upper *bach.score*, rebarred in measure 2 of the lower one.

TENSION RIBBONS: QUANTIFYING AND VISUALISING TONAL TENSION

Dorien Herremans
Centre for Digital Music
School of Electronic Engineering
and Computer Science
Queen Mary University of London
d.herremans@qmul.ac.uk

Elaine Chew
Centre for Digital Music
School of Electronic Engineering
and Computer Science
Queen Mary University of London
elaine.chew@qmul.ac.uk

ABSTRACT

Tension is a complex multidimensional concept that is not easily quantified. This research proposes three methods for quantifying aspects of tonal tension based on the spiral array, a model for tonality. The cloud diameter measures the dispersion of clusters of notes in tonal space; the cloud momentum measures the movement of pitch sets in the spiral array; finally, tensile strain measures the distance between the local and global tonal context. The three methods are implemented in a system that displays the results as tension ribbons over the music score to allow for ease of interpretation. All three methods are extensively tested on data ranging from small snippets to phrases with the Tristan chord and larger sections from Beethoven and Schubert piano sonatas. They are further compared to results from an existing empirical experiment.

1. INTRODUCTION

Musical tension forms an essential part of the experience of listening to music. According to [1], increasing tension can be qualitatively described as "a feeling of rising intensity or impending climax, while decreasing tension can be described as a feeling of relaxation or resolution". However, defining tension in a more quantitative, formalized way is a difficult problem. In previous studies, different characteristics have been used to try to model musical tension. These aspects are usually rooted in either the domain of psychology or that of music. From the psychological point of view, models look at influential factors such as expectation and emotion [2, 3, 4]; and semantic meaning of lyrics [5]. From a more low-level musical point of view, examined features include rhythm and timing [6, 7, 1]; harmonic tonal perception through Lerdahl's tonal tension model [8, 9, 10]); pitch height/melodic contour [11, 12]; dynamics [13, 12]; timbral elements (roughness, brightness, and density) [14, 6]; and pitch register [12, 7]. It must be noted that most of the above mentioned low-level musical features can also be linked to expectation.

No one particular feature, however, seems to be decisive in predicting the experience of tension [14]. Listening to music is an aggregate experience that requires the integration of many different features. A listener's attention can focus on one feature at a particular time and then shift to a different feature or combination of features [1].

In this research we explore tonality as one of the dimensions of musical tension. Three methods for quantifying aspects of tension are developed based on the spiral array [15], a geometric model for tonality. The system developed outputs the results of the methods as ribbons over a musical score. In the next section, these different methods are discussed, followed by an analysis of selected musical fragments, which include snippets previously analysed in an empirical study by [1].

2. THEORY

In this paper, three methods that capture aspects of perceived tonal tension are developed and discussed based on the spiral array, a model for tonality. We first give a brief review of the spiral array, then introduce the three methods: cloud diameter, cloud momentum, and tensile strain.

2.1 Spiral array

The spiral array is a three dimensional representation of pitch classes, chords and keys. Each pitch class is represented as spatial coordinates along a helix [16]. The three dimensional representations allows higher level musical entities such as chords and keys to be embedded in the helix. The exact formula of the pitch class helix implemented in this paper is as follows:

$$x = r \times \sin(t), y = r \times \cos(t), z = a \times t, \qquad (1)$$

where $r = 1$, $a = \sqrt{\frac{2}{15}} \times \frac{\pi}{2}$, $t \in [-\infty, \infty]$ and $t \in \mathbb{R}$.

Close tonal relationships (such as the perfect fifth) are mirrored by their spatial proximity in the spiral array. Figure 1 shows that notes which sound tonally close are in fact positioned close to each other inside the array. This is illustrated by the C major chord, which only consists of spatially close pitches. Notes are positioned one perfect fifth away from each other (a quarter turn in the spiral), which results in notes positioned "above" each other representing a major third [15].

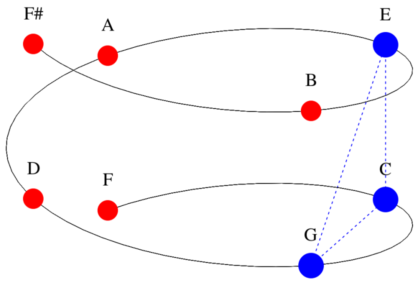


Figure 1: The Spiral Array with a C major chord.

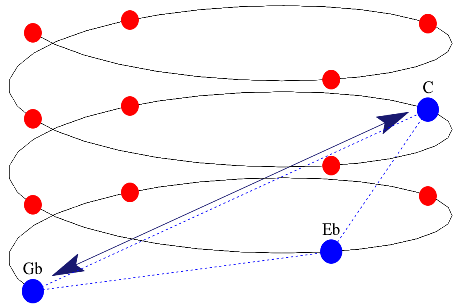


Figure 3: The spiral array with a C diminished triad.

Pitches can be spelled in multiple ways, for instance G♯ and A♭. The spiral array takes pitch spelling into account by assigning a different geometric position to enharmonically equivalent (but differently-spelled) pitches. Because the pitch class representations are a helical arrangement of the line of fifths, pitches with sharps are located above the D and those with flats are located below D.

2.2 Cloud diameter

Tension in musical pieces is a property that varies over time. Therefore, a sliding window approach was used, whereby a musical piece is divided into equal length windows. Within each of those windows, all of the notes can be represented as a *cloud* in the spiral array.

The idea of cloud diameter is to capture the *largest distance between any two notes in a cloud*. When a chord or a cloud of notes contains intervals that are tonally far apart (i.e. dissonant), the distance between these pitches in the spiral array will be large. A first method tries to capture this type of harmonic tension by looking at the largest Euclidean distance within the cloud, or the cloud diameter. To illustrate this, Figure 2 shows the cloud diameter of the C major triad and its diminished counterpart. The larger diameter of the diminished triad can be explained by the large tonal distance between C and G♭, which is a diminished fifth. This is illustrated in the spiral array in Figure 3.

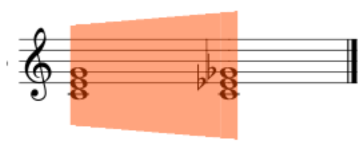


Figure 2: Cloud diameter of the C major triad and the C diminished triad (min: 2.3, max 3.0).

2.3 Cloud momentum

Musical information can be condensed in the spiral array by one set of three dimensional coordinates which represents the centre of effect (ce) of a cloud. This ce has been previously used for key detection [17], whereby the key with the ce closest to the ce of the fragment is selected. It has equally proved to be useful for chord detection [15] and finding key boundaries [16].

For a cloud consisting of i notes, each note has a pitch position p_i in the spiral array and a duration d_i. The centre of effect, ce, of the cloud can be calculated as:

$$ce = \sum_{i=1}^{N} \frac{d_i}{D} \times p_i \text{ whereby } D = \sum_{i=1}^{N} d_i. \tag{2}$$

The idea of cloud momentum is to capture how large the *distance between the centres of effect of two clouds of points* is, thus capturing the movement in tonality. The ce's of tonally similar chords or groups of notes are positioned close to each other in the spiral. When there is a change in tonality, this will cause the centres of effect to shift to a new area from one cloud to the next, thus resulting in a larger cloud momentum. The cloud momentum measures this type of tonal tension by calculating the Euclidean distance between the centres of effect of each window or cloud of notes. In the example in Figure 4, a large movement in tonality between the C chord and the C♯ chord can be seen. This is followed by no movement to the inverted C♯ chord.

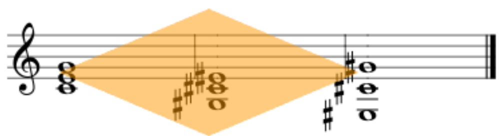


Figure 4: Cloud momentum of C major chord moving towards C♯ major (min 0, max 2.8).

Cloud momentum is a characteristic of movement. It can therefore be seen that its value for the first note of a fragment is non-existent (represented as zero). In the case that the window size is smaller than the duration of a note, it might occur that the cloud momentum drops during the span of that note, as it represents the "movement" in the spiral array and there is no movement within a note or cluster of notes.

2.4 Tensile strain

The previous methods capture the span of the cloud and the distance between adjacent centres of effect. The tensile strain captures the *tonal distance between the ce of a cloud*

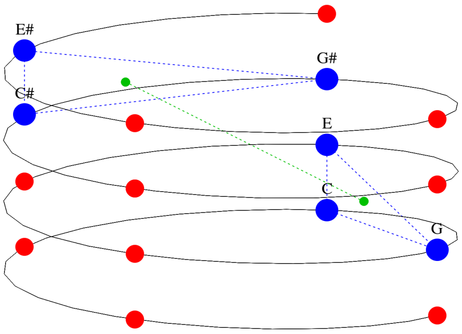


Figure 5: C major and C♯ major chord together with the *ce* (in green).

of notes and the key. By implementing the key detection algorithm developed in [15], the Euclidean distance between *ce* of the cloud of notes and the *ce* of the global key (as described in [15]) is calculated. This distance represents the tensile strength. The short example in Figure 6 has a (given) global key of C major. The tensile strain is largest on the C♯ major chord, which is to be expected since it is tonally more distant than both C major or A minor from the given key. Figure 7 illustrates how the tensile strain was calculated, by marking the distance from the *ce* of all three chords (in green) to the *ce* of the key of C major (in orange) in the spiral array.

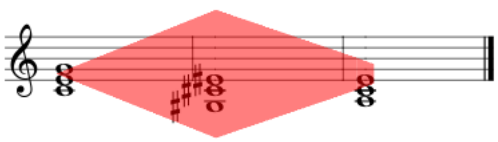


Figure 6: Tensile strain of C major – C♯ major – A minor chord given that the key is C major (min 1.0, max 1.5).

The distance to the *ce* of the key in the spiral array is illustrated in Figure 6.

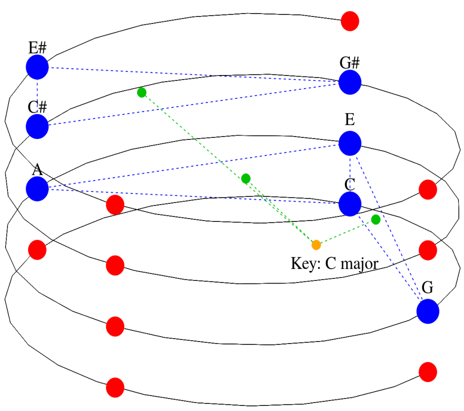


Figure 7: C major – C♯ major – A minor chord together with their distance from the *ce* of the key (in orange).

In the next section, all three methods are studied in greater detail by means of examples.

3. EXAMPLES

In this section a number of example pieces are discussed. Short snippets from an empirical study by Farbood [1] are first analysed, followed by a few famous phrases and more extensive sections of music.

The methods above were implemented in java are available online [1]. The system developed takes a musicXML file as input and outputs an INScore file [18] which represents the results of each of the methods as a coloured ribbon overlaid on the score. To represent these ribbons, the output of each method was normalised within a piece, or, in the case of the examples based on Farbood (Section 3.1.1), over all samples. In order to emphasise the changes in tension visibly, zero values (which occur when there are rests) are not taken into account when normalising.

3.1 Snippets

This section considers the application of the tension measures—cloud diameter, cloud momentum, and tensile strain—to the harmonically-motivated tension examples from Farbood, and compares the results to known user annotations of perceived tension for these samples.

3.1.1 Tension examples from Farbood

The above methods each capture some aspect of harmonic tension. In order to validate this claim, the methods were applied to examples from an empirical study. While each of the proposed methods captures part of the harmonic tension, one could argue that it is not possible to capture tension as an aggregate feature, as the attention of the listener constantly shifts between different features when evaluating tension [1].

Mary Farbood performed an extensive online questionnaire, whereby a total of 2,661 participants (17% of which self-categorised as musicians) annotated the perceived tension after listening to a snippet of music [1]. The participants were asked to select one out of six possible shapes for the perceived tension (see Figure 8).

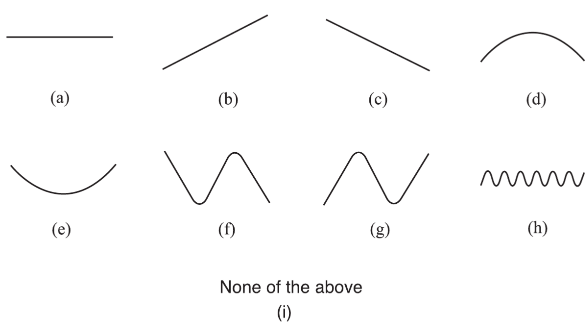
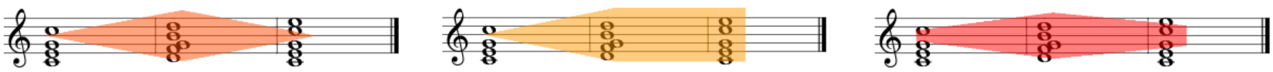


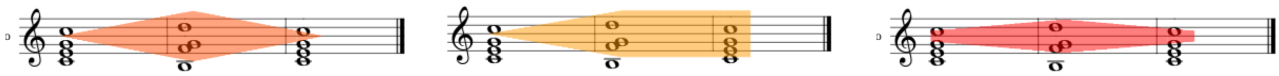
Figure 8: Possible responses in Farbood's study. Figure adapted from [1].

[1] dorienherremans.com/tension

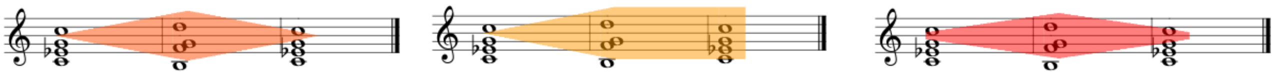


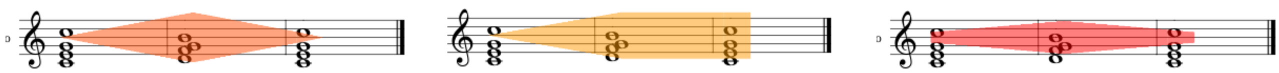
(a) A03

(b) A04

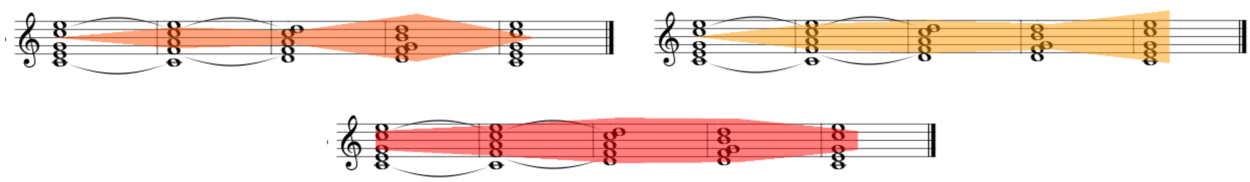


(c) A05

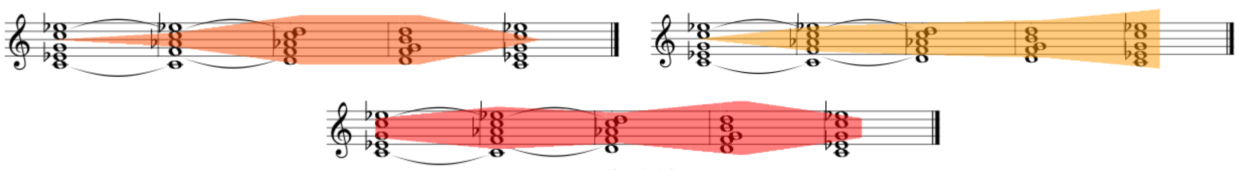


(d) A06

(e) A07

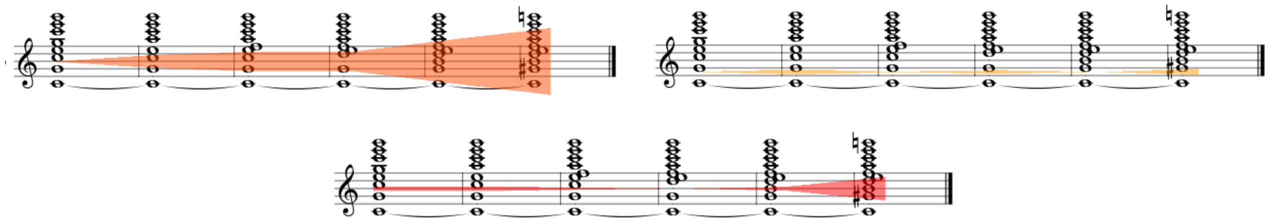
(f) A08



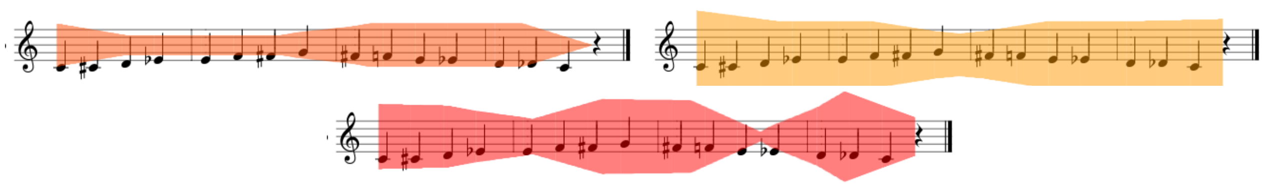
(g) A11



(h) A12



(i) A13



(j) A14

Figure 9: Computed tension parameters for selected stimuli from [1]. In sequence: cloud diameter (orange), cloud momentum (yellow), and tensile strain (red).

In this study, relevant stimuli pertaining to harmonic tension were selected and the results of the three methods were represented as tension ribbons overlaid on the score snippets in Figure 9. The colour-coded ribbons represent (in sequence): the cloud diameter (orange), the cloud momentum (yellow), and the tensile strain (red). The algorithm was run with one window per measure, except for stimulus A14, whereby a window size of two quarter notes was selected. In order to make the changes in the results of the methods clearly visible, the data for each type of ribbon was normalised over the results of all the stimuli.

Stimulus ID	cloud diameter	cloud momentum	tensile strain
A03	(d)	(a)	(d)
A04	(d)	(a)	(d)
A05	(d)	(a)	(d)
A06	(d)	(a)	(d)
A07	(d)	(a)	(d)
A08	(d)	(a)	(d)
A11	(a)(d)	(b)	(d)
A12	(d)	(b)	(a)(d)
A13	(a)(d)	(a)	(a)(b)
A14	(f)	(e)	(f)(g)

Table 1: Correspondence between computed tension patterns and templates in Figure 8.

ID	%	1st	%	2nd	%	3rd
A03	54%	(b)	15%	(a)	11%	(d)
A04	41%	(b)	16%	(a)	13%	(d)
A05	50%	(d)	13%	(g)	11%	(f)
A06	46%	(d)	19%	(g)	14%	(a)
A07	26%	(g)	18%	(e)	18%	(f)
A08	25%	(e)	20%	(a)	16%	(d)
A11	24%	(a)	21%	(g)	17%	(d)
A12	27%	(g)	25%	(a)	18%	(d)
A13	29%	(a)	15%	(d)	15%	(i)
A14	78%	(d)	7%	(g)	5%	(a)

Table 2: Top three responses for perceived tension for each stimulus (shown in Figure 8) as found by [1].

The results in Figure 9 are mapped to the patterns in Figure 8, as presented in Table 1, for direct comparison with the top three responses of Farbood's empirical study, shown in Table 2. In the analysis below, a few results from less popular responses (not in the top three) are used. For a full overview of the empirical results, the reader is refered to [1]. (When examining the cloud momentum, we have to take into account the fact that the first window will not have a momentum, and hence will always be zero. The first value can therefore be ignored when analysing the results.) The comparison confirms that the above defined methods for tonal tension do capture different nuances of tension.

For the first two stimuli (A03 & A04), the cloud diameter and tensile strain have the same movement as the third most popular pattern, (d), identified by 11% and 13%, respectively, of the respondents. Another 15% and 16%, respectively, chose the same response as cloud momentum, (a). It is worth noting here that some of the features that may play a role in listeners' evaluation of perceived tension may not be captured in the spiral array. Therefore it is extremely difficult to simplify the cause of tension perception to just one feature. For example, since the last chord is in essence the same as the first chord, with a doubling of the third at the octave, they are represented by the same cloud of points in the spiral array. An increasing line of tension in the first two fragments might have been captured by applying a method that takes into account the highest pitch or melodic contour [19], both features which are not covered in this paper. Since the spiral array uses pitch classes, our model will not capture tension which arises from chord inversions.

The three methods defined in this paper capture 58% (50% (d) + 8% (a)) and 60% (46% (d) + 14% (a)), respectively, of the responses for stimuli A05 and A06 [1]. For the next two stimuli (A07 & A08), 30% (17% (d) + 13% (a)) and 36% (20% (a) + 16% (d)), respectively, is captured [1]. This decrease can be due to the fact that many participants selected response (e) and (f) from Figure 8. Since a window size of one whole note was used for this experiment, a maximum of three points were calculated for the stimuli. Hence it is not possible to get an output like (e) and (f), both of which require at least four points. More experiments with the window size might be useful in the future to test this influence, however, for now a window size per onset produces reasonable results.

The fact that tension is a characteristic which is perceived through different, alternative features is yet again confirmed in stimuli A11 and A12, where the top response (being (a)) is only selected by 24% and 27%, respectively. In the case of A11, 17% of the participants selected (d) and chose 14% (e) [1]. These two answers could even be seen as a reversed or opposite movement of tension. The calculated tensile strain varies over the fragments, as indicated by 21% and 27%, respectively, of the participants. The cloud diameter captures the perceived increase followed by a decrease very well. This tension profile corresponds to 17% and 18%, respectively, of the responses. These results yet again confirm the multidimensional aspect of perceived tension.

The cloud momentum is (perhaps) surprisingly small in stimuli A13. This can be attributed to the fact that the ce is calculated from many notes at the same time, thus diminishing the effect of changing only one note from one cloud to the next. When many notes are sounding at any given time in a slow changing sequence, cloud diameter and tensile strain might be more sensitive to tonal tension than cloud momentum.

For stimulus A14, a window size of two notes was selected. Due to the chromatic nature of this stimulus, the distance to the key is, as to be expected, very large. Most participants indicate an increase in tension followed by a decrease. This is also apparent when looking at the cloud momentum and cloud diameter. Although these ribbons also register tension in the beginning and ending.

From this analysis, it is apparent that the three measures—

cloud diameter, momentum, and tensile strain—can help us understand different aspects of perceived tension in music.

3.2 Phrases

Next, we consider slightly longer examples in the form of phrases excerpted from Wagner's Tristan and Isolde and Beethoven's Sonata Op. 31 No. 3.

3.2.1 Tristan chord

A famous tension-inducing chord, which typically has an unusual relationship to the implied key of its surroundings, is the Tristan chord. Its unusual composition has been the topic of many musicological works [20]. [21] described it as "that Sphinx-chord, which has already occupied so many minds". The Tristan chord consists of an augmented fourth, augmented sixth, and augmented ninth above a bass note, for example, {F, B, D♯, G♯}. The chord was given its name because the leitmotif associated with Tristan in Richard Wagner's opera "Tristan und Isolde" contains this chord [22]. In the opera, Tristan and Isolde fall in immortal love after drinking a magic potion when they try to commit suicide together. Wagner uses the Tristan chord every time the potion or its effects are mentioned, thus connecting it to the build-up of suspense in the story. When the Tristan chord is represented in the spiral array (see Figure 10) it becomes clear that it is a very dispersed chord in tonal space.

An example of the Tristan chord is displayed in the first beat of bar 3 in the excerpt shown in Figure 11. This figure also displays the results when applying the three methods (with 6 windows per bar) to the fragment. A large increase in cloud momentum and cloud diameter are seen when the chord appears, indicating its tonally disperse nature and large tonal distance to the previous chord/notes. The tensile strain is more difficult to evaluate, as the key of this short phrase is not entirely clear. The example shows the tensile strain with A minor.

3.2.2 Beethoven Sonata Op. 31 No. 3 in E♭ major

The Tristan chord has appeared in other compositions before Wagner wrote the opera. In Beethoven's Sonata No. 18, Op. 31 No. 3, in E♭ major (see Figure 12) the Tristan

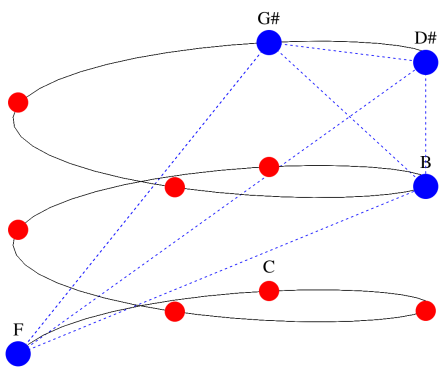


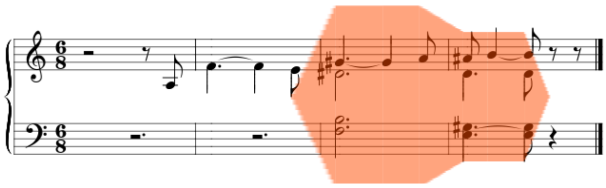
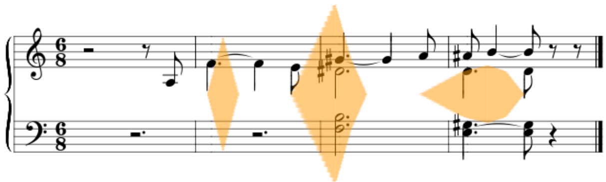
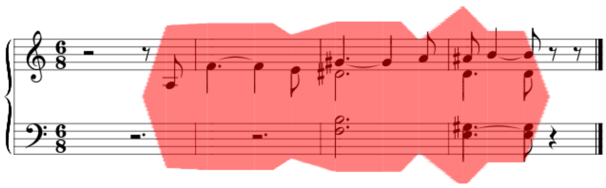
Figure 10: The Tristan chord in the spiral array.

(a) Cloud diameter

(b) Cloud momentum

(c) Tensile strain

Figure 11: The Tristan chord in Wagner's Tristan Prelude.

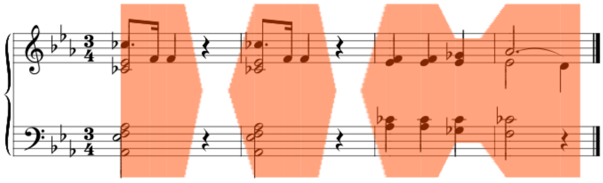
chord appears with the exact intervals albeit with a different spelling in the fourth bar. When enharmonically rewriting the Tristan chord in this way, it becomes less disperse in tonal space, as can be seen in Figure 13. This is reflected by the spiral array, which takes pitch spelling into account. For example, The G♯ in the Tristan chord is much farther away from F than the A♭ used by Beethoven (see Figure 10 and 12). We assume that the composers choices reflect the pitch relations they intend for the listener to perceive. The tension values calculated reflect these choices.

Figure 12 shows the cloud diameter and momentum, and tensile strain behaviour for the Beethoven Op. 31 No. 3 example. Recall that there was a global peak in the tensile strain and a local peak in the cloud momentum values for Wagner's Tristan phrase. The peak tension as evaluated by the tensile strain has now shifted to the chord preceding the Tristan chord. For cloud diameter, although the Tristan chord produces a high value, the value remains high for the following chord. Thus this different spelling may lead to different interpretations of the tension, even though the notes are enharmonically equivalent.

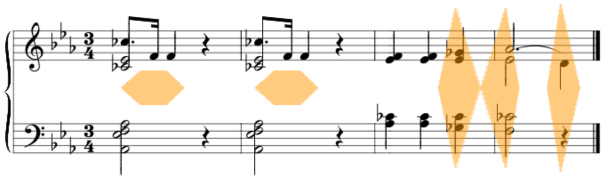
3.3 Sections

Here, we turn to tension in larger sections of music: namely, Adagio (in A-flat major), the second movement of Schubert's Piano Sonata in C minor D958 (beginning and end), and the first sixteen bars of Beethoven's Les Adieux (Sonata No. 26, opus 81a, in E♭ major).

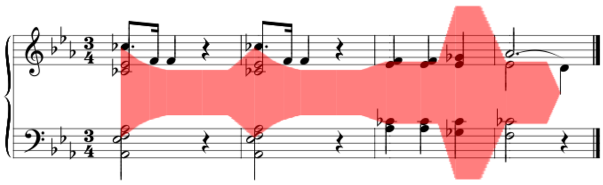
To obtain the results, the methods used eight windows per bar and the results were normalized within each piece. During normalisation, rests were ignored, so as to make the difference between higher tensions more readily visible.



(a) Cloud diameter



(b) Cloud momentum



(c) Tensile strain

Figure 12: Opening bars of Beethoven's Sonata Op. 31 No. 3.

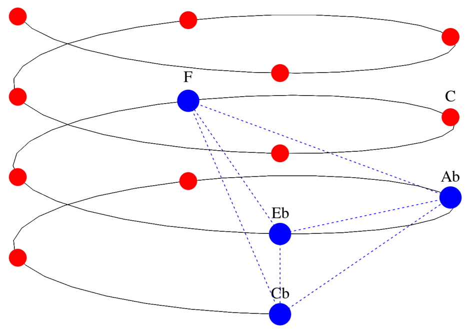


Figure 13: The Tristan chord in Beethoven's Sonata Op. 31 No. 3, enharmonically equivalent to Wagner's Tristan chord, but less dispersed in the spiral array space.

3.3.1 *Schubert*

Figure 14 and 15 display the harmonic tension ribbons for the first 17.25 bars and the last 14 bars of Schubert's Piano Sonata in C minor D958, second movement: Adagio in A♭ major. The last bars contain basically the same material, but are harmonised differently to sound even more tense. It is noticeable that sections with increased tension are mostly captured by one of the three suggested methods, yet not always by all three methods at the same time.

For example, the cloud diameter does a good job of capturing the heightened tension in the middle of the first bar of the ending segment, but less so the cloud momentum or the tensile strain because there is very little movement. Alternatively, in the second and third systems of the ending segment, the tensile strain has elevated values due to the tension of the pull away from the A♭ major key to the more distant regions of D major/minor and A major. For the D major/minor zone, the cloud diameter is low because the pitches are fairly self-consistent even though the keys vacillate between D major and minor, the tonic remains the same and the cloud momentum is only higher at the beginning. The cloud diameter is also high in the A major region, but not the cloud momentum.

3.3.2 *Beethoven*

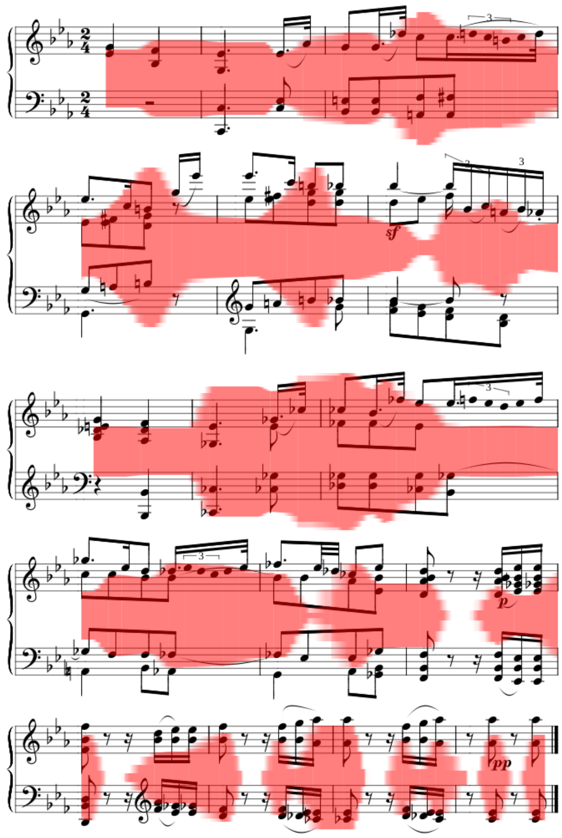
Figures 16 and 17 show the harmonic tension ribbons for the first 16 bars of Beethoven's Les Adieux (Sonata 26, opus 81a, in E♭ major). This is an example that is fraught with tension, as reflected in the high tensile strain throughout. The tensile strain is only not high at regions with the E♭ major chord (as in the second eighth note of bar 6) or E♭ major seventh chord (towards the end of the excerpt). The

(a) Beethoven intro – Tensile strain (min 0.1, max 1.8)

Figure 17: Analysis of Sonata 26, op 81a (first 16 bars).

cloud diameter varies, with the highest values at the right hand's melodic turn in bars 3, 9, and 10, where the accompanying harmonies are also moving chromatically. The cloud momentum turns out to be less informative for this example as the highest values are associated with melodic leaps when there are very few notes, as in the end of bar 4.

4. CONCLUSIONS

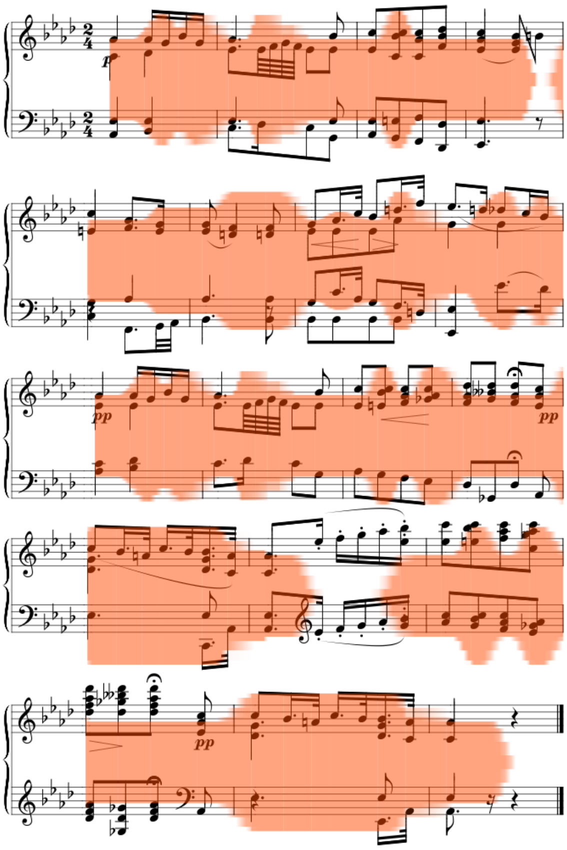
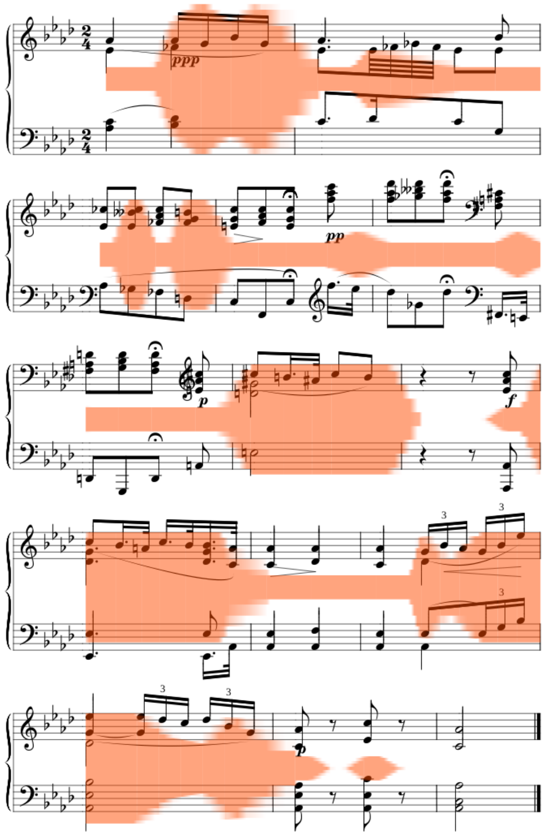
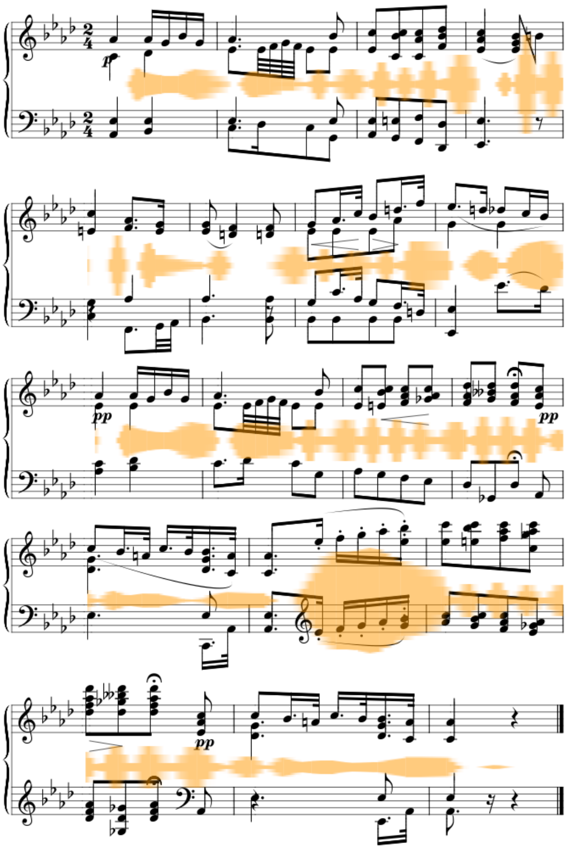
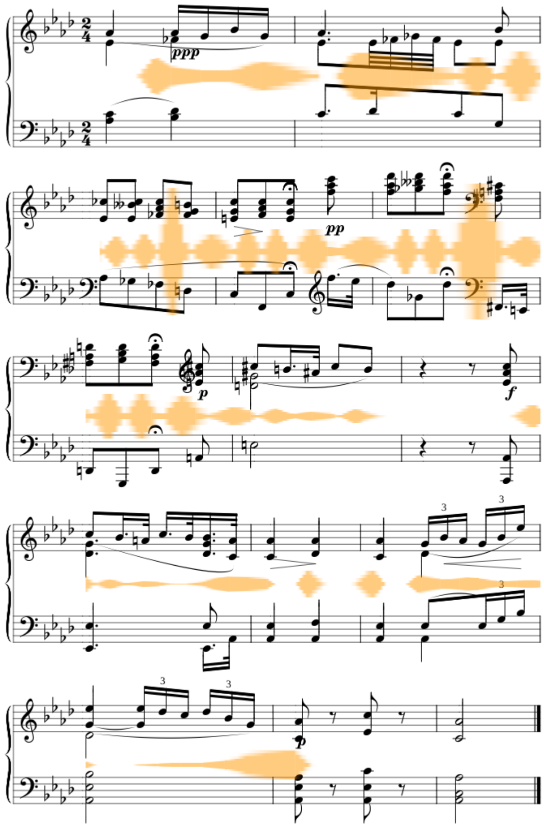
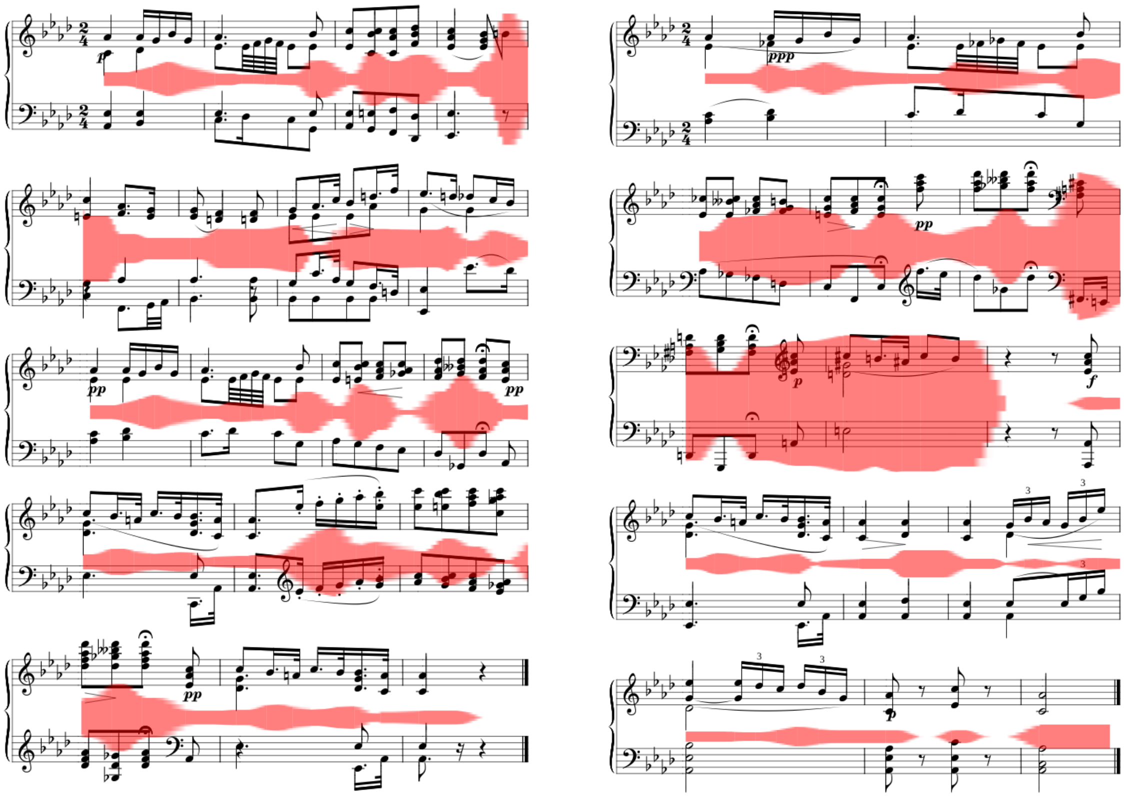
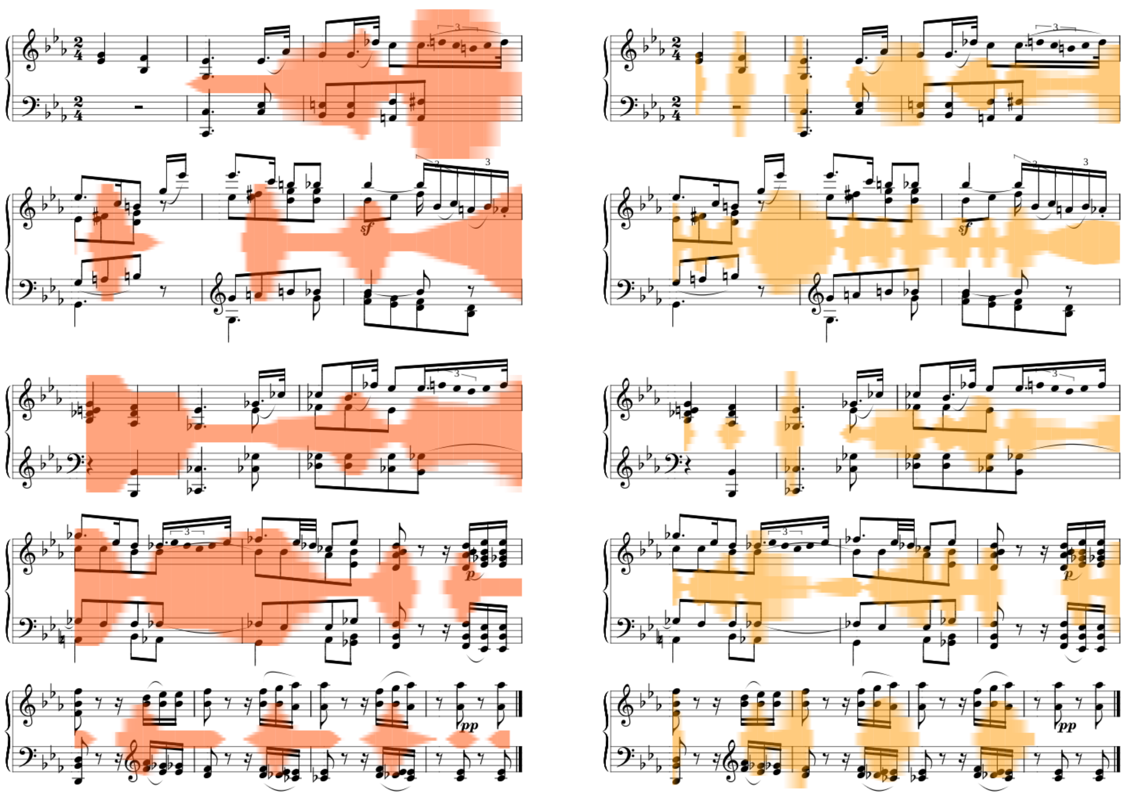
Tension is a complex concept which is not easy to define or quantify. In this paper we have developed and imple-

(a) Schubert beginning – Cloud diameter (min 1.5, max 3.6)

(b) Schubert ending – Cloud diameter (min 0.2, max 3.0)

(c) Schubert beginning – Cloud Momentum (min 0.1, max 2.6)

(d) Schubert ending – Cloud Momentum (min 0.2, max 3.5)

Figure 14: Comparison of the beginning and ending of Schubert's Piano Sonata in C minor D958, second movement: Adagio (in A-flat major).



(a) Schubert beginning – Tensile strain (min 0.2, max 3.0)

(b) Schubert ending – Tensile strain (min 0.2, max 3.0)

Figure 15: Comparison of the beginning and ending of Schubert's Piano Sonata in C minor D958, second movement: Adagio (in A♭ major).

(a) Beethoven intro – Cloud diameter (min 1.5, max 4.2)

(b) Beethoven intro – Cloud momentum (min 0.1, max 1.9)

Figure 16: Analysis of Sonata 7826, op 81a (first 16 bars).

mented three methods for measuring aspects of tonal tension. These methods are based on the spiral array, a model for tonality. The implemented system is able to display tension ribbons over the input musical scores, thus allowing for easy interpretation. An analysis of existing pieces and a comparison with an empirical study [1] revealed that cloud diameter, cloud movement and tensile strain all contribute to capturing the composite feature humans refer to as tension.

This work only attempted to model aspects of tonal tension. The proposed measures relate to perceived distance between notes in a cluster, between consecutive clusters of notes, and between the global and local tonal contexts. They fail, however, to consider tension that is caused by other kinds of expectation, such as that due to delay of cadential closure—modeling this kind of tension is trickier because not all dominant-tonic pairs form cadences. Much work remains to understand the many different parameters that contribute to the perception of musical tension.

Future research includes conduct a more thorough empirical study of how the quantitative measures produced by the methods discussed in this paper correlate with what listeners describe as tension. The current model could be expanded to more completely capture the composite characteristics of tension. Further extensions could take into account features related to melodic contour, rhythm and timbre. Beyond score features, another interesting expansion would be to capture the influence of performance (e.g. timing and dynamics) variations on tension.

Acknowledgments

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 658914.

5. REFERENCES

[1] M. M. Farbood, "A parametric, temporal model of musical tension," *Music Perception*, vol. 29, no. 4, pp. 387–428, 2012.

[2] E. H. Margulis, "A model of melodic expectation," *Music Perception*, vol. 22, no. 4, pp. 663–714, 2005.

[3] C. L. Krumhansl, "An exploratory study of musical emotions and psychophysiology." *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 51, no. 4, p. 336, 1997.

[4] L. B. Meyer, *Emotion and Meaning in Music*, ser. Phoenix books. University of Chicago Press, 1956.

[5] R. S. Hackworth and W. E. Fredrickson, "The effect of text translation on perceived musical tension in debussys noël des enfants qui nont plus de maisons," *Journal of Research in Music Education*, vol. 58, no. 2, pp. 184–195, 2010.

[6] C. L. Krumhansl, "A perceptual analysis of Mozart's piano sonata K. 282: Segmentation, tension, and musical ideas," *Music perception*, pp. 401–432, 1996.

[7] G. Ilie and W. F. Thompson, "A comparison of acoustic cues in music and speech for three dimensions of affect." *Music Perception*, 2006.

[8] F. Lerdahl and C. L. Krumhansl, "Modeling tonal tension," *Music Perception: An Interdisciplinary Journal*, 2007.

[9] E. Bigand and R. Parncutt, "Perceiving musical tension in long chord sequences," *Psychological Research*, vol. 62, no. 4, pp. 237–254, 1999.

[10] F. V. Nielsen, *Oplevelse af musikalsk spænding. Bd. 1-2: Bd. 1: Tekstdel. Bd. 2: Bilag*. Akademisk Forlag, 1983.

[11] E. Bigand, R. Parncutt, and F. Lerdahl, "Perception of musical tension in short chord sequences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training," *Perception & Psychophysics*, vol. 58, no. 1, pp. 125–141, 1996.

[12] R. Y. Granot and Z. Eitan, "Musical tension and the interaction of dynamic auditory parameters," *Music Perception: An Interdisciplinary Journal*, 2011.

[13] M. M. Farbood, "A quantitative, parametric model of musical tension," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[14] F. V. Nielsen, "Musical tension and related concepts," *The semiotic web*, vol. 86, pp. 491–513, 1987.

[15] E. Chew, *Mathematical and Computational Modeling of Tonality: Theory and Applications*. New York: Springer, 2013, vol. 204.

[16] E. Chew, "The spiral array: An algorithm for determining key boundaries," in *Proceedings of the Second International Conference on Music and Artificial Intelligence*. Springer-Verlag, 2002, pp. 18–31.

[17] C.-H. Chuan and E. Chew, "Polyphonic audio key finding using the spiral array ceg algorithm," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 21–24.

[18] D. Fober, Y. Orlarey, and S. Letz, "An environment for the design of live music scores," in *Proceedings of the Linux Audio Conference, CCRMA, Stanford University, California, US*, 2012, pp. 47–54.

[19] E. Narmour, *The analysis and cognition of melodic complexity: The implication-realization model*. University of Chicago Press, 1992.

[20] N. Martin, "The Tristan chord resolved," *Intersections: Canadian Journal of MusicIntersections:/Revue canadienne de musique*, vol. 28, no. 2, pp. 6–30, 2008.

[21] A. Lorenz, "Review of Ernst Kurth: Romantische harmonik und ihre krise in Wagners Tristan.," *Die Musik*, vol. 16, pp. 255–262, 1924.

[22] G. Weitz, "Love and death in Wagner's Tristan und Isoldean epic anticholinergic crisis," *BMJ: British Medical Journal*, vol. 327, no. 7429, p. 1469, 2003.

HYBRID REAL/MIMETIC SOUND WORKS

Lindsay Vickery
Edith Cowan University
l.vickery@ecu.edu.au

ABSTRACT

This paper describes a project to construct a process allowing for data interchange between visual and sonic media: to create a continuum in which sound could be visualized and then resonified through by both live performers and digital means.
A number of processes to aid this visualisation/sonification "ecosystem" were developed. Software was created to create scores based on sonic features of "field recordings" through spectral analysis by rendering the frequency of the strongest detected sinusoidal peak of a recording vertically and its timbral characteristics by luminance, hue and saturation on a scrolling score. Along similar principals a second process was developed to generate a realtime score using graphical symbols to represent detected accents in "found sound" speech recordings. In the other direction software was built to render greyscale images (including sonograms) as sound and a second iteration to generate audio from detected analysis parameters.
The imperfections in the various transcription processes are intriguing in themselves as they throw into relief the distinctions between the various forms of representation and in particular the timescales in which they are perceived. The implied circularity of processes also opened the potential for re-interrogation of materials through repeated transmutation. This discussion explores these implications in the context of the analysis of field recordings to generate visual representations that can be resonified using both performative (via notation) and machine (visual data-based) processes, to create hybrid real/mimetic sound works through the combination (and recombination) of the processes.

1. INTRODUCTION

This paper describes a project to construct means to interchange data between visual and sonic media: to create a continuum in which sound can be visualized and then resonified. *Nature Forms II* (2015) – and the work that has led to it - is discussed as a vehicle for exploring the possibility of recursive re-interrogation of a field recording through visualization and resonification/resynthesis via machine and performative means.
In the work's predecessor, *Nature Forms I*, a score comprising manipulated images of organic shapes derived from photographs of trees, plants and rocks, was simultaneously sonified by performers and software. Three performers and software "read" from the same scrolling score on networked laptops

Four contrasting forms of reading/sonifcation were involved: machine sonification in which spatial position and colour of features from the image were more or less precisely rendered; tablature in which spatial position and colour were recast against the geography of a specific instrument; semantic reading in which the performer's understanding of notational conventions informed the outcome; and aesthetic reading in which the performer's understanding of the conventions of sonic representation were drawn upon [1].

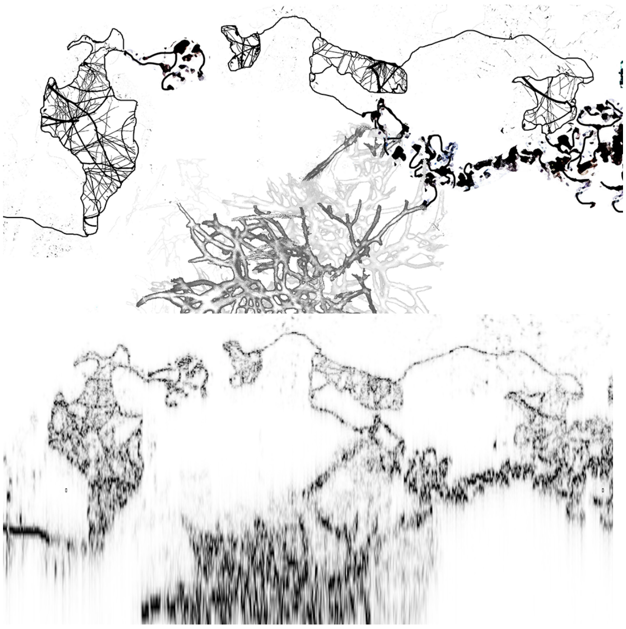


Figure 1. Comparison between an excerpt from the score of *Nature Forms I* [2014] (above) and a sonogram of its sonification (below).

In *Nature Forms II* a range of forms of representation are also explored including semantic graphical notation, percussion notation and a hybrid form of sonogram notation. The concept of multiple notations was also emphasised in another work, *Sacrificial Zones* (2014) in which the performer reads from a rhizomatic score - the notation moves along interconnected vertical and horizontal pathways - that crossfades between five layered images, each notated in a manner corresponding to a different form of visual representation of sound: non-semantic graphical

notation, semantic graphical notation, traditional notation, proportional notation and a spectrogram.
The score confronts the performer (and vicariously the audience for whom it is projected) with the variation in freedom and constraint presented by a range of forms of notational representation. The rhizomatic and layered procedure for rendering the score allows for multiple versions of this work, emphasising different aspects of the relationship between varied notations of the same musical object [2].

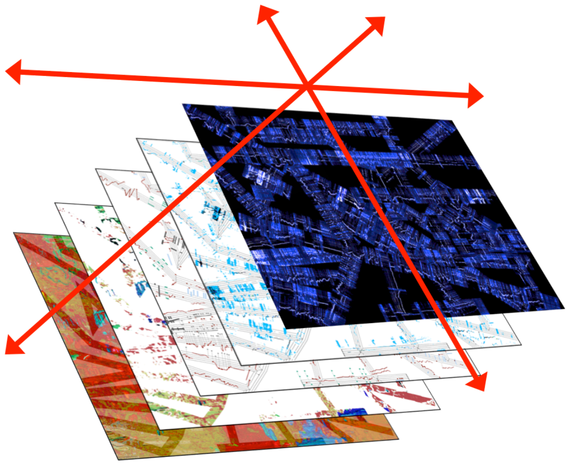


Figure 2. Layers of different visual representation of sound in *Sacrificial Zones*.

This conceptual basis assumes that visual forms of musical representation may be considered to occupy a continuum, between the spectrogram (a precise frequency/time/amplitude representation of sound), proportional notation, traditional notation, semantic graphical notation, non-semantic graphical notation [3]. It is also assumed that notation has inherent semantic implications as a consequence of a degree of 'weak synaesthesia' [4, 5] or cross-modal activation that is present in the population at large, and therefore that graphical symbols can elicit meaning through the inherent semantic qualities of their shape and colour, and that consideration of these qualities is crucial for the development of effective and efficient notation for screen scores [1, 2, 3].

2. VISUALISATION PROCESSES

A number of forms of notation/visualization are employed in *Nature Forms II*. This most literal is the process I have previously termed the "spectral trace" (Vickery 2014b) in which notation is drawn directly onto the spectrogram. This approach was used in the work *acid fury* (2015) in which colour-coded parts for the eight instruments were made from direct transcriptions of a spectrogram of the recordings (Fig. 3.).
Nature Forms II employs this process to represent features of the field recording to be performed by an ensemble of clarinet (orange), viola (red), and cello (green) (Figure 4.). The frequency/amplitude morphology of features of the field recording is communicated to the performer by extracting shapes directly from the spectrogram.

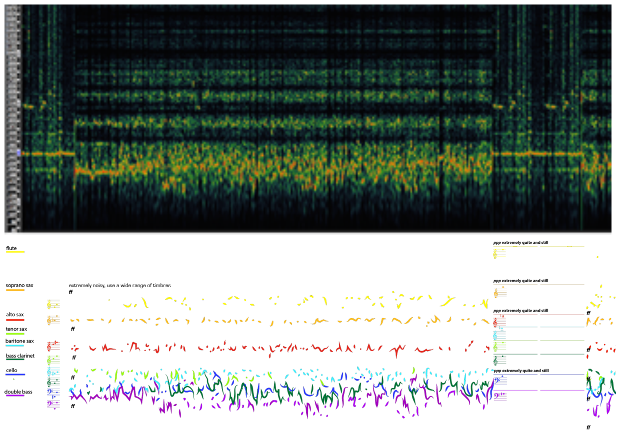


Figure 3. Spectrogram of found-sound recording (above) and annotated graphical transcription (below) of *acid fury* [2015] (excerpt).

A system presenting graphical symbols on a staff that is proportional both horizontally and vertically was created to solve the problem each pitch in a spectrogram occupies a distinct vertical spatial position. The system was developed for the work *here, apparently, there was time for everything* [2015], and attempted to more-or-less retain the topographical layout of the traditional stave, while adding coloured lines to indicate non-natural notes (Fig. 5.). This approach was eventually abandoned in favour of annotation of the score with pitch and articulation information in reference to each spectrally traced graphic.

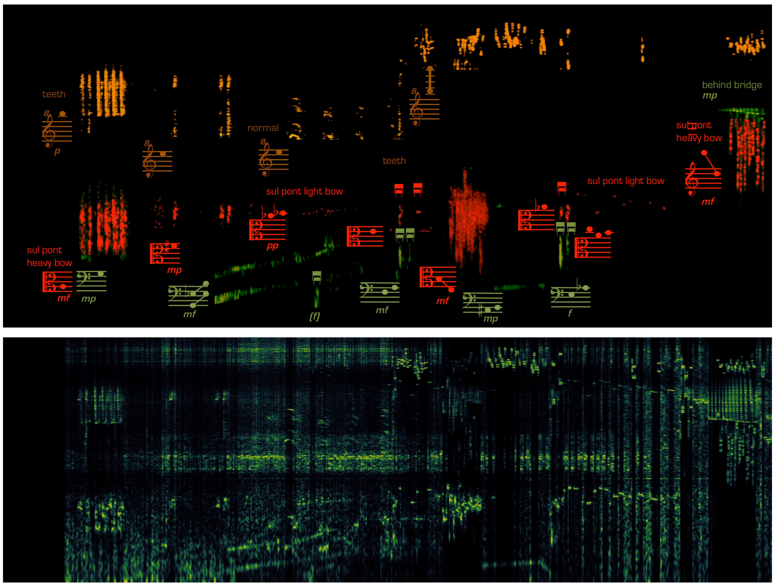


Figure 4. "Spectral Trace" notation from *Nature Forms II* (above), source spectrogram (below).

A second approach drews on the concept and techniques developed in *EVP* (2011) and *Lyrebird: Environment Player* (2014) in which the amplitude, frequency, brightness, noisiness and bark scale[1] of the single strongest

[1] The median of 16 bark scale values (representing the deviations from expected critical bands) is used. This presupposes that the median value refers to the same critical band as the

detected sinusoidal peak in a recording is represented by the vertical height, size, luminance, hue and saturation of rectangles drawn on a scrolling LCD object (in this case *jit.lcd*).

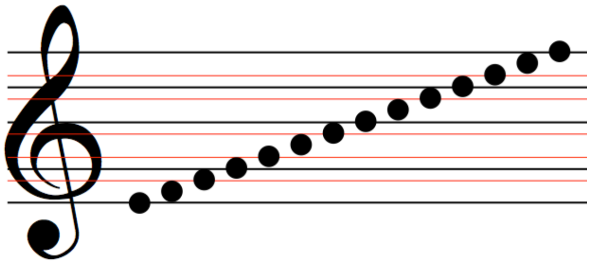


Figure 5. Vertically proportional stave showing a chromatic scale from E3 to F4.

Data is derived from Tristan Jehan's `analyzer~` [6] object in realtime allowing for the scoreplayer to visualise timbral features of the recorded sound.[2] The visualised score depicting the principal features of a source recording is scrolled from right to left across the computer screen and playback of the source recording is delayed (12 seconds in this work) to allow the performer to see a visualisation of the sounds before they appear (Figure 6.)

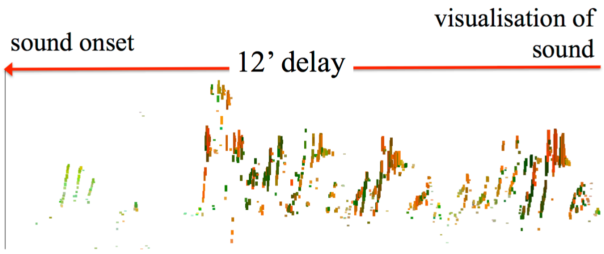


Figure 6. The scrolling scoreplayer for *Lyrebird: environment player* [2014] showing visualized pitch, amplitude and timbral data.

In the work *murmurs trapped beneath the bark* [2014], an this idea is elaborated through the use of a processed recording of a clarinet improvisation as the source audio. Its ambiguous resemblance to real-world natural sound led me to term this an "artificial field recording". In *murmurs* the score produced by *Lyrebird* is also visually processed in *Illustrator* to further the analogy with the idea of an artificial recording (Fig. 7.). It was this work, and this process that suggested the possibility of creating hybrid real/mimetic sound works interchangeably combining field recordings and their machine and human emulations.

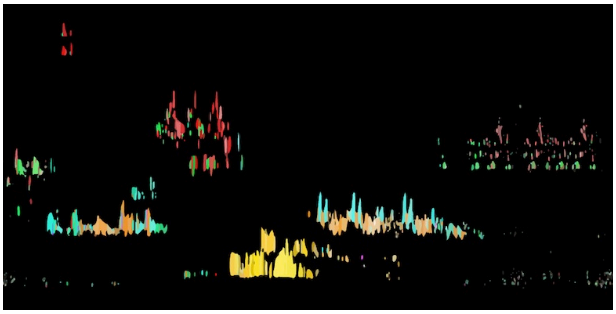


Figure 7. Detail from the score of *murmurs trapped beneath the bark* [2014].

The "lyrebird" generative score process was also employed in *Nature Forms II*, to create visualisations of the field recording. The *lyrebird* visualisations are scalable allowing for renderings of the score that focus upon high, middle and low frequency bands of the field recording.

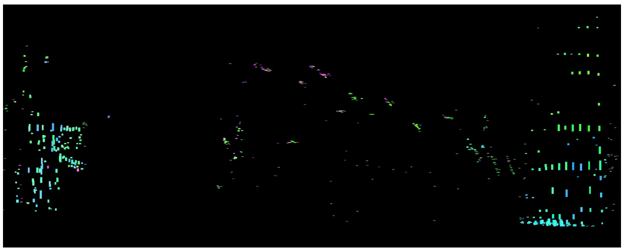


Figure 8. Detail from the high frequency "lyrebird" score of *Nature Forms II* [2015] (excerpt).

In the work *the miracle of the rose* [2015] these ideas were further elaborated combining procedures from the spectrogram score with gestural conventions. The work is based on a passage concerning the time-altering nature of solitary confinement from Jean Genet's novel *The Miracle of the Rose* [1946]. A collage of time-stretched recordings of the text by Australian/French artist Emmanuelle Zagoria was used as the underlying structure of the work. The spoken phrases were transcribed for the two percussionists into gestures exploring their cadence and timbre via varied instruments and notational approaches.

In Fig. 9 the notation for player 2 (which occupies the lower half of the page), indicates the amplitude of the sound (the vertical height), the timbral richness (hue), onset of event (stem) and direction of the bow (beam) for a bowed cymbal gesture that follows the envelope of the fixed media recording. The notation for this figure was again created using *Lyrebird: environment player* software.

In the upper half of Fig. 9 the notation for player 1 indicates muted cymbal strikes (speech rhythms transcribed from the spectrogram). The changing position of the strike is indicated by the direction of beam. In both parts the thin curved beams indicate the movements of the performer's arms between actions. The score is intended to be projected behind the performers, allowing the audience to see the ritualistic gestural coordination between the performers and the score.

strongest sinusoidal component. In future it may be possible to model this parameter more accurately.

[2] A version using externals by Alexander Harker [7], using spectral centroid, spread and skewness is currently being trialed.

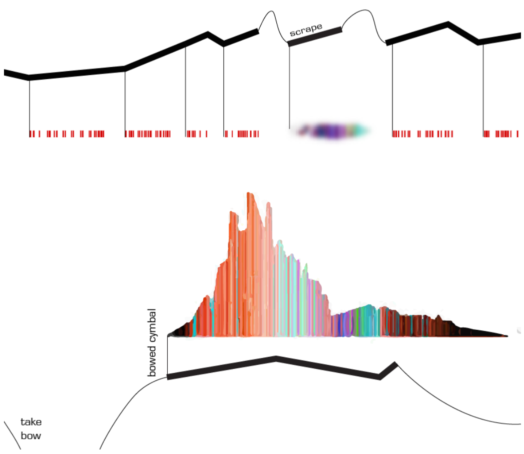


Figure 9. Excerpt from the score of *the miracle of the rose* [2015].

In Fig. 10. player 2 is bowing 5 different keys on the vibraphone, while player 1. is lowering a medium-sized chain onto the keys of a second vibraphone. The red note-heads and the downwardly inclined beams indicate lowering the chain onto the keys and orange note-heads (and upwardly inclined beams) indicate lifting the chain off the keys (both actions produce sound). Again the contours are transcribed directly from a sonogram of the accompanying recording.

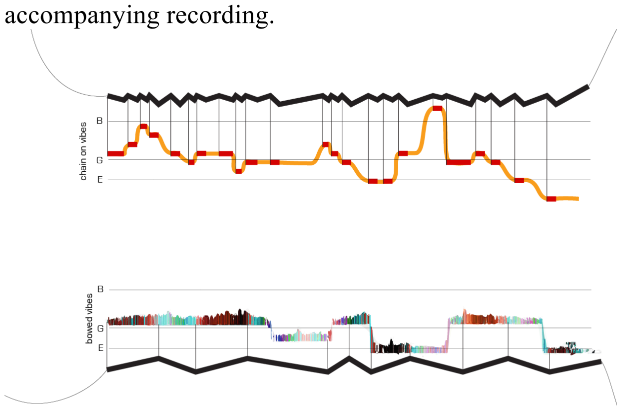


Figure 10. Excerpt from the score of *the miracle of the rose* [2015].

The final form of visualization in *Nature Forms II* is derived from the work *Semantics of Redaction* (2014) in which notation in generated (Figure 11.) by using accents detected in a speech recording in real-time to generate graphical symbols of varying vertical position, size and colour, determined by the frequency, amplitude and timbre of a speech recording at the accent point. In *Nature Forms II* the software detects accepts created by sharp attacks such as the chirping of crickets. This method was employed as the paradigm for the percussion notation for *Nature Forms II.*

These works explore varied aspects of what O'Callaghan terms "mimetic instrumental resynthesis":

> Not only do these works use 'extra-musical' source materials as the starting point of their analyses, but they also attempt to preserve aspects of the source sound through the transcriptive process to engage in a mimetic discourse. [8]

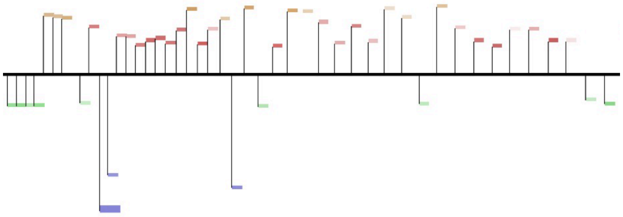


Figure 11. Percussion part for *Nature Forms II* created using the generative score software *Semantics of Redaction* (excerpt).

In particular the works allow for the contrast and interaction of instrumental and machine forms of sonification. The ability to interchange audio and visual representations of a work and to precisely synchronise them with live electronics provides a controllable work environment, including performers and electronic sound, that is not unlike that only previously available to acousmatic composers working with recorded sound alone. Figure 12. shows an excerpt of the workings for the score of *the miracle of the rose*, indicating how the score was developed in reference to spectrograms of the time-stretched recordings. The blue lines below the spectrograms indicate which of the recordings is sounding at any moment and therefore which form of notational representation is employed to represent the sound of the fixed media.

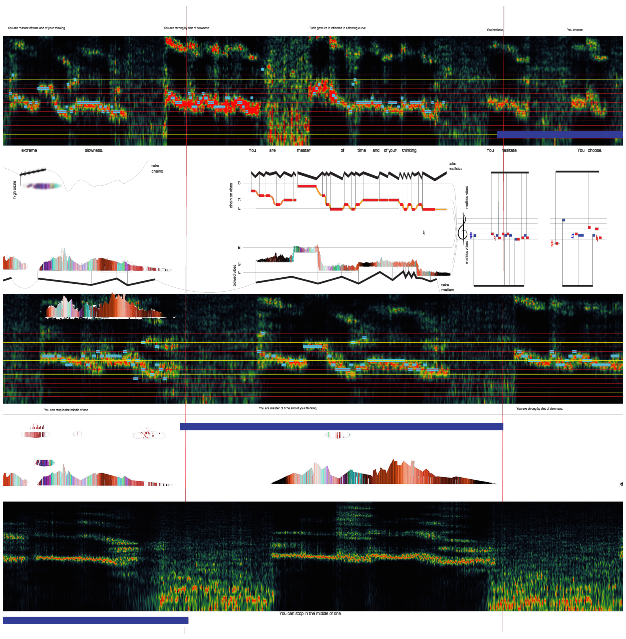


Figure 12. Score workings from *the miracle of the rose* [2015] showing the forms of notational used to represent the sound of the fixed media.

3. SONIFICATION PROCESSES

A number of approaches to sonfication/resynthesis are used in *Nature Forms II*. The first is an additive synthesis approach using a patch called *Sinereader* [1] developed in MaxMSP to resonify greyscale spectrogram images (Fig. 13.). In the patch each vertical pixel of a greyscale version of the spectrogram of a sound is mapped to one of 613 independent sinewave generators. In the patch a

.png file of the spectrogram is loaded into a jit.qt.movie, it is then played through jit.matrix and jit.submatrix that send an image of one pixel width to the jit.pwindow.
Data from the submatrix is split into a list of 613 values in jit.spill and these values are represented in a multislider. The vertical pixels are scaled logarithmically according to the vertical resolution of the spectrogram and each mapped to an individual cycle~ object. The greyscale value of each pixel is scales and mapped to the amplitude of each `cycle~` object. In addition to being a transcription tool, the patch can also be controlled externally as an "instrument" using *MaxMSP Mira* app for iPad.

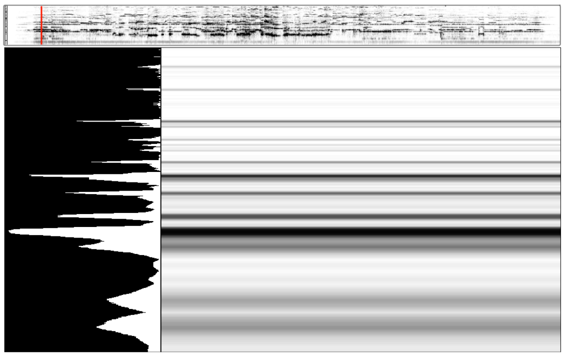


Figure 13. The *Sinereader* patch The complete spectrogram with a "scrollbar" indicating progress through the image is displayed at the top of the image, the greyscale value of each vertical pixel in a one pixel segment is displayed on the bottom right and the resulting amplitude is displayed on the bottom left.

Processes were also developed to resynthesise sounds using spectral analysis. In the first process, the strongest sinusoidal component detected each 40ms of the recording were resynthesized with a sinewave that was then ring-modulated according to the currently detected brightness of the recording (Figure 13.). The aim was to retain the amplitude of sonic features of field recording while maintaining and equivalent brightness.

Figure 13. Spectrogram of Ring Modulation synthesis from *Nature Forms II* (excerpt).

This method is also used in portions of the performance to ring modulate the instruments via live processing. In order to mimic the parametric brightness of the field recording the brightness of the live instruments is subtracted from that of the field recording to derive a ring modulation value.
Subtractive synthesis was also employed by using frequency and amplitude data detected in the recording to bandpass filter white-noise (Figure 14.). The result of these processes are sonic abstractions sharing morphological traits, but sonically distant from the source material

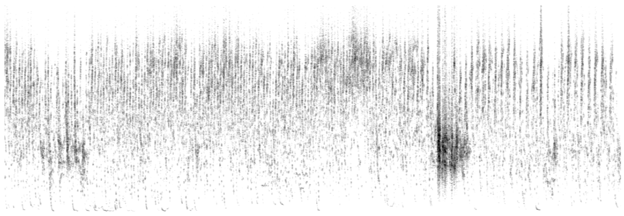


Figure 14. Spectrogram of subtractive resynthesis from *Nature Forms II* (excerpt).

At the opening of *Nature Forms II* "coloured noise" [9] performed by the instrumentalists is gradually shaped into the sonic structure of the field recording using subtractive synthesis, and then cross-faded with the source recording.
A similar process is used by Peter Ablinger in some of his "Phonorealist" works in the *Quadraten* series [8], in which spectral analysis data from recordings is "reconstituted in various media: instrumental ensembles, white noise, or computer-controlled player piano" [10]. A key issue at the heart of *Quadraten* is representation or analogy made between "real" sounds and their reconstituted counterparts. The transmutation through different forms of synthesis causes a loss of resolution and this loss can become interesting in itself.
The work also uses "spectral freezing" of components of the field recording to create spectrally derived chords from features of the recording bird sounds and a rusty gate which are then transcribed into notation for the instrumentalists and temporal manipulation of the recording to allow complex bird calls to be emulated in a human time-scale.
Nature Forms II explores the notion of eco-structuralism, maintaining what Opie and Brown (2006) term the "primary rules" of "environmentally-based musical composition": that "structures must be derived from natural sound sources" and that "structural data must remain in series".
The structure of the original work is conserved using the approach discussed in *the miracle of the rose"*, where the temporal proportionality of the recording is retained by aligning multiple notation and resynthesis versions of the recording in visual representations that can be alternated or combined in the creation process of the score, processing and fixed media (Figure 15.)

4. CONCLUSION

The processes described in this paper constitute a set of possible approaches to engaging with field recordings through machine and performative means. They provide a methodology for manipulating a "found sound" in a relatively precise manner through spectral analysis and synchronisation of visual and sonic elements of the work. Many of the possibilities opened up by the processes described above have been enhanced by developments afforded by the *Decibel Scoreplayer* namely: synchronised networking, communication with external computers via OSC, audiofile playback, cross-fading of layers, random playback of score "tiles" and "nesting" of scoreplayer types [11].
The imperfections in the transcription processes involved here are intriguing in themselves as they throw into relief

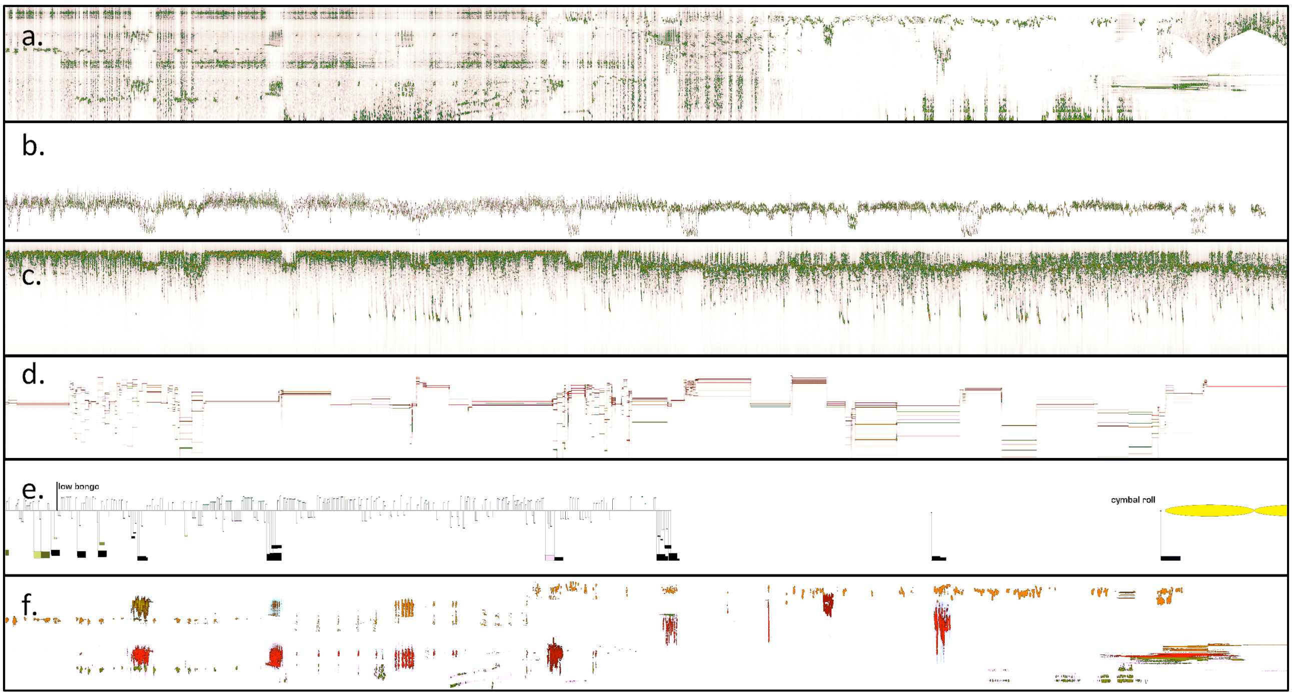


Figure 15. Visual representation of temporally proportional alignment of multiple resynthesis (a.-d.) and notation (e. – f) versions of the recording in *Nature Forms II* (excerpt): a. field recording spectrogram; b. ring modulation resythesis spectrogram; c. subtractive synthesis spectrogram; d. spectral "freeze" sonogram/score e *Nature Forms II* percussion score; and f. *Nature Forms II* instrumental score.

the distinctions between the various forms of representation: highlighting the gaps between "mimetic resynthesis" at different levels of abstraction. The implied circularity of processes opens the potential for re-interrogation of materials through the continuous transmutation of transcription by substituting source recordings for their resynthesised counterparts or live instruments and reprocessing them.

The efforts to extend notation discussed here are part of an ongoing project to better capture nuances of sound such as timbre, temperament and envelope morphology using shape and colour parameters (hue, saturation and luminosity).

5. REFERENCES

[1] Vickery, L. R. (2014b). Exploring a Visual/Sonic Representational Continuum. International Computer Music Conference. A. Georgaki et al. Athens, Greece. International Computer Music Association. 1: 177-184.

[2] Vickery, L., (2014a), Notational Semantics in Music Visualisation and Notation. *Proceedings of the Australasian Computer Music Conference 2014*, 101-112.

[3] Vickery, L., (2014c), The Limitations of Representing Sound and Notation on Screen. *Organised Sound*, 19(3), 215-227, Cambridge, UK

[4] Martino, G., and Marks, L. E. (2001). Synesthesia: Strong and weak. *Current Directions in Psychological Science*, 10, 61–65.

[5] Marks, L. E., and Odgar, E. C. (2005). Developmental constraints on theories of synesthesia. *Synesthesia; Perspectives from cognitive neuroscience*. New York: Oxford University Press. 214–236

[6] Jehan, T. (2004). Event-synchronous music analysis/synthesis. In Proc. Digital Audio Effects Workshop (DAFx), Naples, Italy.

[7] Harker, Alexander (2011) *AHarker Externals.* Alexander J. Harker. (Unpublished)

[8] O'Callaghan, J. (2015). Mimetic Instrumental Resynthesis. Organised Sound (20:2), 231 - 240

[9] Eimert, H. (1955). *Die Reihe* 1. Theodor Presser: Bryn Mawr: 4.

[10] Barrett, D. G. (2007). Music and Its Others. http://ablinger.mur.at/docs/barrett_others.pdf

[11] Hope, C., Wyatt, A. and Vickery, L. (2015). The Decibel Scoreplayer: Enriched Scores For The iPad. Proceedings of the ICMC 2015, Denton, Texas 314-317.

VISUALIZING MUSIC IN ITS ENTIRETY USING ACOUSTIC FEATURES: MUSIC FLOWGRAM

Dasaem Jeong and Juhan Nam
Graduate School of Culture Technology, KAIST
{jdasam, juhannam}@kaist.ac.kr

ABSTRACT

In this paper, we present an automatic method for visualizing a music audio file from its beginning to end, especially for classical music. Our goal is developing an easy-to-use visualization method that is helpful for listeners and can be used for various kinds of classical music, even for complex orchestral music. To represent musical characteristics, the method uses audio features such as volume, onset density, and auditory roughness, which describe loudness, tempo, and dissonance, respectively. These features are visually mapped into static two-dimensional graph, so that users can see how the music changes by time at a look. We implemented the method with Web Audio API so that users can access to the visualization system on their web browser and make visualizations from their own music audio files. Two types of user tests were conducted to verify the effects and usefulness of the visualization for classical music listeners. The result shows that it helps listeners to memorize and understand a structure of music, and to easily find a specific part of the music.

1. INTRODUCTION

Music visualization is widely used in various music activities for many purposes. Because music is an auditory art, its visual representations can contain information that cannot be transferred or preserved accurately with sound. Music notation is a typical example of the visualized music representations. It is designed for communication between composers and performers. The notation systems thus have been evolved to represent and deliver a composer's intention as precise as possible.

For listeners, however, music notation has some limitations. It contains too much information for listeners to interpret and so only a small part can be understood while following the music. Especially, in the case of orchestral music, the score following task is quite difficult unless the listeners are musically trained. Another problem is that the notation does not show the entire structure of a piece of music. The time scope of a music score that can be read in a sight is limited to a few measures. The notation is focused on delivering information about what is happening in a specific time. To understand the global structure, one needs to read through the score for a while, having a certain level of musical knowledge.

As a way of making up the shortcomings of music notation, audio-synchronized music scores have been developed [1]. Synchronized scores automatically follow music on the score so that listeners can easily track where the currently playing measure is or select the measure on the score to play the music from the position. However, such systems require a synchronization process between audio and score. Manual synchronization is too laborious to process a large set of pieces, whereas automatic one, an active research topic in the area of music information retrieval (MIR), is not accurate enough particularly for large orchestral music. Above all, these solutions still cannot show the entire structure of a piece.

In the case of classical music, particularly for long instrumental pieces, visualizing information about the entire structure can be helpful to music listeners in that they do not contain lyrics or clear storytelling to follow. So additional information about the music is required to help listeners to understand the music. A traditional way of providing the information is giving a lecture or writing a program note. But these requires professionals who can explain the music. Many researchers instead have suggested a content-based approach to visualize the entire structure of music from audio. Most of automatic music structure visualization methods are based on self-similarity between each part of a piece [2, 3, 4]. These methods show a repetitive structure of the music based on the self-similarity. Further information about the research is introduced in Section 2. In general, it is not easy to interpret the meaning of the visualizations. Finding the repetitive structure can help music structure analysis, but its usefulness on listeners has been not verified yet.

To address this problem, we present an automatic music visualization method named *music flowgram*, which aims to visualize an entire piece as an easy-to-understand image. It extracts audio features from audio files and visualizes them on a static two-dimensional graph. In our previous research, we found that a simple static graph showing the change of volume of a music piece can help listeners to concentrate more on classical music, compared to spectrum-based real-time visualization [5, 6]. We have improved this concept by adding additional features that can represent other important characteristics of music, and conducted user test to verify its effect on listening to classical music.

The later part of this paper is organized as follows. First, related work on visualizing music structure is briefly reviewed. Then, we present our visualization method in two sections: concept of the visualization and audio features in Section 3, and its implementation in Section 4. The detailed information about user tests are described in Section 5, and results with discussion in Section 6. The last section concludes the paper with a summary and our plan for future work.

2. RELATED WORK

There has been some research on visualizing the structure of a music piece, both in data visualization and MIR areas. The majority of them exploited the repetitive structure of music using self-similarity within a piece. Wattenberg visualized it using an arc diagram that connects each repetitive part with an edge being drawn as a semicircle [2]. Foote visualized the self-similarity as a two-dimensional matrix where each element is calculated from similarity between two audio frames [3]. Müller and Jiang extended it to a scape plot representation that visualizes the repetitive structure with varying segment size. Other researchers combined this self-similarity information with volume transitions over time [7]. There has been also research that applies this structural information to music listening interfaces [8, 9].

Other than those based on repetitions in music, some work visualized the structure using tonality such as key change over multiscale segments [10]. Malt and Jourdan presented a visualization method using statistical characteristics of spectral information, including spectral centroid and standard deviation of the audio spectrum [11]. They illustrated the change of those information over time on a two-dimensional graph, adding amplitude information as a color of the graph. However, the most of the mentioned research have not released an end-user application so that general users can render their own visualization. Furthermore, this research lacks user test or human side experiments that verify its effect and usefulness for listeners.

Besides the automatic visualization methods using audio files or MIDI files, visualization of semantic structure of music is also proposed [12]. This method contains a lot more information than repetitive structure, for example, traditional structure analysis of sonata form, motif development, and how the role of each instrument changes through the piece. But all of the information used in visualization is manually extracted from written explanation of the music, and cannot be automatically computed from audio files.

There is also music psychological research about visualizing whole music [13]. This research tested how people describe short music with graphical representations. Participants are asked to “make any marks” to describe five short orchestral works after listening to the music. The result showed that musically trained participants more tended to describe music with abstract representations such as symbols and lines. Most frequently used mapping was X-axis as time and Y-axis as pitch. The other type was pictorial representations, which were mostly drawn by untrained participants. Among 30 musically trained participants, 24 used an abstract representation and 21 of them were in continuous mode. This result indicates that a two-dimensional graph is natural in human sense for representing whole music piece.

3. MUSIC FLOWGRAM

The idea of music flowgram for music visualization is based on dramatic structure of storytelling. Freytag explained the structure of each story with two-dimensional graph visualization of tension progress [14]. Our idea is applying a similar concept to music: drawing continuous two-dimensional graph that shows the change of music by time. If listeners can see a dramatic structure of music, they could feel more comfortable to concentrate on the music because they can clearly see when the tension will increase or decrease. This is similar to watching an opera, for which people are encouraged to know dramatic structure before watching. The visualization will also help the listeners to recall the sequence of the music, as people remember the order of opera story based on the order of important events.

A similar type of visualization is waveform visualization or volume graph. It shows the volume progress of the music so that users can see which part is loud or quiet. This type of visualization is used in SoundCloud [1] . Though volume is a highly important factor in deciding characteristic of music, there are other quantitative parameters to explain the music. Spectrogram is another way to show the variance of music as a two-dimensional image. However, it contains too much details to deliver meaningful musical information. Thus, more compact representations, which effectively extract musical elements, is needed.

Considering that emotion is the most influential high-level concept on listeners, we focus on musical elements that are associated with the emotional aspects of music. Among many suggested elements in this regard [15], we choose loudness, tempo and harmony. For visualization, we represent them with volume, onset density and auditory roughness, respectively, as below.

3.1 Volume

Unlike other genres of music, classical music consists with many different sub-parts, each of which has a different loudness characteristic. Therefore, temporal differences of loudness can explain the structural information of music effectively. We represent the loudness with volume which is simply calculated as frame-level energy. Though more complex measures of loudness could be adopted, we assume that the volume is sufficiently effective in complex musical sound.

3.2 Onset Density

Emotion of music is highly dependent on the tempo characteristic of music, i.e, whether the music is fast or slow. Beats per minute (BPM) is a typical way of representing it. However, the single speed measure is not sufficient to describe the tempo characteristic of music because note

[1] www.soundcloud.com

passages can vary dramatically in the same tempo. For example, a long note and multiple short notes can be located in a single beat but they produce a different nuance. For this reason, we represent the tempo characteristic with the number of notes per second. Since we need to have overall trend of local note population rather than the exact number of notes for visualization, we use a simple onset detection algorithm which counts note onsets in a selected frame based on amplitude information.

3.3 Auditory Roughness

Quantifying the harmonic feature of music from audio is typically carried out by chord recognition. However, recent work pointed out the limitation of automatic chord recognition [16]. Identifying chords can be arguable even for musicologists, especially for complex classical music. The research shows that the maximum agreement ratio between two chord annotations among four annotations on “I Saw Her Standing There” by *The Beatles* was only 65%. Also, classical music includes atonal music or late-romantic music like Wagner’s “Tristan und Isolde”. This makes hard to employ automatic chord recognition for classical music.

Instead, we use auditory roughness which can represent the tonality with a single value. It is a term used in the acoustics and psychoacoustics literature to describe buzzing sound quality that is produced by two sounds with different pitch that is distinguishable but close to each other like minor seconds interval. This feature is strongly associated with harmonic dissonance. For example, major seconds or minor thirds in a low register, which are usually avoided as dissonant intervals in the western musical tradition, makes high roughness. There are various models to calculate the auditory roughness quantitatively. Among others, we employ a model presented by Vassilakis [17] that uses two sinusoidal components with frequency f_1 and f_2 and amplitude A_1 and A_2:

$$
\begin{aligned}
R &= X^{0.1} \times 0.5Y^{3.11} \times Z \\
X &= A_{min} \times A_{max} \\
Y &= \frac{2A_{min}}{A_{min} + A_{max}} \\
Z &= e^{-b1s(f_{max}-f_{min})} - e^{-b2s(f_{max}-f_{min})} \\
s &= \frac{0.24}{(s_1 f_{min} + s_2)}
\end{aligned}
\tag{1}
$$

where $A_{min} = \min(A_1, A_2)$, $A_{max} = \max(A_1, A_2)$, $f_{min} = \min(f_1, f_2)$, $f_{max} = \max(f_1, f_2)$, $b1 = 3.5$, $b2 = 5.7$, , $s_1 = 0.0207$, $s_2 = 18.96$. The term X represents the dependence of roughness on intensity. For better understanding of this equation, we illustrate how the term Y and Z change over the A_1 and the frequency difference $f_{max} - f_{min}$, and f_{min}, respectively, in Figure 1 and Figure 2. They show that roughness is higher when the amplitude of two sin wave is similar, and the minimum frequency is lower. The roughness of complex sound can be calculated by summing the roughness of each combination of two sinusoidal components in the sound.

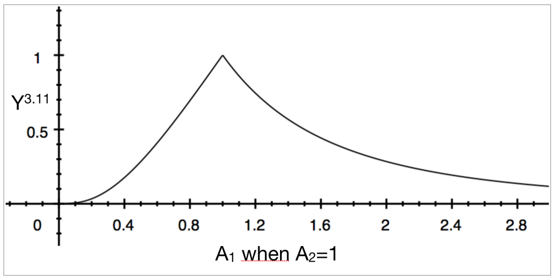


Figure 1. Change of roughness term Y by amplitude difference between two sinusoidal wave, where one amplitude is fixed to 1 and the other varies

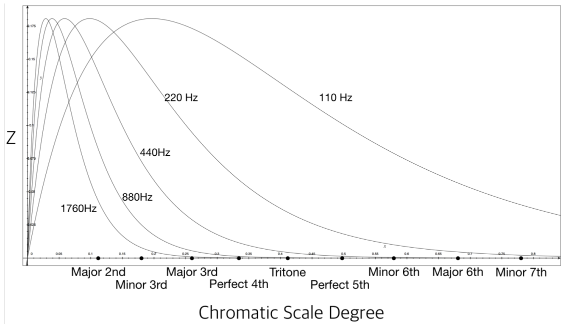


Figure 2. Change of roughness term Z by frequency difference between two sinusoidal wave. The frequency difference is represented as f_{max}/f_{min}

3.4 Visualization Scheme

As we mentioned above, representing music using a 2-D graph with x-axis in time is widely used mapping. Among three features, volume is the most accurate feature to calculate from audio data. Also it is the most dynamic feature. Therefore, we use the volume as a Y value in our visualization. For the other two features, a graph color and a background color are used as mapping targets. Since the auditory roughness is often correlated with the volume, mapping it to the graph color can be somewhat redundant. We thus map the onset density to a graph color and the auditory roughness to a background color.

4. IMPLEMENTATION

Our purpose is building an automatic music visualization system that can be easily utilized by general users. Previous visualization systems use symbolic data such as a MIDI file or pre-analyzed text data, which are not readily obtainable by listeners. Also, many of them ask users to install a stand-alone application, which might have some compatibility issue on user side or require some extra efforts. Considering these problems, we design our system such that, if users can access to audio content on a web browser, the visualization is immediately rendered from the audio file. We used web audio API which was developed for various audio applications under HTML5 specification. Therefore, our visualization system can be run on many web browsers such as Google Chrome and Mozilla

Firefox [2] .

4.1 Feature Extraction

When a user loads an audio file on the system, the file is decoded to linear PCM data and saved as a buffer on the browser memory. Then, samples are segmented by a Hann window. The volume is calculated using root mean square of audio samples for each window. The algorithm for counting onsets uses local maxima of the calculated volume sequence. Specifically, it compares each volume value in the array with the next value. If it increases, the increased amount is saved. This is accumulated if the volume keeps increasing. If the volume decreases, the algorithm compares the accumulated amount and a threshold. If the accumulated amount is larger than the threshold, it is counted as an onset and the accumulated value is reset. Otherwise, only the accumulated amount is reset. Though it is not very sensitive for detecting note onsets in legato passages, it is sufficient for detecting overall onsets in the music.

To calculate the auditory roughness, we use a DSP library for fast offline FFT processing [3] and detect 50 peaks from the local maxima of the magnitude response. We then calculate the auditory roughness from every pair of the peaks and add them all.

The result of these three features are all normalized and scaled to the size of HTML canvas. Since the auditory roughness tends to be somewhat correlated with the volume, we make it up by dividing the auditory roughness by the volume with a constant value. This compensation can emphasize the dissonance in quiet passages.

4.2 Visual Mapping

We visualize the features using a 2-D graph with a single continuous curve on an HTML canvas. The x-axis represents time and its width is fixed regardless of the length of input files. The y-axis represents the volume curve along with two color mappings. We downsample the features such that a set of values are mapped to a pixel by averaging. Onset density is mapped to the color of vertical lines below the volume curve. Lines with high onset density are colored with high saturated red on HSV scale. Auditory roughness is mapped to the color of vertical lines above the volume curve. Lines with high auditory roughness are colored with bright clear blue on RGB scale and so those with low auditory roughness is with dark dim blue.

4.3 Output

Once the visualization is generated and shown on the screen, users can freely navigate the music through clicking on the visualization. It takes a mouse input and changes the playing offset of the music immediately. A progress bar shows the current playing position. Users can make a music flowgram of a very complicate contemporary orchestral work, for example, Salonens Violin Concerto, which is about 29 minutes long as shown in Figure 3.

[2] http://jdasam.github.io/visualization/main.html
[3] https://github.com/corbanbrook/dsp.js/

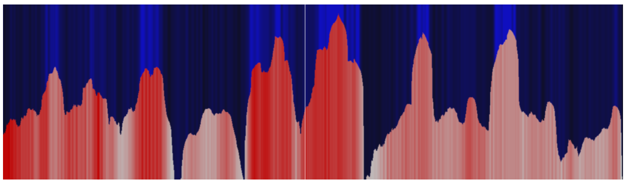


Figure 3. Music flowgram of Salonen's Violin Concerto

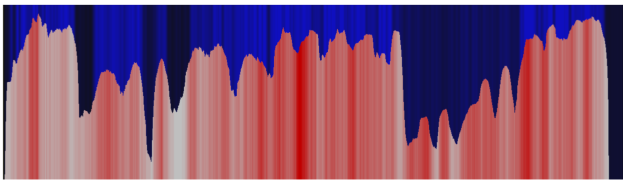


Figure 4. Music flowgram of Sibelius Finlandia

5. EVALUATION

To evaluate the effectiveness of music flowgram, we set up a user test for classical music listeners. The test consists of two scenarios that imitate situations where they listen to music on YouTube. The first case is listening to music without any video, which contain a static image or slide show of images such as an album cover or a picture of composer on YouTube. The second case is searching a specific part of a video when a short extracted audio clip is given.

The participants were 15 undergraduate students from Korea Advanced Institute of Science and Technology. All of them were a member of amateur orchestra, with a regular experience of listening to orchestral music. We divided them into two groups.

5.1 User Test A: Listening and Recall

In our previous research [5], we found that those who listen to the music with a volume graph can identify an extracted audio clip better than those who listened to the same music without it. In this test, we extend it to a more active recall task. That is, we evaluate how much music flowgram will help a listener to concentrate on and memorize the music. After listening to a movement from a symphony, participants are asked to describe how music changed over time in an objective expression. This task requires much more accurate musical memory compared to previous research experiment, because participants need to recall the music without any audio cue.

To design the test environment to be more realistic, we used YouTube as a listening interface. We uploaded two videos for each music, one with a music flowgram and the other with an image of album cover [4] .

Participants listened to three music pieces and described them in three different situations: with an album cover, a music flowgram only while listening, and a music flowgram while listening and writing. Writing a note during the listening was not allowed. That is, participants had to describe music only with their memory or the music flowgram. The music for the tests were selected from rarely

[4] https://www.youtube.com/playlist?list=PLq7cRTjnYEi6fvwS3fSY1PsEDpNQdVRXA

Song (length)	RK1-1 (9:04)	RK1-4 (6:46)	B1-4 (7:18)
Group A	Album cover	Music flowgram (MF)	MF, maintained after the listening
Group B	Music flowgram	MF, maintained after the listening	Album cover

Table 1. The setting of the user test A

performed repertoire so that none of the participants had possibly listened to the piece before. In addition, we selected music with similar style to reduce the effect of difference between the music materials. Selected materials are Rimsky-Korsakovs Symphony No. 1, first movement (RK1-1) and fourth movement (RK1-4), and Borodins Symphony No. 1, fourth movement (B1-4). The detailed setting is shown in the Table 1.

To properly guide participants response, we provided an example paragraph like below:

The piece starts with fanfare of trumpets. Then, cello plays quiet and smooth theme. Some variations of the theme are followed, and the dynamics get stronger. Then orchestra tutti play the main theme in faster tempo. Flute plays fast and virtuosic solo passages. The same orchestra tutti is followed. The key is transposed from major to minor, and woodwinds play march-like melody. This melody developed further by brass and violin in fortissimo. The fanfare from the beginning reappears in a more splendid way. Main theme is played again by cello. The orchestra tutti in the middle appears again with additional coda, which finishes the music. (Dvořák Symphony No. 8, fourth movement, translated into English by the first author)

After the listening and recall test, participants are asked to score each visually represented feature, based on how well it represents the musical characteristics. Also, participants were asked about how the listening experience with the music flowgram was different from that without it.

5.2 User Test B: Searching an Excerpt

The second test was searching an excerpt of the music in the YouTube video for the purpose of validating that a music flowgram can help a listener to find a specific part more quickly.

We rendered a music flowgram from downloaded YouTube videos, and attached it below the YouTube player, as shown in Figure 5. The music flowgram image is linked to YouTube video so that users can select the playing position by clicking a specific position on the graph. We compared this setting to a YouTube player only page that contains the same video. Since the difficulty of searching task is largely influenced by a characteristic of the excerpt, we chose excerpts from movie scenes, rather than choosing arbitrarily. The selected movie is *Lorenzos Oil* and *Birdman*, which used a short clip from the third movement of Mahlers symphony No. 5 (M5-3), and the second movement of Rachmaninovs symphony No. 2 (R2-2), respectively.

Each participant watched the movie clips and searched the excerpted music parts on the YouTube video. There were two movies and a corresponding web page including YouTube videos of the music used in the movie clips. One of the web page included a YouTube player with a music flowgram of the audio of the selected video, and the other page only included a YouTube player. We arranged a different setting for each group, as described in the Table 2.

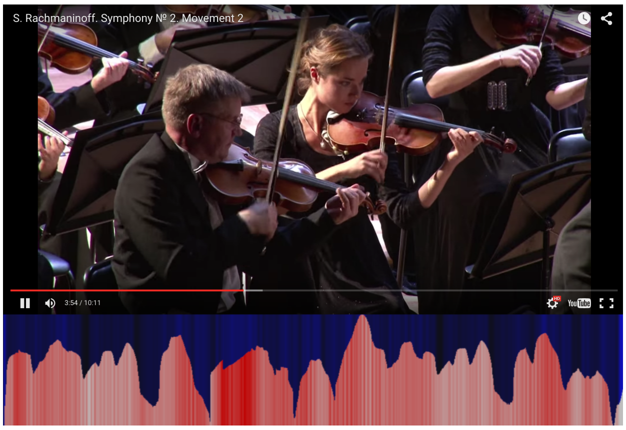


Figure 5. An example of the proposed system

Song (length)	M5-3 (18:05)	R2-2 (10:11)
Group A	YouTube with MF	YouTube only
Group B	YouTube only	YouTube with MF

Table 2. The setting of user test B

6. RESULT AND DISCUSSION

6.1 A. Listening and Recall

We checked the descriptions written by participants and ma-tched the description to the corresponding part in the music. Any phrase or sub-phrase that can specify a rehearsal letter from music was counted as a correct answer. Considering the participants are all non-professional musician and the description was written after only one listening attempt, we allowed a certain level of wrong descriptions, for example, incorrect instrument and melody identification. Many participants had confusions whether the melody was reoccurred or newly introduced, and whether the solo instrument was flute or oboe. But the confusion between string and wind instrument was not allowed. The paragraph below is an example of participants' answer, which is translated into English by the first author. Because Korean does not usually use a definite or indefinite article, melody is translated without an article. A letter in a parenthesis is annotation made by authors, which means a rehearsal letter of corresponding part in the score.

Music starts with brass and bass(A). Main theme starts with bass and cellos, violas, violins takes over the theme and the pitch register goes higher(B). All the instruments play main theme in fortissimo(C). Clarinet plays melody and string plays melody(F). Then, the brass section is added and play majestic chord(G). The same pattern is repeated (Repetition). The flute plays melody and similar pattern is played(I). At the last part, flute and oboe appear(T). After majestic brass, timpani finish music (U).

Song	RK1-1	RK1-4	B1-4
Group A (n=8)	6.75 (SD: 1.49)	6.88 (SD: 2.23)	8.38 (SD:2.45)
Group B (n=7)	9.14 (SD: 1.07)	6.86 (SD: 1.68)	6.71 SD: 1.98)

Table 3. Average score and standard deviation of correct description in user test A

By this criteria, we scored how many parts of the music is described in each description. The rehearsal letters are referenced from an edition of Muzgiz, Soviet State music publishing house, for both symphonies. The data analysis was in blinded name to avoid bias.

The result on the Table 3 shows that the group with the music flowgram recalled more parts of the music than group with the album cover image, regardless of whether the music flowgram was provided until the end of the writing step. This result shows that the music flowgram can help a listener to memorize music more precisely as we expected.

There was almost no difference between two groups in the case of fourth movement of Rimsky-Korsakovs symphony, for which the only difference was the presence of the music flowgram during the writing step. From this result, we infer that the music flowgram was easy to remember and recall, so that there was almost no disadvantage of not watching it again while writing the description.

During the analysis, we found that the ratio of participants who mentioned the repetition of the piece was higher in the group with the music flowgram. Both the first movement of Rimsky-Korsakovs Symphony No. 1 and the last movement of Borodins Symphony No. 1 are in sonata-allegro form that includes a repetition of the exposition. Four out of seven participants of group B mentioned the repetition of the first movement of Rimsky-Korsakovs symphony, while only one out of eight participants mentioned in the other group. In the case of Borodins symphony, six out eight participants mentioned it in the group A, while two out of seven in the group B mentioned it.

Recognizing the repetition is important for understanding a structure of music. Most of musical forms in classical music include repetition of main part. This is one of the reasons why the former research about music structure focused on repetitive structure. The repetition of exposition is an important characteristic of a sonata-allegro form. This result also shows that our music flowgram is helpful for listening and understanding classical music.

Comments from participants, which is shown in the Table 4, also support the previous results. Many participants answered that the music flowgram was helpful for understanding the entire structure of music. There was only one participant who wrote negative comments only. Some of the comments contradicted each other; some participants answered that the listening was more interesting because he can anticipate, but the other answered that it was less interesting because it was easy to anticipate. Three out of fifteen participants answered that they preferred audio only listening environment. But they also commented that just taking a look at the visualization was helpful for understanding the entire structure of the music.

Positive comments	**It was helpful for...** understanding an entire structure of music. anticipating when will music get excited or relaxed. knowing length of each section. recognizing which part is repeating. memorizing change of dynamic. concentrating on music by comparing my own anticipation and the actual music
Negative comments	It was disturbing to concentrate on listening to the music. It made listening less interesting because it was easy to anticipate music. It made me keep thinking about how much the graph is accurate.

Table 4. Examples of comments on listening with music flowgram

Parameter	Average score (1 to 5)
Volume	4.6 (SD: 0.51)
Onset Density	3.4 (SD: 0.74)
Auditory roughness	1.9 (SD: 0.74)

Table 5. Evaluation for individual features by participants

Participants evaluation on each visual parameter is shown in the Table 5. Most of the participants gave the highest rate to loudness. The evaluation of the auditory roughness was not positive. This is partially because of its visual mapping to background color. Some of the participants pointed out that the change of background color was not easily recognizable.

6.2 B. Searching an Excerpt

The result of user test B is on Table 6. Because the task is largely dependent on participants pre-knowledge about the music material, we put answers from participants who know the material well separately in Table 7.

For both pieces, it took less time for the group with music flowgram to find the target part as we expected. In the case of Mahlers symphony, the excerpt was in the very last part of the music, which contains a loud brass section. So it was a tough task to find it by navigating from the beginning of the music. But when the music flowgram was provided, the participants could only concentrate on the loud part. This advantage greatly reduced the searching time, especially for the participants who do not know the piece very well.

On the other hand, the excerpt form Rachmaninovs symphony is in an early part of the music. So many participants can easily find the part by navigating without music flowgram. This is the reason why the difference between the two group is slight. But it is worth mentioning that the shortest record was only 6 seconds, which included the

Song	M5-3	R2-2
Group A (n=6)	170 seconds (SD: 138)	59 seconds (SD: 32.9)
Group B (n=6)	377 seconds (SD: 153)	40 seconds (SD: 29.6)

Table 6. Results of user test B (average of consumed time)

Song	M5-3	R2-2
Group A (n=2)	14 seconds 43 seconds	15 seconds 21 seconds
Group B (n=1)	85 seconds	6 seconds

Table 7. Results of user test B with participants who know well the material

loading time for the YouTube player. One of the participants found the excerpted part with a single click on the music flowgram. Since the excerpt contains legato passage of strings, the participant could easily find it by searching a part with low onset density.

7. CONCLUSION

In this paper, we have presented an automatic visualization method for representing music in its entirety. The goal of our visualization is showing how the music changes from beginning to end. The method visualizes music with three audio features like volume, onset density, and auditory roughness, which are highly associated with loudness, tempo, and dissonance, respectively, in musical characteristics. These features are visualized as a two-dimensional graph. We implemented the method on a web page using Web Audio API and conducted user test for verifying the usefulness of our method in the listening and searching task. The results showed that listening to music with a music flowgram helps listeners to memorize the music more precisely. A music flowgram was also helpful for searching a specific excerpt from music.

Despite of the overall positive results, there is still a large margin for improvement. Auditory roughness, which is intended for representing the harmonic characteristic of music, was not satisfactory for many participants. For the future work, we are planning to improve our algorithm for detecting onset and calculating audio roughness. We are also considering other audio features that can replace auditory roughness such as tonal complexity [18]. Another important challenge will be finding more intuitive and visually pleasing mappings for each parameter.

8. REFERENCES

[1] M. Müller, F. Kurth, and T. Röder, "Towards an efficient algorithm for automatic score-to-audio synchronization." in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, 2004.

[2] M. Wattenberg, "Arc diagrams: Visualizing structure in strings," in *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*. IEEE, 2002, pp. 110–116.

[3] J. Foote, "Visualizing music and audio using self-similarity," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. ACM, 1999, pp. 77–80.

[4] M. Müller and N. Jiang, "A scape plot representation for visualizing repetitive structures of music recordings." in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*. Citeseer, 2012, pp. 97–102.

[5] D. Jeong and Y. Noh, "Music visualization using volume graph and its effect on classical music lstening," 2014.

[6] D. Jeong, "Music visualization using flow graph and its effect on listening to classical music," Master's thesis, Korea Advanced Institute of Science and Technology, 2015.

[7] N. Kosugi, "Misual: music visualization based on acoustic data," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*. ACM, 2010, pp. 609–616.

[8] M. Goto, "Smartmusickiosk: Music listening station with chorus-search function," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*. ACM, 2003, pp. 31–40.

[9] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions." in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. Citeseer, 2011, pp. 311–316.

[10] C. S. Sapp, "Harmonic visualizations of tonal music," in *Proceedings of the International Computer Music Conference*, vol. 1001. Citeseer, 2001, pp. 423–430.

[11] M. Malt and E. Jourdan, "Le ≪bstd≫–une representation graphique de la brillance et de lecart type spectral, comme possible representation de levolution du timbre sonore." in *Proceedings of L'analyse musicale aujourd'hui. Crise ou (r)/'evolution ?*, 2009.

[12] W.-Y. Chan, H. Qu, and W.-H. Mak, "Visualizing the semantic structure in classical music works," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 1, pp. 161–173, 2010.

[13] S.-L. Tan and M. E. Kelly, "Graphic representations of short musical compositions," *Psychology of Music*, vol. 32, no. 2, pp. 191–212, 2004.

[14] G. Freytag, *Die Technik des Dramas*. Hirzel, 1872.

[15] A. Gabrielsson and E. Lindström, "The role of structure in the musical expression of emotions," *Handbook of music and emotion: Theory, research, applications*, pp. 367–400, 2010.

[16] E. J. Humphrey and J. P. Bello, "Four timely insights on automatic chord estimation," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[17] P. N. Vassilakis and K. Fitz, "Sra: A web-based research tool for spectral and roughness analysis of sound signals," in *Proceedings of the 4th Sound and Music Computing (SMC) Conference*, 2007, pp. 319–325.

[18] C. Weiss and M. Muller, "Tonal complexity features for style classification of classical music," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 688–692.

SWARALIPI: A FRAMEWORK FOR TRANSCRIBING AND RENDERING INDIC MUSIC SHEET

Chandan Misra
Indian Institute of Technology Kharagpur
chandan.misra1@gmail.com

Tuhin Chakraborty
Indian Institute of Technology Kharagpur
tuhin.babai@gmail.com

Anupam Basu
Indian Institute of Technology Kharagpur
anupambas@gmail.com

Baidurya Bhattacharya
Indian Institute of Technology Kharagpur
baidurya@civil.iitkgp.ernet.in

ABSTRACT

Creating music in computer system through its music notations requires two primary components. The first one is the mechanisms to encode music notations of respective music genres and the other one is a framework to provide the look and feel of the music written like a published or handwritten music sheet. Popular music scorewriters like Finale, Sibelius, MuseScore can edit, render and playback music transcribed in Staff notation. Being vastly different from the Indic music system in grammar, notation symbols, tonic system and encoding style, the architecture used in the music software for western music cannot cater to the Indic music system. For this reason there is a dearth of such scorewriters for Indic music system which is rich with a variety of musical genres, each different from the others in their unique notation system and language for depicting their lyric. In this paper, we propose a new framework for transcribing and rendering Indic music sheets for different genres of Indic music in computer. This framework is designed to support all major Indic notation systems and Indic language scripts and is explained using three major notation systems and language scripts throughout the paper as a case study.

1. INTRODUCTION

Music notation or musical notation is defined as a system for representing music in written glyphs or characters by encoding its pitch, duration, rhythm, lyric and ornaments. Notation systems have helped in the preservation of musical compositions through the ages and also in spreading them accurately among cultural systems and traditions.

Music and its perception have evolved in a variety of ways over the years in different regions of the world. Hence with change in tradition and culture, the music notations vary, from one country to another. Among others Staff notation is the most popular modern music notation system which was originated in the European classical Music.

Another musical genre of importance in the eastern part of the world is — the music of India. The tradition of the classical genre of *Indic music* has existed for almost a millennium. Its origin can be traced back to Samaveda, the sacred Hindu Mythology [1]. Indic music comprises of myriad varieties of music. Carnatic (South Indian Classical Music) and Hindustani (North Indian Classical Music) Sangeet are two richest varieties, besides the other varieties of *folk*, *Baul*, *Bhajan*, *Rabindrasangeet*, *Thumri*, *Ghazal*, popular or Filmi and pop. However, it is remarkable to note that despite global acceptance of Staff notation, in India, teaching, learning and composing Indic music (particularly classical) are carried out in Indic notation systems even today.

Indic music notation systems have received their present forms from the contribution of three stalwarts - Pt. Vishnu Narayan Bhatkhande, Pt. Vishnu Digambar Paluskar and Rabindranath Tagore. While, notation systems for Hindustani music were introduced by Bhatkhande and Paluskar, Rabindranath Tagore introduced a new musical genre called Rabindrasangeet. Jyotirindranath Tagore, elder brother of Rabindranath Tagore, created *Akarmatrik* notation system for encoding Rabindrasangeet in the year 1905 [2].

Various systems have been developed for displaying and rendering Staff notation in computers. Finale from MakeMusic [3], Sibelius from Avid Technology [4], and MuseScore are some of the leading music notation software. These are used to arrange, notate, display and print engraver-quality sheet music in Staff notation. On the other hand, we know of only one such software, Swar Shala by Swar Systems [5] , for Indic music. However, it lacked in the features required to arrange Indic notation symbols to create music. This absence of a proper system for encoding, composing and preserving Indic music has deprived Indic music lovers and composers from participating in computerized music creation in an environment entirely unique to Indic music. One of the main difficulties of creating such software is the diversity of grammatical structure present in various Indic notation systems. Moreover, the architecture for Staff notation cannot be employed for transcribing Indic music notation systems due to the differences in their notation arrangements. For example, it does not have any bar line; notations are entirely different and depends on Indic language scripts. Hence, there lies a need to develop a robust framework or architecture for arranging and presenting musical components in computer similar to published music sheet, for Indic music notation system.

2. PROPOSED ARCHITECTURE

The proposed architecture helps to encode, arrange, display and render Indic music symbols in computer and in order to do that, we have examined the structure of different music sheets written in popular Indic music notation systems. To demonstrate the test cases we have considered three of the main notation systems presently active in India - *Bhatkhande*, *Paluskar* and Akarmatrik notation system.

Figure 1, 1 and 2 show instances of music sheet written in Bhatkhande, Paluskar and Akarmatrik notation system respectively taken from [6, 7].

Figure 1. (a) Music sheet written in Bhatkhande system taken from [6] and (b) Paluskar system taken from [7]

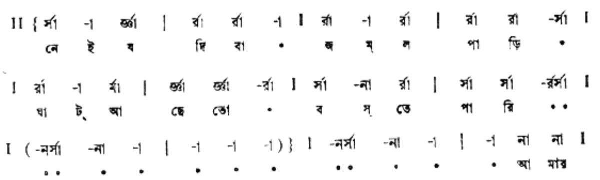


Figure 2. Part of Rabindrasangeet piece "Amar Naiba Holo Paare Jawa" in Akarmatrik Notation System taken from [8]

Minute observations on these music sheets have revealed that each one of them possesses a rectangular structure having a number of rows and columns and some notation symbol/s at each of the intersection of the rows and columns. Based on this similarity a matrix model has been proposed for representing the structure of the music sheet. The detail architecture has been described next.

2.1 The Matrix Model

In order to build the framework, each music piece is converted into a collection of strings of 2-D matrices as shown in Fig. 3. As each line consists of several rows, each line represents a single 2-D matrix and the number of 2-D matrices is equal to the number of lines in the music piece. Due to the fact that each line of a music piece is not necessarily of the same length, the size of all the 2-D matrices are not equal. Each cell thus produced can store either Unicode character(s) or character(s)/glyph(s) of true type or other format necessary to define the notation systems in computer software. The inputs for this model are Taala, Avartana, number of lines of the music piece and position of the notation symbols.

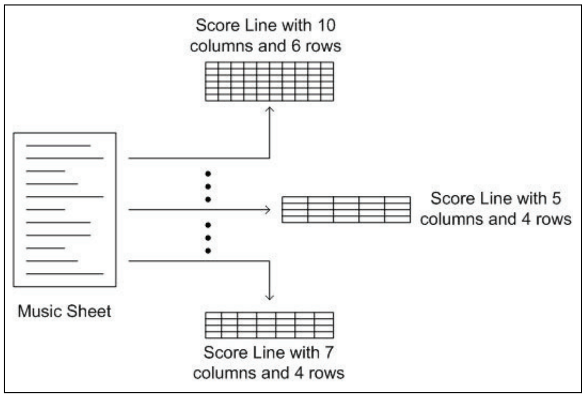


Figure 3. The Visualization of the Matrix Model

For simplicity the entire matrix model has been divided into two sub-models — the row model and the column model. Row model defines the number of rows of the model, the number of rows in each 2-D matrix and the alignment between them. Likewise, column model defines the number of columns for each 2-D matrix. It means that each line of the music piece forms a 2-D matrix and consequently the whole music piece presents a three dimensional rendering. In any 2-D matrix, the number of rows can be selected by the user to encode all necessary features of the music (main notes, lyrics, repetition notes, beat markings, *Meend* etc.). Likewise, the number of columns can be selected depending on the Taala, Avartana and need of initial or terminal phrasings. Each cell contains none, one or more musical glyphs and/or characters.

Figure 4 shows some of the typical lines of an Indic music sheet, specifically Rabindrasangeet. It is scored in the Akarmatrik system capable of depicting Swaras or notes, Sparsha Swaras or grace notes, Maatras or beats, Taalanka or beat markings, equal or unequal Taala Bibhagas or measures, Meend or glide between two notes, Shrutis or microtones, repetition phrase, melody changes in repetition, lyric etc. Some but not all of the musical components mentioned above can be described in Staff notation system. The components which cannot be realized in Staff system are — unequal measures in a Taala, the cyclic nature of

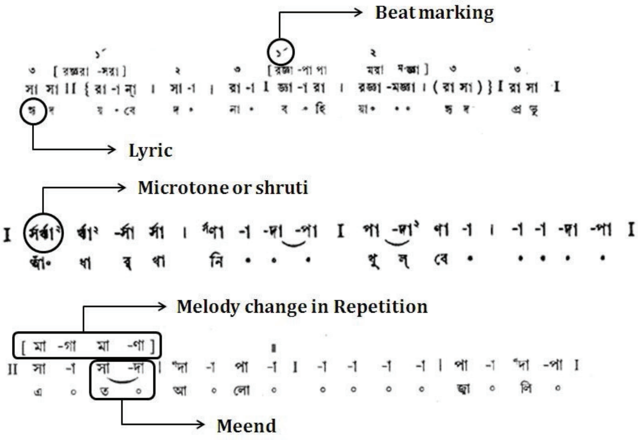


Figure 4. Major Components in Akarmatrik Notation System

Taalas, milestones on the Taala cycle and Shrutis. An appropriate Indic music notation system can describe all the components contained in Indic classical or regional music. To edit and render this kind of music electronically, a new architecture was needed and this is where the present work comes in. We shall present the model with respect to the Rabindrasangeet music sheet as Akarmatrik notation was found to be the most robust and well-content notation system for presenting different components present in Indic music. Then we shall present justifications on other notation systems and music genres as well.

2.1.1 The Row Model

2.1.1.1 The Number of Rows Per Line

It is named row model as it determines the number of 2-D matrices and the number of rows in each line present in the model of a particular music sheet. In the simplest case, a line of a music piece written in any Indic music notation system can have two rows —one for notes and the other for the lyric. But almost always, four rows are required —the lyric row (Fig. 4), the Meend row or the row for accommodating the Meend symbol shown in Fig. 4, the note row or the row for placing notes (Fig. 4) and the Taalanka or beat marking row (Fig. 4), as schematized in the bottom half of Fig. 5. However, if a segment needs to be repeated with a variation, it needs to be shown above the primary melody segment, as shown in the first four rows of Fig. 5. These lines can accommodate other symbols, such as the end of piece symbol as shown in Fig. 4. So altogether each music piece will need at most eight rows —four bottom rows for primary melody lines and four top rows for notes with melody variations.

2.1.1.2 The Meend Symbol

Meend (shown in Fig. 8) is one of the musical ornaments present in Indic music which is an Indic counterpart of Portamento or Glissando [9]. In Meend, one note slides to another note of different pitch over a specified number of beats. It is obvious that the Meend symbol cannot be entered entirely into one cell of the 2-D matrix because two

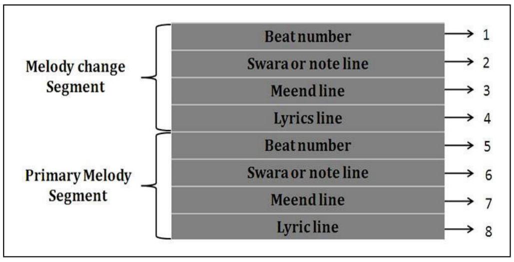


Figure 5. A 2D matrix depicting one line of an Indic music piece, its components and associated row number. The accordance of the components changes when notation system changes

notes between which the Meend slides will be in two different cells of the 2-D matrix. So, the entire symbol is divided into three symbols 1, 2 and 3 as shown in Fig. 6 and can now be entered into the framework. For Meend, we have a dedicated row both for primary and changed melody line of the music piece.

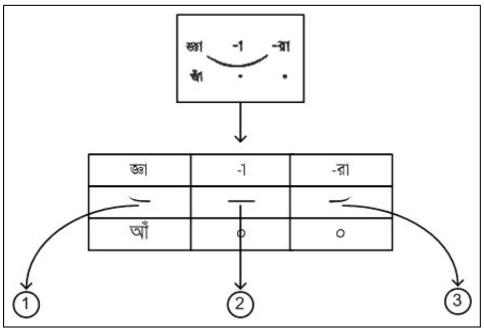


Figure 6. Implementation of Meend symbol in the present architecture

For example, in Fig. 6, Meend starts from 1 (Komal Gandhara), slides through 2 (same as previous note) and ends at 3 (Suddha Rishabha). First part of the Meend symbol will be placed above or below the cell from which the Meend slides. Likewise, the last part of the symbol will be placed above or below the cell to which the Meend slides. If there are multiple notes in between start and end note then the second symbol will be placed above or below every cell of those notes. In Akarmatrik system the Meend curve is drawn below the notes while it is drawn above the notes in Bhatkhande and Paluskar system.

2.1.2 The Column Model

The number of columns of the 2D matrix depends upon the following factors:

1. Taala, i.e., the cyclic beat pattern which specifies the number of Bibhagas (i.e., measures) per cycle, the name of each measure, number of beats in each measure (not necessarily equal, as shown in Table 1)

2. Avartana i.e. the number of cycles per line.

Table 1 shows some of the common Taalas used in Indian classical music and Rabindrasangeet.

Table 1. Some common Taalas used in Indic music system along with their total number of beats and beat pattern

Taala Name	Maatra or Total Number of Beats	Taala Bibhaga or Measures
Dadra	6	3+3
Shashthi	6	2+4 or 4+2
Rupak	7	3+2+2
Kaharba	8	4+4
Jhaanp	10	2+3+2+3
Ektaala	16	3+3+3+3 or 4+4+4
Tritaala	16	4+4+4+4

According to Table 1 Tritaala has one cycle of four measures of four beats each. Not all Taalas have equal number of beats per measure. Each repeated cycle of a Taala is called an Avartana. These two features form the building block of the column model of the architecture.

Figure 7 shows a matrix of a typical line of a music piece of Akarmatrik notation system having Shashthi Taala with two unequal measures having 2 beats in the first measures and 4 beats in the last, 2 Avartanas and the lyric line is written in Bengali script, have been transformed into the architecture having 2 rows and 17 columns.

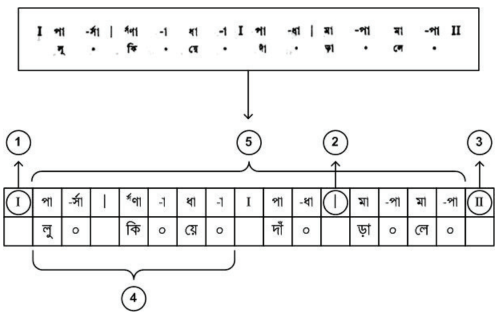


Figure 7. Determination of column number of a line of Rabindrasangeet score in the present architecture

After the empty matrix is constructed, it is populated by music symbols. Each matrix contains cells. The number of cells in each 2-D matrix is the product of the number of rows and columns. A cell may contain no symbol, one symbol, or more than one symbol. Symbol 1 (I) in Fig. 7 signifies the beginning of the Taala and thus occupies a cell before the first beat of the Taala. It is repeated once every cycle of the Taala. Symbol 3 (II) in Fig. 7 signifies the end of a musical phrase after which the first phrase (Aasthayee) must be sung; it also occupies a cell by itself. Symbol 2 (|) in Fig. 7 is the Taala Bibhaga symbol which comes between adjacent measures of the Taala and occupies a cell by itself. Figure 7 shows the Shashthi Taala which has two unequal measures of 2 and 4 beats (part 4 of the Fig. 7) respectively. The shown musical line starts with Symbol 1, shows two repetitions (Avartanas — part 5 of the Fig. 7) of the Taala (separated by symbol 1), and ends with Symbol 3. Because there are only two measures per cycle of this Taala, symbol 2 is used only once per cycle in Fig. 7. In general, the number of columns required to store symbol 2 can be found by the following equation:

$$n = (a \times m) \quad (1)$$

$$t = (b \times a) + (a - 1) + n + 2 \quad (2)$$

where n is number of columns for symbol 2, a is number of Avartana, m is the number of measures of the Taala, t is the total number of column of the matrix and m is the Maatra or total number of beats in the Taala.

For Fig. 7 the number of columns is therefore (6 x 2) +1+ (2 x 1) +2 = 17.

2.2 The Notation Font

While creating the fonts to be used for the implementation of the architecture, we have considered two components — the notation system and the language used for transcription. The task for designing fonts are merely drawing each glyph/character with the help of font creating software and mapping them to a key. But the real challenge is to explore the available notation systems in India and listing down each of their characters, mining intra and inter notational similarities and the language script they use. Our exploration brought into light one useful information — the connection between language script and notation system. We have used this interrelation to design the set of fonts for Indic music. To visualize the inter-relation, first we will give a brief introduction on different notation systems of Indic music. The main objective of this section is not to demonstrate the symbols of various notation systems, but to focus on extracting their common behaviour and gather this knowledge to conclude a common system. This common knowledge has been used to facilitate the guidelines for designing of notation fonts.

2.2.1 Main music annotating styles of Indic Music

As mentioned previously, three major notation systems used in India are Bhatkhande, Paluskar and Akarmatrik. Music sheets written in one of these notation systems may use different Indic language scripts to write the lyric and which in turn describe certain musical symbols of that particular notation system. In this paper we use the term *language base*, elaborated later, to define these language scripts. For a particular language base, we have found some inter and intra notation similarities among these three notation systems.

Figure 8 shows the notation symbols when the language base is Bengali. The properties are written in the leftmost column and their English counter name is given for better understanding.

2.2.2 Intra and Inter notational system similarities

The second, third and fourth column of Fig. 8 describe the symbols of the main properties of Akarmatrik, Bhatkhande and Paluskar system. We have found some similarities between them. They are

1. Pure notes
2. Akarmatrik lower octave and Paluskar flat notes.

Notation Properties	Akarmatrik System	Bhatkhande System	Paluskar System
Pure Notes	স,র,গ,ম,প,ধ,ন	সা,রে,গ,ম,প,ধ,নি	সা,রে,গ,ম,প,ধ,নি
Flat Notes	ঋ,জ্ঞ,দ,ণ	(_)রে,গ,ধ,নি	(ˎ) স্,র্,গ্,ম্,প্,ধ্,ন্
Sharp Notes	ক্ষ	(ˈ)ম	ম্র
Middle Octave	স,র,গ,ম,প,ধ,ন	সা,রে,গ,ম,প,ধ,নি	সা,রে,গ,ম,প,ধ,নি
Lower Octave	(ˎ) স্,র্,গ্,ম্,প্,ধ্,ন্	(.)সা,রে,গ,ম,প,ধ,নি	(˙)সা,রে,গ,ম,প,ধ,নি
Upper Octave	(´) র্স,র্র,র্গ,র্ম,র্প,র্ধ,র্ন	(˙)সা,রে,গ,ম,প,ধ,নি	(ˈ)সা,রে,গ,ম,প,ধ,নি
Single Beat	(।) সা,রা,গা,মা,পা,ধা,না	সা,রে,গ,ম,প,ধ,নি	রে,গ,ম,প,ধ,নি
Half (1/2) Beat	(ঃ) সঃ,রঃ,গঃ,মঃ,পঃ,ধঃ,নঃ	গম	গ ম
Querter (1/4) Beat	(॰)স॰,র॰,গ॰,ম॰,প॰,ধ॰,ন॰	গমপধ	গ ম প ধ
Beat Start	১	×	১
Beat Gap	০	০	+
Division	│	│	│
Beat Mark/Beat Number	২,৩,৪	২,৩,৪	১,৫,৯,১৩
Meend	‿	⁀	⁀
Kan	ধপ,রস,গর	ধপ,রেসা,গরে	ধপ,রেসা,গরে
Khatka		(সা)	(সা)
Enunciation	০	s	-

Figure 8. Notation symbols of Akarmatrik, Bhatkhande and Paluskar notation systems with their properties when the language base is Bengali

3. Bhatkhande upper octave and Paluskar lower octave.
4. Bhatkhande and Paluskar Meend (Portamento).
5. Bhatkhande and Paluskar khatka.
6. Akarmatrik and Paluskar beat start.
7. Kan swaras of all systems
8. Akarmatrik and Bhatkhande beat gap.
9. Bhatkhande flat notes and Paluskar single beat.
10. Bhatkhande sharp note and Paluskar upper octave

The similarities are not only bound in different systems, but similarities do exist in same notation system. The similar symbols or characters are given below

1. Pure, sharp and flat notes and their respective Kan or sparsha swaras in all notation systems.
2. Beat gap and enunciation in Akarmatrik system.
3. Pure, sharp and flat notes and their single beat representation in Bhatkhande system.

The above similarities can be visualized in the following simplified Venn diagram shown in Fig. 9. Here each circle denotes a single notation system. Area denoted by number 1, 2, 3 and 4 describe inter-notational system similarities. They overlap each other in certain cases. There are similarities separately in two different notation systems and some symbols are common in all three notation systems.

Unlike Western Staff notation, Indic music notation systems depend on the languages used to write the lyric of the

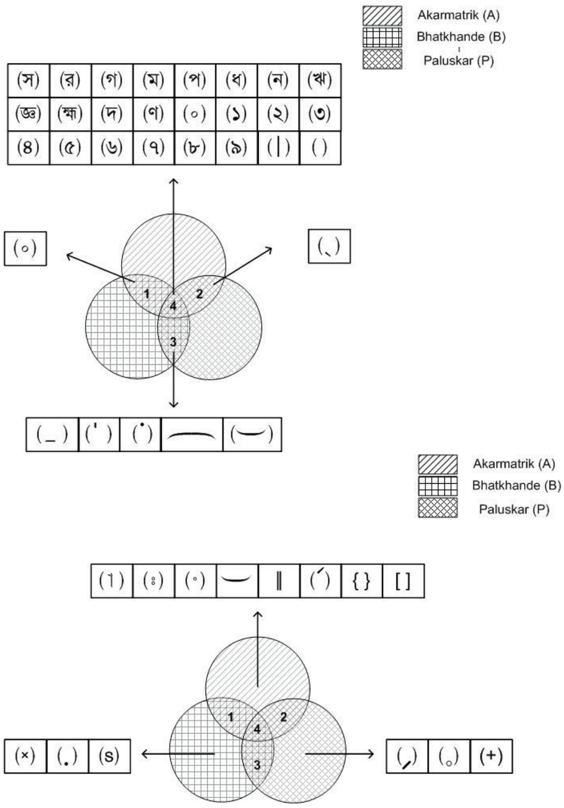


Figure 9. Venn diagram showing inter and intra-notational system similarities and similarities between notation system and Bengali language script present in Akarmatrik, Bhatkhande and Paluskar notation system

Notation System	(Pure Note)-(Unicode Value)-(Pronunciation)						
Akarmatrik	(স)-(09B8)-(*Sa*)	(র)-(09B0)-(R*a*)	(গ)-(0997)-(G*a*)	(ম)-(09AE)-(M*a*)	(প)-(09AA)-(P*a*)	(ধ)-(09A7)-(Dh*a*)	(ন)-(09A8)-(N*a*)
Bhatkhande	(সা)-(09B8+09CD+09BE)-(Sa*a*)	(রে)-(09B0+09CD+09C7)-(Re)	(গ)-(0997)-(G*a*)	(ম)-(09AE)-(M*a*)	(প)-(09AA)-(P*a*)	(ধ)-(09A7)-(Dh*a*)	(নি)-(09A8+09CD+09BF)-(*Ni*)
Paluskar	(সা)-(09B8+09CD+09BE)-(Sa*a*)	(রি)-(09B0+09CD+09BF)-(Ri)	(গ)-(0997)-(G*a*)	(ম)-(09AE)-(M*a*)	(প)-(09AA)-(P*a*)	(ধ)-(09A7)-(Dh*a*)	(নি)-(09A8+09CD+09BF)-(*Ni*)

Figure 10. The Mapping between the glyphs of pure notes of Akarmatrik, Bhatkhande and Paluskar notation system, their Unicode values and respective pronunciations

music piece because, certain music symbols of Indic music are similar to the characters or glyphs of Indic scripts. Observation reveals that the central part (4) in Fig. 9 shows the similarities among all three notation systems which actually describe the similarities between notation symbols and Indic scripts. This common similarity is constant because when the language base changes the same number of similarities can be found between the notation system and the changed language base. We shall describe this constant part with Bengali Indic script next.

2.2.3 Similarities between notation systems and Indic scripts

We have found some similarities among the music notation symbols with Bengali script as shown in Fig. 8. The similar symbols are given below

1. Pure notes and Bengali letters *Sa, Ra, Ga, Ma, Pa, Dha, Na*. There are various pronunciation styles of the main or pure notes. For example — in Akarmatrik system the second and seventh pure note is pronounced or written as *Ra* and *Na*, while in Bhatkhande system it is pronounced or written as *Re* and *Ni*. Figure 10 depicts the mapping between glyphs of the pure notes of three notation systems, their Unicode values and respective pronunciation.

2. Flat and sharp notes and Bengali compound letters

3. Akarmatrik Lower octave and Bengali Hasant symbol (U09CD)

4. Akarmatrik Upper octave and Bengali Ref symbol (proposed in [10])

5. Akarmatrik Single beat and Bengali Akaar symbol (U09BE)

6. Akarmatrik Half beat and Bengali Bisargha (U0983)

7. Akarmatrik Beat marks and Bengali numerical digits (U09E6-U09EF)

2.2.4 Effects of Indic languages on Indic music notation systems

2.2.4.1 Language Base

Music has its own language. But songs which constitute music scores with lyric need additional natural languages. In India different genres of music were born from different places and cultures and they are still in practice majorly in their areas of origin. Every region in India with their unique cultural traditions and languages have influenced their respective notation systems. More particularly, the natural language of the lyric part of the music piece determines the most of the notation symbols. Western music pieces contains no similarity between notes and lyric because they are transcribed in Staff notation system. On the contrary, scores written in Indic music notations have certain characters that resemble with the lyric language. We will call this particular language the *language base* of the corresponding music piece. Some Indic script characters pronounce like pure notes when they are sung. These characters are shown in Fig. 11.

স র গ ম প ধ ন

Figure 11. Seven Bengali script characters pronounced similar with seven pure notes or *Suddha Swaras*

They are pronounced as *Sa, Re, Ga, Ma, Pa, Dha, Na* and the seven pure notes, when sung, produce exactly similar sound. If they are written in Hindi language it will be written Devanagari script and look like Fig. 12. The same is

स र ग म प ध न

Figure 12. Seven Devanagari script characters pronounced similar with seven pure notes or *Suddha Swaras*

true for grace notes and flat and sharp notes. Grace notes or Kan Swaras are similar but smaller in size as pure notes

Notation Properties	Akarmatrik System	Bhatkhande System	Paluskar System
Pure Notes	स,र,ग,म,प,ध,न	सा,रे,ग,म,प,ध,नि	सा,रे,ग,म,प,ध,नि
Flat Notes	ऋ,ज्ञ,द,ण	(_) रे,ग,ध,नि	(ॖ) सा,रे,ग,म,प,ध,नि
Sharp Notes	ह्म	(ॅ) मॅ	म्र
Middle Octave	स,र ,ग,म,प,ध,न	सा,रे,ग,म,प,ध,नि	सा,रे,ग,म,प,ध,नि
Lower Octave	(्) स्,र्,ग्,म्,प्,ध्,न्	(़) सा़,रे़,ग़,म़,प़,ध़,नि़	(ं) सां,रें,गं,मं,पं,धं,निं
Upper Octave	(´) र्स,र्र,र्ग,र्म,र्प,र्ध,र्न	(ं) सां,रें,गं,मं,पं,धं,निं	(ॅ) सॅा,रॅ,गॅ,मॅ,पॅ,धॅ,निॅ
Single Beat	(ा) सा,रा,गा,मा,पा,धा,ना	सा,रे,ग,म,प,ध,नि	रे,ग,म,प,ध,नि
Half (1/2) Beat	(ः) सः,रः,गः,मः,पः,धः,नः	गम	ग़ म़
Querter (1/4) Beat	(॰) स॰,र॰,ग॰,म॰,प॰,ध॰,न॰	गमपध	ग म प ध
Beat Start	१	×	१
Beat Gap	०	०	+
Division	|	|	|
Beat Mark/Beat Number	२,३,४	२,३,४	१,५,९,१३
Meend	‿	⁀	⁀
Kan	^ध^प	^ध^प	^ध^प
Khatka		(सा)	(सा)
Enunciation	०	ऽ	-

Figure 13. Notation symbols of Akarmatrik, Bhatkhande and Paluskar notation systems with their properties when the language base is Devanagari.

as described earlier in Fig. 8. To demonstrate the findings we have created another table as shown in Fig. 13.

2.2.4.2 Unchanged Musical Characters

Certain notations came from certain regions of India and that is why they got the language base of that region when they were invented. Every Indic script character has a counter part in another Indic script. For example, in Fig. 8 and 13 the seven pure note symbols are one to one mapping of Bengali to Devanagari script. As a result, when the language base changes, the similarities between the notation systems and the base language remains constant.

But there are certain characters which are present in one script and cannot be found in similar form in other scripts. For example the upper and lower octave sign in Akarmatrik notation system is Bengali Ref and Bengali hasant. Some other characters are there to denote the same thing in Devnagri language. If some music piece is written in Akarmatrik and the language base is Devanagari, these Bengali characters might be kept intact to denote the upper and lower octave (Nayar1989). That is why in Fig. 13 these symbols are same as in Bengali. These symbols get permanent musical symbol status and not dependent on language base.

We have developed the guidelines out of the findings stated above for designing the notation fonts for various language scripts. These guidelines will help font designers to create fonts and identify the number of symbols to be designed. The guidelines are as follows (Considering three notation systems as in Fig. 9) —

1. There will be a common notation system for each of the language base.
2. It will contain each symbol of the non-intersected area of three systems
3. It will contain one copy of symbols for each intersected area — part 1, 2, 3 in Fig. 9.
4. It will contain one copy of intersected area —part 4 in Fig. 9 corresponding to the language base.

According to the above guidelines, the full set of symbols/characters for designing Indic music notation font for language base Devanagari is shown in Fig. 14.

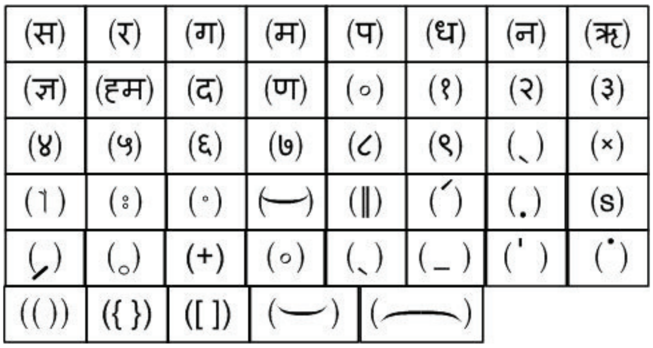


Figure 14. Indic music notation font set for Devanagari language base

2.2.5 Adaptation of New Notation Systems

In West Bengal (state of India), more or less every household has a common practice for music learning, either classical or regional. We have done a rigorous survey in these households and found a notation system called *Dandamatrik* notation system. Books written on music in Bengali

Notation Properties	Dandamatrik System
Pure Notes	স র গ ম প ধ ন
Flat Notes	(Δ) is added above the swara or note
Sharp Notes	(~) is added at the tip of the swara or note
Middle Octave	স র গ ম প ধ ন
Lower Octave	(.) is added at the bottom of the swara or note
Upper Octave	(.) is added at the top of the swara or note
Single Beat	Single line (|) above swara or note
Half Beat	Bengali Chandrabindu (ঁ) symbol (U0981) above the swara or note
Quarter Beat	Cross (x) is added above the swara or note
Beat Start	(+) symbol
Division	Double (||) line
Beat Mart/Beat Number	২, ৩, ৪
Meend/Portamento	⌣
Kan	Same as pure notes but smaller in size
Khatka	⎵

Figure 15. Notation symbols of Dandamatrik notation system with its properties when the language base is Bengali.

also proved the existence of this notation system. It was invented by Kshetramohan Goswami in the year 1868 as stated in [11]. Dandamatrik notation system has a set of notations, some of which have similarities with other notation systems. Figure 15 shows some of them.

Music researchers are continuously experimenting with the notation systems. This experiment is conducted mainly on Bhatkhande and Paluskar notation system. We have found a new notation system called *Ome Swaralipi* or Ome notation system from ome which has been developed by mixing Bhatkhande and Paluskar notation system. Pt. Vinayak Rao Patwardhan, Pt. Shankar Rao and Pt. Omkarnath Thakur had developed this new notation system with little modification in Bhatkhande and Paluskar notation system. The best feature of Ome notation is that it is independent of any linguistic script which makes it universal. It is currently in a validation process in various music schools and institutions.
In order to make a universal notation system for a particular language base we have taken care of these newly found notation systems. We have experimented with the architecture with various combination of notation systems — language base pairs and found that they can be well described by this architecture. Below we describe some of the experimental results.

2.3 Implementation of the Architecture on Other Notation Systems

Figure 16 depicts an example of the implementation of the architecture with a line of Hindustani Sangeet score written in Bhatkhande notation system. As described, the core architecture will be same for this notation system. However, the order of the rows in the 2-D matrix may be different. For example, the beat marking symbols (Taalanka) are placed at the last row instead of first (as in Akarmatrik). Similarly lyrics line is in 3rd row, not in the last row. Figure 17 shows the experimentation on a Hindustani Sangeet music piece written in Paluskar notation system.

The last experiment was on Carnatic Sangeet and traditional Carnatic notation system written in Tamil script. The implementation of the architecture is shown in Fig.18.

To implement the architecture within an application, we need to have two sets of fonts. The first is for writing the music symbols and the other is for writing the lyric. The lyric of the music piece can be written with the help of Unicode transliteration software like Avro Keyboard [12],

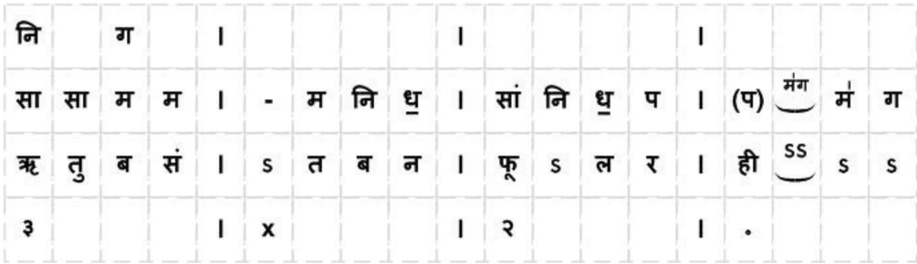


Figure 16. Implementation of a line of Hindustani Classical Music score written in Bhatkhande notation system in the present architecture.

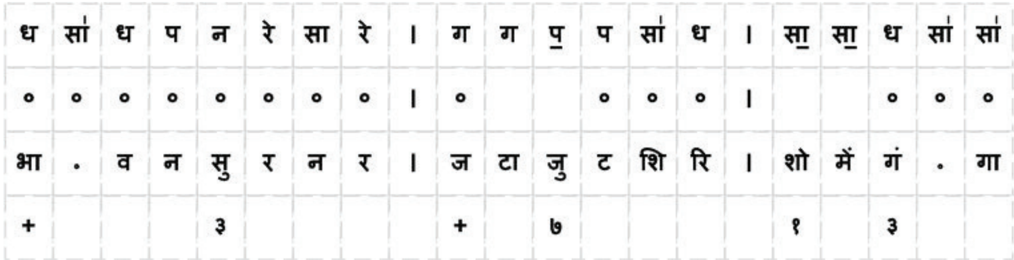


Figure 17. Implementation of a line of Hindustani Classical Music score written in Paluskar notation system in the present architecture

Baishakhi Keyboard [13] for Bengali language. On the other hand, to write music notes we still need to depend on non-Unicode fonts as unlike Staff notation (U1D100—U1D1FF), some of the Indic music notation symbols have not been encoded in Unicode yet. Except some Bhatkhande and Paluskar notation symbols (Bhatkhande Flat Note (U0952), Bhatkhande Sharp Note (U0952), Bhatkhande Lower Octave (U093C), Bhatkhande Upper Octave (U0971), Bhatkhande Beat Gap (U0970), Bhatkhande Abhagraha or Enunciation (U093D) and Paluskar Quarter Beat (U0956)), most of the symbols in Paluskar, Akarmatrik and Dandamatrik systems have not been encoded. Therefore, we need separate non-Unicode fonts for writing music symbols. As a result, to write the score we have to implement mixed format — Unicode and non-Unicode format. Moreover, in the application level the user need to use mixed format into the software and switch between the two for writing notes and lyrics. This cumbersome method can be removed by developing a full set of non-Unicode fonts for each language script along with the language script symbols. This will on the other side makes the music sheet non-standard and without the font set it is not possible to render the score perfectly on different applications.

One solution to this problem is to include all non-Unicode symbols to Unicode. As some of the Bhatkhande symbols have been encoded in Unicode in the Devanagari block (U0900 —U097F), Akarmatrik and Paluskar notation symbols that are not common with the encoded Bhatkhande symbols should be encoded. We have started the process with the proposal of six Akarmatrik symbols among 51 symbols present in the Akarmatrik notation system. These symbols are —Quarter Beat, Meend, Avasan, Taala Bibhaga, Danda and Yugal Danda Symbol as shown in Fig. 8. We proposed eight code points in the Bengali block (U0980 —U09FF) to include these six symbols as described in Table 2 of [10].

We have built a scorewriter implementing the architecture for creating music sheet electronically in Akarmatrik notation system. It has been developed to archive Tagore's songs to set up an online repository. Figure 2.3 shows the original music piece and Fig. 2.3 shows the music sheet rendered by the scorewriter.

3. CONCLUSION AND FUTURE WORK

It is a real challenge to build a software for Indic music and that particularly matches with the framework of the notation systems of different genres and language scripts. The popular Staff system was not a match, as it has a different architecture altogether for representing music. This paper deals with the methods to represent Indic music in computer. The core part of these methods is to design a new architecture for rendering, arranging, displaying an Indic music piece just like published music sheet.

The development of the architecture was carried out using four prominent notation systems in India — namely Bhatkhande, Paluskar, Akarmatrik and Dandamatrik. To make the framework musical genre, notation system and language script independent, we needed a good survey on these three systems. This survey made us collect their similarities and the similarities between them with the language base. We have found the inherent structure of these notation systems which are similar and implemented those similarities to frame the architecture.

The architecture is designed to support only the Indic notation systems and Indic musical genres. There may be other music genres in the world that have same kind of inherent structure as Indic notation systems. Inclusion of these notation systems will make the architecture more sophisticated and universal.

<table>
<tr><td>;;</td><td>;-ப</td><td>ப,-பம</td><td>ரி,ரி,</td><td>|</td><td>ரி;-ஸ</td><td>;-நி</td><td>|</td><td>ஸ,-ரி</td><td>;ப,</td><td>|</td></tr>
<tr><td>ப;-ப</td><td>;-ப</td><td>ப,-பம</td><td>ரி,ரி,</td><td>|</td><td>ஸரி;ஸ</td><td>;-நி</td><td>|</td><td>ஸ,-ரி</td><td>;ப,</td><td>|</td></tr>
<tr><td>பம,-ப</td><td>நி,-பம</td><td>ப,-பம</td><td>ரி,ரி,</td><td>|</td><td>ஸரி;ஸ</td><td>ஸ-நி</td><td>|</td><td>ஸ,-நிப</td><td>;ப,</td><td>|</td></tr>
</table>

Figure 18. Implementation of a line of South Indian Classical Music piece “Bantureethi” written in Tamil script in the present architecture

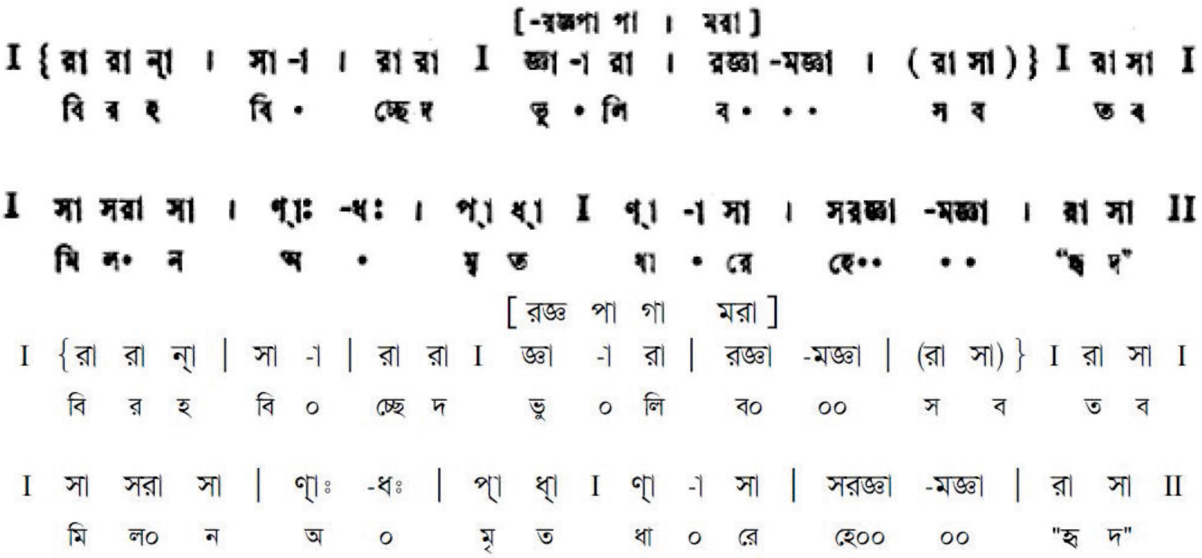


Figure 19. (a) Part of the original Rabindrasangeet music piece “*Hridoy Bedona Bohiya*” written in Akarmatrik notation system and (b) same part as (a) of the music piece generated using the application implementing the present architecture

The architecture provides application developers a model to work on and develop a series of applications on Indic music. Some of them are described below —

1. Music editor that can write, edit, play Indic music with any Indic notation system and with any language. The user can select his intended notation style and language.
2. Application to be used to create and save MIDI files and transfer them between computer and MIDI enabled electronic sitar or tabla.
3. A web environment to learn and teach Indic music system for music lovers and students.
4. A tool that enables social networking sites to write and send music as comments. People can play the written music symbols online and also can download the MIDI version of the music.
5. A portable device that can simulate human humming pattern and with the help of frequency analysis it can map to particular note. With the help of Unicode it can again map notes to respective symbols and finally a printed music sheet.

4. REFERENCES

[1] O. Gupta, *Encyclopaedia of India, Pakistan and Bangladesh*. Gyan Publishing House, 2006.

[2] S. Nayar, *Bhatkhande’s contribution to music: a historical perspective*. Popular Prakashan, 1989.

[3] MakeMusic, “Who we are,” 2014, http://www.makemusic.com/about/.

[4] AvidTechnology, “About avid,” 2014, http://www.avid.com/US/about-avid.

[5] SwarSystems, “About swar shala,” 2014, http://www.swarsystems.com/SwarShala/.

[6] V. N. Bhatkhande and L. Garga, *Hindustani Sangeet Paddhati: Bhatkhande Kramik Pustak*. Sangeet Karyalaya, 1990, vol. 1–5.

[7] Chandra and David, “North Indian Musical Notation - An Overview,” May 2014, http://chandrakantha.com/articles/indian_music/lippi.html.

[8] R. Tagore, “Amar naiba holo pare jawa,” 1905, http://tagoreweb.in/Render/ShowContent.aspx?ct=Songs&bi=1E5F28DE-A4A0-40E5-D51D-6DC060&ti=1E5F 28DE-A4A-46A5-C51D-6DC0607EF5E1.

[9] P. S. Ganguly, *Encyclopaedic Dictionary of World Musical Instruments*, P. S. Ganguly, Ed. Global Vision Pub House, 2008, http://books.google.co.in/books?id=j5oJXHBMBEwC.

[10] C. Misra, "A proposal to encode the akarmatrik music notation symbols in ucs," July 2013, http://www.unicode.org/L2/L2013/13157-akarmatrik-music.pdf.

[11] S. Islam, *Banglapedia: national encyclopedia of Bangladesh*. Asiatic Society of Bangladesh, 2003.

[12] O. Lab, "Avro keyboard," January 2011, http://www.omicronlab.com/index.html.

[13] T. Dasgupta, A. Basu, A. Das, and P. Mandal, "Design and evaluation of bangla keyboard layouts," in *Students' Technology Symposium (TechSym), 2010 IEEE*. IEEE, 2010, pp. 248–254.

NOTATING THE NON-NOTATEABLE: DIGITAL NOTATION OF TXALAPARTA PRACTICE

Enrike Hurtado
Department of Art and Technology
Fine Arts
University of the Basque Country
Bilbao, Spain
enrique.hurtado@ehu.es

Thor Magnusson
Department of Music
School of Media, Film and Music
University of Sussex,
Brighton, UK
t.magnusson@sussex.ac.uk

ABSTRACT

This paper explores notation practices related to the ancient Basque musical tradition of the txalaparta. It presents the txalaparta practice, introduces the improvisational rules of txalaparta playing, and describes our attempts in creating notation systems for the instrument. Due to the nature of txalaparta playing, Common Western Notation is not a suitable notation, and we will present the notation system we have developed as part of the *Digital Txalaparta* project. This system captures the key parts of playing and could potentially serve for both playback and a rich documentation of what players actually perform.

1. INTRODUCTION

The txalaparta is an ancient musical percussion tradition deriving from rural areas of the Basque Country. The instrument belongs to the category of struck idiophones and consists of a variable number of thick wooden planks placed horizontally on two trestles with soft material in between. The planks are struck vertically with heavy wooden batons. The wooden planks typically emit inharmonic sounds, not of any particular pitch, but certain strands in recent developments of txalaparta practice have begun to tune the planks. In txalaparta playing, two or more performers improvise, alternating their beats, through a call-and-response pattern that usually becomes increasingly complex as the performance progresses [1, 2]. The txalaparta is never played by a solo performer: the virtuosity of playing the instrument equally involves the technical skills of the performers as well as the communication established between them, see Figure 1.

The txalaparta is a centuries old tradition, although it lost popularity in the early 20th century, almost disappearing during Franco's dictatorship. However, during the 1960s there was a renewed interest in the tradition, which related to a renewed interest in folk music, diverse projects of preserving Basque culture, and a strong influence of European and American experimental music and avant-garde in the arts, which lead to a fruitful meeting of the ancient tradition and radical modernist art. This further relates to developments in American minimalism, some of whose key protagonists were influenced by the txalaparta [3, 4, 5]. Topics of improvisation, process-based music and algorithmic rules in composition had become frequent, and ideas of using open scores without a fixed results became increasingly popular from the mid-20th century, where an obvious case study would be Terry Riley's *In C,* from 1964. Many of the key elements of txalaparta practice suited this approach to composition, and the reinvigorated interest in the txalaparta in the 1960s can also be traced to Basque musicians and artists engaging with ideological developments in experimental and avant-garde music and art.

Figure 1. A typical txalaparta performance setting. The performers are Felipe and Imanol Ugarte. Picture: Xabier Eskisabel.

Although the topic of some discussion, there is a widespread opinion that the term "txalaparta" refers to the rules, performance style, and the rhythm generated, as well as the physical instrument itself [1, 2]. The argument is that the txalaparta can be played on any material substance, but it has to be an improvisation with two or more performers, following the specific rules of the practice. This division between the rules and the instrument was very helpful when we designed a digital version of the txalaparta, which generates musical events using generative algorithms, since the rules could be represented in a digital system, something the material instrument cannot. The digital txalaparta project is therefore the result of a translation, rather than a qualitative transduction into the digital domain.

2. NOTATING THE TXALAPARTA

Like many musical cultures that are primarily improvisational (e.g., jazz, Indian music, gamelan, or flamenco), the practice of notating the txalaparta serves a very different purpose to that of, for example, Western classical music. The notation is descriptive: it represents patterns and relationships, and the primary purpose is that of explanation, preservation, and communication between performers. This can be seen in contrast to prescriptive notation, where the purpose of the score is to prescribe the musician's actions, as a set of instructions to be followed typically in a strongly linear manner. These categories do not map perfectly to txalaparta notations, as the txalaparta has until recently been an un-pitched instrument (the wooden planks are not of defined musical notes), where the notation describes actions-in-time, not pitch-in-time, like we find in most descriptive notations. In traditional txalaparta the rhythm is non-metric and fluid, and it does not follow bar lines or standard time signatures. It can be defined as additive rhythm as opposed to divisive rhythm. In terms of its fluid nature, performers often play around the beat, exploring elements of rhythmic tension through early, delayed, or silent strokes. This is not mere swing timing, as the divergence from what might be considered a regular meter is quite distinctive. Furthermore, instead of emphasising pitch during playing, the focus is on timbre, where the location of the plank, the force of the mallet, the way the mallet is held, all affect the timbre.

2.1 Scoring the Tradition

There are no notational conventions for the txalaparta. Diverse schools of txalaparta playing have dialects with special symbols and systems to express different characteristics such as which plank to hit. Dynamics are often denoted by the length of the vertical line, and the two players tend to be represented by the respective sides of the line. This type of notation focuses primarily on rhythm and the player relationships, but not on the instrument itself: for example, they do not specify which plank to play or the intended pitch or timbre.

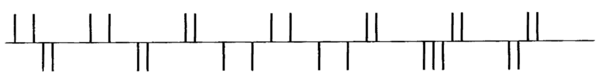


Figure 2. A simple example of Beltran's system of txalaparta notation.

An exemplary system for scoring on the txalaparta is the tablature notation developed by Juan Mari Beltran in the late 1980s (Figure 2). The use of this system resulted in compositions being written for the instrument, for example by Eneko Abad and Sergio Lamuedra, and the system is widely used in teaching at many schools of txalaparta. Here, time is represented with a horizontal line. The two players' events are drawn on each side of the line, since the players are typically facing each other. The strokes are represented by vertical lines, whilst time is represented by white space. This means that '||' represents two close strokes, where '| |' is the same, but more separated in time. Silent hits are often represented by the ':' symbol.

Later efforts in quantising the txalaparta resulted in a different approach to notation where the tempo becomes grid-based, although this grid can be stretched and compressed. Examples of this can be seen in Figure 3.

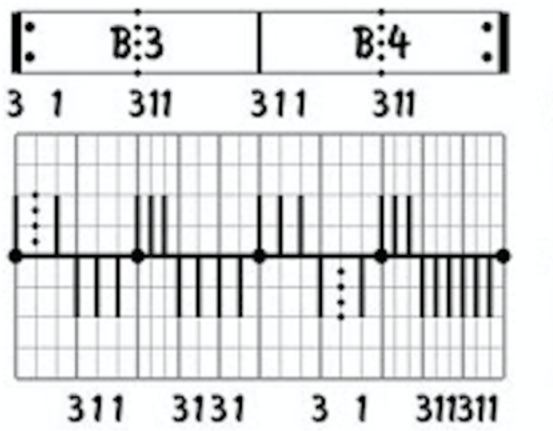


Figure 3. A stretchable grid-notation by Eneko Abad. Here the numbers represent the planks, and the dot symbol are silent hits.

2.2 Rationalisation of Instruments and Scores

During the 1990s, txalaparta practice diversified and reached new audiences. Practitioners started experimenting with pitched wood and certain rhythms become popular. With the pitched txalaparta came the requirement for pitch representation in notation. Some practitioners began drawing scores on paper, using a time-based grid where thick lines on the grid signify strokes, and the pitch is typically represented by a number, or the name of the note the plank is tuned to. This development can be seen as a form of rationalisation of this musical practice, a parallel we find in dance music through the use of quantisation in music software. Here the un-pitched and metrically free txalaparta becomes pitched, and divisive metrical structures are divided into clear units of time. This shift in the nature of the txalaparta from purely percussive towards melody and straight timing is a process Euba has called *xylophonisation* [6]. This process of standardisation aligns with Derek Bailey's description of how traditional musics often lose much of the original characteristics, such as tonal and rhythmical richness, when reduced into the scales and meters of Western classical music [7].

2.3 A Survey and interviews with Performers

We conducted a survey with txalaparta performers, the first of a kind, and also conducted interviews with performers who tested our software. The aim was to collect qualitative data from the personal experience of txalaparta players and how they relate to issues of notation and formal representation of this practice rooted in oral culture. The survey was distributed on online social media groups for txalaparta players, consisting of 280 members. We had 31 responses to the questions, which considered txalaparta practice in general.

The findings will be published at a later time, but on the topic of txalaparta notation, it was clear that the lack of a standardised format allowed interpreters to freely adapt conventions of notation to their needs. It is therefore difficult to find any two practitioners using score systems that are exactly the same. This is also due to the fact that people's needs with regards to notation are very different: some might simply need to make a small drawing that roughly represents the rhythm, whilst others are interested in writing a more complex composition. Many respondents said they used notation for teaching and a discursive analysis of the musical events. In terms of cognitive load, the spatial nature of the score can illustrate patterns that are harder to demonstrate in time.

However, for most of the practitioners who use notation, there is a dedicated space for improvisation. Players do consider the txalaparta as an improvisational tradition, and the use of notation is generally different from that of Western classical music.

A key purpose of the interviews we conducted was to probe reactions to the digital txalaparta – whether the idea of this practice on the digital computer makes sense to practitioners. We were surprised by the general positivity, and we relate that to the fact that the txalaparta is not just an instrument but the rules of its playing. Participants in the survey reported that they found the questions of the nature of the txalaparta introduced by this research interesting, as the practice had not been studied from this perspective before. Many were also intrigued by the novelty of not having to improvise with another human, but with a computer. Some mention that the playing with a computer made them more self-aware but also made them play different to accommodate their play to the computer. One of the players said he started feeling like a machine himself, as he realised he and the computational algorithm were, in essence, performing the same process.

3. THE DIGITAL TXALAPARTA

We present a software system called *Digital Txalaparta,* designed for both performance and analysis of txalaparta. It is a well known fact to software developers that to be able to formalise a practice, a system, or a tool (for example a hospital system, a traffic controller, or an image editor), the developer has to build a representation of the field and be able to categorise it through an ontological process [8]. By so doing, they formalise, make abstractions, and thus have to decide which things to leave out and which to include.

For us, the process of designing software encapsulating the rules and playing of the txalaparta is a method of attempting to understand the practice. In order to program the rules, they have to be made explicit and formalised. This is less problematic in the case of the txalaparta as it is typically defined as a system of rules as well as a physical instrument. Some software applications based on the txalaparta have been developed, but most of them have been playful apps, games or educational tools[1].

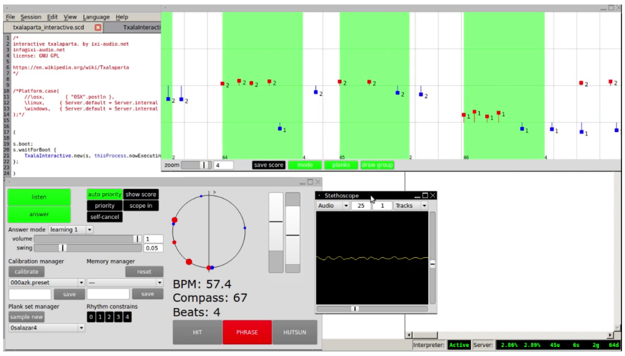


Figure 4. A screenshot of the Digital Txalaparta.

The system's primary function is to serve as an accompaniment for a performer playing a physical instrument. There are two apps: *autotxalaparta* and *interactive txalaparta*. The first one plays rhythmic parts, either using generative algorithms or using playback of known forms. It generates either one or both parts of the txalaparta rhythm and its development allowed us to understand better the options the interpreters face when

[1] The Technotxalaparta was able to listen and respond to the human interpreter adjusting its tempo in real-time through keyboard keystrokes; the computer output was MIDI. Ixi audio released an application called Txalaparta where by dragging four batons around the app's 2D space the user was able to control an ongoing txalaparta rhythm. Finally, the Ttakun was a sequencer that aimed at creating compositions and exercises for the txalaparta. There are also a few apps for mobile media (txalapartapp.com), but none of them suit professional practitioners in understanding and analysing their performance, practice, or produce new output.

they play. The software provides control over the parameters used by the generative algorithms. The interactive txalaparta uses machine listening to analyse and respond to the human performance. To generate the response it uses a sample based system with up to 30 samples per plank, classified by amplitude and location in the plank, which provide a lively timbral output close to that of a real txalaparta. Users can calibrate the system to accommodate to the player's style and they can sample the sound of their own txalaparta in order to get a more realistic timbral response.

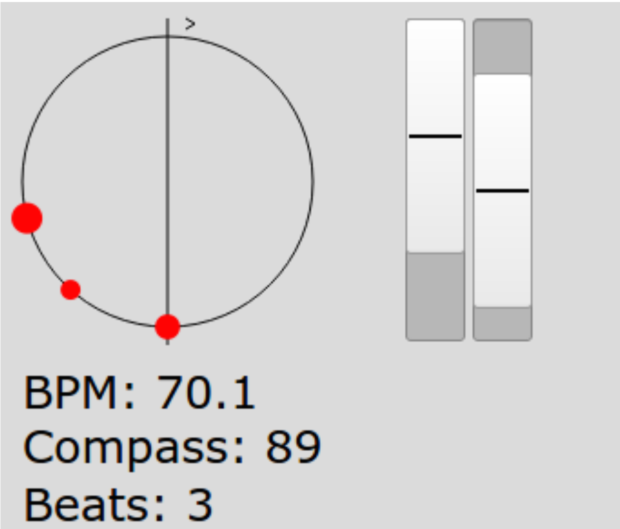


Figure 5. The bar is represented as a circle, with the red dots being the detected strokes. The dots' width maps to amplitude. Time is clockwise. The two vertical bars represent the computer's batons, simulating the movement when raising the stick and stroking.

A key problem for users to understand algorithmic processes is the lack of visual representation [9]. Our user tests corroborated these findings in that visual feedback (even just in peripheral vision) proved to be crucial. Interpreters playing a physical txalaparta in tandem with the autotxalaparta or the interactive txalaparta initially found it problematic that they were not able to see the moving body of their partner. These movements are crucial for txalaparta players to anticipate their partner's actions. To overcome this problem we implemented a graphical representation of the algorithmic process. Two vertical sliders are used to as 'virtual' batons in the computer's performance. This is illustrated in figure 5.

Furthermore, a circular representation of the rhythm is used to show the 'thoughts' of the system, a notation inspired by a diagram Sánchez uses to represent the txalaparta rhythm [10]. This visualises the relationship between the different strokes of the same phrase, as well as the relationship between each phrase and the main tempo. In this system, time is represented as a circular flow with no beginning and no end, visualising the bar in real time. The circle represents the length of the bar split in two by a vertical line that signals where the phrases should of each interpreter should be aligned in the case of the tempo being accurate. In this case the first hit of each phrase is located at the vertical line whereas in the case of any deviation this is shown by their position in relation to the vertical line.

4. THE TXALAPARTA SCORE SYSTEM

As part of our work on the digital txalaparta we implemented a corresponding notation system (see figure 6) that visually represents both the actions by the player and the machine in real time. Since txalaparta playing is typically a turn-taking performance, we also represent the phrases of each. The txalascore is a representation of the play as it happens in real-time and events are written into the score directly as they happen. It represents visually the amplitude, the timing and the plank beat by each hit in each phrase. The score is reminiscent to a piano-roll where the events the system detects and the system's answers flow across: new events appear on the right and move towards the left. Users can zoom into a longer or shorter time spans, ranging between one and twenty seconds. Each plank is represented in a different horizontal line but a color mark differentiates the hits by each player, which can be displayed both on top of the line, or each of them on different sides of the line. This latest method is closer to the one used by the Ttakun sequencer. Furthermore, in order to visualise which strokes belong to the same temporal phrase, these can be grouped with a green transparent field (Fig. 4). While the txalascore was primarily created to help txalaparta players understanding the rhythms generated by the computer by visualising them, it also serves a useful role when analysing recorded performances.

We plan to implement a data format for recording the data extracted from txalaparta performances by this system into a file format that could be further object of analisys by other means. This would be a format of descriptive notation. Moreover, in future versions, through embedding the txalaparta with contact microphones, and accelerometers on the players' batons, we will be able to describe precisely which performer is striking which plank where, with which baton, at what velocity, at the exact moment. This data could be stored in a time-based file format that can be exported to MIDI or MusicXML.

Txalaparta performances are mostly improvised, so the idea of writing prescriptive scores for linear performances does not appeal to many practitioners. People might question the purpose of creating a sophisticated notation system for this reason. However, the fact is that a descriptive notation can be useful in understanding performances, for musicological analysis and for players to study and analyse their playing, even with statistical methods.

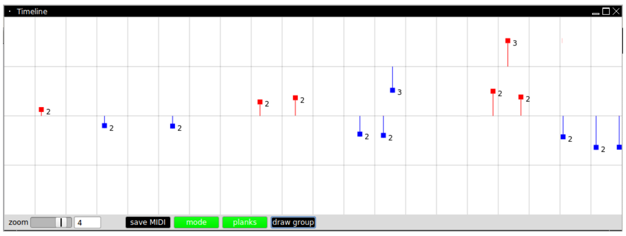


Figure 6. A screenshot of the txalascore. The horizontal lines represent the planks of the txalaparta, whilst the red and blue boxes are musical events of each player. Dynamics are the length of the line. Timbre (location) can be represented with color.

A study by Euba that analysed different methods of transcribing txalaparta performances (unpublished research) concludes that it is practically impossible to transcribe perfectly the actions of two performers: even when using a video recording of the performance, the two players can be playing so fast that it can be very difficult to detect whose stick hit which plank at any event. With a descriptive notation system that picks up amplitude, timbre, location and more, txalaparta performances can be analysed at a much deeper level, for example analysing the relationship between performers, comparing the play of a performer over a longer period of time, comparing different performers' playing, studying the difference between human-human and human-machine relationships and many more.

This can be useful for teaching purposes, and in general it would allow txalaparta users to be more self aware of the different ways they play and it would open up an space for musicologist analysis with greater possibilities than that offered by simple video recording, as we see in Euba's analysis [6]. There is clearly value in precise numerical data here.

5. CONCLUSION

Software development in the domain of music is a highly effective research method for both music and musicology. By having to formalise the rules of the txalaparta in order to create a digital version of it, we had to analyse the play, understand the general practice and the player communication. We had to think about ergonomics, human-machine relationships, and the quality of sound. Reciprocally, when we had early versions of the system running, the computer helped us, but also txalaparta players who used the system, to better understand the rules that govern the playing of the txalaparta.

Through the software development we have become acquainted with different levels of rule sets: on a lower level there are rules that determine the musical material (e.g., how many subdivisions are in the phrase, how to construct the computer response) and on a higher level there are rules that define how the interpreters interact each other during the play to construct long term structures. We have seen that some characteristics of the txalaparta are easy to translate to the digital domain (rhythmical characteristics) while others are more difficult (timbre). Writing software that effectively implements all those rules requires generative algorithmic systems to get closer to the way the txalaparta interpreters interact with each other during the play.

Considering the historical evolution of the txalaparta – in particular the current 'xylophonisation' process where pitch has been added and the rhythm becomes quantised – it is interesting that the digital txalaparta, where the practice is translated into the digital domain, is closer to the origins of the txalaparta in operating with fluid rhythms and non-metric bars, both in its internal algorithms and graphical notation.

Creating a system that analyses and stores the characteristics of the txalaparta play can help understanding better this ancient but modern music. The Txalaparta Score System is still limited but it has already allowed us to get a different and new insight into the way the txalaparta is played. Further developments should provide more detailed data allowing for further research on this matter.

The digital txalaparta is work-in-progress. Future plans include improving the machine listening algorithms in order to make the response system richer and more engaging. We are interested in the cultural reception of the digital txalaparta and studies will be conducted in that area. Finally, since some of the key limitations of improvising with a computer derive from the fact that physical presence is limited and response tends to be audiovisual, we are interested in exploring robotics for both the usability and the cultural studies purpose.

Acknowledgments

This research has been greatly supported by a community of txalaparta players on various social media groups, who have engaged in a dialogue and sent materials that are part of this paper. We would also like to acknowledge the generous support and discussions with Argibel Euba, whose extensive research into the history of the txalaparta greatly benefited this research.

6. REFERENCES

[1] M. Escribano del Moral, *Rhythms of Struggle. Recovery, Revival and Re-Creation of Txalaparta in the Basque Country.* PhD thesis, University of Limerick. 2012.

[2] J.M. Beltran, "Txalaparta". *Aurrera Begira. Udako Euskal Unibertsitatearen 1.988.eko* Ihardunaldiak, 1998.

[3] K. Potter. *Drummed Out? What Steve Reich did after Drumming.* Mitteilungen der Paul Sacher Stiftung, 24, pp. 35-41, 2011.

[4] K. Potter. *Steve Reich handclappers.* London Symphony Orchestra, 15 October 2011, Programme Notes. 2011.

[5] E. Hurtado. *La txalaparta digital: Un análisis de la txalaparta a través del desarrollo de software*. PhD thesis. UPV/EHU. 2015.

[6] A. Euba, "Txalaparta," *Transcription and Analysis*, Goldsmith College, University of London, London. p. 34. 2004.

[7] D.Bailey, "Improvisation, its nature and practice in music". New York: Da Capo Press. Pp. 53-65. 1993.

[8] G. C. Bowker, and S. L. Star, *Sorting Things Out: Classification and its Consequences.* Cambridge: MIT Press. 2000.

[9] G. Weinberg, and S. Driscoll. "The Design of a Robotic Marimba Player – Introducing Pitch into Robotic Musicianship," in *Proceedings of NIME*, New York: New York University, 2007.

[10] F. J. Sánchez, and J. M. Beltran, "Una teoría numérica del ritmo aplicada a la txalaparta," *Txistulari*, no. 173, pp. 27-43. 1998.

S-NOTATION: A COMPLETE MUSICAL NOTATION SYSTEM FOR SCRATCHING AND SAMPLE MUSIC DERIVED FROM "THEORY OF MOTIONS"

Alexander Sonnenfeld
Qbert Skratch University
alexandersonnenfeld@yahoo.de

Kjetil Falkenberg Hansen
KTH Royal Institute of Technology
Södertörn University
kjetil@kth.se

ABSTRACT

In this paper, we present and discuss the *S-notation* system for sample-based music, and particularly for DJ scratching and turntablism. Sonnenfeld developed the system from his *Theory of Motion* where scratch music is seen as constructions of concurrent musical gestures (motion parameters), and not only turntable actions. The detailed symbolic notation was inspired by traditional musical notation, and among its advantages it covers current musical needs, it can be read and played live in performance, it provides a tool for composers to convey musical ideas, it can be expanded towards new styles and techniques, and it is generalizable to other types of sample-based music. In addition to motion parameters, the new notation system involves an analysis of the sampled sound. Finally, S-notation is also applicable for documenting and for teaching situations.

1 INTRODUCTION

In this paper, we introduce the *S-notation* for sample-based music [1], and particularly within turntablism, which is a practice where disk jockeys (DJ) use turntables as musical instruments [2]. There is a growing need for musical notation that can handle performance and composition of turntablism. Since 1998 there have been attempts at creating such notation—and with varying aspiration—but none has been exhaustive. A possible explanation for this is the high musical complexity, which can be expressed as specific challenges:

1. To effectively write and read scores for music with high event density, as scratching averages 5.8 notes played per second [3].

2. To find means of transcribing playing position of the recorded sound that is to be manipulated.

3. To cover the musical diversity of DJ playing techniques including scratching, word play, drumming, beat juggling, and experimental techniques.

This design task is complicated further as each style involves both hands alternately playing the audio mixer and turntables.

The S-notation system proposed here has been developed by Sonnenfeld since 1999 [4, 5]. It represents a reciprocation from the progressively simplified graphical notation systems to a more elaborate symbolic one that extends standard musical notation, see Figure 1. Although the appearance is similar to standard notation, there are a few fundamental differences, mainly related to pitch representation, sound to be played, and rendition of onsets.

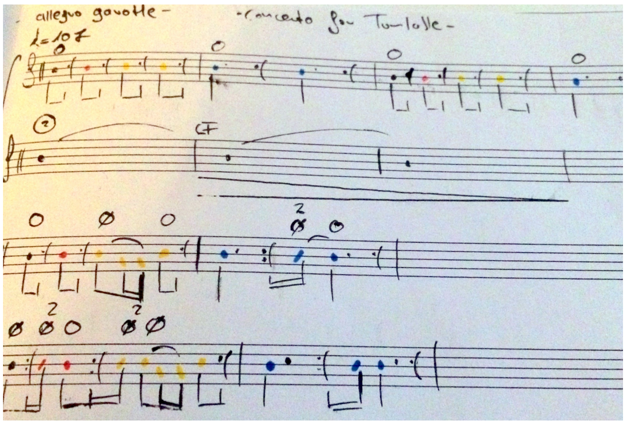


Figure 1. Excerpt from the handwritten S-notation sketches of Gabriel Prokofiev's "Concerto for Turntable".

Because of the balance between complexity and affinity to standard notation, S-notation addresses the above challenges and has a set of advantages:

– it provides means for conveying musical ideas,
– it provides means for transcribing music,
– it can with training be read and played *a prima vista*,
– it covers most of the foreseeable musical needs,
– it is expansible toward new turntablism techniques,
– it is generalizable to other sample-based music.

In the following, we will describe both how the notation derives from the *Theory of Motion* [1] and how the sampled sound material can be transcribed alongside score elements like pitch, note duration and dynamics. Because of their breadth, only a part of the S-notation system and Theory of

[1] Orig. *Bewegungslehre* (German).

Motion can be introduced. [2] We will also discuss possible further developments with some emphasis on tutoring. The emphasis of the notation format, however, is on the now classic performing tradition which dates back to the late 1970s.

2 BACKGROUND

DJ scratching and turntablism have been studied in detail, and are now part of higher education curricula [2, 6]. The musical style has been exposed to a wide audience since Herbie Hancock's MTV video *Rockit* [3] from 1984 until recently in 2011 when Gabriel Prokofiev's *Concerto for Turntable* was featured in the BBC Proms concert series. [4]

The taxonomy of this music is well established [7], and some basic concepts are:

Turntablism Music made using turntables, played by *turntablists*. This does typically not include mere playback, although there is no strict definition. Similar practice with other controllers and other sound sources than vinyl can be considered turntablism.

Scratching The main musical output of turntablism. Performed with a combination of record movement and onset control using the audio mixer's crossfader.

Beat juggling Switching between short phrases from each of the two turntables in the instrument set-up, creating new rhythms, melodies and chord progressions.

Drumming Playing rhythms from a small set of sounds such as a bass and snare drum beat.

The study of hip-hop DJing is to a large part the study of performing gestures, and the importance of such *techniques* is axiomatic. Even the story of the "birth" of scratching is, quite amusingly, the story of the *baby scratch* technique [6]. S-notation is based on Sonnenfeld's Theory of Motion [4], which relates to current research on gestures [8]; in particular, scratching is seen as not only a performing action on the turntable, but as a construction of several concurrent musical gestures. It should however be mentioned that the Theory of Motions was developed outside academia, independently of ongoing research in the area.

Several writers have put forward the need for notation in scratching, and for different reasons. For instance, Taniguchi [9] claims that the audience appreciation will improve; Smith [10, 11] argued that turntable notation is necessary for communicating musical ideas, for documentation, for composition, for analyzing and understanding, and for making scratching a legitimate musical practice; Miyakawa [12] goes even further and says that notation methods are reflections of turntablists' self-conscious efforts to be recognized as musicians; and Ouper [13] and Biederman [14] discusses how orchestral works with turntable soloists are transcribed.

[2] Full descriptions and examples: www.alexandersonnenfeld.com

[3] Official video from VEVO: youtu.be/GHhD4PD75zY

[4] Official BBC clip featuring DJ Switch: youtu.be/38atRejUORM

2.1 Scratch Notation Systems

Even if there are several notation formats for scratching, these have not been used to a large degree: only 23% of the scratchers in a survey had ever used notation [3, p. 41]. Scratch notation was first inspired by musical staff notation, and later based on graphic representation of scratch gestures. Doc Rice [15] and Hansen [16], and later Radar [17] and Webber [18] were among the first to use standard music notation (with additions and alterations) for transcribing scratching. These systems used symbols on the note heads to indicate which scratch technique to play.

DJ A-trak [17] and Raedawn [19] introduced graphical notation systems. DJ A-trak developed his system for personal use, based on sloping lines which represent record movement with crossfader cuts marked on these lines. Raedawn's Turntable Transcription Methodology (TTM) specifically targeted the turntablism community, and was also long the most spread format. Position in the sample is plotted on the vertical axis as a function of time, which is ordered in grids corresponding to beat durations along the horizontal axis. According to interviews and comments from DJs, the system is intuitive and fairly easy; however, there are few reported cases confirming that it has been used.

2.2 Sample-Based Music

Making new music from sampled music is paradigmatic in contemporary music practices. *Paradigmatic* because the practice is everywhere [20]; because sampling music radically changed music industry [21]; because sampled sounds have cultural connotations [22]; and because musical sounds in general can be acoustically [23] and contextually [24] interchangeable. The term *sample* means in this context *the material the musician uses* [25], and is an inseparable component of the instrument, embodied in a tangible medium [26].

Time-coded vinyl, often called Digital Vinyl Systems or DVS, was a game-changer for DJs when introduced in 2001. Instead of playing music stored on vinyl, the record is imprinted with code that can corresponds to playing position in a digitally stored file. In basic use, it is exactly like ordinary vinyl, but with computer processing, the possibilities are limitless. In this paper, we use "sample" as described above, and disregard other uses of DVS.

Music based on samples is fundamentally difficult to transcribe. This is especially true when the samples are manipulated in continuous pitch sweeps and not in incremental steps. Typically the DJ will use a short (less than 0.5s) sample to play, and according to previous studies [27, Figure 5] only the first part of a sample is likely to be played. Playing position and the character of the source sound determine both which techniques that can be used and how it will sound. For instance will playing through a source sound containing several syllables, like the commonly used phrase "making me rich", generate more tone onsets than a sample like "ahhh".

Although we focus on turntable interaction in this paper, sample-based music take on many forms that are related.

The same playing and transcription methodology would for instance apply to music generated using novel interfaces like two-dimensional surfaces [28–30], or in writing scores that include sampled material.

2.3 Playing Techniques and Gestures

As mentioned above, scratching as music has since the very start been defined by a number of *techniques* [16, 31], which also led to extensive collections disseminating these [32]. However, it was shown in a survey that half of the scratching DJs *know* less than ten such techniques or none at all [3, p. 46]. Thus there has been a research bias on techniques that is not incontrovertible.

Instead of looking at named techniques, it is thus more fruitful to consider playing gestures generatively. Each tone (or note) made by a scratching DJ consists of a synchronized movement of pitch control and amplitude control. In the proposed method, these gestures are systematically analyzed together with the playing position and contents of the sample. The theory behind the transcription methods in S-notation is called Theory of Motion, and includes *motions* or gestures in different domains. Common is that the motions are not directly corresponding to performance gestures (as in *forward–backward* movement), but to musical or acoustic gestures.

3 THEORY OF MOTION

Theory of Motion sketches a sort of classification of the DJ's instrument set-up consisting of the audio mixer and turntable to understand how it influences specific musical parameters of the sound material. [5] This systematic is the requirement to transcribe the performance through so-called *motion parameters* by using the S-notation. These motion parameters are grouped into:

Acoustic Motions Movement of the vinyl or control disk (changing the speed or direction) to change the pitch of the sound.

Dynamic Motions Line fader or crossfader movements to change the volume (i.e., for the most part to turn on or off sound).

Frequency Motions Movement of the fader or rotary dial to cut or boost certain frequencies (i.e., equalization, but the features on the mixers vary in this regard).

Panning Motions Movement of the panoramic (panning) dial to spread the sound across the stereo field.

Effects Motions Movement of the fader or rotary dial to change the intensity of applied audio effects (e.g., reverb, delay, distortion).

[5] In this paper we focus on turntablism-style DJs. The theory of motions does however not exclude controllerism in general, which has implications on both the instrument set-ups (other control surfaces) and type of musician (i.e., not only DJs).

While the gestures and movements may be similar for e.g. Frequency, Panning and Effect motions, the musical intention is not. Since the musical intention vary, it is not certain that the actual gesture will be similar either, and therefore the motion parameters are specified. Another reason for having specific controller-based motion parameters is that there are many playing techniques that utilize one controller.

All parameters are further separated into *motion types*: 'Single motions', 'integral motions' and 'groups of motions'. They constitute the fundamental principles of a composition—comparable to melody—and the theory of motion aims to represent them by notational symbols. This principle should also help the player to capture possible playing strategies on the instrument which cannot be defined in the context of classical, tune-based music.

Due to the fact that we cannot transcribe the musical output as ordinary pitched notes, we need to think a little differently about what "melody" is. In turntablism the least common denominators are individual gestures (forward–backward, left–right or up–down). When you make "groupings" of these movements you can conceptually handle a wide range of possible playing strategies. A sonic trademark of these groupings is the "integral motions" which means the connection of both fundamental "single motions" as one pattern. Such a pattern is for example in the *baby scratch* and *1-click flare* techniques. In S-notation we defined this important compositional design element as an "integral motion" and created a separate symbol and term.

Every type of motion is subject to a unique architecture which is defined by a set of motion criteria. These criteria are: *direction*, *time value*, *intensity* and *characteristic*. S-notation is a transcription methodology from which you can read all the motion criteria based on the principles of music theory. As in classical notation, the shape of the symbol and the position inside the staff determines the action the musician should take.

4 S-NOTATION

S-notation is a transcription method which uses an own repertoire of notational symbols to describe the techniques of a turntablist; currently, there are around 20 additional symbols in use, see Figure 2. S-notation basically follows the orthographical rules of classical music notation in order to be able to communicate with traditional musicians outside of turntablism and open doors for composition, education and research. There are also a number of different clefs that serve specific purposes.

The purpose of this type of written music is to enable a musician to repeat performances with consistency, which has two requirements. First, the sound itself must be the same each time. This means there must be a way to annotate playing position of a sampled sound. The second requirement is that the individual techniques on the instrument must be subject to an order, a series of principles which can be understood and applied correctly.

In classical musical notation both prerequisites are firmly

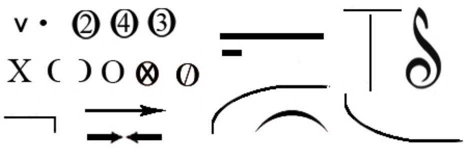


Figure 2. A selection of symbols added to conventional music notation for transcribing turntablism in S-notation. To the right is the "S" clef.

met because the movement on a keyboard or string is always linked to a certain tone or pitch. It is therefore possible to notate the tone based on the positioning of the note inside a musical staff. It is particularly difficult to capture pitch because of the broad pitch range produced by the motion of the disk, the characteristic of the sample, and the difficulty of playing steady-state tones.

4.1 Playing Techniques

S-notation describes only the manual motion on the disk and faders as a sort of gesture. For education and scientific purposes, a base formula allows detailed transcription of all the different playing techniques using a turntable and fader. To assist in this, audio recordings of particular scratches help the player to get familiar with the respective notated patterns.

S-notation provides its own repertoire of symbols to describe the direction of record motion and also the playing style, such as "hand mode" where the hand is in contact with the vinyl, and "release mode" where the record plays forward. It also includes a wide assortment of symbols to describe the velocity characteristics of the record motion (for example constant, logarithmic, exponential). Other parameters such as duration, pauses, articulation, etc., are based on traditional music notation.

4.2 Record Movements

Single movements on the record are divided into forward motion named *Note* with a standard note head, and backward motion named *Eton* which is the mirrored symbol of a Note (i.e. with a mirrored note head), see Figure 3.

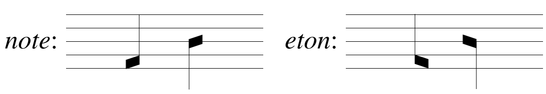


Figure 3. Notation of forward (note) and backward (eton) movements of the record.

Combinations of these two single motion types can produce complex patterns which we will refer to as "groupings", such as the forward–backward–backward movements of a *tear* scratch, see Figure 4.

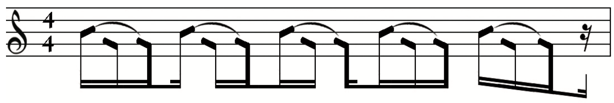


Figure 4. Notation of grouped gestures (forward–back).

4.3 Playing Position and Anatonie

To address challenges of working with recorded music as a sound source, S-notation allows to define all possible positions of the sample (or several samples) by using a given order of colored notes to represent the playing position. [6] Figure 5 illustrates the different color positions on a record sample as played in a phrase, and the mapping between position and color in the waveform representation. The predefined coloring of the sample is part of an analysis of the recorded sound material, and is referred to as the *Anatonie*.

In using this coloring process it is possible to perform specific sound-material based on S-notation (see Figure 5), which is crucial because turntablists work with all kinds of recorded material. The *anatonie* analysis also includes basic values of musical parameters, such as duration, pitch, volume, etc. By using a colored template for the record (there are different methods possible) the player can play through the sound to recognize the colored areas corresponding to chronological positions of the sample.

Just as (most) other music instruments have visual or tactile cues or guides to help the player orientate to find correct pitch (the keyboard layout, buttons, frets, key-holes, etc.), it is important in this context to create a visual aid for the DJ or composer to apprehend the sample. The notation in Figure 6 shows a drum scratching performance by differentiating between a bass drum and a snare drum part.

The black notes (notes 1–4 and 7–12) indicate the record motions of the bass drum; the red notes (notes 5–6 and 13–14) indicate the snare. Round note heads indicate a special playing technique which refers to the so-called *release mode*. The symbols above the staff indicate the crossfader movement and define the wide range of possible playing strategies on faders. The turntable and mixer as being one instrument is defined through the interaction of Acoustic Motion and Dynamic Motion.

4.4 Dynamic and Acoustic Motions

The duration of a Dynamic Motion determines the length of time for which the sample is audible. Normally it is equivalent to the duration of the Acoustic Motion (the movement of the control disk), which is why the time value of the Acoustic Motion matches the action of the crossfader. As the motion durations varies, so do the methods for manipulating the sound material. In each of the examples in Figure 7a–d, a quarter note is cut four times by applying a different playing technique.

The graphical waveforms in in Figure 7 show us the resulting divisions of different techniques on the same sound

[6] The actual coloring scheme is currently under evaluation for maximizing readability, to ensure that for instance color confusion is avoidable through layout settings and templates.

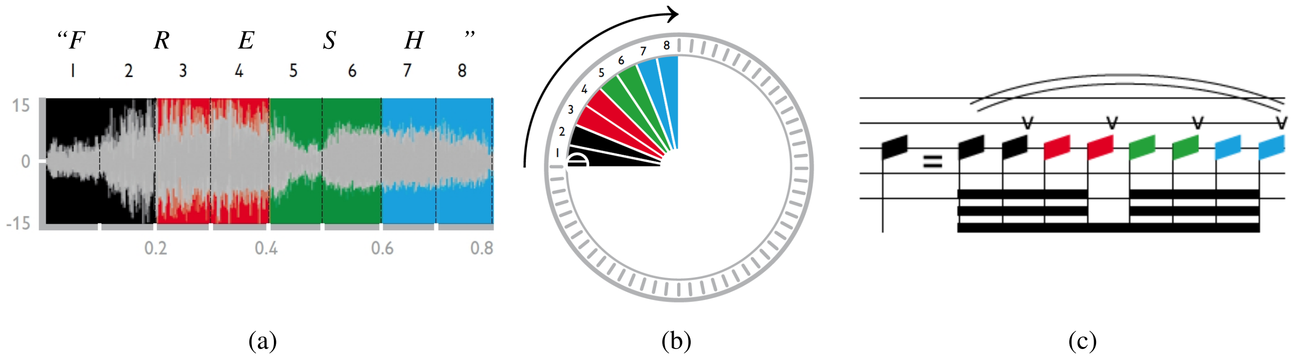


Figure 5. Three representations of a segmented quarter note, sectioned by the colors black [1-2], red [3-4], green [5-6] and blue [7-8]. (a) shows the analysis of a 0.8 s sampled sound (“FRESH”) in a waveform plot. (b) shows a corresponding visual representation of the playing position in relation to a 90° record movement. (c) shows a musical phrase using the same coloring scheme. The double slur indicates that the *forward Notes* should be played in one motion.

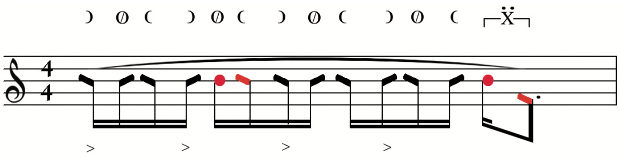


Figure 6. Notation of drumming. Black note heads are bass drum sounds, red note heads are snare drum.

sample (“fresh” of the sentence “ahhh, this stuff is really fresh” [33]). In an accompanying video recording [7], one drum-scratching pattern was repeatedly played using all basic fader techniques that are mentioned with Figure 7.

5 EVALUATION AND DISCUSSION

Like other notation systems for scratch performances described here, S-notation has not been formally evaluated. Indeed, no system to date has been used very much. These matters are further examined below. There is a growing incentive from composers for notation, however, whose opinion is valuable. Until now, there has been little reason for DJs to learn notation. While not a proper evaluation, the actual appropriation of S-notation in new works will function as a proof-of-concept and aid development.

5.1 Appropriation for Composition

As mentioned earlier, the British composer Gabriel Prokofiev wrote “Concerto for Turntable” which has been played worldwide, including a performance by Mr. Switch at the BBC Proms. The notation first used to write the score left however too much room for interpretation for the soloist, and therefore the concerto has now been transcribed using S-notation.

The soloist works in this piece with an orchestral theme sample which was produced by the composer and then prerecorded for scratching with by using a timecoded DVS. The sample was analyzed with Anatonie for transcription, which was important to match pitches and onsets with the orchestral part. All relevant information to the DJ is printed in the score, including tempo (bpm), turntable speed, sound-file names, sample lengths, and pitches.

In the fifth Movement,the sample consists of four transients or onsets: three tones and a reverse tone. To mark all the relevant cue points which are used for performance (generally done by DJs in the DVS software), the wave spectrum is color-coded and transferred onto the notation.

As all motions in the section seen in the sketch in Figure 1 and in the finished score in Figure 8 are in the release mode (sounds are played by letting the record play forward) the note heads are round. The pitch control slider is at 100% turntable speed, so the notes are placed on the middle line. Based on the coloring of the note heads and the position within the bar it is possible to anticipate the arrangement of tones, like one would expect.

5.2 Challenges for Teaching

Few turntablists are educated in traditional music theory or have a musical background outside DJing. Thus, the systematic notation can be challenging to learn and understand. Without tutoring, and compared to students who play traditional instruments that have more formalized learning situations and didactics, the ambition to learn or work with notation can be hard to motivate. A particular hindrance can be the format of scores: it is unlikely that DJs will ever put up a note stand and read printed music, and the computer screen, which is commonplace in today’s gear setup, is already saturated with necessary information to the performer.

Turntablists mostly play and practice in solitude, and seldom in groups; ensemble performances would otherwise have a positive effect towards using notation. The process of education is commonly autodidactic by means of following tutorials or imitating performances from other artists. Nevertheless, many understand the importance of learning music theory because it enhances the possibilities of being more creative and collaborative in making music with this instrument. Having a common notation is critical for the party of teachers that grows with the flourishing music style.

[7] youtu.be/H2dcSpukf6c

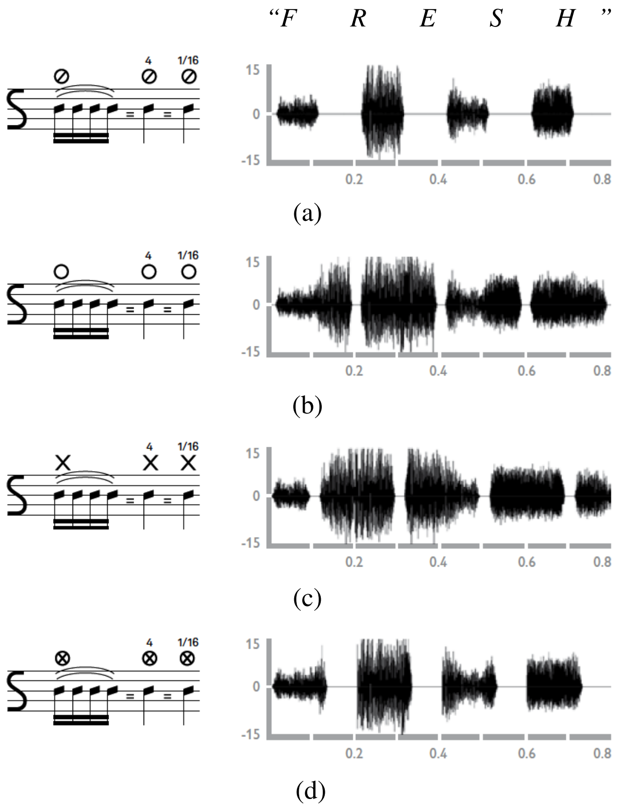


Figure 7. On the left, the breakdown of the patterns is shown in the stave. In order to reduce and simplify notation, the number of cuts or durations are written above the symbols. The resulting sound will depend on the cutting technique used (open–close or close–open with the fader).

5.3 Reception by the Community

By definition, turntablists consider the combination of the turntable and audio mixer as a musical instrument, and typically support the idea of music notation (although with few using scratch notation). Some reactions suggest that notation discourage improvisation, and even taking away the *soul* of scratch music due to the systematical approach of the Theory of Motion; after second thoughts, however, DJs tend to realize that this theorizing constitutes nothing more than the common and essential knowledge which forms the basis for effectively communicating with all types of musicians, a practice that already exist.

Opponents of any notation argue that the best method of analyzing and memorizing a scratch is by studying available recordings. To such learning scenarios one might use waveforms to indicate the acoustical result of a pattern in detail. But S-notation was developed to describe the motions on the turntable and mixer for the purpose of faster comprehension. While one could analyze videos or audio recordings, a more efficient way would be to learn a system that can describe any scratch technique.

5.4 Future Work

The upcoming work with Theory of Motions will tackle the analysis and transcription of other motion parameters such as frequency, panorama and effects motions, and how they influence the musical output. In addition to that, the focus will be on compositional methods, a curriculum based on S-notation, and special techniques such as *beat juggling* or tone playing.

Furthermore, the Theory of Motions will be applied to visual controllerism (often called video DJing or VJing) in order to find new ways of creativity for video performers. The relationship between the current status of technology and live performance of audio and video leads to novel artistic forms which are valuable to explore.

One problem with transcribing scratch music is the rhythmic complexity in terms of onset density and timing. Depending on playing position in the sample, tones can have onset timing that might deviate from the performer's intention, and the acceptance for timing imperfections. A possible approach to analyzing performances could be to use an appropriate automatic transcription method [34].

S-notation was developed with a needs-based—partly *ad hoc*—approach. The design builds firmly on traditional notation: one advantage is that the development process can be quick, and design choices are unlikely to critically contravene established practices. However, the format should be evaluated and harmonized with (contemporary) guidelines for musical notation, and in particular SMuFL [35]. At the moment, S-notation has not been implemented in existing software like MuseScore or LilyPond, but this work is projected.

Design choices concerning how to represent the different motion parameters should be carefully assessed, for instance within the *cognitive dimensions of notation* framework [36]. In this process, the selection of parameters to include should be validated against recent DJ practice studies [e.g. 37]. As the presented system mainly extends standard notation, the need for evaluation is thus arguable. The Theory of Motion is on the other hand more exploratory and will be approached with inspiration from contemporary research on gesture.

6 CONCLUSIONS

Theory of Motion was employed as a theoretical and methodological framework for creating a new notation system for sample-based music, *S-notation*. According to this theory, performances can be transcribed using a set of parameter motions, including acoustic, dynamic, frequency, panning, and effects motions. In addition to the motion parameters, the new notation system involves a time-based analysis of the sound source, named *anatonie*.

The S-notation system was particularly developed for DJ scratching and turntablism, and was designed and implemented with both the musician and composer in mind. Although it is a complex notation format, it can be learned and potentially used in performance. Correspondingly, although turntablism is a complex musical form, S-notation can provide a detailed tool for composers to convey musical ideas. Finally, S-notation is also applicable for documenting and for teaching situations.

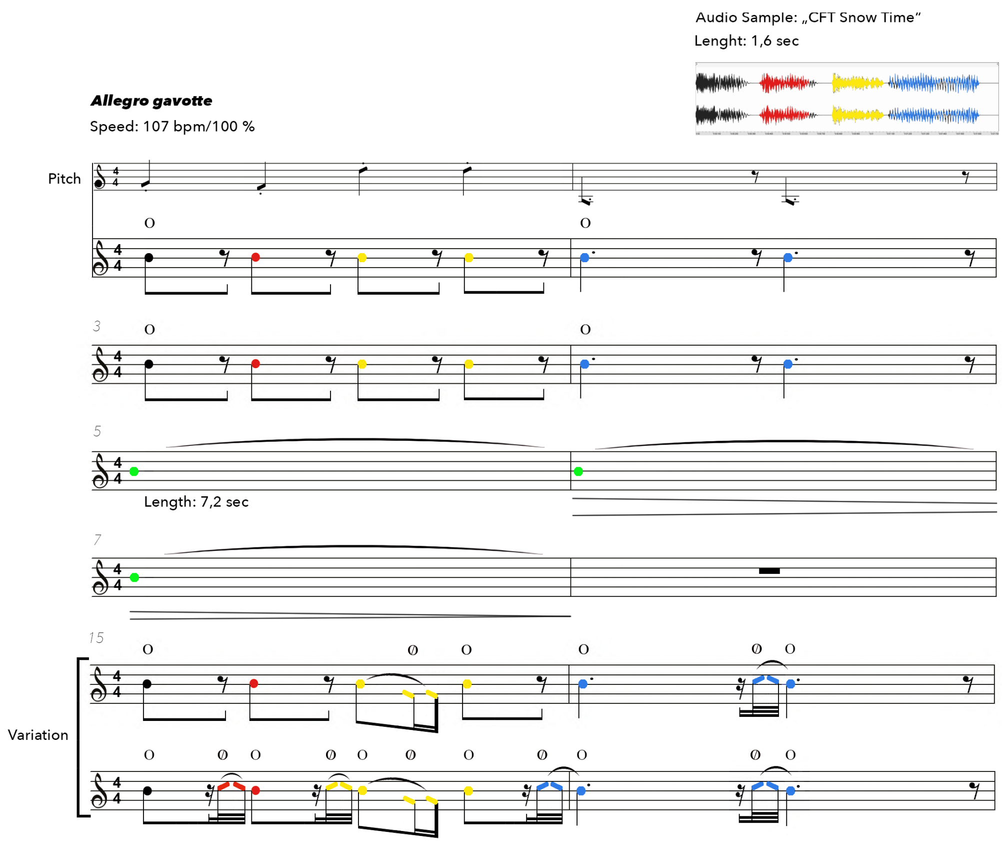



Figure 8. Finished score of the first 10 bars or *Movement V – Snow Time* of Gabriel Prokofiev's "Concerto for Turntable and Orchestra", see video https://youtu.be/38atRejUORM.

Acknowledgments

This study has been kindly supported by Södertörn University and KTH Royal Institute of Technology.

References

[1] A. Sonnenfeld and R. Kalkowski, *Turntablism. Musiktheorie und Instrumentalmethodik*. Tonspielzeug Verlag, 2009.

[2] K. F. Hansen, "The turntable: The musical instrument of hip-hop", in *The Cambridge Companion to Hip-hop*, J. A. Williams, Ed., Cambridge University Press, 2015, ch. 4, pp. 42–55. DOI: 10.1017/CCO9781139775298.006.

[3] ——, "The acoustics and performance of DJ scratching. Analysis and modeling.", PhD thesis, Kungl Tekniska Högskolan, Stockholm, Sweden, 2010.

[4] A. Sonnenfeld, *Bewegungslehre*. Berlin: Alexander Sonnenfeld, 2011.

[5] ——, *Tonspielzeug website*, Last accessed December 2009, 2009. [Online]. Available: http://www.tonspielzeug.de/.

[6] M. Katz, *Groove Music: The Art and Culture of the Hip-hop DJ*. Oxford University Press, USA, 2012.

[7] T. Beamish, "A Taxonomy of DJs, the online taxonomy of DJ practice", 2004, [Online]. Available: http://www.timothywisdom.com/djtaxonomy.php.

[8] R. I. Godøy and M. Leman, *Musical gestures: Sound, movement, and meaning*. New York: Routledge, 2010.

[9] F. Taniguchi, "A musical analysis of DJ performances in turntablism", vol. 7, pp. 15–34, 2003.

[10] R. Smith, "Take notes", *Electronic Musician*, 2005. [Online]. Available: http://www.emusician.com/how-to/1334/take-notes/35123.

[11] S. Smith, "The compositional processes of UK hip-hop turntable teams", PhD thesis, De Montfort University, 2006.

[12] F. M. Miyakawa, "Turntablature: Notation, legitimization, and the art of the hip-hop DJ", *American Music*, pp. 81–105, 2007.

[13] J. Ouper, "From machine to instrument a composer's perspective of turntables composition", PhD thesis, Arizona State University, 2014.

[14] D. Biederman, "The turntable as a musical instrument and the emergence of the concert turntablist", 2015.

[15] Doc Rice, *Proposal for notation system of turntable music—draft three*, Last accessed December 1999, 1998. [Online]. Available: www . wicked - styles.com.

[16] K. F. Hansen, "Turntablisme - His Master's Voice: The Art of the Record Player", Written in Norwegian, Master's thesis, Department of Music, Norwegian University of Science and Technology, 1999.

[17] Thud Rumble, *Skratchcon 2000*, Thud Rumble: SC 2000-DVD-SMIT, DVD, 2000.

[18] S. Webber, *DJ Skills: The Essential Guide to Mixing and Scratching*. Oxford: Elsevier Science & Technology/Focal Press, 2007.

[19] J. Carluccio, E. Imboden, and R. Pirtle, *Turntablist Transcription Methodology*, Online publication, Last accessed December 2009, 2000. [Online]. Available: http://www.ttmethod.com/.

[20] E. Navas, *Remix theory: The aesthetics of sampling*. Springer Wien, 2012.

[21] M. Katz, *Capturing sound: How technology has changed music*. Berkeley and Los Angeles: University of California Press, 2010.

[22] J. T. Demers, "Sampling the 1970s in hip-hop", *Popular Music*, vol. 22, no. 1, pp. 41–56, 2003.

[23] D. Schwarz, "Concatenative sound synthesis: The early years", *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[24] A. Zils and F. Pachet, "Musical mosaicing", in *Digital Audio Effects (DAFx)*, vol. 2, 2001.

[25] S. Hofer, "'Atomic' music: Navigating experimental electronica and sound art through microsound", *Organised Sound*, vol. 19, no. 03, pp. 295–303, 2014.

[26] T. G. Schumacher, "'This is a sampling sport': Digital sampling, rap music and the law in cultural production", *Media Culture Society*, vol. 17, no. 2, pp. 253–273, 1995. DOI: 10 . 1177 / 016344395017002006.

[27] K. F. Hansen, M. Fabiani, and R. Bresin, "Analysis of the acoustics and playing strategies of turntable scratching", *Acta Acustica united with Acustica*, vol. 97, pp. 303–314, 2011. DOI: 10 . 3813 / AAA . 918410.

[28] L. Haken, R. Abdullah, and M. Smart, "The Continuum: A continuous music keyboard", in *Proceedings of the International Computer Music Conference*, International Computer Music Accociation, 1992, pp. 81–81.

[29] P. Kirn, "Faders as instrument: Playing the Vestax Faderboard", *Create Digital Music*, no. 3, 2007.

[30] J. Lee, "The chopping board: real-time sample editor", in *Proceedings of the Conference on New Interfaces for Musical Expression*, N. Schnell, F. Bevilacqua, M. Lyons, and A. Tanaka, Eds., Centre Pompidou: IRCAM, Jun. 2006, pp. 77–78.

[31] S. Webber, *Turntable technique: The art of the DJ*. Berklee Press, 2000.

[32] DJ Qbert, *DJ Qbert's Scratchlopedia Breaktannica: 100 secret skratches*, Thud Rumble: SECRT001, DVD, 2007.

[33] Fab Five Freddie, *Change The Beat*, New York: Celluloid Records, CEL 156, 12" EP, 1982.

[34] M. Rodriguez, "Expressive quantization of complex rhythmic structures for automatic music transcription", in *TENOR 2015 International Conference on Technologies for Music Notation and Representation*, Paris, 2015, pp. 18–22.

[35] D. Spreadbury and R. Piéchaud, "Standard music font layout (SMuFL)", in *TENOR 2015 International Conference on Technologies for Music Notation and Representation*, Paris, 2015, pp. 147–154.

[36] C. Nash, "The cognitive dimensions of music notations", in *TENOR 2015 International Conference on Technologies for Music Notation and Representation*, Paris: Institut de Recherche en Musicologie, 2015, pp. 191–203.

[37] A. E. Greasley and H. M. Prior, "Mixtapes and turntablism: DJs' perspectives on musical shape", *Empirical Musicology Review*, vol. 8, no. 1, pp. 23–43, 2013.

PITCHCIRCLE3D: A CASE STUDY IN LIVE NOTATION FOR INTERACTIVE MUSIC PERFORMANCE

Tom Hall
Anglia Ruskin University
tom.hall@anglia.ac.uk

ABSTRACT

Recent decades have seen the establishment of computer software live notations intended as music scores, affording new modes of interaction between composers, improvisers, performers and audience. This paper presents a live notations project situated within the research domains of algorithmic music composition, improvisation, performance and software interaction design. The software enables the presentation of live animated scores which display 2D and 3D pitch-space representations of note collections including a spiral helix and pitch-class clock. The software has been specifically engineered within an existing sound synthesis environment, SuperCollider, to produce tight integration between sound synthesis and live notation. In a performance context, the live notation is usually presented as both music score and visualisation to the performers and audience respectively. The case study considers the performances of two of the author's contrasting compositions utilising the software. The results thus far from the project demonstrate the ways in which the software can afford different models of algorithmic and improvised interaction between the composer, performers and the music itself. Also included is a summary of feedback from musicians who have used the software in public music performances over a number of years.

Keywords: notation, interaction, algorithms, music performance, improvisation, software, SuperCollider

1. INTRODUCTION

Musical notation can have various functions, including as a mnemonic, an analysis or transcription. The most common function is prescriptive, in the form of 'instructions' for performers [1, p.100]. Such instructions in musical scores are by convention *symbolic representations* of the musical elements to be sounded, which are then interpreted by performers as actions to be undertaken to perform the required sounds. In the early 1960s Cornelius Cardew saw notation in terms of a hierarchy of rules, lamenting that interpretation of Western classical music relied on many of these rules being implicit [2]. This learned implicit (and indeed often embodied) knowledge can be considered as notationally contextual information, for example stylistic and/or historical performance practice and so on.

Live notation and digital scores in most cases are created using computer technologies and offer advantages for some kinds of music over 'analogue' (paper) scores. Reliability is a key advantage of paper scores over digital notation (technology often breaks in concerts). Paper scores are also convenient for annotation (musicians bring pencils to rehearsals), however, once printed, are otherwise notationally fixed. On the other hand, live, dynamic notation can enable a 'just-in-time' notational approach, allowing notations digitally assembled in realtime. This is useful for improvised or algorithmic music, in which the notation is able to reflect algorithmic procedures or composer live mediation, for example. Computer technology also has the advantage in making *sharing* notations with an audience trivial via projection, of which more below.

2. TYPES OF DYNAMIC DIGITAL SCORES

The notation of a 'musical score' is often synonymous with Common Western Notation (*CWN*), a highly evolved and efficient method of indicating musical intention within the Western music tradition [3]. However aside from notations involved in non-Western music and those of early Western music, there is now a century of 'non-standard', often experimental scores usually known as graphic scores. The degree to which non-standard notation employs elements or conventions from CWN varies considerably. Whilst on the one hand graphic scores often require lengthy textural performance directions, on the other they may also rely on a performer's more general ability to read 'iconic depictions' of various sorts [1, p.130]. Mortan Feldman's 'graph paper' scores of the 1950s, for instance, can be understood within a long tradition of human culture around 'grids' [4]. Of course, different approaches to graphic music notation offer design trade-offs in terms of representation of musical parameters, as discussed further below.

Live digital notation for music goes under various names: 'realtime-score' [5], dynamic digital scores, 'screen scores' and others. Vickery correctly draws a historical connection here to traditions within experimental 20th Century scores, including the 'mobile scores' of Earle Brown [6]; in the computer music domain, experiments by Max Mathews at Bell Labs in onscreen musical notation by computer are also relevant here. The number of approaches available to screen scores are clearly numerous. Vickery categorises two main types, distinguishing between scrolling and 'segmented scores' on the one hand, and 'realtime scores' (permutative, generative and transformative) on the other [p.131]. Whilst this is a useful start, it could be argued that these are ultimately overlapping categories and live notations may often as not observe such distinctions in the breach as otherwise. This is implicit in the following discussion, which introduces the software used in this project, before contextualising its use in performance projects.

3. PITCHCIRCLE3D

PitchCircle3D is series of custom classes written in the SuperCollider programming language [7]. SuperCollider is an established and sophisticated interactive programming environment for realtime computer music synthesis. Whilst PitchCircle3D can simply be used to visualise note music, a stronger motivation for its development is to enable the sharing of a form of non-specialist music notation with performers and audience alike. Elsewhere I argue that this is broadly inline with a post-war desire towards transparency of communication in art [8]. PitchCircle3D is implemented within SuperCollider as a 'system within a system' [9]. A motivation for this is to allow, to use Leman's term, 'micro-integration' [10, p.3] with the SuperCollider's audio synthesis engine. This in turn enables responsive electronic music performance with very little latency.

PitchCircle3D uses SuperCollider's cross-platform GUI environment (implemented in Qt) to display live, animated non-standard music notation in the form of notes and chords in 12-tone equal temperament (12-TET), as shown in Fig. 1. The notation view is animated at a customisable frame rate which can be updated in realtime within the PitchCircle3D class via SuperCollider's interactive programming environment. The PitchCircle3D class knows nothing of SuperCollider's audio synthesis, use of the class is usually incorporated into individual SuperCollider code as required for each composition. The allows PitchCircle3D to be used in conjunction with most of the many coding styles available to within the SuperCollider language, sclang. Whilst all the music discussed in this paper has been coded in SuperCollider, it should be noted that PitchCircle3D also can be easily configured within SuperCollider to respond to external control through eg MIDI or OSC messages. In the case of the latter, this is discussed further below.

PitchCircle3D currently has three notational views available. A 3D spiral helix illustrates relative register, shown over three octaves in Fig. 1. A 'pitch clock' shows pitch-classes, omitting registral information (see Fig. 2, which shows a pitch-class set view of the pitches in Fig. 1). A third view flattens the view to a 2D spiral, as seen from above and shown in Fig. 3. These different views are relatively trivial to achieve within the class as they are derived from matrix operations on an identity matrix. [1]

Animation of PitchCircle3D's views includes code methods to smoothly tilt, rotate and zoom views, programmatically or by mouse interaction (the former are achieved using 'easing' functions). In each view, small discs represent potentially sounding pitches, by default connected by a line passing through each. Ordinarily the notation indicates corresponding pitch-classes with a separate colour for each pitch/pitch-class (the colour scheme is customisable). Pitch-class or note names are also indicated in each view, allowing the role of note colours to represent other musical parameters if desired (eg dynamics). An additional small circle around a disc is available (shown in Fig. 1 on the lowest note C) to indicate a point of focus or emphasis, for instance a tonic or area of pitch centricity.

[1] This part of the class is based on the SuperCollider Canvas3D quark by Jonatan Liljedahl and Fredrik Olofsson. In the PitchCircle3D classes, however, mathematical transformations are decoupled from SuperCollider's Pen drawing class.

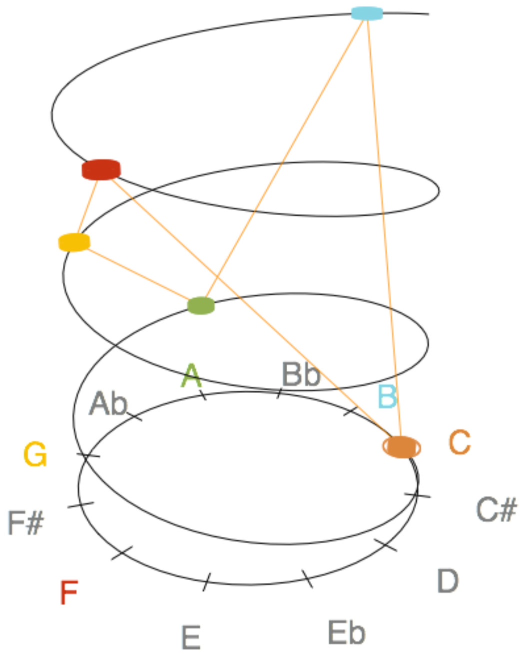


Figure 1. PitchCircle3D spiral notation view.

Notes can be entered and removed individually or in groups, faded in and out at a desired rate (in seconds) and displayed for a specified duration, starting either immediately or at some future point in time. These operations create a series of 'time points' to structure musical progression according to predefined or algorithmic sequences. As a small and simple example, the following SuperCollider code can be used to begin an instance of the PitchCircle3D class within SuperCollider and fade in, over two seconds, a number of notes (as MIDI numbers) to the view. These are connected by default via a line, in the oder listed in the array. The rotate and tilt methods can be used to then generate the view shown in Fig. 1 (these shifts can also be animated over time, something which will be discussed further presently). To clear the view and close it, the methods shown below are used.

```
p = PitchCircle3D.new(numOcts:3);
p.front;
p.addComplex([60, 95, 69, 79, 89], 60, 2);
p.rotateTo(9.5, 1); // rotate over 1 second
p.tiltTo(5.9);
p.clearAll; // remove discs immediately
p.close; // close window
```

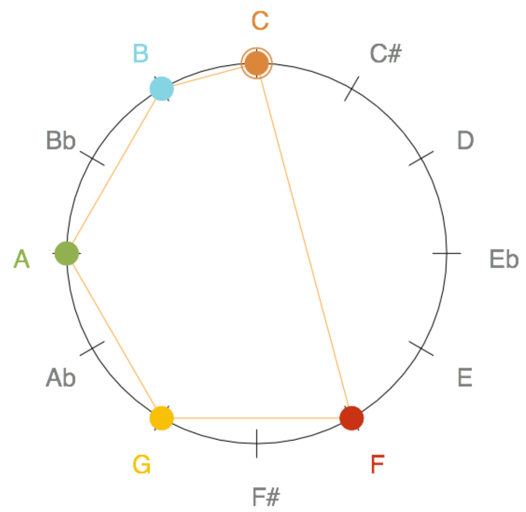


Figure 2. PitchCircle3D clock notation view.

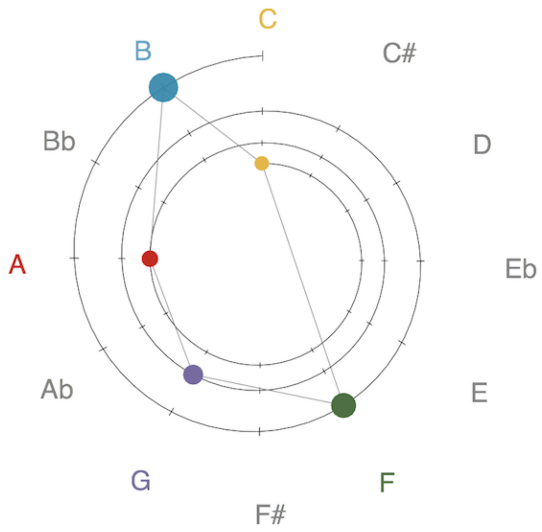


Figure 3. PitchCircle3D 2D spiral notation view.

The notational representations available in PitchCircle3D are in themselves not novel, although their particular implementation and the software's performative context offers affordances. Related software include iPhone apps *Music Set Theory* [11], which presents an interactive pitch-class clock view for the classification of set-class names and the display of their complements according to Alan Forte's naming system. The closest relation to PitchCircle3D is perhaps Chew and François' software MuSA.RT, which also displays pitches around an animated 3D spiral helix, and can do so using live MIDI input [12]. [2] Whilst it shares many of the notational concerns of PitchCircle3D, it focuses on illustrating a specific theory of the analysis of tonal music. It thus appears that MuSA.RT may be categorised more as a visualisation tool for live musical analysis rather than as software for displaying live notation as a digital musical score for performance.

[2] PitchCircle3D can respond both to external MIDI and Open Sound Control (OSC) messages via core SuperCollider capability.

4. CASE STUDIES: PITCHCIRCLE3D IN PERFORMANCE

4.1 *All the Chords*: interaction and improvisation

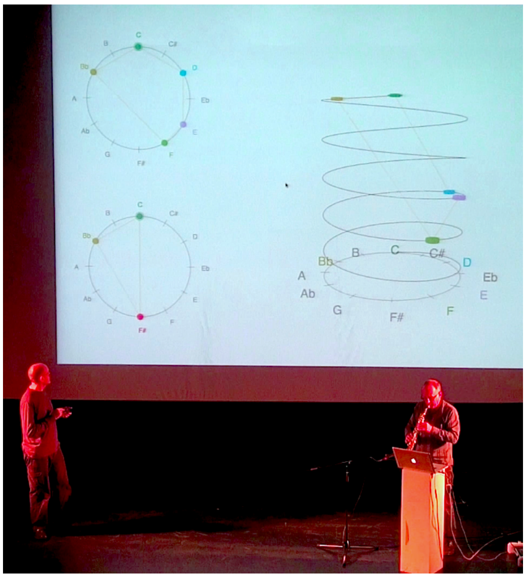


Figure 4. *All the Chords* on-stage configuration of performers, computers and notation.

In 2014 PitchCircle3D was used in a configuration for performance of the author's composition *All the Chords*, involving an instrumental musician, Kevin Flanagan (saxophone), and computer performer (the composer). In performance, PitchCircle3D was used in full-screen mode, and a 'mirrored' screen projected on the rear of the stage. The instrumental performer mediated aspects of the performance, viewing the notation on the screen of the laptop running SuperCollider, whilst the author as computer performer referred to the notation on the rear projection, as shown in Fig. 4. SuperCollider was used to both display the live notation using PitchCircle3D, as well as to synthesise a computer music part. This computer music part comprises a drone, an occasional bassline and pulse emphasised by a synthesised percussive layer. The main musical material in *All the Chords*, however, is a predetermined sequence of collections of notes ('chords'), all the possible subsets of a superimposed major and parallel harmonic minor scale algorithmically ordered according to SuperCollider's powerset method.

To aid sight-reading by the instrumental performer, during the performance, two collections are displayed. The current collection of notes are shown in spiral helix form and also in pitch-clock on the top left of the screen (see Fig. 4). On the bottom left of the screen, the *upcoming* collection of notes is shown, allowing the performer(s) to prepare as necessary for the next note collection of the composition before it arrives.

The computer-performer communicates with the main computer laptop via the Open Sound Control (OSC) protocol. Rather than achieving this via a second laptop, communication with SuperCollider is via a mobile device in order to enable direct on-stage communication with both the instrumental performer *and* appear most present to the audience. To achieve this, the mobile device runs a customised layout of the TouchOSC app [13], and amongst other things allows interaction with the stratified compositional layers of electronic musical elements (chords, drones, arpeggiations, bass notes and overall mix volume). There is also a button to toggle between manual advancement of chord collection frames and a chordal 'autopilot' setting, using a predetermined algorithmic sequence of time intervals specified within the SuperCollider code.

As a representation of musical events, PitchCircle3D's notations combine aspects of both indeterminate and fully determined events. The software's default spiral helix view is fully determinate in terms of its notation of pitch-space, whilst its clock view, as shown in Fig. 2, presents pitch-class space only, necessarily omitting registral information (this figure represents the same pitches as those in Fig. 1). Whilst it might be expected that the choice between displayed views would depend on the level of pitch determinism required, experience in performance has found a friction in reading the spiral helix view quickly and without error (see performer feedback section below). Thus even where pitch is fully determined, in *All the Chords* the spiral helix view is presented simultaneously with the corresponding clock view. This configuration can be seen on the top left hand side of Fig. 4.

The rhythm and tempo of musical events are notationally unspecified in *All the Chords*. This notational constraint within the current PitchCircle3D model of musical representation, however, encourages certain approaches to rhythm and temporality which have been exploited in the music written for the system by the author. In *All the Chords*, musical rhythm (in the sense of sequence) is represented at a higher level in the timing of display transitions ('time points'). Within these time points, rhythms are freely improvised by the instrumental performer around the note collections displayed. In this way, the notes shown are also freely interpreted as material for melodic improvisation, since ordering of each collection is not indicated. Musical continuities are created in the piece by linking these collections via common-tones across time points.

As can be seen in Fig. 4, lines are drawn between notes in each collection, in order to emphasis interval relationships. (This feature works most clearly in PitchCircle3D's pitch-class clock view and is turned off in *Untitled #1*, the composition discussed below, as it exclusively used the spiral helix view.) The pitch-class of the current drone sounding in the electronic part is indicated via the ring around relevant 'note' in the displayed notation. Dynamics for the instrumental performer are notationally unspecified in *Untitled #1*.

In this overall approach to musical temporality, the design of PitchCircle3D sits well with established models of musical improvisation [14]. The following very brief summary introduces the key ideas: Pressing's model divides improvised music into sequential 'event clusters' divided by time points, usually demarcated by 'local musical boundary criteria' including pauses and other phrase junctures [14, p.153]. Musical continuation within and between clusters is determined by 'associative' or 'interrupt' generation across musical parameters [14, p.155].

Overall the project sits between other recent approaches in digital notation that are more indeterminate (graphic notation), or fully determinate—for example employing CWN (such as [15]). Design decisions behind PitchCircle3D offer clear constraints for performance (what notes to play), but leave others relatively open (when and how to play), a mode of performance well documented since at least the 1960s [16]. This notational indeterminacy of PitchCircle3D can be regarded as an affordance, leaving as it does considerable room for collaborative musical improvisation, as noted above.

In the 2014 performance using the software, temporal constraints of the musical improvisation were partially determined by the duration over which each collection was displayed. As discussed above, the duration was in turn determined either algorithmically, or through mediation by the computer musician. Likewise, timings of musically noted material and the relationships between this material thus influenced whether event cluster continuations were associative or interrupt-driven.

These musical decisions and outcomes were the result of the collaborative nature of the musical improvisation, the music of which reflected the musical interests of both parties. In particular Flanagan's improvised jazz vernacular was a clear feature of the performance, constrained as it was by the harmonic material presented to him, yet hopefully still affording the 'intrinsically explorative nature' of improvisation [17, p.53]. Fig. 5 illustrates the feedback between the two performers, the digital notation, and the sounding musical performance (influenced by Nash and Blackwell's approach to diagramming user interaction within music software [18]). Note that the majority of these interactions function as iterative feedback loops which may operate on multiple timescales in relation to the Pressing's model of improvisation presented above.

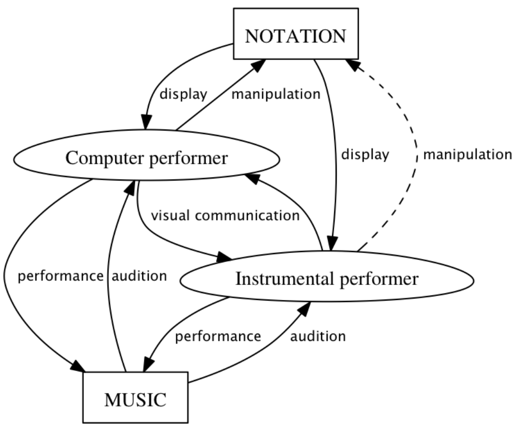


Figure 5. Interactions between performers.

4.2 *Untitled #1*: algorithmic processes

Untitled #1 (2015) is a 15 minute composition for computer performer with computer sound, and two amplified instruments played monophonically. The 2015 first performance at Anglia Ruskin University (2015-04-17) paired electronic guitar with E-bow, performed by Will Crosby, and amplified 'cello sounding only natural and artificial harmonics, performed by Cheryl Frances-Hoad. In this performance the author as composer-computer performer stayed off-stage, using a MIDI controller keyboard attached to a laptop. The electronic music part of the composition was coded in SuperCollider and used PitchCircle3D to display the live notation via a projector. The two instrumental performers were on either side of the stage, forming a triangle with the live score, as is shown in Fig. 6.

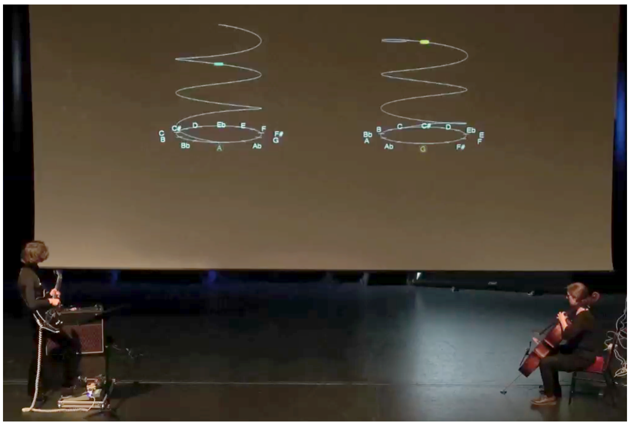


Figure 6. Stage layout for Hall's *Untitled #1 (2015)*.

As should be clear from Fig. 6, the live notation is laid out as two musical parts, one for each performer. This was deemed necessary as overlapping instrumental entrances and exits would have made reading both parts from a single spiral helix overly difficult. Dividing the screen real estate into two in this way made a separate 'look ahead' event cluster frame as used in *All the Chords* impractical. However this constraint was offset by the very slow rate of musical change in this composition, enabling performer cognition of the next note to be played.

Each instrumental part is monophonic, enabling the playing of artificial harmonics in the 'cello part and the use of the E-bow to achieve infinite sustain on the electric guitar (E-bows can only play a single note at a time). The onset and duration of notes is indicated notationally in two connected ways in in *Untitled #1*, which also indicates relative dynamics within notes to the performer. For each part, all notes slowly fade in and then fade out. This is indicated by the changing transparency in real time of the displayed note. The helix is rotated such that when the note displays full colour / amplitude, the horizontal alignment of the helix displays the note at the front of the 3D view.

The musical structure of *Untitled #1* takes particular advantage of the live notation enabled by PitchCircle3D. The algorithmic nature of the composition centres on a relatively simple series of harmonic progression rules which determine in real time the sequence of harmonies within an overall simple modal harmonic framework. These sequences of harmonies occurr either in the live performer parts, the electronic accompaniment, or both.

		Performance	
		Every Chord	*Untitled #1*
Frame	Sequence	determined	algorithmic
	Timing	determined	algorithmic
Intra-frame	Sequence	improvised	determined
	Timing	improvised	determined

Table 1. Algorithmic and improvised musical structures afforded by live notation

Unlike *All the Chords*, in the newer composition a fixed electronic pitched drone establishes a dorian musical centricity throughout. Further differences between the approaches to notation and structure between the two compositions are discussed in the next section.

5. NOTATIONAL STRUCTURES AND INTERACTIONS USING PITCHCIRCLE3D

The foregoing has discussed approaches to using the PitchCircle3D software for the performance of two compositions employing live notation in two different and distinct musical ways. These approaches to musical structures and musical performance has required rethinking the role of the software as live notation for each composition.

Table 1 presents a high level summary of these differences and shows the switch in determinate (fixed) and indeterminate (algorithmic and improved) musical elements in the structural levels of the two compositions under discussion. Here the notion of a 'frame' is one delineated by time points, and 'intra-frame' that of possible event clusters within each animated view (for instance, improvisation on a note collection). At a lower level, relative dynamics ('hairpin' fades) are represented in one of the compositions, but not in the other.

Just as algorithmic music in general challenges the notion of the fixed work, live notation further blurs lines between the composition of a work and its performance, opening up possibilities for structured improvisation between performers coordinated by composer mediation or algorithmic control. Fig. 7 shows the role of live notation in this situation, in which brackets show optional characteristics within compositional and performance processes.

6. PERFORMER EXPERIENCES OF USING PITCHCIRCLE3D IN PERFORMANCE

Four musicians who have performed using PitchCircle3D in public responded to a short written questionnaire about their experience using the software. All had had some previous experience performing using non-standard notation. Each performer had also had some experience with algorithmic and/or semi-improvised music and there were a number of interesting reflections on this. For instance performer A commented on 'serendipitous moments that would have

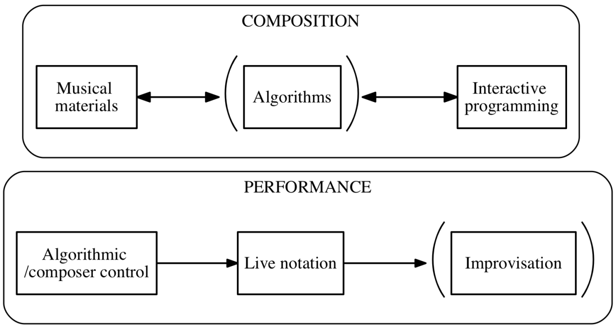


Figure 7. Role of live notation in composition and performance.

never occurred [using other methods]', while performer B claimed that there 'tends to be unspoken stylistic assumptions with each piece' which an improviser must position herself with respect to.

The performers were also asked about the general experience of performing with PitchCircle3D. Performer A stated that, 'once you get your head around it, it feels very intuitive', however commented that tracking note changes required intense concentration at times—being on 'high alert'. Similarly performer B experienced the demands of live notation as enjoyable and viewed this in terms of a challenge around musical (improvisational) inventiveness which 'added to the spontaneity of the performance'. Performer C also considered the novelty of interacting with the live notation 'logical' yet challenging, and experienced this in terms of embodied responses (the physicality and instinct required in performance). Performer D also considered the physicality suggested by the notation, relating the polygons shapes of the 2D form in terms of hand shapes for a keyboard 'guided improvisation'.

Performers were asked about their ability to distinguish between aspects of the notation in each of the two PitchCircle3D views, the 3D spiral helix and the 2D clock. Performers A and C reported that identifying the indicated note correctly was helped by the individual note colours, and the slow rotation which centred the upcoming note to the front of the view. For performers B and D correct identification was reported as more difficult, however it should be noted that the rate of change and number of notes in those performances was much greater. There were mixed responses as to the ease of identifying the correct octave of a note in the spiral helix view.

In some performances, as noted above, performers were presented with both 2D and 3D views of the current note collection for performance, as well as a 2D representation of the upcoming collection. There was a varied response from all four performers as to the usefulness of both of these features as experienced or imagined. Performer D made the observation that the efficacy of these features would depend on on the time available to use them in performance, i.e. the overall level of musical activity. Likewise, performer B commented on this in relation to the potentially different kind of continuities that might be achieved or desired, depending on whether a performer was able to 'read ahead' from the upcoming collection view.

7. CONCLUSION

Writing in 1961, Cornelius Cardew suggested that 'notation should put the player on the right road' [2, p.31]. This metaphor for moving in time in a defined direction sits well with the aims of PitchCircle3D, which has proven to be a flexible tool for displaying and sharing live notation in different musical contexts. PitchCircle3D's implementation in SuperCollider allows tight integration with audio synthesis, and the resulting realtime capabilities have affordances for live algorithmic computer music in tandem with live instrumental performance.

Ongoing software development of PitchCircle3D forms part of a reflective shared practice-led project between the software's author and instrumental performers who use PitchCircle3D in performance. Investigation of the effectiveness of this environment for both flexible and specialist means of communication and sharing between performers and audience forms part of this research context. The results thus far have demonstrated a number of models of performative and compositional interaction as outlined in the case study above. Questionnaires conducted with musicians who have used the software have demonstrated generally positive results, with, however, a common experience being that its use in performance can make cognitively high demands on the musicians.

Future work will aim to reduce some of the friction in the reading of this notation by the performing musicians, as well as explore further options for collaboration between performers. In particular, it might be interesting to involve performers in reciprocal musical interaction with the software—including the ability of the instrumental performer to influence the live notation and possibly the electronic music if desired. Such developments could also be judged against audience reaction analysis as well as further feedback from performers.

The most obvious inbuilt notational constraint within the PitchCircle3D model is that its notations currently provide no rhythmic information except through the realtime temporality of time points structuring a performance. Whilst such indeterminacy might be regarded as an inbuilt limitation of the system as a performance notation, the approach taken maps well onto existing practices of musical improvisation as discussed above. Nevertheless, future research will leverage further possibilities for musical parameter representation within PitchCircle3D. This is intended to enable more varied musical improvisation around flexible live musical structures.

Acknowledgments

Thanks to the performers Will Crosby, Kevin Flanagan, Cheryl Frances-Hoad and Richard Hoadley for permission to include their questionnaire responses in this paper. Thanks also to Alan F. Blackwell, who contributed to an earlier stage of this research, and to the three anonymous reviewers for their helpful suggestions.

8. REFERENCES

[1] S. Davies, *Musical Works and Performances / A Philosophical Exploration*. Oxford: Oxford University Press, 2001.

[2] C. Cardew, "Notation, Interpretation, etc," *Tempo*, vol. 58, pp. 21–33, Summer 1961.

[3] R. Rastall, *The Notation of Western Music / An Introduction*. London: J.M. Dent, 1983.

[4] H. B. Higgins, *The Grid Book*. Cambridge, MA: MIT Press, 2009.

[5] G. E. Winkler, "The Realtime-Score. A Missing-Link in Computer-Music Performance," in *Sound and Music Computing '04*. Paris: IRCAM, October 2004.

[6] L. Vickery, "The Evolution of Notational Innovations from Mobile Score to Screen Score," *Organised Sound*, vol. 17, no. 2, pp. 128–136, 2012.

[7] J. McCartney, "Rethinking the Computer Music Language: SuperCollider," *Compuer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002.

[8] T. Hall, "Sharing Electronic Music Performance," in *Visualise: Making Art in Context. Cambridge*, B. Ferran, Ed. Anglia Ruskin University, 2013, pp. 60–63.

[9] J. Rohrhuber, T. Hall, and A. de Campo, "Dialects, Constraints and Systems within Systems," in *The SuperCollider Book*, N. Collins, D. M. Cottle, and S. Wilson, Eds. Cambridge, Mass.: MIT Press, 2011.

[10] M. Leman, *Embodied Music Cognition and Mediation Technology*. Cambridge, Mass.: MIT Press, 2007.

[11] P. Couprie, "Music set theory," http://logiciels.pierrecouprie.fr/.

[12] Chew, E., and François, A. R. J., "Interactive Multi-Scale Visualizations of Tonal Evolution in MuSA.RT Opus 2," *ACM Computers in Entertainment*, vol. 3, no. 4, p. 16, 2005.

[13] Hexler, "Touchosc," http://hexler.net/software/touchosc.

[14] J. Pressing, "Improvisation: methods and models," in *Generative Processes In Music / The Psychology of Performance, Improvisiation and Compostiion*, J. A. Sloboda, Ed. Oxford: Clarendon Press, 1988.

[15] R. Hoadley, "Calder's Violin: real-time notation and performance through musically expressive algorithms," in *Proceedings of the International Computer Music Conference*, IRZU, Ljubljana, Slovenia, September 9-15 2012, pp. 188–193.

[16] D. Behrman, "What Indeterminate Notation Determines," *Perspectives of New Music*, vol. 3, no. 2, pp. 58–73, 1965.

[17] D. Bailey, *Improvisation / Its Nature and Methods*. Boston, Massachusetts: Da Capo Press, 1992.

[18] Nash, C. and Blackwell, A. F., "Liveness and Flow in Notation Use," in *Proceedings of New Interfaces for Musical Expression NIME*, University of Michigan Ann Arbor, MA, 2012, pp. 28–33.

PERFORMANCE PRACTICE OF REAL-TIME NOTATION

Seth Shafer
University of North Texas
SethShafer@my.unt.edu

ABSTRACT

This paper addresses the performance practice issues encountered when the notation of a work loosens its bounds in the world of the fixed and knowable, and explores the realms of chance, spontaneity, and interactivity. Some of these performance practice issues include the problem of rehearsal, the problem of ensemble synchronization, the extreme limits of sight-reading, strategies for dealing with failure in performance, new freedoms for the performer and composer, and new opportunities offered by the ephemerality and multiplicity of real-time notation.

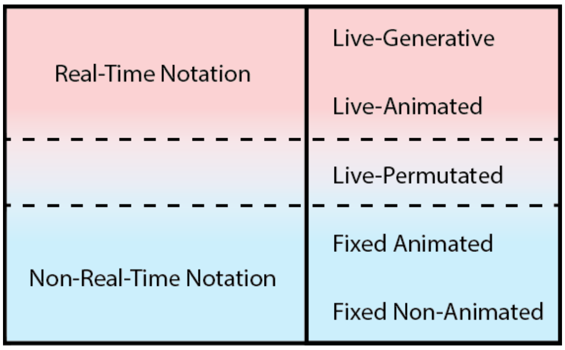


Figure 1. Categories of real-time and non-real-time music notation.

1. REAL-TIME NOTATION

The issue of permanency in notation immediately evokes a continuum bounded by pre-determined paper scores at one end and free improvisation on the other. Gerhard Winkler suggests that between these two extremes lies a "Third Way" made possible by recent technologies that support various types of real-time notation [1]. This emerging practice of using computer screens to display music notation goes by many names: animated notation, automatically-generated notation, live-generative notation, live notation, and on-screen notation. These new notational paradigms can be separated into two categories: real-time notation and non-real-time notation (see Figure 1). Real-time notation encompasses scores that contain material open to some change during the performance of the piece. Many works fit this definition, from those that use predetermined musical segments that are reordered in performance to those that are completely notated in the moment of performance. Non-real-time notation accounts for all other uses of the computer display as a notational medium. Both static and animated scores occupy this category. The boundary between these two primary approaches to notation on the computer screen is not rigid and a technique like the live-permutated score can be argued to fit in either category.

It is useful to further categorize an on-screen work by its attributes. These attributes are found in both real-time and non-real-time scores: notation style, interpretive paradigm, time synchronization and location tracking management, degree of on-screen movement, whether the performer reads from a part or a score, and if there is non-notational interactivity (see Figure 2). Notation style refers to the spectrum between traditional symbolic notation and graphic notation. Many real-time notation scores use graphic notation or a combination of traditional symbols and abstract graphics. The interpretive paradigm of a piece determines whether the performer does strict music reading or uses some degree of improvisation to interpret the notation. The method of time synchronization, location tracking, and the amount of on-screen movement can be important in solo and ensemble pieces reading from a computer screen. Relying on eye-movement research, Lindsay Vickery [2] and Richard Picking [3] conclude that common approaches like the playhead-cursor and the scrolling score are unnatural for the performer to follow. I argue for a bouncing-ball-type tracker that embodies expressive and anticipatory tempo information drawing on a performer's skill of following a conductor [4]. The question of whether the performer reads from a part or score has implications for ensemble coordination and the visual size of the music. Works using real-time notation often incorporate non-notational interaction through audio or video processing. In addition to the challenge of real-time notation, the performer must grapple with the issues associated with *musique-mixte* and interactive electroacoustic music.

2. ON THE LACK OF PERFORMANCE PRACTICE GUIDES

The performance practice issues of real-time notation share connections with open form music, indeterminacy, complexity, free improvisation, and interactivity. These issues and their associated challenges pose a formable hurdle for many performers. Many composers have incorporated real-time notation in their practice despite the inherent difficulties. Some have written extensively on the topic of real-time

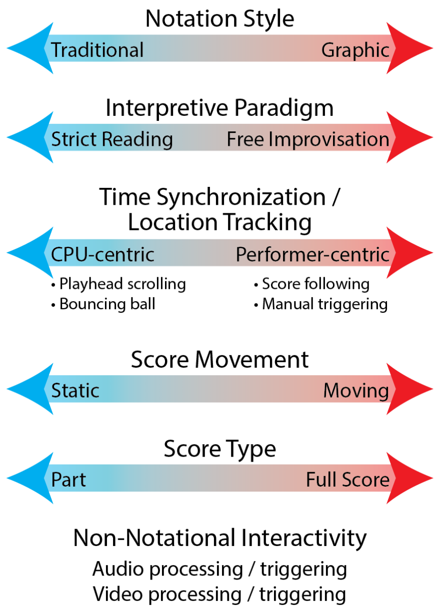


Figure 2. Attributes of the real-time score.

notation in an effort to detail new software in the field or to explain the technological or theoretical underpinnings of a new work. With some notable exceptions, few have presented the problems and newfound freedoms that the performer faces in performing such works. Jason Freeman's "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance" is an excellent first attempt at developing a comprehensive guide for the performer [5]. However, Freeman fails to go far enough when describing performer psychology in both the rehearsal and performance experience. In addition, his definition of real-time notation is limited to synchronized ensemble improvisation and audience participation. Freeman largely ignores real-time scores that employ traditional notation symbols.

Many composers and technologists include small reports of performance practice in their research, often mentioned as an ancillary issue. Such remarks read like the following: "The best way to approach the playing of a Real-Time-Score seems to be that of a relaxed, playful 'testing' of the system" [6]. This type of suggestion ignores the real barriers for performers approaching real-time notation and often comes across as composer-knows-best. The trust required between a composer, performer, and a work that exhibits notational agency is not a thing to be taken lightly and requires an in-depth study.

3. NEW FREEDOMS FOR MUSICAL EXPRESSION

3.1 Freedom From Replication

The composer or performer viewing real-time notation from a distance might rightfully wonder in what ways the added challenges of real-time notation can ultimately benefit a composition. Real-time notation affords both composer and performer with new freedoms in live performance and new means for musical expression.

One freedom is the release from the burden of replication. Since the advent of the phonograph, recorded performances have imparted an increasingly weighty tradition on the shoulders of each generation of performers. Not strictly relegated to the hallowed ranks of common practice music, recordings of contemporary compositions by esteemed new music performers become authoritative in a way that was perhaps unintended. Issues related to the archival worth of such documents aside, composer-endorsed recordings become a type of *urtext* (an *urklang* perhaps) and an immediate arbiter of what is an "authentic" performance of a piece.

Remarking on authenticity and values in common practice music, Bruce Haynes lists ideals that are ever increasingly found in new music:

> The shortlist of "Masterpieces" that it plays over and over, repeatability and ritualized performance, active discouragement of improvisation, genius-personality and the pedestal mentality, the egotistical sublime, music as transcendent revelation, *Absolute Tonkunst*…ceremonial concert behavior, and pedagogical lineage [7].

Those ideals contrast those that Haynes asserts ruled musical events before the nineteenth century:

> That pieces were recently composed and for contemporary events, that they were unlikely to be heard again (or if they were, not in quite the same way), that surface details were left to performers, that composers were performers and valued as craftsmen rather than celebrities…and that audiences behaved in a relaxed and natural way [7].

By extension, these ideals might have something to say about works written today. Paul Thom affirms this line of thinking when he says, "An ideology of replication leaves no room for interpretation; and yet interpretation is a necessity…in performance" [8]. Works using real-time notation offer freedom from the shackles of authenticity and the burden of being measured against recordings by creating a situation that defies (even undermines) replication.

3.2 Improvisational Freedom

While the variable nature of real-time notation guarantees diversity in the source material, it also grants a degree of creative license to the performer through improvisation. Many real-time notation works use graphic notation to guide a performer through improvisation. Karlheinz Essl's *Champ d'Action* (1998) uses a combination of on-screen text and graphic symbols to elicit group improvisation (see Figure 3). Written for an unspecified ensemble of between 3 to 7 soloists, the musicians respond to live-generated universal parameter instructions that must first be translated to their instruments before attempting the loftier goal, "to create relationships by listening and reacting to the sounds that are produced by the other players which could lead to dramatic and extremely intense situations" [9]. Essl describes the piece as a, "real-time composition environment for computer-controlled ensemble," [9] indicating the open-form nature of the work and his relinquished compositional agency to computer spontaneity and performer creativity.

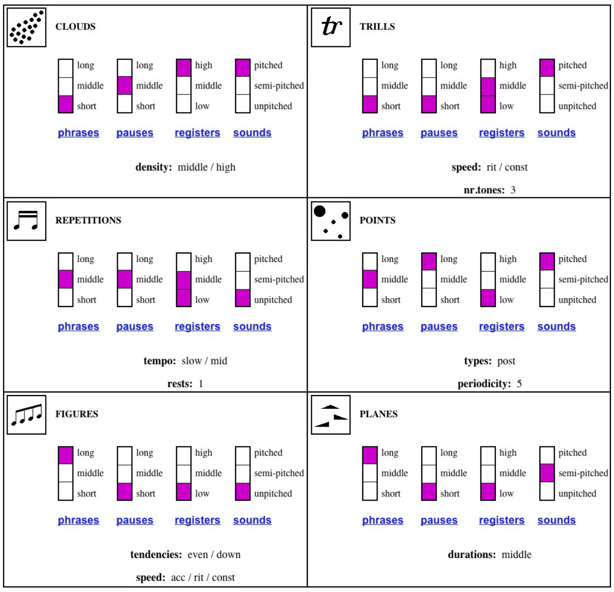


Figure 3. Computer-generated instructions in Karlheinz Essl's *Champ d'Action*. Used with permission.

3.3 Interactive Freedom

The freedom of direct interaction between computer-generated notation and performer is related to improvisation. Given the appropriate circumstances, the performer can assume direct control over the content of their own notation or the notation of another performer. This is the case in Jason Freeman's *SGLC* (2011) for laptop ensemble and acoustic instruments, in which the laptop ensemble chooses and modifies pre-composed musical fragments for the instrumental ensemble to perform in real-time [10]. While Freeman urges each performer to familiarize themselves with the pre-composed material, he gives complete agency to the laptop performers to create loops, add or subtract notes, change dynamics, transpose, and otherwise alter the notation. In this particular piece, the relationship between laptop performer and instrumental performer can appear adversarial; the instrumental musician is at the mercy of the laptop "re-composer." Freeman counters this initial impression by encouraging pairs of laptop and instrumental performers to rehearse separately, becoming familiar with each other's behaviors and abilities, before attempting an ensemble rehearsal: "This unusual setup encourages all of the musicians to share their musical ideas with each other, developing an improvisational conversation over time" [10].

Freeman's approach to notational improvisation is representative of new interactions made possible in real-time notation. This type of interaction can be labeled permutative interaction, where pre-composed segments are reordered. Other new categories of interaction include formal interaction, where the performer can influence aspects of the large-scale structure of a piece; temporal interaction, where rhythmic augmentation and diminution or tempo modulation can change dynamically; and local interaction, where surface details of a piece like pitches, rhythms, dynamics, articulations, and other expressive elements become dependent on performer input. These are but a sample of the new types of notational interaction made possible by abandoning fixed notation.

3.4 Ephemerality and Multiplicity

In the age of abundant documentation, societal pressures to package, brand, and sell a finished artwork choke out the ephemerality of music making. While space limits a fuller discussion of the beauty of impermanence, real-time notation offers a solution to this philosophical and moral problem in the form of multiplicity: each performance presents only one possible version of a piece that exists in plurality. To know one performance is to know only part of the whole. From the performer's standpoint, each performance is unique, free from any historical burden of the past and any comparative critique in the future. The music exists only as it is performed, as any documentation inherently fails to fully represent the work.

Winkler compares the composer of a real-time score to a gardener, "who plants 'nuclei' or germs, and watches them grow, depending on influences from the environment, in this or that way. All versions are welcome" [1, p. 5]. John Cage remarked about his *Concert for Piano and Orchestra* (1957–58) that every performance contributes to a holistic understanding of the work: "I intend never to consider [the work] as in a final state, although I find each performance definitive" [11]. Richard Hoadley asserts that the process is similar to mapping the landscape of a geographic territory without describing every rock, tree, and bush [12]. In this way, the composer acts as cartographer, creating a landscape and releasing the performer to explore its details.

4. PROBLEMS IN REHEARSAL AND PERFORMANCE

4.1 Traditional Purposes of Practice and Rehearsal

With new freedoms for interaction and improvisation and without concerns about replication in light of the ephemerality and multiplicity of real-time notation, come the practical issues that face musicians in rehearsal and performance. Before exploring some new ways to approach practice and rehearsal, the obvious and less apparent purposes of traditional, fixed notation works should be stated. The most prominent purpose of practice is to learn the details of a piece. Some performers describe their practice trajectory as first translating notational language into physical gestures, gradually linking larger and larger musical units together, culminating in a large-scale coherent interpretation [13]. Other performers may follow the opposite path, beginning from a theoretical understanding of the entire work and moving towards mastering the details of each moment. In either case, what is necessary is an understanding of both the specific and the general, the micro and the macro.

The rehearsal process involves other players and presupposes the micro-macro knowledge gained in private practice to develop an understanding of ensemble interaction. Rehearsal with an electronic component or interactive computer part adds complication. Often in the case of interactivity, rehearsal time is spent navigating the technological prosthetics involved (microphones, loudspeakers, pedals, sensors, and other devices), the temporal modalities employed (fixed, fluid, or interactive accompaniment), and the behaviors of the computer agent (traditional score following, coordinated live-input processing, active human-computer joint improvisation, and so forth.) [14].

4.2 New Purposes of Rehearsal with Real-Time Notation

Many of the traditional purposes for practice and rehearsal fall away in works using real-time notation. One of the primary hindrances for newcomers to real-time notation is the unfamiliar process of rehearsing. Why rehearse when the notation changes in the moment of performance? The answers that follow do not pertain to every existent work, but are a list of possible reasons for and approaches to rehearsal.

Instead of practicing a work to transcend the physical actions of the surface details to an informed interpretation of the whole, the performer must engage with the real-time score paying attention to behaviors. Much like how the performer of interactive computer music rehearses with the computer to investigate the designed functions, a work using interactive notation can be built with specific responses to human input or a temporally-cued score. These behaviors can be studied in two ways: with an eye for general local detail and with an eye for general large-scale form. The local detail can be as simple as discovering a set of pre-composed fragments, or it can be as complex as deducing the frequency of rhythmic figures, probability of pitches, or variety of graphic indications. In my quartet for viola, bass clarinet, marimba, and computer, *Law of Fives* (2015) [15], a limited number of predetermined pitches are probabilistically selected and assigned to algorithmically-designed rhythmic structures (see Figure 4). In this piece, the pitches are predictable while their order and associated rhythms are variable. Local details can depend on performer input and the rehearsal process defines the way in which the input affects the notational output. In *Law of Fives*, increased dynamic input from one instrument influences the likelihood of rests and random ordering of pitches for another instrument (see Figure 5).

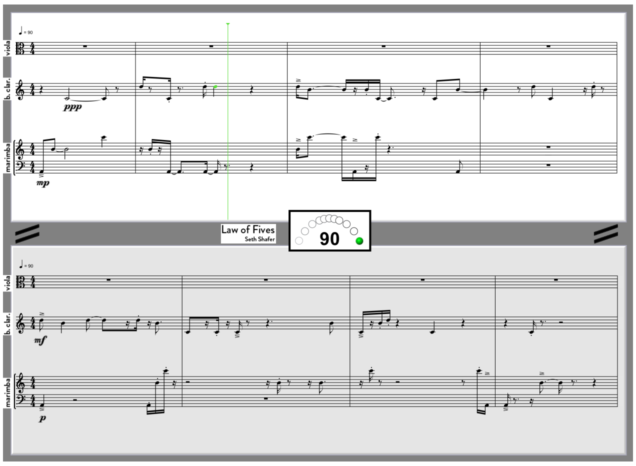


Figure 4. Predictable pitch behavior in Seth Shafer's *Law of Fives* (2015).

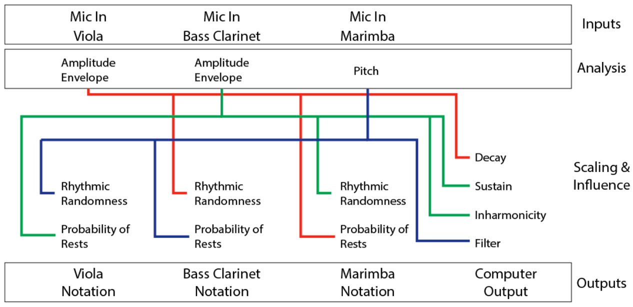


Figure 5. Notational variability from live performer influence in Seth Shafer's *Law of Fives* (2015).

Some local detail defies the predictability described previously. In such cases, the performer can benefit from studying the large-scale form. Rehearsal should afford the performer time to play the piece multiple times to gain a sense of any pre-planned or emergent forms. One possibility is that the notational behavior changes significantly at certain time-points. This is a strategy employed in *Law of Fives,* where one can outline predictable large-scale changes in tempo, texture, orchestration, and tessitura over time. In other works, one might find that behavior y always follows behavior x, or some more sophisticated formula. Another attribute that one can study is the general difficulty level and the modulation of that difficulty throughout the piece.

Some behaviors lie outside of either composer or performer control. A work like Nick Didkovsky's *Zero Waste* (2001) creates a performer-computer-notation feedback loop that highlights inaccuracies in human performance, errors in the computer analysis of the performance, and inadequacies in symbolic notation [16]. Even in a chaotic system certain behaviors can emerge. In *Zero Waste*, the trajectory of cumulative error in the system is toward an increased number of rests near the beginning of each notational output due to performer hesitation and the accumulation of chord clusters due to rhythmic quantization.

Another situation that evades composer and performer control is that of audience participation. Works like Kevin Baird's *No Clergy* (2005) [17] and Jason Freeman's *Graph Theory* (2005) [5] crowdsource certain compositional decisions, making the rehearsal of such works difficult. In this case, simulating audience feedback in rehearsal can clarify which parameters can be anticipated and which are subject to chance. Whatever strategy the composer employs, a major purpose of rehearsal is deducing notational behavior.

A common thread in real-time notation is that some amount of sight-reading is necessary. One purpose of rehearsal is to practice sight-reading the notational output from the system. Even performers confident in their abilities can balk at the prospect of sight-reading live in front of an audience. Substantial time must be dedicated to this task to aid in both the behavioral analysis described previously and developing quick music reading skills. Performers must keep in mind that every repetition of the work that they practice sight-reading is an equally valid version of the piece. Anything displayed in rehearsal can be in the version performed live.

Another important rehearsal consideration is the extent of improvisation involved in the work. Some pieces, particularly those with graphic elements, require a great deal of improvisation. Others do not ask for improvisation. Whether as a direct result of the notational design or the pressures and human limits of fast music reading, most pieces requiring live sight-reading involve possible improvisation. The composer and nature of the piece are the performer's guide. In the heat of performance, mistakes will occur and the musician must know which elements take priority and which can be neglected. Perhaps the general effect of the work is of prime importance and some brief moments of improvisation are preferable to silence if the performer's sight-reading skills falter. Conversely, perhaps formal connections should be sacrificed to meet the demands of local detail. These realities must be faced directly, ideally with composer input, so the performer knows what options exist when the inevitable mistake occurs.

A practical consideration for the performer during rehearsal is to become familiar with the on-screen graphical user interface. Every piece is different in this respect and the performer must acclimate themselves and glean every useful bit of information they can from the screen. The notational display might follow one of several paradigms. The notation might move: Does it constantly scroll horizontally or vertically? Does it slide periodically every beat, bar, system, or pre-determined span of time? The notation might remain stationary: Does the notation have virtual page turns? Does the screen refresh with new notation periodically? How far can the performer read ahead? The timekeeping and location tracking system can behave one of the following ways: a smooth scrolling tracker, a tempo-quantized tracker, or a bouncing-ball type tracker. The performer must be able to read the notation comfortably from their desired playing position, meaning the music size and distance from display must be adjusted. Other practicalities such as who or what triggers the piece to start, how the piece ends, and if the performer interacts with the screen or software in any unusual ways must addressed in rehearsal.

4.3 Performer-Composer Trust in Performance

A successful performance of a work using real-time notation hinges on the trust a performer places in the composer and computer-mediated notation. While there is no formula for building relational confidence, the following factors can help create a more optimal situation for the performer and composer.

Many factors that lead to an ideal real-time notation experience for the performer revolve around the difficulty of the score and the sufficiency of information about the piece provided by the composer. Ideally, the notation should strike a balance between several competing factors: the difficulty of the mechanical instructions like pitches, rhythms, dynamics, and articulations; the visual layout of the score (including the size of the notation font), the use of non-standard symbols, and whether the performer reads from a part or a score; the clarity of the timekeeping mechanism and how tempo modulations are implemented; the amount of expressive interpretation desired by the composer; the amount of improvisation; and the difficulty of ensemble coordination. As the complexity one parameter increases, the remaining parameters must correspondingly decrease in complexity to let the performer divert maximal effort to the most difficult elements. The performer can be best prepared if the composer provides clear and ample information about hardware and software requirements, the graphic user interface, notational conventions, a formal behavioral outline, sample scores, and/or documentation of past performances.

The balance between complexity and simplicity breaks down if performer failure is a conceptual component of the work. Failure in performance is a theme explored by many composers in what some have termed the "post-digital" aesthetic [18]. Any performer can understandably be alarmed at such a prospect. Although it falls outside the scope of this paper to address this aesthetic issue, the optimal experience for a performer put in that situation is one that does not make them appear foolish, even though this is a difficult standard to determine.

For the benefit of the performer, imagine what the ideal performance of a real-time score looks like from the point of view of the composer. A composer wants trust and engagement from the performer, treating the work as musically viable and as expressive as any traditionally-notated piece. A composer wants a performer who is willing to risk sight-reading from the stage, who makes mistakes and continues to engage, and who knows that some performance errors are apparent to the audience while others are not. Above all, the composer wants a musician who attempts to transcend the high demands of sight-reading and ultimately makes music.

5. THE COMPLEX SCORE AND THE FUTURE OF NOTATION

A brief examination of the complex score and the associated musical movement called New Complexity provides historical and aesthetic perspective on the issues presented in this paper. The complex score shares some striking similarities to real-time notation. Composers such as Iannis Xenakis, Brian Ferneyhough, and Richard Barrett often ask the player to perform near the limits of what is possible. This is often accomplished by presenting the player with conflicting instructions or goals. The result is a collision of actions, often represented in meticulous, high-density detail. Overloading the performer with notational information often guarantees that every performance is inherently short of perfection.

In a similar way, real-time notation presents the player with conflicting goals: relinquish the security of a fixed score while embracing new performance freedoms, sight-read in front of an audience while performing musically, expose the limits of ability while performing confidently. It also celebrates the beauty of ephemerality and difference. Both the complex score and the real-time score present ensemble coordination issues. Both present problems in rehearsal strategies. In some ways, the real-time score is a logical extension of the complex score in which Barrett's concepts of notation as freedom and improvisation as a method of composition can be realized [19].

Just as the proliferation of fixed paper notation was the product of incremental advancements in printing technology throughout the last few centuries, so real-time notation is a natural outcome of our current technology. As technology becomes more powerful and accessible, the body of real-time notation works and their associated approaches will likely continue to expand and differentiate. It is the author's hope that this paper builds upon the foundation established in the performance practice of real-time notation and provides a platform for further exploration by seasoned performers of such works.

6. REFERENCES

[1] G. Winkler, "The Realtime-Score: A Missing-Link in Computer-Music Performance," *Sound and Music Comput. Conf.*, p. 1, 2004.

[2] L. Vickery, "The Limitations of Representing Sound and Notation on Screen," *Organised Sound*, vol. 19, no. 3, pp. 215–227, 2014.

[3] R. Picking, "Reading Music from Screens vs Paper," *Behaviour and Information Technology*, vol. 162, pp. 72–78, 1997.

[4] S. Shafer, "VizScore: An On-Screen Notation Delivery System for Live Performance," *Proc. of the 2015 Int. Comput. Music Conf.*, pp. 142–145, 2015.

[5] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Comput. Music J.*, vol. 32, no. 3, pp. 25–41, 2008.

[6] G. Winkler, "The Real-Time Score: Nucleus and Fluid Opus," *Contemporary Music Review*, vol. 29, no. 1, p. 99, 2010.

[7] B. Haynes, *The End of Early Music*, Oxford University Press, p. 223, 2007.

[8] P. Thom, "Authentic Performance Practice," from *The Routledge Companion to Philosophy and Music,* ed. T. Gracyk and A. Kania, Routledge, p. 97, 2011.

[9] K. Essl, "*Champ d'Action*: Realtime Composition Environment for Computer-Controlled Ensemble," 1998. [Online]. Available: http://www.essl.at/works/champ.html [Accessed: 31-October-2015].

[10] J. Freeman, "*SGLC* for Laptop and Instrumental Ensemble," 2011. [Online]. Available: http://distributedmusic.gatech.edu/jason/music/sglc-2011-for-laptop-and/ [Accessed: 31-October-2015].

[11] R. Kostelanetz, ed., *John Cage*, Praeger, p. 131, 1970.

[12] R. Hoadley, "Notating Algorithms," Symp. for Performance of Electronic and Experimental Composition, Oxford University, 6-January-2012.

[13] E. McNutt, "A Postscript on Process," *Music Theory Online*, vol. 11, no. 1, 2005. [Online]. Available: http://www.mtosmt.org/issues/mto.05.11.1/mto.05.11.1.mcnutt.html [Accessed: 31-October-2015].

[14] E. McNutt, "Performing Electroacoustic Music: A Wider View of Interactivity," *Organised Sound*, vol. 8, no. 3, pp. 297–304, 2003.

[15] S. Shafer, *Law of Fives*, 2015. [Online]. Available: http://sethshafer.com/law_of_fives.html [Accessed: 12-May-2016].

[16] N. Didkovsky, "Recent Compositions and Performance Instruments Realized in Java Music Specification Language," *Proc. of the 2004 Int. Comput. Music Conf.*, pp. 1–2, 2004.

[17] K. Baird, "Real-Time Generation of Music Notation Via Audience Interaction Using Python and GNU Lilypond," *Proc. of the 2005 Conf. on New Interfaces in Musical Expression*, pp. 240–41, 2005.

[18] K. Cascone, "The Aesthetics of Failure: 'Post-Digital' Tendencies in Contemporary Computer Music," *Comput. Music J.*, vol. 24, no. 4, p. 12, 2000.

[19] R. Barrett, "Notation as Liberation," Symposium: Notation in Contemporary Music, Goldsmiths University of London, 18-October-2013.

Representing atypical music notation practices: An example with late 17th century music

Rodolfo Zitellini
Swiss RISM Office
rodolfo.zitellini@rism-ch.org

Laurent Pugin
Swiss RISM Office
laurent.pugin@rism-ch.org

ABSTRACT

From the 17th century to the first decades of the 18th century music notation slowly loses all its mensural influences, becoming virtually identical to what we would consider common modern notation. During these five decades of transformation composers did not just suddenly abandon older notation styles, but they used them alongside ones that would eventually become the standard. Void notation, black notation and uncommon tempi were all mixed together. The scholar preparing modern editions of this music is normally forced to normalise all these atypical notations as many software applications do not support them natively. This paper demonstrates the flexibility of the coding scheme proposed by the Music Encoding Initiative (MEI), and of Verovio, a visualisation library designed for it. The modular approach of these tools means that particular notation systems can be added easily while maintaining compatibility with other encoded notations.

1. INTRODUCTION

Mid to late 17th century musical notation was already very similar to what we know and use nowadays. The conventions began to be the same and many vestiges from the past were being lost. The Late Baroque period should not be considered as uniform, but as one of great transformation. This transition affected musical notation, which retained some specific features and idiosyncrasies.

Many particular features found towards the end of the century, such as void notation and coloration (see Sections 2.1 and 2.2), are sometimes considered to be just left over from the past with little practical use.

Nevertheless composers and printers used these alternative notation types extensively, making it important for modern editions not to lose them.

Up to the second half of the twentieth century, the most common custom for critical editions has been to transcribe the notation to the modern standard. Nowadays, on the other hand, more and more editions try to reproduce faithfully the features of the original sources [1]. Indeed, even if features such as void notation seem at first glance just a curiosity, completely removing them in transcriptions makes the edition less useful for philological work and scholars frequently have to reach out for difficult to find sources just to clarify some small passages.

There is a problem with the symbolic digital representation of music written in these non-standard notations, as currently no system is capable of encoding them properly. Since no notation system directly supports void or black notation (hand adjustments are necessary in many cases), the resulting encoded data is often invalid since, as we will see, it is often necessary to use over- or underfilled measures to represent the music.

In this paper, we investigate the use of the Music Encoding Initiative (MEI) scheme together with the rendering library, Verovio, for better encoding and visualising unusual notation features of the late 17th century. The following section explains more precisely some of these uncommon features we find in the music notation of that time, and then we look at previous work for both encoding and visualising them. Our proposed approach with MEI and Verovio is then explained and illustrated.

2. LATE 17TH CENTURY MUSIC

2.1 Black notation and colorations

Black notation is a feature of late 17th century music reminiscent of its mensural past. As in the mensural system it indicates particular rhythmic values, and its use is generally limited to simple cases. It is often mixed with white void notation, requiring, in many cases, great manual intervention to be typeset with modern systems.

Black notation is similar but not identical to modern notation. It was used in triple meter, 3/1 or 3/2, to indicate rhythmic alterations. The corpus in which such notation is found is vast and varied, and composers went to great pains to specify such notational details. It is therefore important that such information not be lost in modern editions, and such editions should encode it properly.

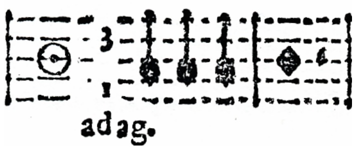


Figure 1. Black notation found in a sonata by Giovanni Maria Bononcini [2], bars 34-35 (excerpt from *Violino Primo* partbook).

Figure 1 shows a typical example of black notation in a late 17th century source. Typesetting it in software such as Finale requires adjustment to each note head or encoding it with an approximate symbolic representation. It is also important to note that the default music font in Finale does not include a black semibreve note head, so one is forced to use the smaller note head for quarters and other notes with flags.

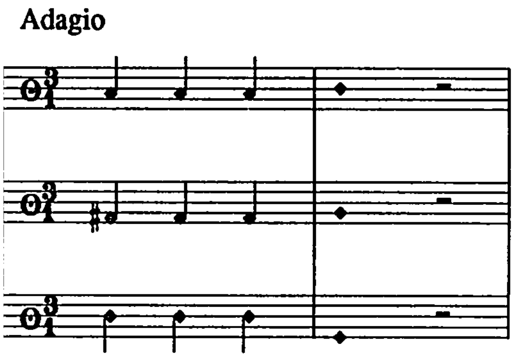



Figure 2. Full-score modern edition Figure 1, bars 34-35,[3].

Figure 2 shows a the same passage as Fig. 1 transcribed into modern notation. In the introduction to the edition the author laments the lack of easy support for 17th century notation in the (unspecified) typesetter she is using [3]. Coloration of single notes is encountered much more often and is used often in triple meter. It is a leftover expression from the old mensural system, and, as coloration indicated at the time, it specifies that a perfect note loses its perfection:

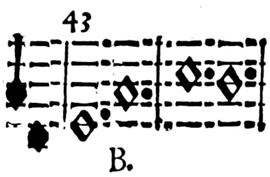


Figure 3. Bars 92-96 in a Mass by Maurizio Cazzati [4], excerpt from *Organo* partbook.

In the small extract shown in Fig. 3 the author employs coloration to visually underline a different rhythmic subdivision. This is an extremely common example, and modern editions have employed various strategies over the years to deal with it.

Normally brackets are used to signal coloration, but in some cases no indication at all is given in the score, only resorting to a small footnote in the critical apparatus.

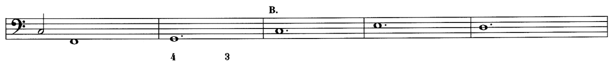


Figure 4. Basso of a full-score modern edition of Cazzati's *Messa Concertata* from op. 14, bars 92-95 [5].

As we can see from Fig. 4, although the modern notation used is correct and clear, it completely loses the visual cue given in the source. This kind of cue is not only useful to performers: they are also important to scholars particularly interested in notational problems. Burying them in lengthy critical notes makes this information inaccessible to the reader.

2.2 Void notation

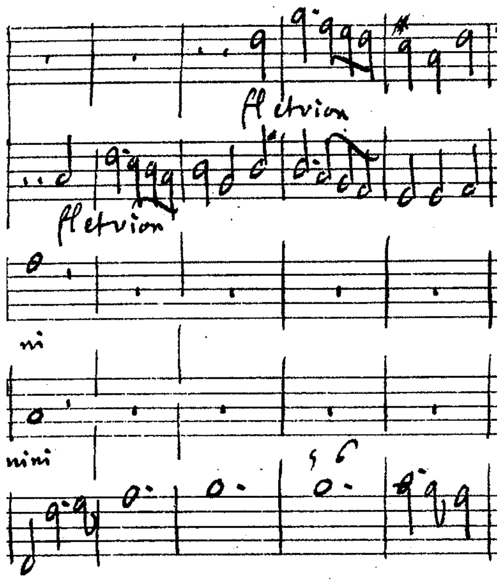


Figure 5. Marc-Antoine Charpentier, *In Honorem Sancti Xaverii canticum*. Facsimile of autograph ms., bars 110-114.

The use of void notation at the end of the 17th century is probably one of the great riddles still left for musicologists to figure out. Figure 5 gives us a common occurrence of this style of notation. In 3/2 time, quarter notes are written out as eighths with a white note head, and, if appropriate, beamed as "normal" eighths.

Various studies seem to show that it is simply an alternative to proper modern "black" notation, arbitrarily applied and without a specific meaning. Nevertheless it appears in copious prints and manuscripts [6]. Most well-known are the examples by Marc-Antoine Charpentier, but many others come from printed music in northern Italy, in the region of Bologna more specifically, a trend lasting until the mid-1710s [7].

Through a closer investigation of the issue it becomes apparent that void notation is not limited to a couple of sources here and there, but was a deliberate choice made by composers. Recent studies [8] have brought to light the fact that music printers were explicitly requesting movable type in void notation. This is the case, in the final years of the 17th century, of Silvani in Bologna, who purchased his characters in Venice. In his order [8], Silvani specifically laid out the details for void notation characters.

Void notation had commonly been used in the music printed in the City until the late 1650s, when Maurizio Cazzati, *Maestro di Cappella* in the Basilica of S. Petronio, produced the first examples of it [9]. Many later editions were produced in Bologna with this notation.

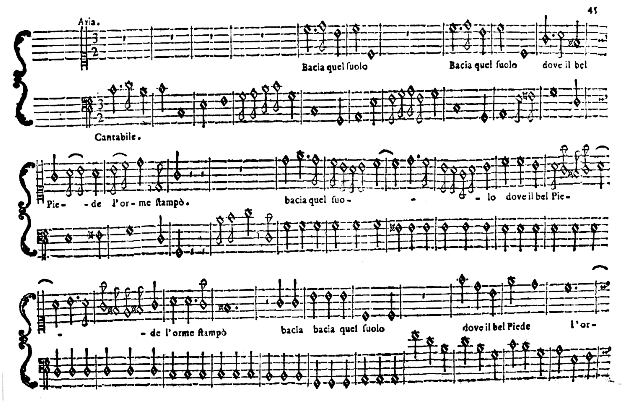


Figure 6. A late example of void notation in a collection of cantatas by Pirro Capicelli Albergati [10]. Many collections of music from this city are visually very similar to this setting [11].

Since the total corpus of music in void notation is somewhat restricted, scholars have mostly dismissed it in modern transcriptions. In recent years scholars wanting to maintain this original aspect of the sources have generally been discouraged by the lack of easy support for it in commercial music notation software.

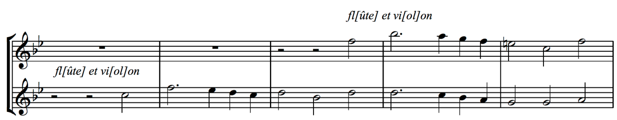



Figure 7. Marc-Antoine Charpentier, *In Honorem Sancti Xaverii canticum*. Modern edition [12], bars 110-114.

In Figure 7 we see clearly how the void notation from Fig. 5 is reproduced using modern notation. A note from the editor at the beginning of the 3/2 section informs us that the original is in void notation, which is not maintained in the transcription. We cannot blame the editor, as a proper transcription would require manually editing each note to change the note head to a white one. Also, the original manuscript source clearly beams the white eighth notes. In this case it is not sufficient merely to change the note head, although, in the notation software, the beamed notes need to be set at actual eighth notes (to obtain the beam), resulting in an insufficient number of notes in the measure. Not only does this require a deal of manual adjustment but the resulting symbolic representation of the music is invalid.

3. PREVIOUS WORK

A plethora of music codes have been developed [13]. Mostly they have a specific application or notation type in mind. For common western music notation (CWMN) and music notation applications, the most widely used is MusicXML [14]. It is designed as an interchange format between computer music applications. Therefore, it is meant to be sufficient for most applications but not optimal. MusicXML has no support for early music notation (before 1600) and is not designed to be customizable. For earlier notations, one of the first solutions proposed was DARMS with a dialect for mensural notation [13]. More recently, an XML code was developed as part of the Computerized Mensural Music Editing (CMME) project [14]. It is very comprehensive and supports the encoding of philological information. However, it was designed primarily as a file format for a critical edition software application developed for this repertoire and is not widely used outside it.

Many musical codes have been designed together with the development of music notation software applications. Most of these solutions target printed output and the encoding design and rendering tools are tightly bound together. Such is the case for MUP [16] and LilyPond [17]. In LilyPond, the code acts as a set of typesetting instructions to be interpreted by a compiler. This makes it a very powerful and flexible solution for producing a tailored output, including for specific repertoires. It can be used for Renaissance music and is highly customizable [18]. One limitation of LilyPond, however, is that it is very difficult to parse outside its own environment. One reason is that the structure of the data, which can be of arbitrary complexity, is driven more by the desired visual output than by the underlying logical structure of the notation.

Guido is another solution that follows a similar approach where the design of encoding structure is developed hand-in-hand with a rendering software component [19, 20]. It is designed to be embedded into a wide range of environments and is more flexible than LilyPond in that regard. To our knowledge, however, no extension for early music notation is available.

Quite sophisticated rendering engines are embedded in music notation software applications such as Finale [21] or Sibelius [22]. Most of these are so called 'wysiwyg' ('what you see is what you get') desktop applications. Since many of them are closed-source commercial applications, the musical data representation, both in-memory and in the files, is usually kept undisclosed. Ultimately, what counts for the user is the final (printed) result. For representing uncommon music notation features, users have continuously been tweaking the use of these applications to obtain the desired output. To do so, custom symbols can be used, or the music font can be changed. While this can work for producing printed scores, it is cumbersome to create since the applications are pushed beyond their design scope. Furthermore, it significantly reduces the interoperability of the data. Exporting them, for example to MusicXML, mostly produces musically nonsense data.

Remaining in the commercial domain, software packages such as SCORE [23] can produce highly customizable and fine-tuned output, and have indeed been used for decades by the music publishing industry. The principal drawback in this case is not only the difficulty of using the software, which is a specialized task requiring special training, but also that SCORE does not encode symbolically the music to be typeset. Its input format is focused on the precise description of the elements to print on the score, not what they mean. Extracting simple symbolic data, such as the notes on their own, is a complex and not completely accurate translation process.

4. MEI AND VEROVIO

4.1 Encoding

MEI is a community-driven effort to define a common encoding scheme for describing music notation documents. One the principal goals of MEI is to model music notation and how to represent it digitally in a structured and meaningful way. This approach differs from other initiatives where the goal is to encode the music notation for it to be usable by existing computer software applications, or for it to be typeset. In that sense, MEI acts as an application agnostic music encoding framework for representing music notation documents [24].

One characteristic of MEI is that it is organized into modules. There is an 'MEI' and a 'shared' module for common elements that form the basis of the MEI schema. Then each module groups the definition of the XML elements and attributes of a specific notation or application sub-domain. For example, CWMN and mensural notation are defined in two distinct modules. The advantage of having modules is that the schema definition is kept structured and can be adjusted according to needs. When necessary, valid values for some attributes can be defined differently in two modules, making their intended use more precisely defined and clearer. However, if all modules are activated, all possible values will be valid. The note duration is a good example since it has two distinct data types for CWMN and mensural notation.

Additional modules provide editorial markup capabilities – 'critapp' and 'edittrans'. With editorial markup, it is possible to encode alternative content using a parallel segmentation approach. This means that, when necessary, the encoding tree is divided into two or more alternative sub-trees. This is used for encoding variants between sources in critical editing with <app> and <rdg> elements. In a more generic way, the <choice> element can be used for encoding different representation options.

MEI is application agnostic and aims to be as comprehensive as possible. To this end, different representation domains are defined, although making a clear cut between the different domains in music notation is sometimes impossible. MEI makes the distinction between the visual domain, the logical domain and the gestural domain, the latter referring to how the music notation is expected to be rendered in sound. This separation is quite powerful and makes it possible to encode different domains simultaneously.

4.2 Rendering

Verovio is a music rendering engine written in C++ based on MEI [25]. Its goals are to be small and self-contained, without complicated external dependencies, and to be easily embeddable in other applications. It can work as a standalone tool or as a linked library, or, by compiling it using Emscripten, directly as a JavaScript library for in-browser rendering. This latter option enables it to build rich and responsive web-based music applications. This also relies on the output format, SVG, which greatly facilitates user interaction with the underlying encoding in web-based environments.

The internal data representation of Verovio is based on MEI. This means that music encoded in MEI is not transcoded before being rendered (for example to MusicXML for use in Finale or Sibelius) but is directly interpreted by the rendering engine. The MEI structure is preserved as far as possible in the SVG output of Verovio. This also means that Verovio inherits MEI's modularity and can easily be extended to support the different modules of the specification.

Verovio follows the SMuFL specification for its music font, making it easy to change [26].

4.3 Black notation

Encoding black notation is quite straightforward in MEI. It is encoded with the @colored attribute on the <note> element that indicates coloration, i.e., inverted note heads. The duration encoded in the @dur attribute is expected to be the duration of the corresponding uncolored note. In the Cazzati example of Figure 3, for the first measure, this means a value of "2" for the colored half note and "1" for the colored whole note.

```
<measure n="1">
  <staff n="1">
    <layer n="1">
      <note pname="c" oct="3" dur="2"
        colored="true"/>
      <note pname="f" oct="2" dur="1"
        colored="true"/>
    </layer>
  </staff>
</measure>
```

Figure 8. Black notation in MEI. The <note> element has a @colored attribute that can be set to "true".

The encoding of Figure 3 can be visualised as is with Verovio (Figure 9). The appropriate note heads are selected for the colored notes. For the second note of the first measure, this is the SMuFL code U+E0FA for a filled whole (semibreve) note head (noteheadWholeFilled). This character is not the same as the note head used for quarter notes, as it is specifically designed for coloration.

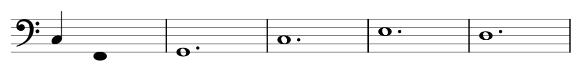



Figure 9. Black notation with Verovio. The appropriate note head is displayed.

4.4 Void notation

Void notation is more complex because it actually introduces a gap in the scale of note durations in the visual domain. There are indeed no quarter notes in the visual domain of void notation since they are visualised as void eighth notes. However, their actual (sounding) duration is still the duration of a quarter note. This means that void notation introduces a dichotomy between the visual and gestural domains from the quarter notes on. Figure 11 illustrates how the second staff of the second measure of the Charpentier example from Figure 5 can be encoded in MEI. The visual duration of the voided notes is encoded in the @dur attribute, namely with “8”, since they need to be visualised as eighth notes. Their gestural duration is encoded in the @dur.ges with a value of “4”. The voided characteristic is encoded with the @colored attribute.

```
<staff n="2">
  <layer n="1">
    <note pname="f" oct="5" dur="2" dots="1"/>
    <beam>
      <note pname="e" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="d" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="c" oct="5"
        dur="8" dur.ges="4"/>
    </beam>
  </layer>
</staff>
```

Figure 10. Void notation in MEI. Because we have a dichotomy between the visual and the gestural domain, both the @dur and @dur.ges attributes need to be used.

The encoding of Figure 10 can be visualised as is with Verovio, with a correct interpretation of the duration of both the visual (voided eighth note) and gestural (quarter note) domains.

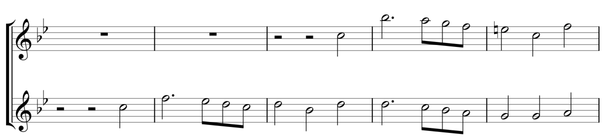



Figure 11. Void notation in Verovio. The appropriate note head is displayed with the duration of the gestural domain.

In some cases, it might be desirable to be able to switch from the original notation to a normalized one. In the case of void notation, one way to do this would be to have the rendering tool being aware of this practice and making it visualise the notes with a @coloration attribute set to “true” by only looking at the @dur.ges attribute value (i.e., by ignoring the @coloration and the @dur attributes). However, this would be a very specific implementation and a more generic solution is highly preferable. One way is to act at the encoding level using a parallel segmentation with a <choice> element. With such an approach, the original void notation is encoded in an <orig> element with the original notation, and a normalised version is encoded in parallel in a <reg> element.

```
<choice>
  <orig>
    <beam>
      <note pname="e" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="d" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="c" oct="5"
        dur="8" dur.ges="4"/>
    </beam>
  </orig>
  <reg>
    <note pname="c" oct="5" dur="4" />
    <note pname="d" oct="5" dur="4" />
    <note pname="c" oct="5" dur="4" />
  </reg>
</choice>
```

Figure 12. Alternate encoding in MEI. In some cases, it is desirable to have both the original notation and a normalised version in parallel.

Verovio has a ‘choiceXPathQuery’ option that can be used to select a specific child of the choice element for visualisation. By default, the first child of a choice is selected. In our example, in order to select the normalized version, the option would need to be set to ‘./reg’ for selecting the <reg> element instead of the first child.

Since the exact meaning of void notation is still unclear, and since composers used it quite extensively, it is important to encourage proper modern encoding of it. This not only allows the notational particularities desired by the author to be retained, but will also hopefully facilitate further investigation of the problem using a bigger and more coherent dataset once a proper corpus of modern transcriptions has become available.

5. CONCLUSIONS AND FUTURE WORK

The examples we have shown in this paper focus on uncommon rhythmic notations. They show how unusual features can be represented out of the box with MEI. There are also harmonic specificities in the notation of the time that we would like to cover in the future.

5.1 Figured bass

The most idiomatic notational element in the music from the 17th to the late 18th century is figured bass. Yet some high-end musical typesetters, such as Finale, completely lack support for it: a special font is provided that is inserted as lyrics. This approach not only requires an incredible amount of fine-tuning to the score to obtain an acceptable figured bass, it is also nonsensical as symbolic representation.

Other software, such as LilyPond, have a complete figured bass support, albeit with a complex encoding method. The single figures in LilyPond are not directly attached to any notes, but are a free-form independent voice, which is then superimposed on the music notation. While this solution works very well for typesetting music, in spite of the complexity of inserting the single figures, it is very difficult actually to associate the numbers with the notes to which they are attached, making the system useless as a symbolic representation of music plus figured bass.

To complete the support for 17th century music a module for figured bass in MEI is to be proposed. The purpose will be to have a complete representation so the single figures can be analysed in relation to the notes to which they are associated, if needed, and to provide high-quality typesetting output.

5.2 Scordatura

Virtuoso violin music from the 17th century often employs a technique called *scordatura*, in which the single strings of the instrument are tuned to other notes.

Different methods were used to write down music requiring *scordatura*, and most often the written notation marks the notes the player would have performed on a normally tuned string. In this way the hand position does not change in respect to the written note, however the note sounds different from what is actually written. It is a technique somehow in-between tablature and normal notation.

The challenge to typeset *scordatura* is that the written notes no longer correspond to the sounding notes. With commonly available notation systems there is no way to obtain at the same time sounding pitch and written pitch. It is necessary to encode each separately.

5.3 Conclusions

Many specialist software notation packages exist, and a number of these have been developed with specific requirements in mind. Unfortunately such specialized software is often unavailable outside large editors or projects, putting them out of the reach of many researchers. Moreover specialized software often requires special training to use, which can be an unachievable hurdle to master. Lastly a plethora of different systems encodes its data in a plethora of different manners, often not interchangeable between one and another. This impedes theconstitution of large and accessible collections of encoded music, as no shared standard poses an obvious barrier to anyone not using the particular software created for a particular collection.

As illustrated in this paper, the separation of different representation domains offered by MEI together with a tool that can take them into account is the perfect basis for encoding at the same time the visual information (the note position) and the gestural one (the sounding pitch).

Encodings such as MusicXML strive to achieve compatibility across systems, but were created on purpose with no support for ancient music. On the other hand MEI proposes to be a unique container for all western notation styled – extensible to what is not currently supported – with the same compatibility across systems attempted by MusicXML.

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNSF) and the Swiss Academy of Humanities and Social Sciences (SAHS).

6. REFERENCES

[1] J. Grier, The Critical Editing of Music, Cambridge University Press, Cambridge, 1996.

[2] Giovanni Maria Bononcini, *Sonata Ottava* from *Sonate da chiesa à due violini*, Op. 6 (Venezia: Magni, 1672).

[3] A. C. Yeung, "A study of late seventeenth-century compositional practice: The "Sonate da Chiesa" (Opus 6) of G. M. Bononcini", PhD diss., Columbia University, 1994.

[4] Maurizio Cazzati, *Messa Concertata* from *Messa e Salmi a Quattro voci,* Op. 14, (Venezia: Vincenti, 1653).

[5] Masses by Giovanni Pietro Finatti, Maurizio Cazzati, Giulio Cesare Arresti; edited with an introduction by Anne Schnoebelen (Seventeenth-century sacred music; v. 6), p. 103.

[6] S. Thompson, "Once more into the void: Marc-Antoine Charpentier's croches blanches reconsidered", Early Music XXX (1), pp. 83-93, 2002

[7] C. Churnside, "Music printing in early eighteenth-century Bologna: the case of Giuseppe Antonio Silvani and Pirro Albergati (1716-1717)", in *Fonti Musicali Italiane*, no. 17, pp. 105–34, 2012.

[8] G. Giovani, “Un capitale vivo e morto ad uso di samparia. Nuovi documenti sull'editoria musicale bolognese del Seicento”, in Giorgio Monari (ed) *Studi sulla musica dell’età barocca*, Libreria Musicale Italiana, Lucca, 2012

[9] R. Zitellini, “A study on the auto-revision activities of Maurizio Cazzati in his music reprints”, Schweizer Jahrbuch für Musikwissenschaft, serie 33 (2013), Peter Lang, Bern 2016.

[10] Pirro Capicelli Albergati (Bologna: Giuseppe Antonio Silvani, 1717).

[11] H. van der Linden, “Early eighteenth-century music type for the printer Giuseppe Antonio Silvani: Correspondence and other documents”, Tipofilologia, nr. 5, pp. 27–68, 2012

[12] T. G. Merrill, Canticle in Honor of Saint Xavier. <http://www.sscm-wlscm.org/main-catalogue/browse-by-composer/45-wlscm-no-13>

[13] E. Selfridge-Field (Ed.), *Beyond MIDI: The Handbook of Musical Codes*. The MIT Press, Cambridge, 1997.

[14] MakeMusic, *MusicXML*. <http://www.musicxml.com>

[15] T. Dumitrescu and M. van Bechum, *Computerized Mensural Music Editing*. <http://www.cmme.org>

[16] Arkkra Enterprises, *Mup*. <http://www.arkkra.com>

[17] H. W. Nienhuys and J. Nieuwenhuizen: “LilyPond, a system for automated music engraving,” *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, pp. 167–72, 2003.

[18] LilyPond, *Typesetting mensural music*. <http://lilypond.org/doc/v2.18/Documentation/notation/typesetting-mensural-music.en.html >

[19] H. Hoos and K. Hamel, *The GUIDO Music Notation Format*. <http://guidolib.sourceforge.net/doc/GUIDO-Music%20Notation%20Format.html>

[20] C. Daudin, D. Fober, S. Letz and Y. Orlarey, “The Guido Engine: A toolbox for music scores rendering,” in *Proceedings of Linux Audio Conference*, 2009, pp. 105–11.

[21] MakeMusic, *Finale 2014* <http://www.finalemusic.com/>

[22] Avid, *Sibelius*. <http://www.avid.com/en/sibelius>

[23] San Adreas Press, *SCORE*. <http://www.winscore.info/>

[24] A. Hankinson, P. Roland and I. Fujinaga. “The Music Encoding Initiative as a Document-Encoding Framework,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 293–8.

[25] L. Pugin, R. Zitellini and P. Roland, “Verovio: A library for engraving MEI music notation into SVG,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014),* pp. 107–12.

[26] Steinberg, *SMuFL*. <http://www.smufl.org>

THE EXPRESSIVE FUNCTION IN WOR SONGS

Helena Lopez Palma
University of A Coruña
helena.palma@udc.es

ABSTRACT

We study some musical and expressive features of traditional Wor vocal music, an ancestral gender of the Biaks (Indonesia). A core aspect in Wor songs is the expression of wonder, which Biaks have developed into an Aesthetics of Surprise [1, 2]. We describe some key structural features in the pitch and time domain used as means to express such an aesthetics. We represent the acoustic and prosodic features encoding expressive content by means of an Expressive Function which contains expressive indices with internal structure [3, 4]. We propose an augmented expressive score [5] for the transcription of unaccompanied Wor songs.

1. INTRODUCTION

We study the expressive content conveyed by traditional Wor songs. We aim at representing the musical features encoded by Expressives by means of an augmented musical score. Our data consists of vocal music from Yapen Island collected and recorded by the author and Alfons Arsai and translated into English by Izak Morin. It includes some 20 *wor* songs in the Biak language (ISO 639-3: bhw) [6, 7, 8, 9] performed *a capella* by Hendrik Arwam (tenor) – a gifted singer expert in *Wor* music – and his daughter Sara, and 4 Serewen (ISO 639-3: pmo) songs performed by Obaja Tarami (baritone), who was also the composer of some of the songs. [1] The transcription of the songs onto musical notation was done by ear. The music notation languages used to transcribe our data were "abc", [2] "lilypond" [3] and "guido". [4] *Wor* is a cover term for a gender of traditional vocal music of the Biaks [1, 2]. [5] Wor songs (*dow*) are performed as part of a ceremonial event or a feast in which singing is combined with dancing accompanied by tifa drums. Wor feasts are related to many situations of Biaks' life: Wor is used to honor an ancestor important to the community, to call for protection for children or for a person in a transition in his or her life-cycle, to evoke sympathy or sorrow, to raise anger or support, to prepare for a battle or to celebrate the victory of some warrior.

Wor comprises many different subtypes: Kankarem (introduction song), Beyuser (narrative song), Erisam (expressionist style), dow Mamun (war song), Dance songs (Sandia, dow Arbur), Kajob, Morinkin, Wonggei. For this paper we just focus in four Wor songs belonging to the narrative (Wo nayro), dance (Forine, Woresa) and expressionist (Aya ma) subtypes.

Wor music is believed to have a magical power which grants welfare and protection to Biaks. Wor ceremonies attempt to use such a power to summon the forces of nature and to tighten social bonds. To that effect, wor uses a singing style that has been described as an "Aesthetics of surprise" and wonder [1, 2]. We will return to that in section 3. A legend attributes the discovery of wor to a magical origin, related to the sound of a vine heard in the forest by an old man from the Mnuwon clan. [6] Wor has been transmitted thereafter within clans by expert singers who teach wor to their children. Biak singers believe Biaks have been protected by the wor sung by their ancestors, and they feel obliged to sing wor to protect their own children. As they express it, "If we don't sing wor, we die." [10].

Although in real performance most of the songs of our corpus would be sung by a soloist and a learners-choir in heterophonic style [1], and accompanied by tifa drums, we decided not to include drum accompaniment in our study so we could concentrate on the vocal technique of solo skilful singers who would bring out the artistic features of the

[1] A sample of the songs can be heard at: http://www.udc.gal/grupos/ln/music_research/indonesia/biak.html

[2] http://abcnotation.com/, https://abcjs.net/;
"abc" notation is widely used for transcribing traditional music. Digital music archives keep a large number of traditional music scores in "abc" language, which can be worldwide queried and retrieved by the incipit of the melody.
http://drawthedots.com
http://music.gordfisch.net/montrealsession/editor.php
http://poorfox.com/hymns/abc2gif.html

[3] http://www.lilypond.org, http://lilybin.com

[4] Guido Scene Composer IDE: http://guidolib.sourceforge.net/

[5] Nowadays, Yospan songs, accompanied by a band of string instruments and tifa, are more popular than wor among young people.

[6] The legend says that after that experience, Mansar Mnuwon became the first expert of wor. The story has been described as follows:

> "Late one night, while he was hunting in the forest, the man suddenly heard voices high in a tree. In vain, he scanned the branches for the source of the noise. When he sat down to rest, the music swelled. Startled, he grabbed a vine that was coiled around the tree, and the voices dividid into two choruses. The vine's flowers were singing the song! To keep the voices from sinking into the soil at sunrise, the man cut down the vine. He took it home and ate the leaves and became the first Biak clever at singing wor." [2, 90]

	mode	interval	song
1	C D E G A	2 2 3 2 3	Ayama
2	D E G A C	2 3 2 3 2	Wo nayro; Ae yasoba
3	E G A C D	3 2 3 2 2	–
4	G A C D E	2 3 2 2 3	–
5	A C D E G	3 2 2 3 2	–

Table 1. Pentatonic scales.

music.

2. MUSICAL FEATURES

The features that contribute to build and articulate the structure of our songs are: The scaling of the octave interval, the intervals with a functional value, and syntactic units such as motives, phrases, intermediate and final cadences.

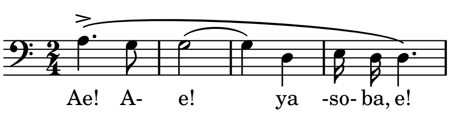
2.1 Scales, Pitch-Class Sets and Interval Vectors

The songs in our corpus are built using anhemitonic pentatonic and tetratonic scales, with no intervals of a semitone between any two consecutive notes. Although intervals of a semitone or even of a smaller size may occur as ornamental notes, they do not have a structural role. In section 2.1.1 we focus on pentatonic scales and in section 2.1.2 on tetratonic ones.

2.1.1 Pentatonic scales and modes

The anhemitonic pentatonic scale is built from the section 0-4 in the cycle of fifths (C G D A E), pitch class set $\{0, 2, 4, 7, 9\}$. The smallest interval between any two consecutive notes is a tone interval. [7] Scales are built with a pattern of alternating intervals with size of 2 and 3 semitones. Starting with a pitch class 0, the structure of interval sequence is 2 2 3 2 3. Depending on which note is taken as the first one of the scale, those sounds can be organized in five different modes or rotations, each mode with a particular flavour associated. In a pentatonic scale there are 5 possible different modes (cf. table 1). In wor songs, a mode is an arrangement of sounds around a nuclear tone, taken as the ground or tonic sound. [8] A tone is prominent if it is the last sound in a final cadence of a song. The nuclear tone can be viewed as the pitch that marks the point of rest in a song.

(1) Ae ae! Yasoba (mode 2: d e g a c)



(2) Ayama (mode 1: f g a c d)

[7] However, our songs do not use the whole pentatonic scale with intervals of the same size (C D E F# G#).

[8] Although any of those modes can be transposed, in Wor songs transpositions depend on the natural register of the voice of a singer rather than on musical composition criteria.

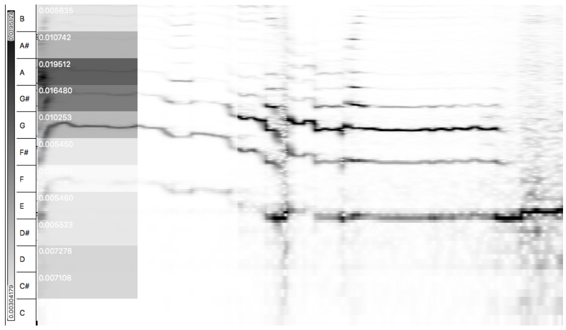
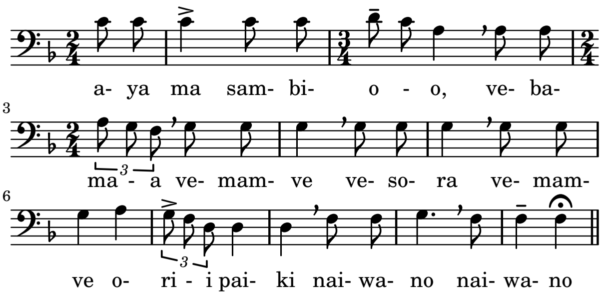


Figure 1. Ae yasoba e: chroma.



Our songs use the anhemitonic pentatonic mode 1 (“Aya ma”) and mode 2 (“Ae! Yasoba”), both symmetrical. Mode

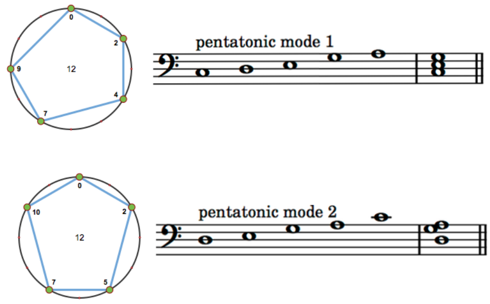


Figure 2. Pentatonic scales.

2 is very frequent in the songs we collected. The implicit tonic chord in mode one is CEG, and in mode two is the suspended DGA.

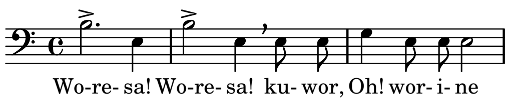
2.1.2 Anhemitonic tetratonic

Tetratonic scales seem to be generated as gapped anhemitonic pentatonic [11]. Our songs use the tetratonic asym-

	mode	interval	song
1	C D E G (A)	2 2 3 5	Forine
2	D E G A (C)	2 3 2 5	Yasoriso
3	E G A C (D)	3 2 3 4	–
4	G A C (D) E	2 3 4 3	–
5	A C D E (G)	3 2 2 5	Woresa

Table 2. Tetratonic scales.

metrical mode 1 (“For ine”), the symmetrical mode 2 (“Yasoriso” and the asymmetrical mode 5 (“Woresa”).

(3) Woresa (tetratonic 5: e g a b)

Mode 1 has the implicit major chord CEG(D). Mode 2 the suspended tonic chord DGA. Mode 5 has the implicit minor chord ACE(D).

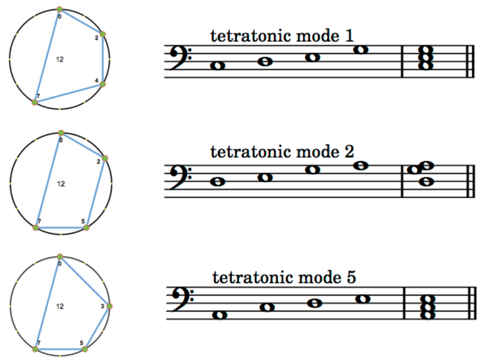


Figure 3. Tetratonic scales.

In table 3 we summarise the interval vectors of pentatonic and tetratonic scales used in our songs.

prime form	scale	i-vector	sym
{0,2,4,7,9}	penta 1	[032140]	+
	penta 2		+
{0,2,4,7}	tetra 1	[021120]	-
	tetra 5		-
{0,2,5,7}	tetra 2	[021030]	+

Table 3. Interval vectors of pentatonic and tetratonic scales

2.2 The structure of the songs

The songs are generated from some motive with a melodic or rhythmic prominent characteristic. Such motives are subject to variations which make them grow into articulated phrases.

Phrases
The phrase is a syntactic unit consisting in some integrated musical events. We use the definition of phrase in [12, 3]:

> “...a unit approximating to what one could sing in a single breath.”

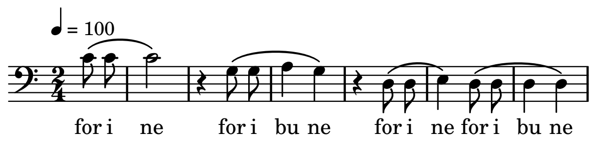
The phrase ending may be marked by a combination of features:

1. Rhythmic reduction. The notes at the end of a phrase may have a longer duration.
2. Melodic relaxation through a drop in pitch.
3. The use of some characteristic descending sequence associated to the end of a phrase.
4. The use of a nuclear tone in a mode.
5. A fading-out amplitude envelop.

Rhythm
The rhythm of the songs follows the metrics and the expressive structure of the words.

Range
The range of the rising and falling contour of the phrases is between a pentachord, and a heptachord in the final cadences.

Motives
In the song “Forine”, the main motive is an anapaest rhythmic feet ○ ○ ●. The motive is repeated three times. Each time with a descending pitch

(4) “For ine”: rhythmic motive

The initial phrase of song “Wo nayro” is made of 2 motives: a wonder motive, expressed by an exclamative surprise-like utterance on E (cf. figure 4), which is answered at a perfect fifth below by a motive with an ascending-descending contour (ABA).

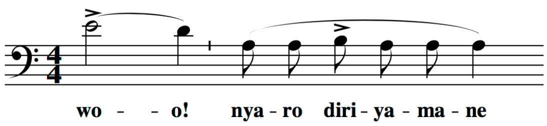


Figure 4. Wo! Nyaro diriyamane.

3. THE EXPRESSIVE FUNCTION

The meaning conveyed by Wor songs is structured in two hierarchical layers: The lexical and the expressive structure. The Lexical Structure is generated from predicative items that describe events denoting truth conditional content. The Expressive Structure conveys affective or communicative contextual content with no truth conditional value. It is built from an expressive function that projects the lexical structure onto an expressive utterance. [9] Both lexical and expressive structures differ musically in pitch and intensity features and in pauses marking phrasal structure.

Some of the linguistic items encoding expressives documented in wor lyrics include: interjections (*wo*, *ae*) expressing wonder, surprise, or desire. [10] The expression

[9] The expression structure has been related to an allegedly early stage in the evolution of human language, which could be shared by some animal vocalisations, such as the language of birds [13].

[10] Interjections and affective vocalisations are holistic expressions that cannot be analysed in subcomponents. They show the property of descriptive ineffability [4], which captures the fact that it is difficult to describe their content by means of a linguistic paraphrase.

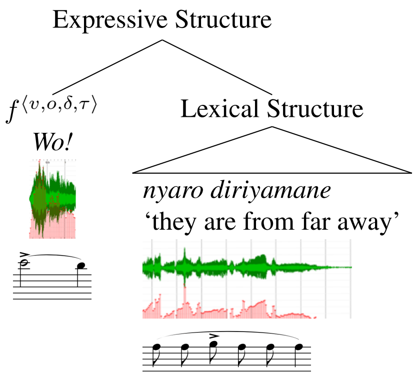
of wonder is a prominent feature in the aesthetics of wor songs, characterised as an "aesthetics of surprise" [2, 1]. Attention-getters [14] expressed with a verb in imperative mood (*woresa* 'stand up'; *wafyeri* 'dance') or by nominal phrases (*for ine* '(light) this fire'). Those are invitations to dance or to participate in some way in the wor ceremonial feast. Topic markers (*ma*) pointing to the experiencer of a contemplative event (*aya ma* 'I-TOPIC').

We represent the expressive function f_{exp} by means of expressive indices with an internal structure [3, 4]. Indices form the tuple $\langle v, o, \delta, t \rangle$ with v standing for the voice of the agent uttering the expressive, o the object (who, what) to which the expressive is orientated, δ the degree of the expressive, t the time interval [15] (onset, offset) of the duration of the expressive. The expressive function f_{exp} takes an utterance - projected from the lexical structure u_{lex} - and yields an expressive utterance u_{exp}.

$$f^{\langle v,o,\delta,\tau \rangle}(u_{lex}) = u_{exp} \quad (1)$$

In the tree below the expressive structure is obtained by projecting a lexical context C_{lex} – which conveys the content of an utterance generated from items in the lexical structure with truth-conditional meaning – onto an expressive context C_{exp}.

(2)



The musical features of expressives are the ones associated with call-like vocalisations (attention-getters), screams, or conative speech acts (incitations to action). The indices v, o, δ, τ of the expressive function may be related with some of the following emphatic features:

- High register within a scale.
- Sustained mora sounds bearing ornamental fluctuations in pitch, as the one expressing *Wooo!* in the song "Wo! nyaro" (cf. figure 5).
- Intensity of the signal. In figure 6 the intensity of the signal of the expressive is measured as the RMS value of the waveform.
- Repetitions, sometimes with an additional reinforcing segment (for ine, foribune).

By contrast, the pitch range of the melodic contour for the lexical structure is almost monophonic and in a low register within a scale. Furthermore, the phrasing and the rhythm follows the metrics of the words. In the table 4 we summarise musical features differentiating expressives from lexicals.

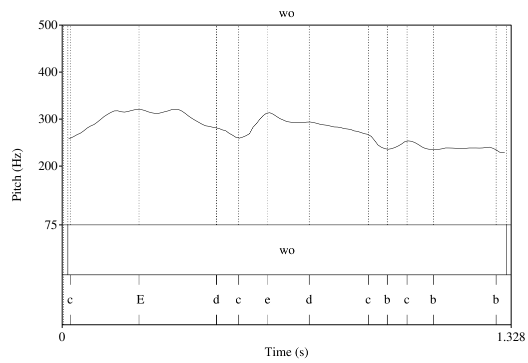


Figure 5. Melismatic ornamentation in "Wo! nayro".

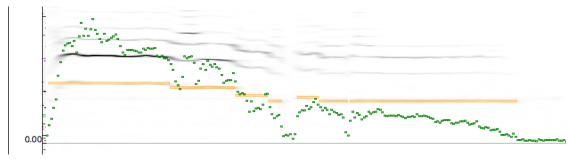


Figure 6. Intensity of the Expressive energy in "Ae! Ae! Yasoba".

4. LYRICS

(5) Gender: Erisam. Tribal expression of the self.

Aya *ma* *sambio* *ve-ba-ma* *ve-mam* *ve*
1SG TOP myself who-big who-look to

ve-so.ra *ve-mam* *ve* *or-i* *paik-i*
who-follow who-look to sun-3SG moon-3SG

na-i-wa *no* *na-i-wa* *no*
3PL.INAN-SPC-over.there LOC 3PL.INAN LOC

'Here I am myself grown up looking at that distant sun and moon.'

(6) Gender: Erisam

Ae *ae* *ya-so* *ba* *e* *awin* *e*
EXP EXP 1SG-follow not EXP mother EXP

be-o *marbui* *ker* *o* *ma* *ya-far-fnak*
give-me marbui little.piece EXP so 1SG-play

ya-frar *kamsar* *o* *ya-so* *ve* *Kurudu* *ve*
1SG-run kamsar EXP 1SG-go to Kurudu to

va-ri *vari*
side-the side

'Mother, let me go to the other side of the island, to the village of Kurudu, so I can play with a piece of marbu.'

Musical Feature	Expressive	Lexical
Pitch register	high	low
Duration	emphatic	speech metrics
Accent	intense	regular pulse
Envelop	sharp	not sharp

Table 4. Musical features of expressive and lexical structure

(7) Gender: Ceremonial wor

For	*ine*	*for*	*ibune*	*insos-e*	*wa-fyeri*
fire	3SG.SPC	fire	2SG.SPC	girl-PL	2SG-dance

wa-susu	*wa-kababen*	*o*
2SG-move.backward	2SG-make.burn	EXP

'This fire, light this fire. Girls, dance, move forth and backwards, make a big fire and dance.'

(8) Gender: Ceremonial. The singer incites a woman to dance and sing around the fire.

Wores-a	*wores-a*	*ku-wor*	*o*	*wor*
2SG-stand	2SG-stand	1DU.INC-sing		song

ine	*bae*	*bin*	*ve-na*	*wore*	*wo*
3SG.SPC	EXP	woman	who-has	songs	EXP

wores-a	*ku-wor*	*o*	*wor*	*ine*	*bae*
2SG-stand	1DU.INC-sing		song	3SG.SPC	EXP

'Stand up; lets you and I sing this song. Come on, woman who sings. Stand up, lets sing this song.'

5. AN AUGMENTED EXPRESSIVE SCORE MODEL

The interconnected research of composers, performers, artists scientists and engineers taking place since 1945, and the need for designing an accurate language to represent the results of such interactive research has stimulated the invention of new music notation technologies. Recent scoring systems that have created a language capable of meeting such artistic and scientific demands can be classified according to the following aims:

a) Notation languages aiming at being a tool for the design of contemporary composition scores. Examples of such systems are BACH [16], implemented as a library in Max/MSP; [11] Another is ENP [17], built in the lisp orientated visual programming language PWGL. [12] ENP has a GUI with direct editing capabilities. ENP enables to construct scores in both mensural and non-mensural time. The score can be augmented with graphical annotations of a large kind of different sorts: expression marks for performance directives, or analysis annotations for motives, harmonic progressions, or Schenker style graphs, pitch-class set. Non-standard expressions include groups, canvas-expressions, score-BPF. It is a powerful tool for the design of contemporary compositions. However, in its actual state it is not possible to synchronise mensural and non-mensural notation or symbolic notation and audio signal.

b) Notation systems aiming at representing musical analysis: Those include the powerful signal visualiser and artistic graphic designer tool EAnalysis. [13] Another recent analysis model [18] is a computer implementation of Lerdahl and Jackendoff's GTTM. [14] The interactive GTTM analyser and the GTTM database with 300 monophonic pieces are available online. [15] c) Notation systems designed to follow agents in live performances: Antescofo. [16] d) Notation systems capable of synchronising diverse multimedia objects. One of such systems is INScore [19], which provides an OSC API for designing interactive augmented music scores. The music score, symbolise in guido language, is projected into a scene, where it may be augmented with audio or video signals, bitmap or vectorial images. Time is the core driving feature and the interactive elements are related by means of a mapping algorithm. This system can be used in multi-agent live scoring performances.

The aims that have guided our modelling of a music score system for Wor songs have been:

- To provide an analysis of the compositional techniques used in traditional Wor songs.
- To account for the relation between speech and song instantiated in Wor vocal music.
- To fix the music and the text of Wor songs in order to contribute to preserving the rich musical heritage of the Biaks.

We model the music score of Wor vocal music as an Augmented Expressive Score (AES) with the dynamic structure of a bottom up directed tree. Figure 7 illustrates AES applied to the song "Woresa". In the AES model, the indices of the expressive function $f^{\langle v,o,\delta,\tau \rangle}$ are located at the vertices of the tree.

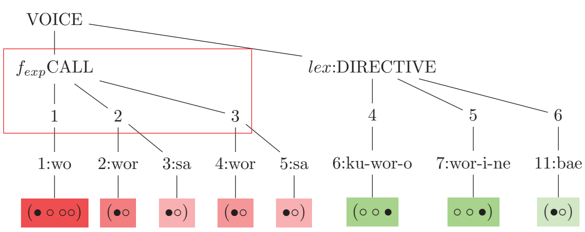


Figure 7. Augmented Expressive Score for "Woresa".

[11] http://www.bachproject.net/home

[12] http://www2.siba.fi/PWGL/

[13] http://logiciels.pierrecouprie.fr/?page_id=402

[14] GTTM provides a generative abstract representation of classical tonal music in 4 levels: (a) Grouping structure; (b) Metrical Structure; (c) Time-span binary tree, which captures the core melodic items; (d) Prolongation binary tree, which captures tension and relaxation.

[15] http:www.gttm.jp

[16] http://repmus.ircam.fr/score-following

0.038895	1.400212	wo
1.487725	2.168383	wor
2.259138	3.007862	sa
3.056481	3.798723	wor
3.850582	4.589583	sa
4.657649	4.800263	ku
4.800263	5.020667	wor-o
5.429062	5.578158	wor
5.610571	6.158339	ine
6.158339	8.112801	bae

Table 5. Time intervals

- The voice index (v) names the highest vertex (VOICE).
- The expressive function item (f_{exp}) is represented at a vertex annotated with a label specifying the semantic content of the expressive (f_{exp}CALL).
- The time index (τ) is represented at the numbered vertices. The τ level relates an audio signal interval with a graphical score segment. The number n labelling each τ vertex (1 through 6 in the example) stands for an index in a relational database, which specifies the intervals (onset, offset) of the temporal duration of the audio signal for each syllable of the lyrics (table 5). We apply the mapping algorithm propose in [20] to relate the time interval and the graphic segment.
- The red and green coloured nodes at the bottom of the tree represent the δ index of the expressive function, with the gradation of colour standing for the degree of energy of the expressive.

We implement our AES model in Max/MSP.

6. CONCLUSIONS

A key aesthetic feature of Wor traditional songs is the expression of wonder, conveyed through linguistic and musical means. We have proposed that the linguistic and musical expressive items are part of an Expressive Structure which is built on top of a Lexical Structure bearing truth-conditional meaning. The expressive structure is obtained by applying an Expressive Function f_{exp} to an utterance conveying truth-conditional lexical content u_{lex}. We represent the expressive function by means of indices with internal structure $f^{\langle v,o,\delta,\tau\rangle}$. We map those indices onto the musical features represented in an augmented score.

Abbreviations

1 = first person; 2 = second person; 3 = third person; AN = animate; DU = dual; EX = exclusive; EXP = expressive; INAN = inanimate; INC = inclusive; LOC = locative; PL = plural; POS = possessive; REL = relativiser; SG = singular; SPC = specific; TOP = topic; VBLZ = verbaliser; VOC = vocative;

Acknowledgments

We would like to express our gratitude to Mercedes Bernacer, Willem Burung, David Gil, Carmen Lage, Francesca Merlan and Alan Rumsey.

7. REFERENCES

[1] P. Yampolsky, Ed., *Music dari Biak, Irian Jaya 99057-10 MDCD; Music of Indonesia*. Smithsonian Folkways, 1996, no. 40426.

[2] D. Rutherford, “Of birds and gifts: reviving tradition on an indonesian frontier,” *Cultural Anthropology*, vol. 11, no. 4, pp. 577–616, 1996.

[3] D. Kaplan, “Demonstratives: An essay on the semantics, logic, metaphysics, and epistemology of demonstratives and other indexicals,” in *Themes from Kaplan*. New York: Oxford University Press, 1989, pp. 481–614.

[4] C. Potts, “The expressive dimension,” *Theoretical linguistics*, vol. 33, no. 2, pp. 165–198, 2007.

[5] D. Fober, S. Letz, Y. Orlarey, and F. Bevilacqua, “Programming interactive music scores with inscore,” in *Sound and Music Computing*, 2013, pp. 185–190.

[6] H. Arwam, *Memahami Kekerabatan Bahasa Barapasi-Tefaro-Sauri di Teluk Saireri Waropen*. Manokwari. West Papua: Sasako Papua Publisher, 2011.

[7] ——, *Dinamika Bahasa Biak*. Yogyakarta. Indonesia: Kepel Press, 2013.

[8] S. Mofu and M. Dalrymple, “On-line language documentation for biak,” http://biak.clp.ox.ac.uk, 2010.

[9] W. Van den Heuvel, *Biak, description of an Austronesian language of Papua*. Netherlands Graduate School of Linguistics, 2006.

[10] F. Kamma, *Koreri Messianic Movements in the Biak-Numfor Culture Area.* Springer Netherlands, 2013.

[11] M. McLean, *The structure of Tikopia music*. Auckland. New Zeland: Archive of Maori and Pacific Music, Department of Anthropology, University of Auckland, 1991.

[12] A. Schoenberg, *Fundamentals of musical composition*. London: Faber & Faber, 1967.

[13] S. Miyagawa, R. C. Berwick, and K. Okanoya, “The emergence of hierarchical structure in human language,” *Frontiers in psychology*, vol. 4, 2013.

[14] F. Ameka, “Interjections: The universal yet neglected part of speech,” *Journal of Pragmatics*, vol. 18, no. 2, pp. 101–118, 1992.

[15] B. H. Partee, “Some structural analogies between tenses and pronouns in english,” *The Journal of Philosophy*, pp. 601–609, 1973.

[16] A. Agostini and D. Ghisi, “A max library for musical notation and computer aided composition,” *Computer Music Journal*, vol. 39, no. 2, pp. 11 – 27, 2015.

[17] M. Kuuskankare and M. Laurson, "Expressive notation package," *Computer Music Journal*, vol. 30, no. 4, pp. 67–79, 2006.

[18] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing methods for analysing music based on lerdahl and jackendoff's generative theory of tonal music," in *Computational Music Analysis*. Springer, 2016, pp. 221–249.

[19] D. Fober, Y. Orlarey, and S. Letz, "Inscore. an environment for the design of live music scores," in *Proceedings of the Linux Audio Conference, CCRMA, Stanford University, California, US*, 2012, pp. 47–54.

[20] D. Fober, F. Bevilacqua, and R. Assous, "Segments and mapping for scores and signal representations," Technical report, GRAME, Tech. Rep., 2012.

IS THERE A DATA MODEL IN MUSIC NOTATION?

Raphal Fournier-S'niehotta
CNAM
fournier@cnam.fr

Philippe Rigaux
CNAM
philippe.rigaux@cnam.fr

Nicolas Travers
CNAM
nicolas.travers@cnam.fr

ABSTRACT

Scores are structured objects, and we can therefore envisage operations that change the structure of a score, combine several scores, and produce new score instances from some pre-existing material. Current score encodings, however, are designed for rendering and exchange purposes, and cannot directly be exploited as instances of a clear data model supporting algebraic manipulations. We propose an approach that leverages a music content model hidden in score notation, and define a set of composable operations to derive new "scores" from a corpus of existing ones. We show that this approach supplies a high-level tool to express common, useful applications, and can easily be implemented on top of standard components.

1. MOTIVATION

The digital encoding of music notation is a long standing endeavour, and has given rise to many proposals [1]. Nowadays, leading encodings are those which rely on the XML format to represent music notation as structured documents. MusicXML [2] is probably the most widespread one, due to its acceptance by major engraver software applications (Finale, Sibelius, and MuseScore) as an exchange format. This interoperability motivation yields an encoding which simultaneaously conveys structural, content and rendering information in a somewhat intricate representation. Another issue is the dependency of notation syntax and its interpretation on locations, periods, styles, and cultural contexts. Designing a format apt at capturing this high variability in a single and consistent representation is quite challenging. The MEI initiative [3, 4] attempts to address this challenge with an extensible format [5]. It relies on pre-defined components such as, for instance, the Common Music Notation (CMN) module. The initial discusions held in the recent W3C Music Notation Community Group [6] , launched in Sept. 2015, also point to the difficulty of a general, consistent, encoding framework that would capture the syntactic and semantic nuances of music notation throughout the specialized context of its use.

If we consider a specific notational context, and assume the existence of a specialized format that accurately covers the musical idiosyncrasies of this context (for instance an MEI module as stated above), then it makes sense to assume that this format encapsulates a data model for content encoding, at least for this particular part of the music repertoire. Focusing on Common Music Notation for instance, this data model can partly be identified through the commonalities of distinct formats such as, say, MusicXML, MEI and Lilypond. Beyond their different initial motivations and approaches, they share a basic set of features that characterize the music material to be represented.

Obviously, this notion of "content model" is controversial in the context of music notation: most of the elements that describe a score rendering can, to some point, be considered as significant and part of the global meaning conveyed by the notation. Although the separation of content and rendering components is a recurring topic of discussion for the designers (see for instance [7]), no score encoding, to our knowledge, has yet been designed with such a motivation in mind. One of the greatest benefits would indeed be the ability to "style" score contents, akin to what has been achieved for the rendering of HTML/XML content in the area of Web documents.

In the present paper, we address another motivation for separating content/rendering concerns. Being able to identify a content model opens the way to the vision of score corpora as a collection of structured objects, and makes it possible to envisage operations that "plays" with the structured content in order to extract useful parts, combine several scores, and derive new content from existing ones.

There are strong motivations for enabling such a system. It would indeed provide, via a high level language, a number of useful operations.

- *Automatic content management*. Split a score in parts, distribute them to digital music stands, apply transpositions and add decorations (directives) as needed; conversely, merge distinct parts as a single score;
- *Search and compare*. Search scores which satisfy some criteria; extract the matching fragments; align those fragments in a new score for further investigation;
- *Advanced analytic*. Derive analytic features (e.g., harmonic progression); annotate scores with these features; produce new representations emphasizing structural or compositional aspects.

Our perspective is to equip a Digital Score Library (DSL) with such an algebraic language, in order to derive "intentional" scores from "extensional" ones (the Library), in a direct correspondence with relational databases and the relational algebra [8]. We focus on operations that apply to the model space *in closed form*, i.e., which map instance(s) of the model to other instance(s). With closure comes composition: if s is a model instance (a "score") and o_1, o_2 are two operations, then $o_1(s)$ is a "score", $o_2(o_1(s))$ is also a "score", and we obtain an algebraic structure (in the mathematical sense) that lets us manipulate score material in order to produce new representations.

This approach brings, to the design and implementation of applications that deal with symbolic music, the standard and well-known advantages of specialized data management systems. Let us just mention the few most important: (i) ability to rely on a stable, well-defined and expressive data model, (ii) independence between logical modeling and physical design, saving the need to confront programmers with intricate optimization issues at the application level and (iii) efficiency of set-based operators and indexes provided by the data system.

In summary, we expose in the rest of the paper the following contributions

1. *A vision*. We describe in Section 2 a conceptual setting where digital libraries of score encodings can be leveraged to support structured content manipulation.

2. *A model*. Section 3 proposes a model that captures the most common features of CMN, along with a high-level, algebraic query language.

3. *Implementation guidelines*. Finally, we provide in Section 4 a technical discussion, based on our current implementation choices, showing the limited efforts required to achieve our vision.

Section 5 briefly discusses related work and Section 6 concludes the paper and discusses ongoing and future work.

2. VISION

Figure 1 summarizes the envisioned system. We propose in Section 4 a technical discussion related to our current implementation choices, but the figure exposes the main conceptual features at a convenient level of abstraction.

The bottom layer is a Digital Score Library (DSL) managing corpora of scores in some encoding, whether MusicXML, MEI, or any other legacy format (e.g., Humdrum). As explained in the introductory part, such encodings are not designed to support content-based manipulations, and, as a matter of fact, it is hardly possible to do so. Access to *explicit* music content information is intricate, due to the complex interleaving of content-oriented and rendering-oriented elements. Extracting a mere sequence of notes from MusicXML or MEI for instance is not a trivial task. Using implicit features that could be derived from the encoded content is even more difficult.

We therefore *map* the encoding toward a model layer where the content is extracted from the encoding and structured according to the model structures. One mapper has to be defined for each possible encoding, as shown by the figure which assumes that MusicXML and MEI documents cohabit in a same DSL. Adding a new source represented with a new encoding is just a matter of adding a new mapper. Each document in the DSL is then mapped to a (virtual) instance of the model. These define virtual – no materialization occurs – corpora of music notation objects that we will call *vScores* in the following.

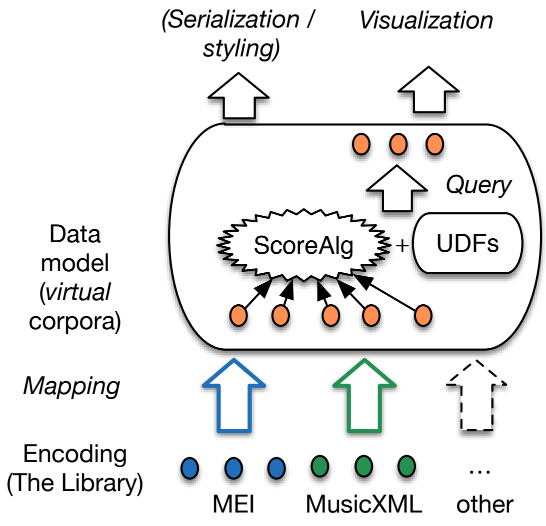


Figure 1. Envisioned system

The data model layer encapsulates both data representation and data operations. We further distinguish two kinds of operations: *structural operators* and *user-defined functions* (UDFs). The former implement the idea that structured score management corresponds, at the core level, to a limited set of fundamental operations, grouped in a score algebra, that can be defined and implemented once for all. The latter acknowledges that the richness of music notation manipulations calls for a combination of these operations with user-defined functions at early steps of the query evaluation process. Modeling the structural operators and combining them with user-defined operations constitutes the operational part of the model. This yields a query language whose expressions define the set of transformations that produce new *vScores* from the base corpora.

The result of a query is itself a new, intentional corpora. This gives rise to several potential exploitations. First, the result can be kept in the user space, as a "view" (using database terminology) over the base corpora. A performer could for instance keep a set of parts for her next rehearsal/concert. Second, the query result can be visualized, possibly with representations that emphasize analytical aspects computed from the scores.

Finally, derived *vScores* can be in serialized in a permanent storage, in a format compliant to one of the encoding standards. This is where styling could take place, in a process which is the exact opposite of the mapping which abstracts a content from MusicXML or MEI documents. Serialization of *vScores* implies the "decoration" of pure content with rendering features. Voices must be assigned to staves, the clef must be chosen based on the voices range, alterations must be displayed according to some general policy, etc. We do not elaborate on this process which, as explained above, is directly related to the complex issues of concerns separation in the music notation domain and falls beyond the scope of our current work. We note that, in some sense, the vision outlined above constitutes a possible framework to investigate this issue. A way to provide a meaningful distinction between content and rendering would indeed be to define a pair of (mapping / serialization) operations that produce an alternative rendering of a score while preserving its content.

In the subsequent sections, we implement this vision with a conceptual model applied to CMN, and expose our technical choices to make the whole approach practical.

3. THE DATA MODEL

We now present a simple data model that extends the relational model with the concepts of voices and events. The model features a core algebra which is mostly illustrated via examples expressed in a high-level language. A formal presentation of the algebra can be found in [9].

3.1 Schema: events, voices, scores and opera

CMN scores are modeled as polyphonic pieces composed of "voices", each voice being a sequence of "events" in some music-related domain (notes, rests, chords, syllabs) such that only one event occurs at a given instant for a given duration. The concepts of voices and events (with non null duration) are shared by most of the encodings we are aware of, in the field of CMN.

3.1.1 Events

An event e is some value v observed during an interval $[t_1, t_2[$. For our purposes, v is any value taken from a domain **dom**, and we note $\mathcal{E}$(**dom**) the set of events on **dom**. Of particular interest are the following (musical) domains, with some internal operators.

- Sounds (**dsound**): represents n simultaneous tones, $n \geq 1$). This covers simple sounds (notes, $n = 1$) and composed sounds (chords, $n > 1$).
- Syllables, (**dsyll**).

Sound is a complex notion that can be decomposed in several components (height, intensity, timbre). In practice, we are limited to those captured by the notational system, mostly the frequency (pitch and octave). Other aspects are sometimes indirectly represented (for instance, timbre by the instrument name).

We do not restrict the events to musical domains. For instance, an event in the **dint** domain might represent the value of an interval between two voices at a given timestamp. Such events can be inferred from the notation, and can enrich the representation. Beyond this simplistic illustration, this permits the definition of generalized scores that extend the usual concept by combining musical events with non-musical domains representing, for instance, some analytic feature.

3.1.2 Voices as Time Series

A musical time series (or *voice* to make it short) is a mapping from the time domain $\mathcal{T}$ (a discrete, ordered set isomorphic to $\mathbb{Q}$) into a set of events $\mathcal{E}$(**dom**). We denote by **Voice(dom)** the type of a voice, where **dom** is the domain of interest.

A voice is an instance of a voice type. So, for instance:

- v_1: **Voice(dsound)** denotes a voice v_1 which represents a function from $\mathcal{T}$ to "pure" music events.
- v_2: **Voice(`dint`)** denotes a voice v_2 which represents a function from $\mathcal{T}$ to integers, such as the intervals between two (music) voice.
- v_3: **Voice(dsyll)** denotes a voice which represents a function from $\mathcal{T}$ to text, such as lyrics.

Since a voice is a function, there is exactly one event at each instant (in other words, events cannot overlap). We can partly relax this constraint by adding to each domain a distinguished *null value* $\perp$ which denotes the "absence" of event (see [9] for a detailed discussion).

3.1.3 Scores as synchronized time series

We can now define scores. At a basic level, a score is a synchronization of voice(s). We extend this definition to capture a recursive organization of scores built from subscores.

- v a voice, then v is a score.
- if $s_1, \cdots, s_n$ are scores, the sequence $< s_1, \cdots, s_n >$ is a score.

The type of a score is the enumeration of voices that constitute a score, associated with their names. For instance:

1. The type T_q of a quartet is

```
[violin1: dsound, violin2: dsound,
     alto: dsound, cello: dsound ]
```

2. The type T_v of a vocal part is:

```
[lyrics: dsyll, monody: dsound]
```

3. The recursive structure of a score with a vocal part of type T_v and a figured bass is

```
[vocal: Tv, bass: dsound]
```

Instances of these types are time series from $\mathcal{T}$ to, respectively, **dsound**^{4}, **dsound** $\times$ **dsyll**, and (**dsound** $\times$ **dsyll**) $\times$ **dsound**. Conceptually, the first one represents a function which associates to each timestamp a 4-tuple of music events, the second one a function which associates to each timestamp a pair (sound, syll). Essentially, a score extends the concept of voice (i) by allowing several events to occur simultaneously and (ii) by labelling events with names, providing a "hook" to refer to them with operations. Unifying the model for voices and scores makes it easy to define operations that remain in a consistent setting.

3.1.4 Corpora as extended relations

Finally, an *opus* is a tuple of values which can either be atomic values (strings, integers, floats) or scores. Opuses with similar structure can be grouped in a *Corpus*. If we compare with the standard relational approach, a corpus is a container of similar objects, akin to a table, and an opus is an element in the container (a row in the table). A database is a set of corpora.

Since a corpus gathers opera with similar type, this type can be summarized as a corpus schema. The following example shows a possible schema for a *Quartet* corpus.

```
Quartet (id: int,
         title: string,
         composer: string,
         published: date,
         music: Score [v1: dsound,
                       v2: dsound,
                       alto: dsound,
                       cello: dsound
                      ]
         )
```

3.2 User query language

We need a concrete syntax to express our score manipulations. Since we want to limit as much as possible the extension required to adapt our model to an existing system, this leaves two main options: SQL and XQuery. The examples below are based on XQuery which presents several features of interest, including the ability to incorporate functions, and fits naturally with the hierarchical nature of our data items (opuses, made of scores, made of voices, with possibly intermediate levels).

In order to avoid formal developments, the main characteristics of the language are introduced with examples. The interested reader is referred to a companion paper [10] that details the design of the query language and the related implementation issues. The examples below *cannot* be directly evaluated as XQuery expressions, since they are interpreted over virtual instances of the above score model. The actual evaluation relies on a lightweight query rewritting presented in the next section.

The examples rely on the *Quartet* corpus (refer to the previous section for its schema). Our first example creates a list of the Haydn's quartets, reduced to the violin's parts.

```
for $s in collection("Quartet")
where $s/composer="Haydn"
return $s/title, Score($s/music/v1, $s/music/v2)
```

Recall that `music` is an attribute of type **Score** of the *Quartet* corpus. This first query shows two basic operators to manipulate scores: projection on score/voices with the standard"/" XPath syntax, and creation of new scores with the `Score()` synchronizer operator.

A third operator that allows the derivation of new score contents is MAP: it represents a higher-order function that applies a given function f to each event in a voice, and returns the voice built from f's results. Here is an example: we want the quartets where the violin1 part is played by a B-flat clarinet. We need to transpose the `v1` part 2 semitones up.

```
for $s in collection("Quartet")
where $s/composer="Haydn"
let $clarinet := Map ($s/music/v1, transpose (2))
let $clrange  := ambitus ($clarinet)
return $s/title, $clrange,
         Score($clarinet, $s/music/v2,
               $s/music/alto, $s/music/cello)
```

This second query shows how to define variables that hold new content derived from the stored scores via *user defined functions* (UDFs). For the sake of illustration we create two variables, `$clarinet` and `$clrange`, calling respectively `ambitus()` and `transpose()`.

In the first case, the function has to be applied to each event of the violin voice. This is expressed with MAP which yields a new voice with the transposed events. By contrast, `ambitus()` is directly applied to the voice as a whole. It produces a scalar value (not a voice).

MAP is the primary means by which new voices can be created by applying all kinds of transformations. MAP is also the operator that opens the query language to the integration of *external* functions: any library can be integrated as a first-class component of the querying system, providing some technical work to "wrap" it conveniently (see next section).

By "mapping" a Boolean expression e to a voice, we can filter out the events that do not satisfy e, replacing them by the `null` event $\perp$. Note that this is different from *selecting* a score based on some property of its voice(s). The next query illustrates both functionalities: we select all the psalms such that the vocal part contains some word,"nullify" the events that do not belong to the first ten measures, and trim the voice to keep only non-null events.

```
for $s in collection("Psalters")
let $sliced := trim(select ($s/air/vocal/monody,
                   measure(5, 10)))
where contains ($s/air/vocal/lyrics, "Heureux")
return $s/title, Score($sliced)
```

We can take several opuses as input and produce an opus with several scores as output. The following example takes three chorals, and produces an opus with two scores associating respectively the alto and tenor voices.

```
for $c1 in coll("Chorals")[@id="BWV49"]/music,
    $c2 in coll("Chorals")[@id="BWV56"]/music,
    $c3 in coll("Chorals")[@id="BWV12"]/music
return <title>Excerpts of chorals</title>,
         Score($c1/alto, $c2/alto, $c3/alto),
         Score($c1/tenor, $c2/tenor, $c3/tenor)
```

Finally, our last example illustrates the extended concept of "score" as a synchronization of voices which are not necessarily "music" voices. The following query produces, for each quartet, a score containing the violin 1 and cello voices, and a third one measuring the gap (interval) between the two.

```
for $s in collection("Quartet")/music
let $intervals := Map(Score($s/v1,$s/cello),
                   interval())
return Score ($s/v1, $s/cello, $intervals)
```

Such a "score" cannot be represented with a traditional rendering. Additional work on visualization tools that would closely put in perspective music fragments along with some computed analytic feature is required.

4. IMPLEMENTATION

Our system has been fully implemented in NEUMA. It integrates an implementation of our score algebra, a mapping that transforms serialized scores to vScores, and off-the-shelf tools (a native XML database, BASEX [1], a music notation library for UDFs, MUSIC21 [2] [11]). This simple implementation yields a query system which is both powerful and extensible (only add new functions wrapped in XQuery/BASEX). We present its salient aspects.

4.1 Architecture and query processing

Figure 2 shows the main implementation modules. Data is stored in BASEX in two collections: the semi-virtual collection (e.g., Quartet) of music documents (called *opus*), and the collection of serialized scores, in MusicXML or MEI. Each virtual element `scoreType` in the former is linked to an actual document in the latter. Those collections are managed by the NEUMA digital library[12].

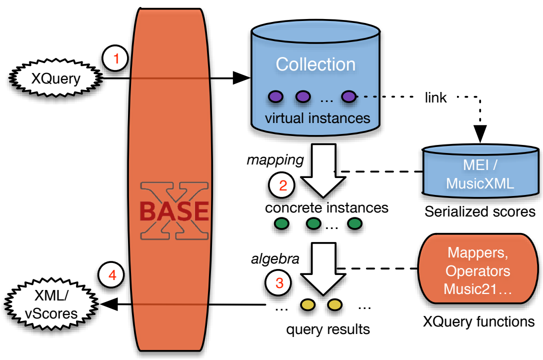


Figure 2. Architecture

The evaluation of a query proceeds as follows. First (step 1), BASEX scans the virtual collection and retrieves the opus matching the `where` clause. Then (step 2), for each opus, the embedded virtual score element has to be materialized. This is done by applying the mapping that extracts a vScore instance from the serialized score, thanks to the link in each opus.

Once a vScore is instantiated, algebraic expressions, represented as composition of functions in the XQuery syntax, can be evaluated (step 3). We wrapped several Python and Java libraries as XQuery functions, as permitted by the BASEX extensible architecture. In particular, algebraic operators and mappers are implemented in Java, whereas additional, music-content manipulations are mostly wrapped from the Python Music21 toolbox.

The XQuery processor takes in charge the application of functions, and builds a collection of results, finally sent to the client application (step 4). It is worth noting that the whole mechanism behaves like an ActiveXML [13] document which *activates* the XML content on demand by calling an external service (here, a function).

[1] `http://basex.org`
[2] `http://web.mit.edu/music21`

4.2 External components

How do we integrate *functions* that manipulate the score representation? In general, we need to resort on an external component. Getting the highest note of a voice for instance is hardly expressible in XQuery. In general, getting such features would require awfully complex expresssions. This is due to very detailed decomposition of any XML encoding which makes very difficult the reconstruction of high-level features.

XQuery is extendible to user-defined functions, and the point is technically harmless. In our current implementation, we simply "wrap" relevant functions in an external library compliant to BaseX. The following example retrieves all the quartets such that the first violin part gets higher than e6, using a *highest()* UDF.

```
for $s in collection("Quartet")
where highest($s/music/v1)  > 'e6' return $s
```

A naive, direct evaluation would load the MusicXML (or MEI) document from the underlying storage, pass it to the function and get the result. This works with quite limited implementation efforts. Such an evaluation raises, however, strong efficiency issues. In general, any function will need to access to the whole score encoding (or to put it differently, we cannot in general anticipate the part of the score it needs to access). This has to be done for each score in the collection: a clearly unacceptable burden, likely to make the full query process highly inefficient.

A solution is to materialize the results of User Defined Functions as metadata in the virtual document and to index this new information in BASEX. This can directly serve as a search criteria without having to materialize the vScore. The result of the *highest()* function is such a feature. Index creation simply scans the whole physical collections, runs the functions and records it result in a dedicated `index` sub-element of each opus, automatically indexed in BASEX. To evaluate the query above, it uses the access path to directly get the relevant opus.

```
for $s in collection("Quartet")
where index/v1/highest   > 'e6'
return $s
```

5. RELATED WORK

Music Information Retrieval has mostly considered so far unstructured search, and notably similarity search [14]. Unstructured search is convenient to the end user, and avoids intricate considerations related to music notation structure. A limitation is that the granularity of results stays at the document level, and cannot access to finer internal components. Our work allows such a fine-grained inspection.

An early attempt to represent scores as structured files and to develop search and analysis functions is the HumDrum format. Both the representation and the procedures are low-level (text files, Unix commands) which make them difficult to integrate in complex application. We are only aware of a few other approaches. An attempt to transpose database principles to score management is presented in [15]. The authors of [16] study how XQuery may be directly used over MusicXML. XQuery is a general-purpose query language which hardly adapts to the specifics of symbolic music manipulation. Besides, by ignoring the issue of the inherent underlying data model, closure of operations becomes undecidable, and the query language misses the essential properties that makes it safely usable in applications.

We make the case for a clear identification of the data model that underlies the operations on “scores”. This allows to abstract from useless details, brings a support to the definition of closed operations, and enforces to review what kind of content we aim at manipulating. We might always (rightly) complain that part of the meaningful content is lost, and that rare features (e.g., chords with varying note durations) are not adequately captured by an abstract model, but this seems the price to pay for a clear understanding of the stakes. As a side effect, this allows to integrate distinct encodings in a consistent setting.

The mapping process by which this is achieved is reminiscent from mediation architectures used for data sources integration [17, 18, 19, 20], and can be seen as an application of method that combines queries on physical and virtual instances. It borrows ideas from ActiveXML [13], and in particular the definition of some elements as “triggers” that activate external calls.

Abstracting an agnostic score content from XML formats is a design shared by several earlier proposals, including NEUMA [21], Music21 [11] and formal approaches such as Euterpea [22] that attempt to model music content for generative or analytic purposes. This allows in particular to develop manipulation primitives independently from serialization concerns. We can re-use for instance in our implementation some of the analytic functions supplied by Music21, and combine these functions to the structural database operators that constitute the core of our contribution.

6. CONCLUSION AND FUTURE WORK

We propose a new approach to treat music notation as a structured source of information apt at supporting modern query techniques inspired by object and relational databases. A debatable aspect of the approach is the lossy mapping that extract “content” from the notation. There is no well-defined answer to the separation of score content from score rendering, which can be perceived as an encouragement to further investigate the issue.

On the other hand, having a high-level specification language to combine, change and derive scores offers quite promising perspectives for performance, teaching and analysis of music content. We are in particular keen to explore the following ideas:

- Maintain a tight synchronization between the parts of a score and the full ensemble, in order to reflect any change (e.g., an annotation).
- Propose new visualizations of music notation, dictated not by performance issues, but by the need to grasp some analytical aspect.
- Study styling mechanisms which can map an abstract music notational content to sheet representation.

Our implementation in NEUMA is available to the community of scholars, musicologist and data scientists who aim at investigating the corpora of this library for analytic purposes. We hope that the design presented in the paper is generic enough to inspire similar endeavours. We will be glad to provide our software components to anyone wishing to exploit these ideas in a similar system.

Acknowledgments

This work is partially supported by the Musica Consortium, `https://humanum.hypotheses.org/503`.

7. REFERENCES

[1] E. Selfridge-Field, Ed., *Beyond MIDI : The Handbook of Musical Codes*. Cambridge: The MIT Press, 1997.

[2] M. Good, *MusicXML for Notation and Analysis*. W. B. Hewlett and E. Selfridge-Field, MIT Press, 2001, pp. 113–124.

[3] P. Rolland, “The Music Encoding Initiative (MEI),” in *Proc. Intl. Conf. on Musical Applications Using XML*, 2002, pp. 55–59.

[4] “Music Encoding Initiative,” http://music-encoding.org, Music Encoding Initiative, 2015, accessed Oct. 2015.

[5] A. Hankinson, P. Roland, and I. Fujinaga, “The Music Encoding Initiative as a Document-Encoding Framework,” in *Proc. Intl. Conf. on Music Information Retrieval (ISMIR)*, 2011, pp. 293–298.

[6] “W3C Music Notation Community Group,” https://www.w3.org/community/music-notation/, 2015, last accessed Jan. 2016.

[7] L. Pugin, J. Kepper, P. Roland, M. Hartwig, and A. Hankinson, “Separating Presentation and Content in MEI,” in *Proc. Intl. Conf. on Music Information Retrieval (ISMIR)*, 2012.

[8] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Addison-Wesley, 1995.

[9] R. Fournier-S'niehotta, P. Rigaux, and N. Travers, “An Algebra for Score Content Manipulation,” CEDRIC laboratory, CNAM-Paris, France, Tech. Rep. CEDRIC-16-3616, 2016, submitted for publication. [Online]. Available: http://cedric.cnam.fr/index.php/publis/article/FRT16d

[10] ——, “Querying XML Score Databases: XQuery is not Enough!” CEDRIC laboratory, CNAM-Paris, France, Tech. Rep. CEDRIC-16-3612, 2016, submitted for publication. [Online]. Available: http://cedric.cnam.fr/index.php/publis/article/FRT16c

[11] M. S. Cuthbert and C. Ariza, "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proc. Intl. Conf. on Music Information Retrieval (ISMIR)*, 2010, pp. 637–642.

[12] P. Rigaux, L. Abrouk, H. Audéon, N. Cullot, C. Davy-Rigaux, Z. Faget, E. Gavignet, D. Gross-Amblard, A. Tacaille, and V. Thion-Goasdoué, "The design and implementation of neuma, a collaborative digital scores library - requirements, architecture, and models," *Int. J. on Digital Libraries*, vol. 12, no. 2-3, pp. 73–88, 2012.

[13] S. Abiteboul, O. Benjelloun, and T. Milo, "The active XML project: an overview," *VLDB J.*, vol. 17, no. 5, pp. 1019–1040, 2008. [Online]. Available: http://dx.doi.org/10.1007/s00778-007-0049-y

[14] R. Typke, F. Wiering, and R. C. Veltkamp, "A Survey Of Music Information Retrieval Systems," in *Proc. Intl. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[15] Z. Faget and P. Rigaux, "A Database Approach to Symbolic Music Content Management," in *Intl. Symp. on Exploring Music Contents (CMMR)*, 2010, pp. 303–320.

[16] J. Ganseman, P. Scheunders, and W. D'haes, "Using XQuery on MusicXML Databases for Musicological Analysis," in *Proc. Intl. Conf. on Music Information Retrieval (ISMIR)*, 2008.

[17] H. Garcia-Molina, J. Ullman, and J. Widom, *Database System Implementation*. Prentice Hall, 2000.

[18] A. Doan, A. Halevy, and Z. Ives, *Principles of Data Integration*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[19] S. Abiteboul and et al., "WebContent: Efficient P2P Warehousing of Web Data," in *VLDB'08 Very Large Data Base*, August 2008, pp. 1428–1431.

[20] N. Travers, T. T. D. Ngoc, and T. Liu, "Tgv: A tree graph view for modeling untyped xquery," in *(DASFA) 12th International Conference on Database Systems for Advanced Applications*. Springer, 2007, pp. 1001–1006. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-71703-4_92

[21] P. Rigaux and Z. Faget, "A database approach to symbolic music content management," in *Exploring Music Contents - 7th International Symposium, CMMR 2010, Málaga, Spain, June 21-24, 2010. Revised Papers*, 2010, pp. 303–320.

[22] P. Hudak, *The Haskell School of Music – From Signals to Symphonies*. (Version 2.6), January 2015.

THE ONTOLOGY OF LIVE NOTATIONS THROUGH ASSEMBLAGE THEORY

David Kim-Boyle
Sydney Conservatorium of Music
The University of Sydney
david.kim-boyle@sydney.edu.au

ABSTRACT

This paper uses assemblage theory to help develop an ontological framework for better understanding live notation practice. Originally developed by Deleuze and Guattari across a range of theoretical writings, assemblage theory is more fully explicated in the work of Manuel de Landa in the more focused context of social ontology. This paper examines the basic concepts of assemblage theory such as material components, expressive capacities, and relations of exteriority and how they may provide useful insights in the analysis of music which explores the creative potential of live notation. The temporal dynamics of nonlinear musical forms are discussed and assemblage theory is shown to be a powerful tool for promoting a better understanding of how the various interactions between material and expressive components help catalyze the emergent properties of the assemblage and through it, the ontological identity of a live notation aesthetic practice.

1. INTRODUCTION TO ASSEMBLAGE THEORY

In *A New Philosophy of Society*, De Landa uses assemblage theory to develop a social ontology for better understanding the complex dynamics of social structures. [1] Drawing heavily from Deleuze, [2] De Landa describes assemblages as constructs defined by what Deleuze refers to as *relations of exteriority*. *Relations of exteriority* ascribe defining characteristics to the relations that exist between an assemblage's component parts. Indeed, the ontological identity of an assemblage becomes an emergent property of those relationships rather than a reductive one -

> *...the reason why the properties of a whole cannot be reduced to those of its parts is that they are the result not of an aggregation of the components' own properties but of the actual exercise of their capacities.* [1]

To that end, and especially in the context of social ontology, assemblage theory refers to objects and relations between them that are ostensibly real. [3] In the course of his investigation, De Landa applies assemblage theory in the analysis of a variety of social structures from interpersonal networks through to the organization of institutions. Each of the social constructs which De Landa examines comprise interchangeable components which have both material and expressive capacities.

The materiality of an assemblage's components is constituted by its spatial presence. De Landa offers numerous examples within the framework of societies including bodies, food, physical labor, tools, machines, and buildings.

A component's materiality is complemented by its expressive capacity. These expressive capacities encompass both linguistic and non-linguistic forms of social expression. The content of an interpersonal conversation is given by De Landa as an example of linguistic expressivity while accompanying facial expressions or bodily gestures are of a non-linguistic form. In each case, both forms of expressivity are a realization of the expressive capacity of material components. It is worth recognizing as well, that these expressive capacities can only be realized through the interaction of material components and to that end, expressive capacities are a second-order property.

Defining the materiality and expressive capacities of an assemblage's components constitutes a type of analytic reduction. The interaction between these components, however, acts as a synthetic complement, helping to stabilize the ontological identity of an assemblage through processes of territorialization and deterritorialization. Both Deleuze and De Landa describe how territorialization is most simply defined by the physical networks formed between component elements. Once again using the example of a conversation, De Landa demonstrates how that conversation territorializes a space through the physical presence and interaction between two people. Conversely, a deterritorialization may occur when the physical presence is less material or spatial boundaries are blurred such as might occur when that same conversation is enabled through the modulation of electromagnetic waves over a telecommunications network.

Territorialization and deterritorialization are a first-order synthetic process in the respective stabilization and destabilization of an assemblage. A second-order articulation is formed by a process of *coding* or *consolidation*,

[2] which consolidates "…the identity of the assemblage or, on the contrary, allow(s) the assemblage a certain latitude for more flexible operation while benefiting from genetic or linguistic resources." [1] Deleuze and Guattari further define this process of consolidation as one of interiorization based on processes of reinforcement (intercalary events), distribution, and articulations of superposition. [2] These various processes of homogenization within assemblages are summarized in Figure 1.

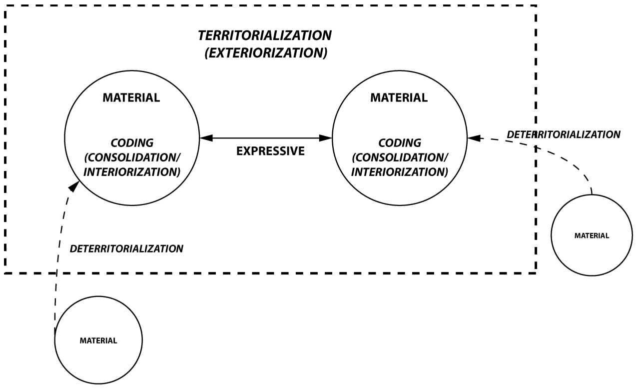


Figure 1. Schematic of the basic concepts of assemblage theory.

While assemblage theory has been developed to better understand social interactions and constructs, to what extent might these concepts be useful in the development of an ontological framework for understanding live notation practice?

2. LIVE NOTATION PRACTICE – A BRIEF REVIEW

Live notation practice is a relatively new area of creative inquiry which encompasses many different artistic practices and aesthetic styles. It includes work in which scores are generated live but also work in which scores are largely predetermined prior to performance but only recalled at the instance of performance. Unifying all these approaches is a move away from paper-based to screen-based media and an embrace of animated graphical typographies. Along with this transition come a number of visual design constraints that influence formal structural elements of the music in ways largely unknown to paper-based media. [4] While live notations often test the limits of a performer's sight reading ability, [5] the graphical schemata employed in such notations typically remains stable from realization to realization. Similarly, the way the notation develops during the performance does not typically fall outside the bounds of predetermined constraints established by the composer. In Kim-Boyle's *point studies no. 2*, for example, for any two pitched instruments, the two performers interpret a grid of moving, interconnected colored nodes. The spatial distribution and movement of nodes, their colors, sizes, and separation, are determined through various stochastic processes which result in different nodal configurations for each performance. Despite these different manifestations of the score, the manner in which its various components are interpreted and the graphical schemata itself, remains fixed.

Similarly, in Ryan Ross-Smith's *Study no. 41 [rr:___]* for nine or more instrumentalists, one of over forty studies by Ross-Smith exploring live notation, performers interpret a kinetic shell of nodes with individual instrumentalists not knowing which nodes the other instrumentalists have chosen. Despite this uncertainty, along with which comes a tremendous variety in musical expression, the graphic schemata used in the work and the way performers interpret its flowing movements remains stable.

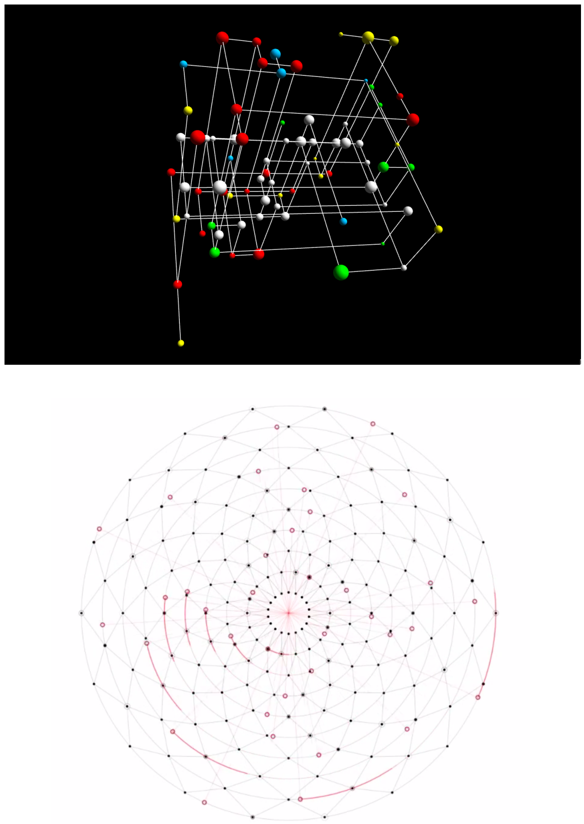


Figure 2. Upper) Snapshot from the score of Kim-Boyle's *point studies no. 2*; Lower) Ross-Smith's *Study no. 41 [rr:___]*.

In both Kim-Boyle and Ross-Smith's works the key or code that establishes how the score is to be interpreted remains constant across performances as does the graphic typography employed in the score's visual design. Performers are never presented with symbols they have not previously encountered, nor does the movement of symbols present unique transformations. In this respect this presents performance challenges no different to those involved with the interpretation of works written in common practice notation or which employ more overtly graphic typographies which have clear rules or guidelines regarding their interpretation. The only difference, of course, is that the real-time manifestation of the score contains kinetic components the low-level organisation of which may differ from performance to performance.

Unlike fixed media scores, live notation enables more complex nonlinear processes to be integrated into the formal structure of musical works. In Kim-Boyle's *Valses and Etudes* for pianist and computer (MaxMSP/Jitter), for example, the pianist is presented with a series of score fragments from established works in the piano repertoire by composers such as Chopin, Ravel, Webern, Debussy, and Schoenberg. The order in which the fragments are presented is based on a series of weighted probabilities that determine the likelihood that one score will follow another, i.e. a first-order Markov chain. Had the pianist been asked to determine the succession of musical fragments, as they might in a similar open form work such as Stockhausen's *Klavierstücke No. XI* (1956) it is unlikely they would be able to implement such successions as are derived through the Markov chain selection process. In addition, by delegating the ordering process to a Max patch, the risk of the performer choosing fixed, and subjectively preferred orderings of material is also avoided.

Nonlinear processes can be integrated in many levels of a musical score other than structural ordering. Rebelo, for example, has explored how notation can be made responsive to live elements of performance [6] and composers such as Ross-Smith, Vickery, Kim-Boyle and others have integrated various nonlinear processes into lower-level musical structures such as pitch selection or rhythmic articulation. Given that assemblages are defined by relations of exteriority rather than by their component elements, assemblage theory is particularly well suited to helping develop an ontological framework for live-notation practices exploring such nonlinear processes.

3. NOTATION AS ASSEMBLAGE

Musical scores assume many forms but usually adopt either a descriptive function through describing musical structures to be interpreted by performers, a prescriptive function in prescribing a course of performative actions or some combination of the two. It is important to distinguish between these contrasting roles as the manifest sonic outcomes of each may be quite different, subsequently broadening the ontological identity of the work. In common practice notation, which is inherently descriptive, a score's material components include graphic symbols which denote various structural elements such as pitch, rhythmic values, dynamics and articulations. In such notation, the symbols used to define these elements has remained relatively stable for hundreds of years while the manner in which their expressive capacities are realized has also helped to stabilize the ontological identity of the works they are intended to articulate. Through the expression of these material components, the traditional (common practice notation) score thus territorializes a musical space through stabilizing relationships between its material components and their expressive capacities. Conversely, a prescriptive notation establishes stable relationships between performative gestures the expressive realization of which may result in quite different sonic outcomes from one performance to another.

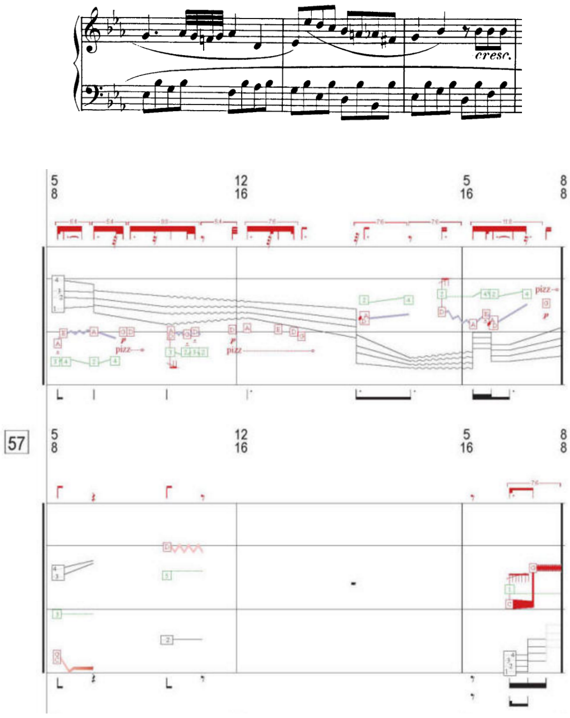


Figure 3. Upper) Descriptive notation in which a score's material components represent stable musical properties such as pitch, rhythm or dynamics (extract from score for W. A. Mozart's *Piano Sonata in C Minor*, K.457); Lower) Prescriptive notation in which a score's material components represent stable performative gestures which may result in a wider variety of sonic results (extract from the score for Aaron Cassidy's *Second String Quartet*).

The relationship between a score's material components, whether they have a prescriptive or descriptive function, and how those components are expressed, i.e. the expressive capacity of those components, can only be strongly related when that relationship exists within an understood code of practice. This *decoding* from the material to the expressive is traditionally informed by the conventions of performance practice. When the expressive realization of a score's material components is not strongly coded, however, the ontological identity of a work becomes less strongly bound to sonic outcomes. In Christian Wolff's *Edges* (1964) for example, the score presents the performer with a series of graphic symbols spatially distributed on a single page. The performers are free to musically interpret the symbols themselves and the order in which they are performed. This naturally provides each realization of *Edges* with a spontaneity and variety not bound by conventions or codes of strict performance practice. Similarly in Cardew's celebrated *Treatise* (1963-67), performers are free to determine how they interpret the score's diverse range of graphic symbols. Both *Treatise* and *Edges* thus become defined not so much by any manifest sonic outcome but by the interrelationships, or relations of exteriority, that emerge between atomic musical gestures.

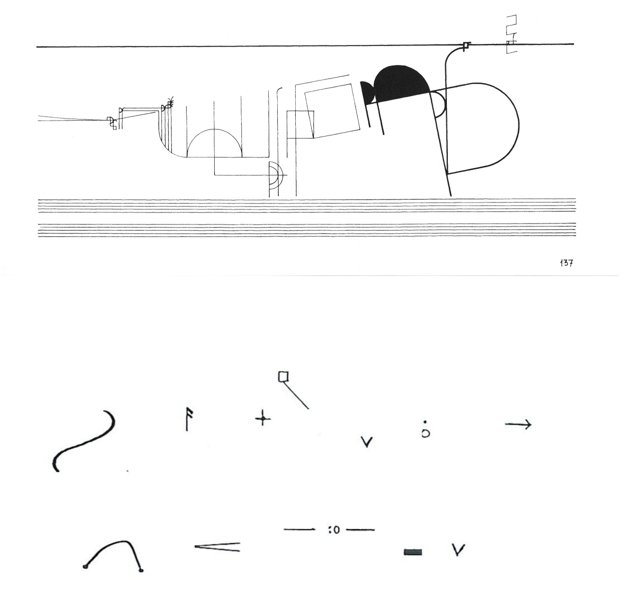


Figure 4. Upper) Excerpt from the score for Cardew's *Treatise* (1963-67); Lower) Detail from Wolff's *Edges* (1964).

The material components of a graphic score may be articulated at many different levels, have different referential allusions, and be parsed into various different aggregates. The *Treatise* excerpt in Figure 4, for example, makes strong allusions to common practice notation through the prominent use of staves which brings with it common practice notation's inherent linear associations but also helps frame the prominent use of long horizontal, vertical and curved lines elsewhere on the page. The deconstructed staves and various other shapes from which this page of the score are constructed, can be grouped into different low-level assemblages or aggregates, providing different expressive capacities to these material components as their interrelationships shift. The graphic shapes in the score excerpt shown in Figure 4, for example, can form various different aggregates, see Figure 5, each of which suggest unique expressive possibilities.

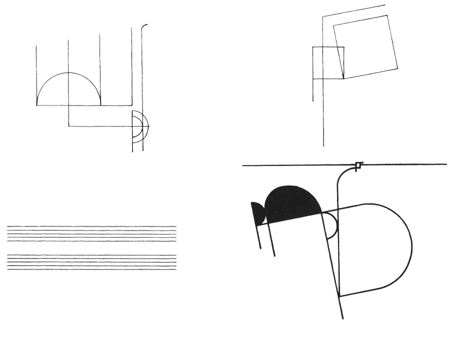


Figure 5. Possible aggregates within Cardew's *Treatise*.

Paper-based scores such as *Treatise* or *Edges*, in which the distribution of material components is fixed despite the relations of exteriority that may pertain to their expressivity also present the distinct likelihood that performers will establish certain expressive preferences. Ironically, this tendency to stabilize relationships has the affect of prioritizing sonic outcomes in a similar way to that of common practice notation. In other words, the ontology of the work territorializes a musical space through *habitual repetition*. [1]

The manner in which a score's material components are decoded in live notation practice follows similar trajectories to those experience in fixed, paper-based notation but establishes less of a likelihood that preferential expressive capacities of the score will be habitually established. In the two works cited earlier, Kim-Boyle's *point studies no. 2* and Ross-Smith's *Study no. 41 [rr:___]*, for example, the material components of the score remain stable as does the manner in which they are expressed by the performer. The decoding mechanism, in other words, is clearly defined and remains consistent from performance to performance. The live notation of both works, however, establishes a greater opportunity for performers to explore unique expressive possibilities that emerge from constantly shifting relations of exteriority. Similarly, in works employing live notation in which the material components can be interpreted in many different ways, such as in Pedro Rebelo's *Netgraph* (2010), see Figure 6, the performance issues related to the broader expressive capacities of the notation are not that dissimilar to those involved in graphic scores on fixed media, although the opportunities for playful exploration of a musical space are considerably enhanced through the live, dynamic notation.

Figure 6. Still shots from the live score of Pedro Rebelo's *Netgraph* (2010).

The fundamental and perhaps defining ontological distinction between fixed and live notation schemas to be explored is of course that pertaining to the temporal dynamics of each form of practice.

4. TEMPORAL DYNAMICS

As previously noted, common practice notation presents assemblages in which the material components are structured according to strictly linear relationships of exteriority. The understood decoding process through which these components are expressed territorializes the musical space of the work with a highly homogenous identity further stabilized through *habitual repetition*. [1] While De Landa uses Deleuze and Guatarri's term *territoriali-*

zation to refer to the social processes through which assemblages are spatially defined, it can also refer to the assignment of ontological identity. Thus, the territory established by a Beethoven piano sonata, for example, is such because the material components that comprise the work (pitches, rhythmic values, dynamic shadings, tempi etc.) exist in defined relationships that are reinforced and unvaried through repetition. Similarly, in works that are realized through a more prescriptively notated score, a gestural language is organized according to pre-established formal rules and repeated from one performance to another.

As noted by Bryant, within an assemblage, “Time and space should not be conceived as containers or milieus within which events take place, but rather as meshes of connective relations.” [7] To that end, as the relationships between the material components of an assemblage become less strictly linear, the ontological identity of the assemblage becomes defined by the relationships of exteriority that emerge between the expressive capacities of those material components. Even within nonlinear musical forms, these relationships can be codified and stabilized as Christian Wolff attempts in his *For 1, 2 or 3 People* (1964) which require performers to listen and respond to each other. Within live notation practice, synchronicities within essentially nonlinear temporal dynamics can occur in complex configurations as in Australian composer Lindsay Vickery’s *UBahn* (2012) for two violas, two cellos, double bass, percussion, and electronics where performers read scores from networked iPads and synchronicities between performers are determined by computer.

The temporal dynamics within an assemblage are not always strictly linear or nonlinear, in the same way that musical scores can combine both descriptive and prescriptive modes of notation. Roman Haubenstock-Ramati’s *Mobile for Shakespeare* (1960), for example, with its integration of common practice notation figures and graphical notation schema, blends the two. The work thus presents assemblages within assemblages with its material components connected in both linear and nonlinear relationships.

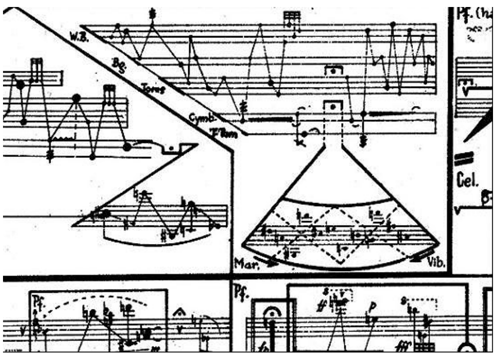


Figure 7. Detail from Haubenstock-Ramati’s *Mobile for Shakespeare* (1960).

Live notation practices often eschew the requirement for performers to determine how musical fragments are ordered. In some respects this parallels the approach taken in a work such as Earle Brown’s *Available Forms I* (1961) for orchestra in which the conductor determines the succession of discrete sections of musical material, but as previously noted the ordering process in live notation practices can allow more complex successions and distributions of musical components to be realized and help dissuade a tendency for performers to establish preferred orderings. This type of nonlinearity does not necessarily lead to a deterritorialization of the assemblage and corresponding destabilizing of the ontological identity of the work as assemblages are only destabilized through exogenous forces. Recalling the author’s *Valses and Etudes* in which the succession of musical fragments is determined by a first-order Markov chain, the stabilizing effect of habitual repetition is not present but this does not mean that the connective relationships between musical fragments results in a destabilization of the work’s identity. Rather, it highlights the fact that those relationships are more multifaceted than those of simple linear succession, i.e. they exist as a “mesh of connective relations.” It is through the consolidation of those relations, rather than their stabilization that the work’s identity is established.

Nonlinear relationships between the material components of live notation can be extended to lower levels of musical order. In some respects, this is not that dissimilar from the nonlinearity called forth in works such *as Mobile for Shakespeare*, but again, more complex nonlinear relationship can be realized in live notation practice. In Kim-Boyle’s *point studies no. 1* for any four musician, for example, the material components of the score comprising pitches, durations, and dynamic levels are stochastically distributed and related to each other, falling within certain boundaries but never entirely predictable. The relationship between the material components of the score change as the work develops through rotation and extension of arcs which determine the duration of notes and affect how the performers navigate through the score, and the appearance and gradual disappearance of nodes, which denote particular pitches. It is doubtful whether the same types of nonlinear relationships between the material components of the score could be so easily achieved in fixed media.

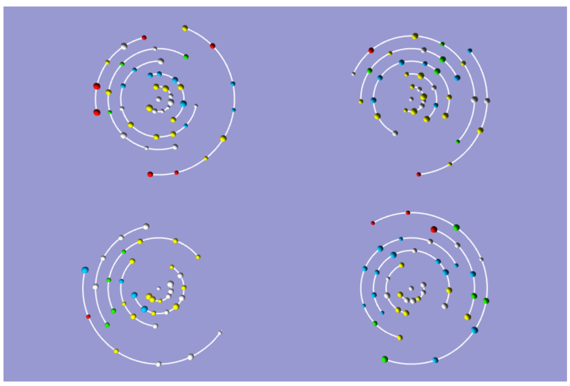


Figure 8. Score excerpt from the author’s *point studies no. 1*.

Like social ontologies in which relations of exteriority between material components can be one of exchange (such as that between a consumer and seller), the relationship between material components in a live score may also be related to the expressive capacity of performance. In Pedro Rebelo's *Netgraph*, cited earlier (see Figure 6), the material components of the score are responsive to the expressive capacities of other material components. In performance, the performers are spatially distributed across different physical locations and their interpretation of the score's graphical schema modulates that schema for other performers. These dynamic relationships are a unique feature and possibility of what Rebelo refers to as *reactive scores*. [6]

5. CONCLUSIONS

Assemblage theory presents a useful ontological framework for better understanding live notation practice. Through prioritizing relations of exteriority such a framework is particularly well suited to the analysis of nonlinear processes which live notation practices. It is hoped that this brief paper helps illustrate some ways in which assemblage theory can be applied in the analysis of live notation practices and provide useful insights into this rich field of creative enquiry.

6. REFERENCES

[1] M. De Landa, *A New Philosophy of Society – Assemblage Theory and Social Complexity*. Continuum: New York, 2006.

[2] G. Deleuze and F. Guattari, *A Thousand Plateaus*. Trans. B. Massumi. University of Minnesota Press: Minneapolis, 1987.

[3] G. Harman, "DeLanda's Ontology: Assemblage and Realism," *Continental Philosophical Review* 41, pp. 367-383, 2008.

[4] D. Kim-Boyle, "Visual Design of Real-Time Scores," *Organised Sound* 19(3), pp. 286-294, 2014.

[5] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal* 32(3), pp. 25-41, 2008.

[6] P. Rebelo, "Playspace – 3D Reactive Notation," June 9, 2010. Accessed January, 2015. Available at <https://pedrorebelo.wordpress.com/2010/06/09/playspace-3d-reactive-notation/>.

[7] L. R. Bryant, "DRG: 'Assemblages Against Totalities'" September 8, 2010. Accessed September, 2015. Available at <https://larvalsubjects.wordpress.com/2010/09/08/drg-assemblages-against-totalities/>.

[STUDY NO. 50] [NOTATIONAL BECOMING] [SPECULATIONS]

Ryan Ross Smith
Rensselaer Polytechnic Institute
ryanrosssmith@gmail.com

ABSTRACT

The use of animation in contemporary notational practices has become increasingly prevalent over the last ten years, due in large part to the increased compositional activities throughout Europe, the United Kingdom, and North America, and in particular Iceland and Western Australia.[1] The publication of several foundational texts,[2] and the materialization of focused scholarly meetings[3] and online consolidation projects[4] have also contributed to the expansion of this growing field of animated notational practice. The range of compositional ideas represented by these scores is vast, encompassing a wide variety of stylistic approaches and technological experimentation. While these ideas often demonstrate intriguing compositional directions, and the unique dynamic functionalities and visual characteristics of animated scores are clearly distinct from traditionally-fixed scores, it is the real-time generative processes of these scores that represent a shift in the very ontology of the musical score. In this paper I speculate on one possible framing for this ontological distinction by focusing on several attributes that, in combination, most explicitly demonstrate this distinction. These include the real-time, process-based qualities of generative animated notations, the *openness* that enables these procedural functionalities, the displacement of interpretive influence, and the *timeliness* of these processes in respect to the temporal relationship between generation, representation as notation, and sonic realization. A new work, *Study no. 50*, will be examined as a practical demonstration of these attributes, and will function as a jumping off point for a speculative discussion of the concept of *Notational Becoming.*

[1] Reykjavík-based S.L.A.T.U.R. and Perth, Australia-based Decibel Ensemble.
[2] Contemporary Music Review, Vol. 29:1, 2010 and Leonardo Music Journal, Vol. 21, 2011.
[3] TENOR 2015, Paris and NIME 2014, London [Interactive Music Notation and Representation Workshop]
[4] animatednotation.blogspot.com & animatednotation.com.

1. FRAMEWORK

1.1 The Animated Score

An Animated Score is any score that contains perceptibly dynamic characteristics that are essential to the symbolic representation of the compositional idea. The symbols and dynamic functionalities that populate these scores are here designated as Animated Music Notation [AMN]. The range of approaches to the design and functionality of animated scores is varied, but it is generally possible to make a broad distinction between those scores that are fixed prior to their dynamic activation, and those that generate and represent notational information in real time *as* the score is functioning. Cat Hope and Lindsay Vickery have noted that these *generative* animated scores "construct(s) components of the score in real-time," [1] determining the local and global symbolic and functional characteristics of these components [AMN] as it is produced, and nearly simultaneous with its realization in performance. The simultaneous, real-time generation and representation of notation in generative scores are often based on "dynamic systems [that] have the role of a '*nucleus*' of *relations*" that provide the foundational musical and notational content for a "*set of potentialities*" that may or may not occur in any given instantiation of the score. [2] Within a generative animated score, the *nucleus of relations* are left *open* in order to enable the continuation of these processes for the duration of the work, and the processes by which these potentialities are selected, represented as notation, and subsequently realized by the performer, occur in close temporal proximity, and generally disappear shortly thereafter. [5]

1.2 Openness

The concept of *openness* in musical works ranges from the interpretive expectations of traditionally-notated works, to the modular and malleable scores that began to emerge in the mid-20th century. Umberto Eco describes these *open works* as works designed in such a way that "considerable autonomy [is] left to the individual performer in the way he chooses to play the work." [3] The performer is not restricted to the traditional, and often limited, mode of interpretation, but has agency to impact

[5] While the following text may be applicable to animated scores that are fixed prior to their activation, this paper focuses primarily on qualities that are specific to *generative* animated scores.

the realization of a work on many levels, from its atomic characteristics to its broad formal structure. In some open works, the *field of possibilities* is presented to the performer in such a way that its inherent openness is constrained to the degree that each realization of the score represents an identifiable concept, preventing its dissolution into "an amorphous invitation to indiscriminate participation." [3] These works are further regulated by their music-historical context and any relevant "notational conventions." [4] "In order to interpret correctly the instructions in the score, the performer needs to know the notational conventions used in it and the performance practices that are assumed without being explicitly indicated. The naive performer who considers only the score and who takes it 'literally' would misunderstand its instructions." [4] But even with these regulative influences, the field of possibilities may be represented by a notational form that temporarily, and perhaps intentionally, defies contextualization, and in doing so, may only provide enough information to elicit a suggestive, largely unregulated (at least regarding notational and contextual) interpretation. Without prior knowledge of a work's potentialities, even notational forms that strictly regulate the performer's interpretive range may not provide the *listener* with any foundation upon which to gauge the effectiveness or intrigue of the work's openness; without a common or shared foundation, "there are no privileged points of view, and all available perspectives are equally valid and rich in potential." [3] In other words, an open work can effectively write-out the wrong by not describing the right. The *right*, in this case, is made up of the musical *codes* that are perceptible to the listener [3], and whether or not these codes are similarly understood by the composer and the performer, so long as some code is perceptible, divergences from and within the code (drawn from the field of possibilities) will create a perceptible formal construct by its very difference. Eco describes the foundation upon which these differences can be identified as the *Ur-code*, which includes the harmonic, melodic, and rhythmic aspects of the Western musical tradition, and may also include the specific sonic characteristics of a composition and the musical context within which it exists. [4] Still, without an understanding of the music-historical context from which a particular work emerges, and an ability to perceive the code that it is based on, or diverges from, the openness of the work is aurally insignificant. In short, the work's openness may only be intriguing at the performer level: readily available to the performer's eyes, but meaningless to the listener's ears.

The generative animated score is open prior to its realization, but unlike traditional notions of openness, the score's openness is contained by, and often restricted to, the computational processes of the score. The performer still engages with the notations selected from the *field of possibilities*, but has little to no impact on the selection process itself. The notational representation of these selections is often specifically prescribed in real-time, further limiting performer intervention and diminishing the possibility for preparation. [5] Following this, the openness of generative animated scores is equally insignificant to both the performer *and* the audience. One need not know what these potentials are, nor determine which of these potentials are selected, because neither the performer nor audience member can influence what potentialities are actualized.[6] Following Davies, the performer is *naïve* in this regard, as the potential for any interpretive intervention is made unavailable by the processes of the score application, and the prescriptive specificity of the AMN.

With a fixed score, its tangible rigidity necessitates some action beyond it in order to determine which possibilities, and their respective qualities, are selected in performance. These actions are generally the performer's manual responsibility. The generative score displaces this selection/interpretation process by embedding it within the functionality of the score application. So while the score is open prior to its realization, because this realization occurs simultaneously with the emergent notational representation, the score simply indicates to the performer what to do, and when to do it, with little to no room for interpretive extrapolation. The score's openness is effectively inaccessible.

1.3 Time

From the low-level relationships formed between adjacent sonic minutiae, to the high-level, formal compartmentalization of the composition, *time* is the container within which the aforementioned musical codes, and the representational results of the selection process are held. But although time contains these codes, the tendencies of these codes control the flow, shape and *size* of these containers. Jonathan D. Kramer notes that within the coded tonal system, for example, time is linear, "always in motion toward tonic resolution," [6] subservient to the melodic and harmonic tendencies of the Western scale. When the tonic is destabilized, time finds release, and becomes increasingly multivalent: "nondirected linear music moves by a variety of means and with varying degrees of localized stability at cadences, yet it avoids the implication that certain pitches can become totally stable." [6] This is a nondirectional, but certainly not direction*less* kind of time, nor is it lacking in a stable foundation or code. [3,6] Rather, the singular, magnetic qualities of the tonic are dispersed across a series of candidates that may or may not be related on a functional melodic or harmonic level. When these points of interest appear to serve some functional purpose, but their feeling of displacement within the larger structure is perceptible, the piece exists in *multiple time*. [6] As in linear music and nondirected linear music, one's perception of multiple time is dependent on one's understanding of the underlying musical code, in order to "comprehend the function of a musical gesture even when it occurs in the 'wrong' part of a composition." [6] So long as this code is known,

[6] Contrarily, the interactive works of Jason Freeman, Gerhard E. Winkler, Harris Wulfson and others successfully integrate audience and performer activities with the actualization of the score.

some semblance of structural clarity may still be perceptible, despite the composer's intentions to do otherwise. In all three cases, one's perception of the passage of musical time is contingent on the hierarchical relationships between a series of events in a specifically-coded Western context (or any musical context for that matter). Musical time, in these contexts, is not clock time, but is based on the relationships between events that are perceived as musically structural or "important." In these three types of musical time, the performer is, ostensibly, responsible for the perceptible, sonic expression of the composed directionality, non-directionality, and multiplicity. Contrarily, "'Moment time,' after Stockhausen's formulation of moment form," describes a music in which the ordering and qualities of the musical content are not causal. [6] Each *moment* does not have a functional relationship to those adjacent to it, nor to the composition as a whole: "a work in moment time does not really begin; rather, it simply starts." [6] Even with the dissolution of beginnings and endings, internal form is still perceptible, but rather than perceiving compositional form based on hierarchical tensions, "the self-containment of moments allows the listener to process them as individual entities," each a formal contributor despite a lack of relational function. [6] In a sense, each moment becomes a temporary structural tonic, defined by a local logic that extends its influence only to the beginning of the next autonomous moment. The performer still maintains some interpretive responsibility: to express the structural-autonomy of each moment. Lastly, Kramer describes *vertical time* as "a single present stretched out into an enormous duration, a potentially infinite 'now' that nonetheless feels like an instant." [6] A composition in vertical time contains any number of sonic events, but unlike the aforementioned examples, including moment time, each of these events is an anti-landmark, equal in (non)importance as the others. The even distribution of musical importance across all events effectively flattens the significance of each event, disrupting the potential for emergent hierarchies and structures. [6] More so than Kramer's other distinctions of musical time, several aspects of the generative animated score reflect the concept of vertical time. Perhaps most obvious is the capacity for the notational generation of indefinite duration. As described above, the computational processes that access and select from the field of possibilities can be designed to function autonomously from human intervention. The notational flow will continue as long as the score application is running, and because the notational flow runs uninterrupted, any performance, which is likely shorter than the potentials of an endless score, has a quality of *nowness*, as the ephemerality of the score provides no past or future temporal or structural boundaries. In addition, the visual representation of these notations demonstrate a *visual verticality*. In many animated scores, points of attack are often contained to a small, immovable visual space (scrolling score), or a dynamic object which the eye follows (swiping playhead, tablature). [1,7] The eye is *moving*, in a sense, but fixed in a correspondent relationship with the behaviors of the dynamic symbol. Each instant is relegated to the necessarily controlling visual representation of the animated score, extending each sonic moment by the symbol's dynamic movement toward the next event. In order to maintain an adequate correspondence with the score, the performer is more or less *forced* into a continuous engagement with the functionality of these notations. Thus, each sonic event is visually extended by the notation as it leads the performer through a "single present" of constant movement. [6]

1.4 Sound

In *Sonic Philosophy*, philosopher Christoph Cox describes sounds as "peculiarly temporal and durational, tied to the qualities they exhibit over time. If sounds are particular or individuals, then they are so not as static *objects* but as temporal *events.*" [8] The temporal characteristics of sound influence not only the musical or sonic identity of the sounded, temporal event, but reflexively influence the qualities of the temporal container within which these events are framed; the quality of musical time, for instance. [6] The particular qualities of the sounded events contained within a work in vertical time, for example, will be designed to exploit the particular qualities of this container, not the other way around. In other words, extended duration and a quality of *newness* is only enabled by the sonic container. But even though these sounds are contained by the temporal framework that support its extended durational qualities, these sounds are still not *objects* distinct from the durational flow of its container. Rather, "Sound […] affirms an ontology of flux [in] which objects are merely temporary concretions of fluid processes. This flux ontology replaces objects with events." [8] The concept of *openness* seems to mirror this ontology of flux. The possibilities inherent in an open work, for example, do not contribute to the compositional identity of the work unless they emerge during the process of its realization; there is only the potential for their momentary *concretization,* and their absence does not disrupt the identity of the work. In *From Music to Sound: Being as Time in the Sonic Arts*, Cox suggests that the "shift from 'music' to 'sound' marks an ontological shift from being to becoming, and a temporal shift from time (le temps) to duration (la durée)." [9] Framed by Bergson's distinction between quantified time and "time as a qualitative process," and Nietzsche's rejection of *being* in favor of "ceaseless becoming and change," Cox cites Cage's *0'00"* as an example of emergent behaviors that, in their *becoming*, occupy a space unadorned by "musical" expectation, or as the realization of scored musical "objects." [9] The events in *0'00"* exist despite their framing by the score, and the score simply *repositions* their soundings as a *scored* event. For music, in Cox's distinction, "constitutes a domain of beings, time-objects that spatialize sound and that mark a pulsed time," and sound as "not being in time but being as time." [9] In a sense, the fixed, *closed* score relegates music to a sonic reflection of an immovable object, a relationship that is maintained despite the ephemerality of sound. Increasing the degree to which a score is left *open* loosens the structural and temporal holds on what sonic events might occur. But still, the *tangibility* of the score enables reference, repeatability, even reverence, despite the composer's intentions to (in theory) subvert these kinds of interactions. The open qualities of the generative animated score, in tandem with the computational processes that select and represent the notational information autonomous of human interaction, demonstrate a scoring process that is much more akin to a *temporal event* than a *static object*. [8] These processes, like the realization that follows, does not *exist* in the same tangible sense as any traditional score does, open, closed or otherwise: "*Before* and *after the moment of performance the piece*, - in the historical sense -, *does not "exist"*, there is nothing […] where you can *refer to*." [2] The performer is still reading notation, but is doing so *as* it is generated. There is no fixed object, but instead a momentary reflection of the underlying, generative processes *as* notation. This uniquely temporary manifestation of these processes as notation demonstrates a clear ontological distinction between the tangible *being* of the fixed score, and the open and ephemeral notational *becoming* of the real-time, generative animated score.

2. STUDY NO. 50

2.1 Introduction

Study no. 50 was composed in December, 2015 for the Williams College Percussion Ensemble, under the direction of Matthew Gold. *Study no. 50* was developed in tandem with the exploration of the aforementioned concepts, and its purpose here is as a practical demonstration of how these concepts informed the compositional and notational process.

2.2 Compositional Intentions

Prior to composing *Study no. 50*, many of my works were designed in such a way that the real-time notational processes demonstrated some perceptible musical code, including phase processes, discernable poly-temporal relationships, and hocket. While the musical results were personally satisfying, the impact of the process as a perceptible compositional factor had begun to elicit an unwelcome sense of novelty. With *Study no. 50* I endeavored to build a framework in which the generative processes that create and control the notation were functionally autonomous across all levels. In short, I wanted to reduce the possibility of emergent perceptible structures by creating a set of potential actions that were unlikely to create any local or global structural form regardless of their ordering and/or combination, and to limit the performer's interpretive agency regarding what form these potentials might ultimately take. Following this, the score for *Study no. 50* is designed to create a consistently-inconsistent flow of events for an extended (indefinite) duration, and to evenly, although randomly, distribute these events throughout the piece. Furthermore, the selection of these events from an open field of possibilities is not governed by any high-level structures, or performer influence. In this sense, the compositional intention and notational representation in *Study no. 50* explicitly echoes Kramer's elucidation of *vertical time*: "The motion is so consistent that we lose any point of reference, any contact with faster or slower motion that might keep us aware of the directionality of the music. The experience is static despite the constant motion in the music." [6]

2.3 Instrumentation

"Respecting self-imposed boundaries is essential because any move outside these limits would be perceived as a temporal articulation of considerable structural import and would therefore destroy the verticality of time." [6]
In order to avoid the emergence of any perceptible timbral, rhythmic, or pitch-based *structural articulation,* the potential for instrumental variation is limited. The instrumentation for *Study no. 50* included 42 pieces of wood [planks], 7 per player, each only slightly larger or smaller than those adjacent to it. The similarities between each plank effectively limited their perceptible distinction. Each player was permitted two sets of mallets, hard and medium, and were instructed to switch mallets as often as they pleased, so long as these changes were irregular (i.e. to avoid a structural pulsation), and that mallet usage should be evenly distributed over the course of the performance. Furthermore, each player was instructed to vary their dynamics between *MP* to *F* over the course of the performance, and similarly, to distribute this range evenly over the course of the performance. These instructions produced a narrow timbral and dynamic range with only minor perceptible changes.

2.4 Notation

Each performer's aggregate contains seven nodes and one attack cursor (see figure 1).

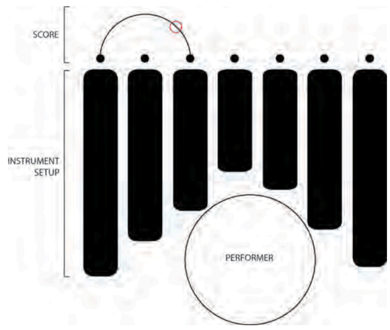


Figure 1. *Study no. 50* Aggregate and Performer diagram.

In figure 1, each node is represented by a small black circle, which corresponds to a single plank, represented by the black rectangles of varying lengths. Because the score is projected downward, each plank can be lined up with the corresponding node (see figure 2).

Figure 2. Performance detail.

There are four possible functionalities for the attack cursor that determine which planks are to be played, and *when* they are to be played. The node that has most recently been engaged by the attack cursor is the *current node*, and the node that the attack cursor is moving toward is the *target node.* The primary notational functionality simply represents which plank to play, and when to play it, indicated by the arrival of the attack cursor at the corresponding node (see figure 3).

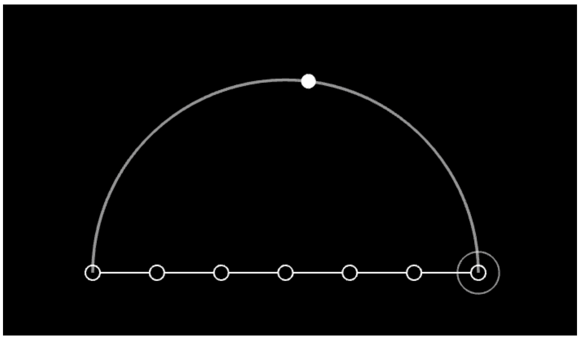


Figure 3. Function 1: Current Node [far right] to Target Node [far left].

In figure 3, the attack cursor is en route from the rightmost node to the leftmost node. The performer will strike the plank that corresponds to the leftmost node at the moment the attack cursor makes contact with that node.

The second functionality occurs when the target node is the same as the current node. Because the attack cursor is already at the target node, a notation called the *repeat spinner* is utilized (see figure 4).

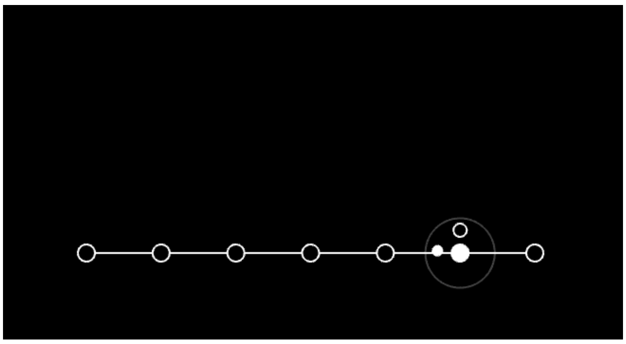


Figure 4. Function 2: Repeat Spinner.

At the completion of the event that precedes a repeat spinner, a small *attack point* appears above the current node, followed by a similarly-sized attack cursor rotating in clockwise motion around the node. The point of attack is when the rotating attack cursor makes contact with the attack point at 12 o'clock.

The third functionality is represented by a single arc, similar to the first functionality, but with a number displayed at the top of the arc (see figure 5).

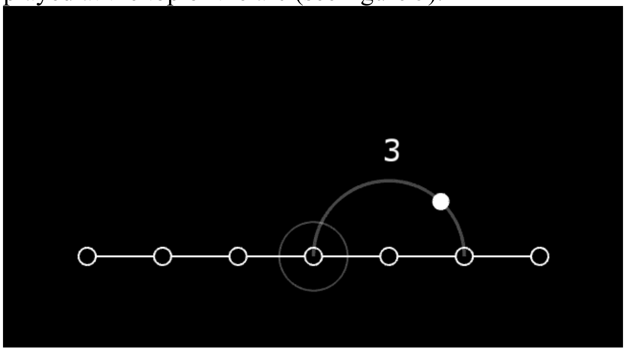


Figure 5. Function 3: Open Repeats.

This indicates that the player should repeat the current node, or target node's corresponding plank *that* number of times before the attack cursor reaches the target node. These attacks should occur within the duration it takes for the attack cursor to move from the current node to the target node, and the target node's corresponding plank should *not* be played upon the arrival of the attack cursor.

The fourth functionality is the *flourish*, in which a series of arcs extend from the current node to the target node, and every node in between (see figure 6).

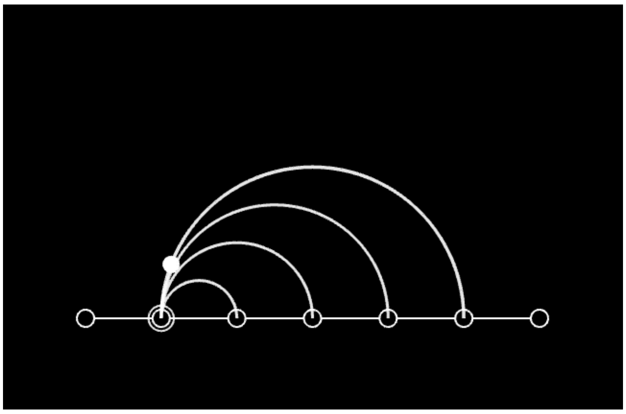


Figure 6. Function 4: Flourish.

This notation indicates that the performer play each plank corresponding to the nodes leading to the target node. These gestures can be played at any speed, but should be rhythmically consistent between attacks, and should end as the attack cursor reaches the target node.

2.5 Notational Processes

The score for *Study no. 50* is generated in real-time from an application written with openFrameworks, and will continue to run indefinitely once executed. While the symbolic elements of each performer's aggregate are identical, and contain the same functional potential, the processes of each aggregate are autonomous from the others, and are not governed by any high-level structure. The processes that determine the behavior's of each performer's aggregate are based on a simple set of *if-then* statements, and the particulars of these functionalities are randomly determined within a narrow range of possibilities. The functionality of the attack cursor is determined at the completion of each event (ie. the moment the attack cursor makes contact with its target node). In this sense, the potential functionality of the attack cursor is *open* until the next moment it is selected. This selection process proceeds as follows:

1) Determine the target node. The target node is determined randomly, and is equally weighted across all nodes, including the current node.
2) If the target node is the same as the current node, skip to step 6.
3) If the target node is immediately adjacent to the current node, choose between functions 1 and 3. Function choice is determined randomly, and is equally weighted between functions 1 and 3. If function 1 is selected, skip to step 6. If function 3 is selected, skip to step 5.
4) If the new target node is not the current node, and the distance between the current and target node is greater than one, choose between functions 1, 3 and 4. This functionality is determined randomly, and is equally weighted between these three functions. If function 1 or 4 is selected, skip to step 6.
5) Select a number between 1 and 4.
6) Determine traversal duration.
7) Draw arc(s) or repeat spinner and activate the attack cursor.

Step 6, "determine traversal duration" is randomly determined within a range of 500 to 1600 milliseconds.

2.6 Presentation

The score for *Study no. 50* was designed to be projected onto the floor, with each node positioned directly above its corresponding plank (see figures 2 & 7). This alignment creates a direct correspondence between the notation and the instrument, facilitating legible clarity.

2.7 Discussion

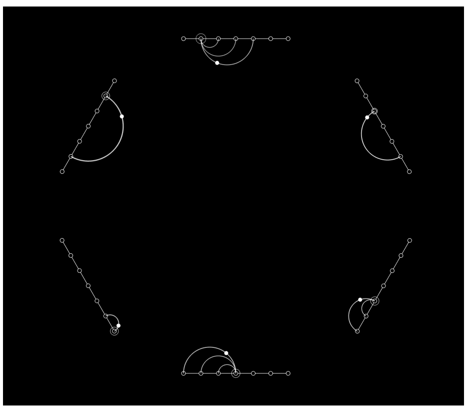


Figure 7. Study no. 50 [score detail].

As mentioned earlier, the processes that control the behaviors of the attack cursor, including the four possible functionalities, and the duration of these functionalities, represent the field of available possibilities. The random processes that select these possibilities are largely unweighted, and are determined one at a time at the completion of each event. To this end, the score for *Study no. 50* is effectively *open*, and based on the autonomous, random functionality of the selection process, and modest combinatorial possibilities, will likely generate a unique, consistently-inconsistent flow of events each time the score is activated.

This functional openness was an essential component toward the creation of a persistent, durational performance. Because a new notational function is generated at the completion of each event, and the animated music notation was designed to be sightreadable with a high degree of accuracy, a performance of *Study no. 50* can last for any duration without *running out* of notational material, while preserving the compositional identity of the work. Furthermore, the limited range of event durations, the even distribution of dynamic and timbral changes, and the general similarities between the 42 planks, creates a sustained gestalt that is devoid of any perceptible musical landmarks or structural intentions. Recalling Kramer, vertical time can be described as "a single present stretched out into an enormous duration, a potentially infinite 'now' that nonetheless feels like an instant." [6] Each *present* corresponds to the execution of each notational function, and the inconsistent, but temporally regulated concatenation of these events drastically reduces the possibility for the emergence of pulsed or structurally significant time, while maintaining a high degree of rhythmic activity, reducing the potential for structural silences.

One of the primary motivating factors for *Study no. 50* was to create a process-based work in which the process itself was perceptible only in its representation as notation. I did not want the audience to *hear* the process as it unfolds. Rather, to distinguish between the sonic realization of the score, and the visual representation of these processes as notation in their real-time becoming.

3. DISCUSSION

3.1 Notational Becoming: Speculations

The speculative concept of a *notational becoming* suggests an ontological distinction between open or closed, fixed scores, and generative animated scores. This distinction is primarily based on the location of the score's openness (including a displacement of performer agency while maintaining prescriptive notational specificity), the timeliness of these processes as temporary *concretions* of legible notation, and the unique temporality of the processes that form the compositional, notational and functional foundation of the score.

The real-time generative processes that demonstrate this notational becoming also suggest a method for the real-time production of an infinite flow of prescriptive, through-composed notation. The score becomes not the execution of an extended-duration process regulated by performers, but the realization of notations indefinitely produced through the real time processes of the score itself. The real-time becoming process may enable distinctly durational compositional identities that can be well-maintained by the prescriptive specificity of the notation. Winkler notes that “A mixture of ‘installation’ (where one can enter, move around and go out at will) and ‘concert-situation’ (with fixed start and endtime, focused sitting and listening) seems to be the best environment for the presentation of this type of music.” [2] But there is no reason to engage with start and stop times: these notations have a continued *presence* regardless of interaction, like that of Kramer’s sculpture [6], and unlike the traditional score, when projected, the animated score maintains a unique visual presence. For although the notational content of these scores is ephemeral, the persistent notational flow maintains a notational image of the sonic qualities it represents.

Clearly, practical limitations (human biological function, live performance expectations, economic considerations) impact *actual* duration, [6] but this potential for extended duration introduces a unique compositional question: if performance duration can only be determined by practical considerations, is there a minimum durational threshold that a performance must pass in order to fully represent the compositional idea? Can the *infinite* nature of vertical time be represented in *realistic* time? Kramer notes that “Once we have entered the vertical time of the composition, we have apprehended its limits. The piece has defined for us its context; it will not step outside its boundaries.” [6] But no matter how well-defined, well controlled, and shielded from interpretive disruption the generative animated score might be, by enabling and embracing an endless durational flow as a compositional characteristic, the work, like the sculpture, is durational only to the degree that the listener decides to engage. In this sense, notational becoming represents a “ceaseless becoming and change,” [9] that is only contingent on its autonomous processes, is timely, and demonstrates a (non)structural ephemerality of notational and sonic flux.

3.2 Conclusions

This paper has speculated on how the open, timely, and ephemeral aspects of generative animated scores demonstrate qualities that are ontologically distinct from musical scores that are fixed prior to their performance-ready representation. I have described these qualities as a *notational becoming*, an extrapolation, if not bastardization, of Christoph Cox’s demonstration of the ontological difference between music and sound. These speculations are only temporarily concretized, and are subject to immediate revision.

Acknowledgments
Thanks to Matthew Gold and Daniel Steffey for their valuable input.

4. REFERENCES

[1] Cat Hope and Lindsay Vickery, “Screen Scores: New Media Music Manuscripts,” in *Proceedings of the International Computer Music Conference 2011*, University of Huddersfield, UK, 2011.

[2] Gerhard E. Winkler, “The Realtime-Score: A Missing Link in Computer-Music Performance,” in *Sound and Music Computing*, Paris, 2004.

[3] Umberto Eco, *The Open Work*, Harvard University Press, 1989.

[4] Stephen Davies, *Musical Works and Performances: A Philosophical Exploration*, Oxford University Press, 2001.

[5] Jason Freeman, “Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance,” *Computer Music Journal*, vol. 32, no. 3, 2008.

[6] Jonathan D. Kramer, “New Temporalities in Music,” *Critical Inquiry*, vol. 7, no. 3, pp. 539-556, 1981.

[7] Lindsay Vickery, “The Limitations of Representing Sound and Notation on Screen,” *Organised Sound*, vol. 19, no. 3, 2014.

[8] Christoph Cox, “Sonic Philosophy,” *Artpulse Magazine,* vol. 4, no. 16, 2013.

[9] Christoph Cox, “From Music to Sound: Being as Time in the Sonic Arts,” *Sonambiente Berlin 2006: Klang Kunst Sound Art*, ed. Helga de la Motte-Haber, Matthias Osterwold, Georg Weckwerth, pp. 214-23, 2006.

AUTONOMY, CONTROL, AND NOTATION IN INTERACTIVE MUSIC

K. Michael Fox
Rensselaer Polytechnic Institute
foxk@rpi.edu

ABSTRACT

This paper proposes a conceptualization of notation for interactive musical environments. The notational approach describes the relationship between both human and non-human agents, instead of actions to be taken or sounds to be made. Of critical importance in contemporary networked culture is the degree to which technological devices and networks constrain (or control) the actions of their users. The author has developed a conception of interactivity and notational considerations which instead foreground the autonomous potential of participants and the computational systems. The author analyzes three case studies that demonstrate either a direct connection or a broader conceptual link to the described notational approach. The larger implication is a need for notational systems which do not constrain the identity of the users of interactive systems while also acknowledging and representing the agency of the systems themselves.

1. INTRODUCTION

In Western art music, classical composers have used notation to express the intent of their music to the performer, who then communicates this intent to the audience. The performer and composer have been trained to speak a common language that forms the heart of the score. This system prevailed over the centuries, until about the 1950's, when composers began to seek new relationships between composer and audience as mediated through notation. One such new relationship expands the scope of "performer" to include audiences – who may typically lack the training to interpret standard notation fluently. Enrolling audiences as performers, or even co-composers, is among the potential challenges facing the composer of interactive musical systems. While some forms of interactive musical systems may model themselves upon the traditions of instruments, requiring some degree of mastery, and use notation in ways familiar to common practice, others may explicitly call for performers without needing to rely on the assumption of specialized knowledge of the common practice.

This article presents one account of a particular framing of interactivity and the role notation plays within it to afford audience participation. This conception refrains from positioning the computer directly as either an instrument or tool to be controlled, or as a proxy for a human performer. Instead, it incorporates the computer along with humans in the work as a part of a network that privileges effects from localized contextual relationships between actors. [1] I will then make a case for why and how notation can still function at the heart of these systems with potentially non-musically literate audiences as participants. Three case studies serve to test the validity of the theory, chosen for similarities in some of their notational concerns: Thor Magnusson's code score, the *Threnoscope*; Arthur Clay's interactive music system, *Book of Stamps*; and my own musical installation for networked mobile phones, *Parallel*. The following issues or topics will be examined in each case: the role of its symbology as passive or active; the target of the notation and assumed skill; the model for interaction in the system; the role or representation of the observer; the concepts that the notation express; and the identity of the interpreter. Using these case studies, I will argue for a unique conception of the role of notation, that represents and characterizes the relationship between users, observers, and computer agents in interactive works.

2. CONCEPTS AND SCOPE

2.1 Interactivity

Interactivity in my own composition is constrained to the domain of distributed, networked systems that foreground the computer as a compositional collaborator on the same ontological level as the human. A full account of the motivations for this constraint, rooted in ideas about Actor-Networks and the ubiquity of computers as mediating devices, is beyond the scope of this short paper [2]. While computers are human creations, their processes, upon which we increasingly rely, are ever more black boxed, fragmented, modular, distributed, and networked. Whatever computers may "be", I claim, we can't truly grasp their essence (if such a thing exists) or observe the fine details of their processes, but we can recognize our relation to them and the resulting effects at the time and in the context of their use. My creative goal as a composer is to approach the computer as an unknowable collaborator, so I strive to design interactive systems which neither seek to mimic the activities of humans directly, nor do they exist as instruments

[1] "Network" is used in this context not in the technological sense but rather the Actor-Network sense put forward principally by Latour, Callon, and Law. These networks are characterized by a few key properties: networks are comprised only of the actors or actants they contain and the relationships between them; these networks can be infinitely black-boxed or unpacked, or, zooming in or out does not reach an "end"; and there are no invisible components between networks – to connect disparate networks is but to zoom in or out to reveal the empirical connections. For a more exhaustive clarification, see: [1].

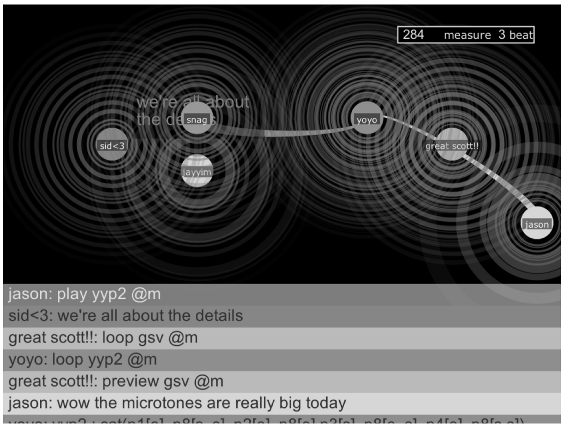


Figure 1. LOLC audience view.

to be directly controlled by the agency and intention of the human performer. To create systems that directly mimic or elevate the human perspective is to unduly limit the scope of their efficacy within the work. As George Lewis has demonstrated, these mimetic designs also make implicit assumptions about the user which can severely constrain their cultural identity as well [3].

2.2 Autonomy and Control

According to Felix Stalder, the Enlightenment's *reason* brought about a division of human existence into the inner and outer worlds – the outer world of appearance and social behavior, and the inner world of contemplation and formation of *personal* opinions based on logic. This new divide enabled individuality that creates a sense of autonomy, which Stalder defines generally as "the ability for people to lead their own lives according to their own plans" [4]. By extension, this would correspond to agents in interactive musical settings acting in ways that match or reflect their own contextual relationship to that musical setting. Stalder's subsequent contention is that, if the Enlightenment's *reason* did in fact create this divide, then the digital, networked age collapses it. In the "network society", as Stalder refers to it, "in order to create sociability in the space of flows people first have to make themselves visible, that is, they have to create their representation through expressive acts of communication" [4]. What could be more social and expressive an act than interactive music-making? More importantly, in the case of a networked and interactive music setting, what would it mean to have autonomy?

George Lewis illustrates the interactive autonomy of human and machine in performances of his improvisation machine listening system, *Voyager*. The key aspect of the *Voyager* system is that it is explicitly created with the cultural and musical identity of an African-American improviser. Further, this identity is retained throughout performance while improvising with humans [5]. The same is true of the personal identity of any human improvising with *Voyager*. The music resulting from this improvisational interaction is the conversation between the autonomous agents present. Faced with the prospect of including the general audience, ostensibly comprised of untrained musicians, this autonomy would require configurations of the system which allow conversations to occur which did not require a priori knowledge of certain musical traditions. In some sense, the consequence of leaving these pathways closed, and creating only interactive potentials of an excessively narrow definition illustrates the darker counterpoint to autonomy, which is to say *control*. "The intellectual and musical problem endemic to structure-generating activities such as improvisation (or any other musically generative or creative activity) is that we are not always aware of the constraints that we are functioning under as we work, or why we decide upon certain actions" [5]. This is certainly true of the Anglo-centric Western musical practices that Lewis was critiquing in *Voyager*, and he notes that it is equally true of computational systems. Herein lies the abusive potential of interactive systems: Interactive systems – even those for music – that do not enable some degrees of collaborative autonomy are implicitly configured to control their users. The subtext of this control is that certain interactions are valid, while others are not. In the extreme case, this implicit control can lead to a reinforcement, if not exacerbation, of the conditions which lead to George Lewis's critique in *Voyager*.

2.3 Need for Notation

Beyond the traditional role of notation as a communication method between the composer and the performer, there have been a number of attempts to apply notational schema to explicitly help audience members understand the otherwise hidden processes of laptop or computer-based performance. Jason Freeman's LOLC and SGLC network music performance frameworks contain network visualization and chat feed components which are projected to aid in the audience's understanding of otherwise invisible interaction between laptop performers [6]. Likewise, Alex McLean and colleagues have documented several experiments in visualizing the code processes of live coding performances [7]. Thor Magnusson questions the ability for visualization or secondary notations to effectively capture the functionality of algorithms in meaningful ways for audiences. Instead, he posits that code itself is the best representation for the actual algorithms or blocks of code, and any visual or notational element describing them should represent their presence in the context of other algorithms or processes performed on them, as in his *Threnoscope*[8], described in section 3.

In each of these cases, the notational nature of the visualization is related closely to the way Bruce Haynes has differentiated between the descriptive score and the prescriptive score. For Haynes, the descriptive score communicates the idea of a piece to a performer who provides an interpretive realization, by contrast, the prescriptive score provides detailed instructions that, if rendered correctly, will reveal the composition at the time of performance [9]. Examples are not hard to find that seem to occupy both sides of this divide simultaneously. In the "extensible open" works of David Kim-Boyle, the score itself is realized at the time of performance and is thus inaccessible for complete a priori comprehension. At the same time, in works such as *tunings* (2006), *music for 2* (2010), and

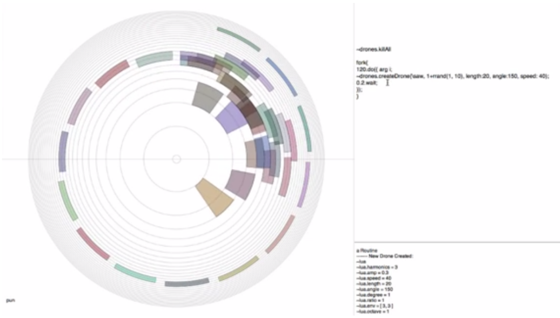


Figure 2. The *Threnoscope* screen interface.

music for 4 (2011), the scores are also designed in such a way as to elicit more interpretive responses from the graphical scores, lacking the level of formal and detailed instruction to perhaps totally qualify as a prescriptive score [10]. This point seems to be mirrored by Kim-Boyle's own description of his "compositional strategy, whereby it becomes more useful to think of the nature of composition as being that of the design of graphical environments explored through sound" [10]. Further, these complications still arise when scores are presented for the trained performer.

In the case of scores presented for the audience's comprehension, how does the inclusion of the presumably untrained public as receivers of the score or even potential performers further complicate the distinction or function between the prescriptive and descriptive? Or, alternatively, what is the function of the score for these audience members? The answer seems to be in many cases that they are descriptive in that they transduce and represent actions (perhaps in the form of code executed) taken at the moment they occur by performers in the work. This is the means by which the abstract computer performance, which may not otherwise resemble usual forms of musical communication, can be made a little more discernible to the viewer or participant.

The extension that I would provide, especially in the case where the audience becomes the performer, is the possibility for these representations to also represent the relationships between performers at any given moment in the work. These representations need not require specific prescriptive responses, but rather can suggest a context that may elicit a personal response from the individual user observing it. These relationships may also be created by the system and cast upon the audience/performer, proposing a momentary transformative potential within the work. From this we can imagine a notational approach that lies somewhere between the descriptive and prescriptive score – but the approach seems to drift closer to the former. The following cases analyze how three specific works demonstrate notational similarities within these considerations.

3. CASE STUDY 1 - THE *THRENOSCOPE*

Magnusson's *Threnoscope* (2013) is conceived simultaneously as a notational framework and a "composed system" for live coding performance [8]. [2] As a part of the larger live coding practice, it functions as a constrained system which represents the code that is functioning over a period of time. Magnusson distinguishes between representing the presence of code and, as others have tried, representing the functionality of that code [8]. As he notes, "notation is a way of communicating abstract ideas to an interpreter, and in live coding that interpreter is typically a compiler called the 'language interpreter'" [12]. So the code written by the performer, which represents processes resulting in sound or changes to sound, is interpreted by the computer and rendered into sounds and visualization. In this way, the code becomes prescriptive notation to the *Threnoscope* software. At the same time, the graphical representations of the code are displayed to the performer, any co-performers, and the audience. To these observers, the notation becomes a reference for representations of actions taken by the performer or the autonomous "machines" in the system, and the relation of each bit of running code to the others. [3] The system itself and the performer have the skill to respond to these representations or code, but the representational notation for the audience is an abstraction to assist in comprehension.

Though the notation does not explicitly "place" the observer or the performer within the score, the score creates a number of cues that can orient the observer. To some extent, observing the score's alteration by the performer, at the time of performance, allows the audience to cognitively grasp that affordance. Though, it is also clear to the audience that they do not have the same capacity to change the score that is afforded to the performer. This creates a kind of distance between the audience and the work. Moreover, with regard to the spatialization of sound, the score's radial design does implicitly place both the audience and the performer at the center of the circle. As the geometric representation of code moves around the plane, its sonically spatialized position in the multichannel speaker field correspondingly shifts, mapping the virtual space of the score onto the real space of the performance location.

4. CASE STUDY 2 - BOOK OF STAMPS

Book of Stamps (2009) is an interactive installation for what Arthur Clay describes as a "new audience" – the audience which has been empowered to effect meaningful change in the work, due to the composer's efforts to "create a fluid transformation from basic passivity to intense participation" [13]. Like the *Threnoscope*, sounds are prescribed to be made by the computer, a reactive and composed system, by placing a stamp of symbolic meaning on a page within view of a computer vision camera. The computer is given some semantic understanding of the sounds associated with each stamp's symbol. However, unlike the *Threnoscope*'s code notation, there is no performer who

[2] Here, Magnusson is using "composed system" to describe a system with a performance interface which is rearranged "with specific musical intent... [such that the software system embodies some aspect of the maker's musical intent, and acts (like a score) as a vehicle for sharing musical ideas across culture." For more discussion of this and context of the quote, see: [11].

[3] Magnusson uses the term "machines" to describe software agents within the system that perform actions independent of the performer's control. An example might be the radial rotation of a "satellite" drone.

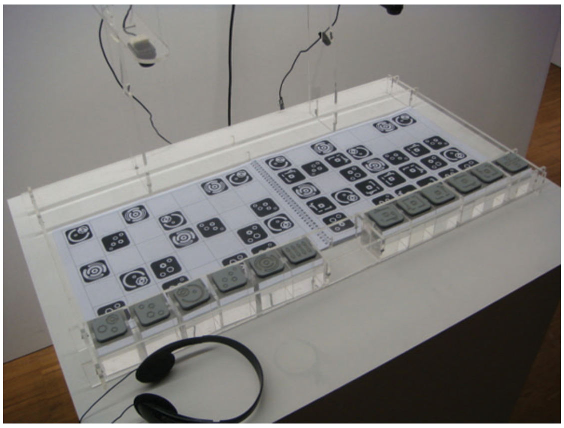


Figure 3. Book of Stamps.

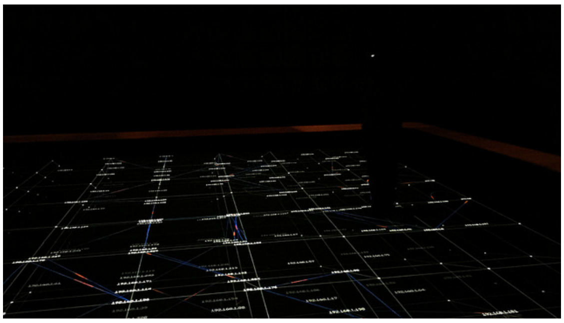


Figure 4. A view of the floor projections in Parallel - IP addresses representing audience members appear on the grid.

shares the computer's semantic understanding of the symbols in the *Book of Stamps*. The stamps are applied by Clay's New Audience, the casual on-looker who has decided to participate. So, also like the *Threnoscope*, the primary interpreter is the computer system, but the score also acts as an abstract, representational palimpsest upon which the user may add their own contribution.

The fact that the score both maintains a record of the previously performed actions and also invites audiences to invoke additional changes leads the audience to directly identify their place as part of a group effort. However, in the absence of seeing previous actors' contributions, it could be nearly impossible to understand any rationale, intention, or motivation behind particular contributions to the score, as in: "Who put that stamp there, and why?" The one strong exception to this is Clay's own design intention behind the symbology, which is based upon collections that suggest architectural or structural relationships and forms. Clay has deliberately designed symbols that seem to imply a particular relationship to each other, where "the visitor can 'construct' building layouts in endless variation" [13]. Thus, the score does not prescribe direct action from the onlooker, but rather offers the possibility of contributing and implies a relationship between the available means of input.

5. CASE STUDY 3 - PARALLEL

Parallel (2015) is an installation-based musical piece for distributed, networked mobile phones by this author and collaborator Raven Kwok. The notational components of this piece emerge from a series of weakly-tied visualizations. [4] As participants, not trained performers, enter the installation space, they are allowed to connect using a iOS device with the ANMPlatform app installed. At the moment the app is connected to the network from within the application itself, an audio response is heard from loudspeakers and an IP address associated with the user's phone is added to graphical projections appearing on the floor. Further, when more than one phone is connected to the installation, data flows from one phone to another in a topology assigned by the network itself. This data flow process is also visualized by bezier curves extending between IP address nodes and strobing in the direction of the flow of data. Meanwhile, the users are allowed to manipulate an abstract interface on their iOS device, the state of which – along with incoming data from other iOS devices – determines some synthesis parameters for sound which is emitted from their mobile device itself. The ultimate effect of this network topology and interaction is that no single element has direct or complete control over the system, and all influences are assimilated into a collective system state.

From this vantage, it is perhaps possible to see how the computer and user are forced to actively reinterpret the meaning or intentions of each other, as one of the few directly causal actions in the work is the moment when the user joins the network. But the users are also forced to actively interpret their relationship to the other users present by the mediated computer network and visualizations. Together, these connections drift closer to what has been described by Werner Rammert as "framed interactivity", or a relationship which seeks to create locally and contextually coherent interaction [14]. In addition to their embodied presence within the installation setting, observers are also presented with abstract representations within the system. This provides audience members a way of grasping their invisible virtual connections to the other participants in the context of the work. Though the mobile devices they use to make the connection may have an interface with some causal influence over some elements within the work, the direct effects of that interaction have been obfuscated by distributing them elsewhere within the system. The system can produce sound even in the absence of user input. Therefore, the notation serves to privilege the relationship between actors within the system and the effects of actions as a result of those relationships, as opposed to actions or sounds themselves.

6. CONCLUSIONS

This article has discussed a particular framing of notation within the context of distributed and networked interactive

[4] I use the term "weakly-tied" here to mean that, while many different visualized forms may be related in one way or another, none of them can be considered the "parent" or "primary form". Neither can these visualizations be directly linked with a principal causal outlet for the audience. Each form presents one angle or way of viewing a particular actor or process in a complex web of influence between many actors.

musical systems. The interactivity upon which that ontology relies is defined narrowly, though the author has attempted to illustrate that the notational consequences are in fact present in other contemporary works with varying degrees of similarity. Among the main concerns of this notational schema is the concern for how each actor, human or otherwise, regards their situation within the context of the work at any moment, or even how they are to be regarded by other actors. In some ways, it functions as a bridge between certain types of compositional practice which may have one foot in each of the virtual or real worlds.

The collision or oscillation of influences between the virtual and real worlds can occur within a broader, more inclusive audience-performer hybrid. It is not hard to imagine obvious extensions to augmented, fully immersive, virtual, or video game worlds that are oriented around or inclusive of musical composition. I leave these as open questions for later exploration. To conclude, it is possible that interactive systems and the participatory audience may situate the role of notation somewhere between the descriptive and prescriptive score, or possibly somewhere new entirely.

Acknowledgments

I would like to thank Michael Century for the detailed conversation on this topic. I would also like to acknowledge Jason Freeman, Thor Magnusson, and Arthur Clay for their support.

7. REFERENCES

[1] B. Latour, "Actor-Network Theory: A few clarifications and more than a few complications," *Soziale Welt*, vol. 47, no. 4, pp. 369–81, 1996.

[2] K. M. Fox, "Actor-Network Music: A paradigm for distributed, networked music composition," Master's thesis, Rensselaer Polytechnic Institute, 2015.

[3] G. E. Lewis, "Improvising Tomorrow's Bodies: The Politics of Transduction," *E-misférica*, vol. 4, no. November, 2007. [Online]. Available: www.hemi.nyu.edu/journal/4.2/eng/en42_pg_lewis.html

[4] F. Stalder, "Autonomy and Control in the Era of Post-Privacy," *Open. Cahier on Art and the Public Domain*, vol. 19, pp. 81–83, 2010.

[5] G. E. Lewis, "Interacting with latter-day musical automata," *Contemporary Music Review*, vol. 18, no. 3, pp. 99–112, 1999.

[6] S. W. Lee and J. Freeman, "Real-time music notation in mixed laptop-acoustic ensembles," *Computer Music Journal*, vol. 37, no. 4, pp. 24–36, 2013.

[7] A. McLean, D. Griffiths, N. Collins, and G. Wiggins, "Visualisation of Live Code," in *Proceedings of the Electronic Visualisation and the Arts Conference*, 2010.

[8] T. Magnusson, "Scoring with Code: Composing with algorithmic notation," *Organised Sound*, vol. 19, no. 3, pp. 268–75, 2014.

[9] B. Haynes, *End of Early Music: A Period Performer's History of Music for the Twenty-First Century*. Oxford: Oxford University Press, 2007.

[10] D. Kim-Boyle, "Real-time Score Generation for Extensible Open Forms," *Contemporary Music Review*, vol. 29, no. 1, pp. 3–15, 2010.

[11] O. Bown, A. Eldridge, and J. McCormack, "Understanding Interaction in Contemporary Digital Music: from instruments to behavioural objects," *Organised Sound*, vol. 14, no. 2, pp. 188–96, 2009.

[12] T. Magnusson, "Code Scores in Live Coding Practice," in *Proceedings of the International Conference for Technologies for Music Notation and Representation*, Paris, 2015, p. 5.

[13] A. Clay, "You Can Play It Too: The Virtuoso Audience," *Contemporary Music Review*, vol. 29, no. 1, pp. 63–73, 2010.

[14] W. Rammert, "Where the Action Is: Distributed agency between humans, machines, and programs," in *Paradoxes of Interactivity: Perspectives for Media Theory, Human-Computer-Interaction, and Artistic Investigations.*, U. Seifert, J. H. Kim, and A. Moore, Eds. Bielefeld, Germany: Transcript Verlag, 2008, pp. 62–91. [Online]. Available: https://www.ts.tu-berlin.de/fileadmin/fg226/TUTS/TUTS_WP_4_2008.pdf

MUSICAL INSTRUMENTS AS SCORES: A HYBRID APPROACH

Enrique Tomás
University of Art and Design Linz
Enrique.Tomas@ufg.at

ABSTRACT

The development of new approaches to instrumentality during the decade of 1960 contributed to the dual perception of instruments as scores. For many performers, the instrument became the score of what they played. This artistic hybridization carries substantial questions about the nature of our scores and about the relationships among instruments, performers and musical works. This paper contextualizes the historical origins of this instrumental development within Drucker's theory of performative materiality. Then we examine the nature and notational scheme of this type of scores making use of the concept of *inherent score*. Finally, through the analysis of two examples (*tangible scores* and *choreographic objects*) and the notions of *affordance* and *constraint*, a compositional framework for shaping the inherent instrument score is presented.

1. INTRODUCTION

1.1 Inherent Scores: Origins of a Form

The idea that a musical instrument can be considered a musical score too, or that *the instrument is the score* has come to the academic discussion partly in the field of interfaces for musical expression [1], music notation [2] and performance of electronic music [3]. Certainly, when a performer approaches a musical instrument a number of limitations or constraints will be revealed. These characteristics of the instrument are often considered a score in virtue of its property of shaping the musical work. The following section contextualizes historically the origins of some musical practices fully supporting this idea.

It is widely considered that the creative interpretation of musical instruments as scores has its roots in the Sixties. Composers like Gordon Mumma, David Tudor or David Behrman built electronic music instruments that, once configured, can afford enough performative potentials to reveal a musical work. Alvin Lucier [4] describes how within many of the works produced by the Sonic Arts Union *there were no scores to follow; the scores were inherent in the circuitry*. In David Behrman's *Runthrough* (1968) an undefined number of performers interact with the instrument by illuminating parts of a light sensitive audio mixer. Conceived as an improvisational piece, Behrman allowed ample time for the possibilities offered by his circuit to unfold and explore the acoustics of the actual room used. In *Runthrough* performers are not provided with instructions about the type of sound sources to use, their durations, sections of the piece, etc. The general rules for performing are delegated to player's musical exploration.

In the same year, David Tudor composed *Rainforest* (1968) (Figure 1). In this work, a set of sculptural speakers are suspended in the installation space and act as unique resonant loudspeakers with sound emanating directly from the sculptural objects (each having a unique sound source). In *Rainforest* the compositional idea is that if you send sound through materials, the resonant nodes of the materials are released. It is a kind of physical filter. Visitors are encouraged to wander around and physically interact with the work. Tudor's notation of the composition is, in a deliberate way, only a circuit diagram (Figure 1). Like in *Runthrough*, *Rainforest* can be played without further instructions about durations, sound sources or number of sculptural speakers.

It is during this historical decade when an intense research on alternatives to traditional musical notation was produced. A seminal example is *Notations* (1969), a printed compendium of musical notation edited by John Cage and Alison Knowles. It is remarkable that among 269 compositions, in *Notations* we only find three musical works making use of circuit schematics as notation: Gordom Mumma's *Mesa* (1966), Max Neuhaus's *Max-Feed* (1966) and Fredic Rjewski's *Piece with Projectors and Photocell Mixer* (1966). Certainly, if circuits are configured in a specific way for their artistic use, the role of the composer, at an equidistant point among designers, composers and performers, would start with the configuration of the technical system behind the actual instrument. Indeed, performing becomes the creative exploration in freedom of the musical affordances, musical reactions or acoustic relations to the physical space performed, without the need of any kind of dedicated musical composition.

When Lucier exposes that *"the score is inherent in the circuitry"* we are facing the origins of a new compositional practice, often known as *composing inside electronics*. And in this sense it constitutes a new way of understanding instrumentality. Performers would not need an external cause, a precondition to play the instrument like in the case of traditional scores. The musical work would not only be only defined by the instrument, but more importantly, by the act of playing the instrument. The performer's role would be to reveal instances of the musical work inherently integrated in the circuitry. This type of embedded-in-the-instrument scores we will call *inherent*

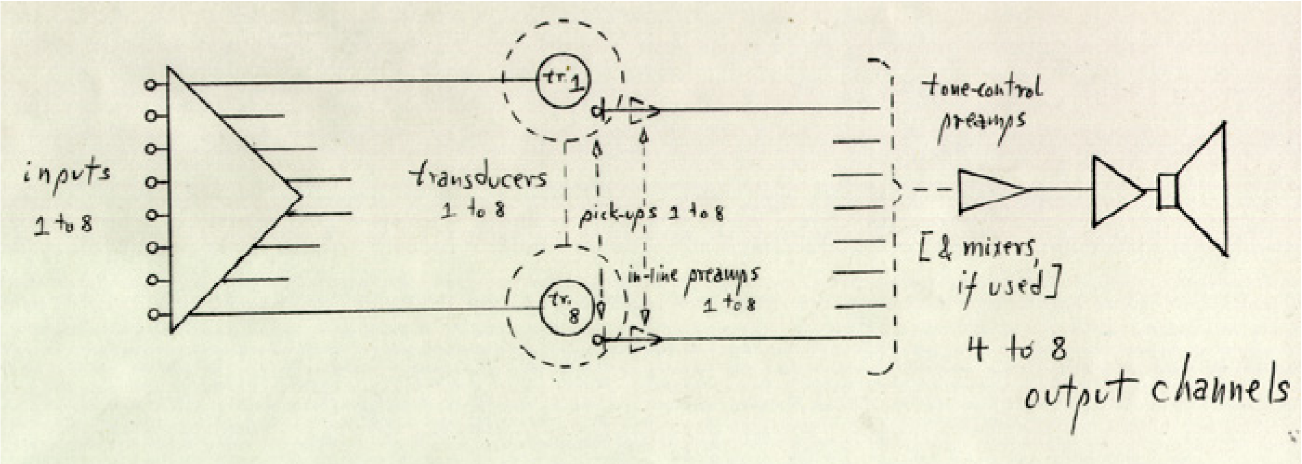


Figure 1. David Tudor's Rainforest (1968).

scores.

Through the concept of *inherent scores* we can better analyze the complex and mediated relationship between composers, performers and their instruments, especially in the case of electronic musical instruments. It has been often stated that electronic music instruments are *open-ended systems* [25]. Many times it is difficult to define where the electronic instrument ends and the composition starts. Certainly, for defining the instrument it is necessary to implement some input to output mapping strategies. But normally these strategies are fully affected by the characteristics of the composition to which the instrument is dedicated. Remarking this observation, Schnell and Battier introduced the concept of *composed instruments* [5]. This term serves to explain how our digital instruments equally *"carry the notion of an instrument as that of a score"*, in the sense of determining various aspects of a musical work itself. This is coherent with the fact that during the technical implementation of the instrument, being hardware or software, we often incorporate many ideas of composition into the programmed system.

A substantial question to this new musical practice would be if it resulted from a compositional, instrumental or technical development. We can illustrate an answer through the analysis of David Tudor's *Bandoneon!* (1966). In the words of its author, *Bandoneon!* is a *combine* of programmed audio circuits, moving loudspeakers, TV images, and lighting, all controlled through the live sound of a bandoneon played by Tudor. From the program notes of this work's premiere we read that *"Bandoneon! uses no compositional means, since it composes itself out of its own composite instrumental nature"* [6]. Kuivila also asserts that we were in fact facing a new way of understanding instrumentality. In these self-composed instruments, Tudor acts as the interpreter and performer of a composition that composes itself out of these constituent parts. Or using Lucier's arguments, the composition is created from the inherent scores that can be found in the structural elements of a particular electronic configuration. This concept carries an extraordinary rendition: the acceptance that an electronic instrument is an entity that can display itself without the need of a composer or a composition. Probably the most important characteristic of these inherent scores is that they can reveal or display themselves to their performers only at the exact moment of being performed.

Therefore, we can now assert that the origins of this new musical practice, under the influence of electronics and the germinating attitudes of post-modernism [1], trace their roots mainly on the appearance of a new approach to instrumentality. Thus, it does not mainly rely on the mere evolution of an existing compositional practice. The introduction of electronic components in composition definitely changed the understanding of what until that moment was defined as "playing". Many of this electronic circuitry was able to synthesize sound or modify sound without the need of direct manipulation. Then, instead of playing, performers "control" their instruments. John Fulleman [6], a frequent collaborator of John Cage, attributes David Tudor an *"ability to assert just enough control over the equipment to get through a concert"*.

In a lecture-talk given at the Oxford University [3] James Mooney explains how within an interview to the English experimental music band *Gentle Fire* in 1970, the band member Richard Bernas describes how he plays a custom sensor-based electronic music instrument called *qHong*. Bernas assures that: *"the instrument is the score of what we are playing"*. On his talk called *"the instrument is the score"* Mooney develops a framework where the relationships between instrument and score can be defined through shaping the affordances the instrument creates. For being more exact, its range of affordances. The concept of affordance in musical instruments will be explained and contextualized later in this paper. In addition, Mooney recognizes performers as another active shaping element of the musical composition. Consequently, for Mooney performers would have a crucial role in defining the musical work. Later this paper will recover Mooney's observations for proposing a theoretical framework for designing instrument-scores.

[1] Many artists labeled as postmodernists i.e. Frank Zappa or John Zorn declared how they were deeply influenced by this musical practice

2. PERFORMATIVE MATERIALITY

An important conclusion from our previous section is that inherent scores would be the result of an instrumental practice. Inherent scores only exist in virtue of a performer's commitment on interpreting some type of materiality as performative, being of physical, virtual or mixed origin. In order to explain the foundations of this instrumental practice, we will make use of the theory of *performative materiality*.

Jacucci and Wagner [7] have explained why the materiality of electronic musical instruments is not only a mere support for acoustic or digital sound machines. This materiality is performative too: *"material artefacts have a history, emerge as part of specific events in time and become part of performative action"*. Physical materiality has always a performative potential. The theory and application of performative materiality within Human Computer Interaction has been extensively studied by Johanna Drucker [8]. Drucker suggests that the materiality of a system *"only occurs when we action it, and only and at that moment we perceive and discover it, always distinct in each instance"*. For Drucker, *"material conditions provide an inscriptional base, a score, a point of departure, a provocation, from which a work is produced as an event"*. Certainly, as Brown and Duguid [27] have emphasized, material features, in their peripheral, evocative, and referential function, provide border resources for interaction. But can these features be considered scores?

In contemporary performative arts, scores can take diverse forms and materials: graphic scores, action scores, computational, sculptural, etc. Then, can anything be a score? The choreographer William Forsythe pointed out [9] that a score *"represents the potential of perceptual phenomena to instigate action, the result of which can be perceived by a sense of a different order"*. Following this idea, in traditional Western music a score would be the instigator of a transition from the visual to the aural via our body. In the case of non-traditional notation, how does a score define the way we interpret a musical work? Forsythe explains that *"a score is by nature open to a full palette of phenomenological instigations because it acknowledges the body as wholly designed to persistently read every signal from its environment"*. Forsythe lays more importance on the embodied relation with the performer: scores appear in the exact moment when a performer finds a performative potential within an object or a concept, deciding its phenomenological outcome on an open basis. Thus, each object would be an embodied device open to the phenomenological interpretation: a potential available to its conversion into a performative event. We could conclude that under this vision, any object has the potential to be a score. And definitely our musical instruments incorporate this potential too. Finally, if musical instruments are perceived as scores that would be essentially in virtue of their physical or performative materiality, as well as their high evocative power to instigate musical actions.

Under this generalized definition of score, it is now possible to interpret electronic musical instruments as scores. Not mainly because of being musical instruments, but essentially from the performative materiality they engage. The generalization of this fact would suggest that any object can be a score. For exemplifying this idea, James Mooney [6] attributes the creation of improvisations out of found objects to composer Hugh Davies, who *"re-purposed (objects) as musical scores"*, in a way that *"any visual stimulus can be interpreted as a set of instructions that shape the development of music"*.

Additionally, we should clarify that physical materiality is not the only in charge of shaping the musical work. Nowadays, a great part of our digital instruments base their functionalities on the control of graphical user interfaces (GUIs). In this case, their performative materiality cannot be expressed through physical artefacts. From Thor Magnusson's research [10] we understand better how the inherent affordances and constraints of the constituent elements of graphical interfaces mediate on screen-based musical instruments. In a certain way, GUIs and tangibles can be unified by the theory of performative materiality.

3. NOTATION

3.1 Preliminary Questions

The idea that an instrument can be a score affords creative relationships. But, it also carries very substantial questions. For example, if an instrument is a score, is it true that a score is an instrument too? If one thing can be the other at the same time, are both the same thing? If an instrument is a score, can one part be separated from the other? Where can we physically find this inherent score within the instrument's body? Is an inherent score the addition of both forms or is it a new synthetic thing? How is an instrument-score interpreted or shaped by its performers?

For fully understanding the nature of inherent scores, they have to be contextualized within the ontology of notation, arts and music.

3.2 Notational Systems

Along this section, we are adopting Nelson Goodman's notation principles taken from *Languages of Art* [11] in order to spell out the kind of notation behind inherent scores.

If inherent scores are scores, it is because they manifest symbols to their readers. A symbol in a notational system refers to something (literal, metaphorical, indirect) and its interpretation depends on the system of symbolization. Furthermore, the sort of symbol it is -linguistic, musical, pictorial, diagrammatic, etc.- will be in virtue of its belonging to a specific system.

A symbol system, say, the English language, actually consists of a symbol scheme -i.e., of a collection of *characters*- with rules to combine them into new, compound characters associated to a field of reference. For Goodman, symbol systems are notational when:

1) the characters are correlated to the field of reference unambiguously (with no character being correlated to more than one class of reference, or compliance class)

2) what a character refers to -the compliance class- must not intersect the compliance class of another character (i.e., the characters must be semantically disjoint)

3) it is always possible to determine to which symbol an item in the field of reference complies (i.e., the system must be, semantically, finitely differentiated).

Languages like English have a notational scheme but fail to be a notational system because of ambiguities (in English, cape refers to a piece of land as well as to a piece of clothing) and lack of semantic disjointness (man and doctor have some referents in common).

Finally, let's apply these definitions to our artistic field. Sculptural or pictorial systems fail on both syntactic and semantic grounds so they are non-notational systems. Within Goodman's approach, a musical score is a character in a notational system only if it determines which performances belong to the work and, at the same time, is determined by each of those performances. Given the notational system and a performance of a score, the score is recoverable. This is ensured by the fact, and only by the fact, that the language in which a score is written must be notational, so it must satisfy Goodman's stated requirements.

3.3 Inherent Scores Notational Scheme

For bearing out the notational scheme of an inherent score we need to examine its symbol system and rules. If musicians consider that instruments, through their constraints and affordances, are scores too, then some kind of symbols and rules must exist. In this section we will first clarify where these symbols can be found. This analysis will help us to conclude if this system is notational or not. It is important to remark that our task here is not showing if it is possible to create a Goodman's notational system for electronic instruments. Our focus is on understanding what kind of notation is the one of an instrument, or the notation of an inherent score.

The initial and probably main complication consists in the total absence of rules within the field of reference. Normally, within instruments materiality i.e. a cello, elements are not discretized. The space of affordances against materiality is continuous. For this reason, traditional Western musical notation establishes a radical discretization on this space of affordances. For example, in Western musical notation, the space of frequencies has been entirely discretized with the use of notes and scales. Or as another example, among all the possible sounds that i.e. a cello can produce, our traditional Western notation has filtered out all kind of noises, being centered on the production of tones.

Additionally, within this continuum of materiality, if a constituent is defined as a symbol or not, it is a decision left to the actual performer of the inherent score. Goodman explains that these kind of systems are essentially *analog systems*. For every character there is an infinite amount of others such that referring to the same mark. We cannot possibly determine that the mark does not belong to all and such that for some object we cannot possibly determine that the object does not comply with all. A system like this kind is obviously the very antithesis of a notational system.

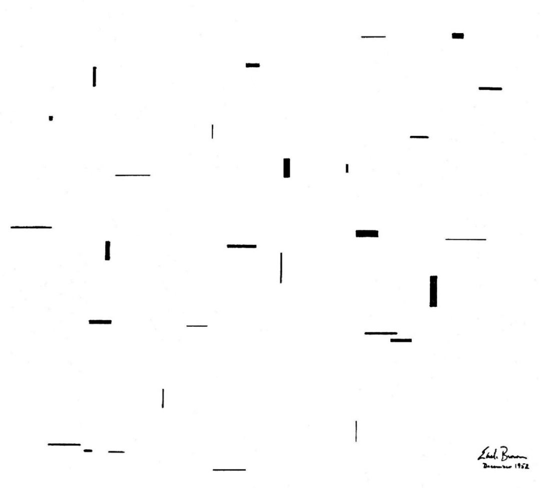


Figure 2. Earle Brown's December 1952

3.4 Non-Notational Systems and Musical Graphs

In the search for defining the properties of an inherent score and its symbols, there is a clear lack of terminology to apply here. For this purpose, we propose first analyzing the notational scheme of graphic scores, which nowadays are an accepted format of scores while they have been extensively studied within the ontology of music. From these results it would be easier to extrapolate some parts of our analysis. Although graphic and inherent scores are not the same thing they share many instrumental similarities. Later in this paper we will explain some interesting differences applied to instrumentality. Nevertheless, their terminology can be used for incorporating our inherent scores within the ontology of music.

Graphic scores appeared in the musical avant-garde as a way to release composers from the constraints of writing their music using the notation of a traditional Western score. Consequently the representation of a musical idea opened to the personal and subjective selection of graphic figures that inspire new and imaginative ways of interpretation. One of the first examples of graphic scores is Earle Brown' *December 1952* (Figure 2).

Are graphic scores notational systems? Earle Brown did not specify how his graphical symbols should be interpreted. Therefore, depending on just how the symbols are interpreted, syntactic and semantic disjointness may be lacking. In cases like *December 1952*, composers are using systems that only slightly restrict the performer's freedom to play what and as he pleases. The system furnishes no means of identifying a work from performance to performance. Furthermore, we can say that the system of *December 1952* is non-notational, like inherent scores.

An early but fundamental contribution describing, illustrating and classifying the symbols used by modern composers was Erhard Karkoschka's *Notation in New Music* (1965) [12]. Karkoschka developed the following typology of musical systems:

- Precise Notation: where every note is named
- Range Notation: where for example, only the limits of

the ranges of notes are set

- Suggestive Notation: where at most relations of notes, or approximate limits of ranges, are specified.

- Musical Graphics

Certainly, musical graphics are non-notational because they lack both syntactic or semantic articulation. We should note Karkoschka's intuition in not calling them *musical graphics notation*. In *Languages of Art* Goodman explains that Musical Graphics are another example of *analog systems*. For every character there is an infinite amount of others referring to the same mark.

We must remark that *musical graphics* -as they were coined by Karkoschka- are non-notational systems but they are still scores. The implications of classifying traditional Western scores as notational systems and musical graphs as non-notational do not restrict us from saying that *December 1952* is a score. The appreciation of -what is- and -what is not- a score has changed historically with the introduction of new musical poetics. Finally, it is a fact that Earle Brown's *December 1952* has inspired hundreds of musical realizations. Thus it must be a score.

A very important property of graphic scores is that they are usually not created with the interest of substituting a "normal" score. As discussed by Rebelo [13] or Vlagopoulos[14], and by Earle Brown himself on his seminal *On December 1952*, graphic scores are usually created as improvisational scores. They appear with the mere intention of guaranteeing an unique way of performing. But graphic scores are not "the performance". Furthermore, the graphic score is a trigger for the interrelation among performers in the rehearsal phase, if it exists. A graphic score is a provocation to solve a musical challenge with our own poetics on communication with ourselves and the other performers. This strategy would be congruent with the practice of musical improvisation. For suggesting an open improvisation, certainly a non-notational system can be a very valid creative trigger. Any effort in the direction of discretizing the system of symbols used during a performance (e.g. with a notational system) would lead to discretization of the musical response as well.

Updating Karkoschka's typology, nowadays our musical graphs can adopt any form and any dimension. It is remarkable that historically a big percentage of these graphic scores have eminently used paper as the medium, in an inexplicable and non necessary conceptual analogy to the traditional format of the traditional score. We had to wait until the advent of digital interfaces to see musical notations that can be interactive, dynamic, fragmented or non linear. Examples would include the animated scores of Miyashita [15] or the three-dimensional scores of Berghaus [16].

3.5 Instrumentality of Graphic and Inherent Scores

Graphic and inherent scores are non-notational systems manifesting interesting differences in their instrumentality.

The first difference we can observe is that graphic scores exist in physical or virtual forms. They can adopt diverse forms. Even they can be virtual or dynamic. But they are always perceived through our senses and they can be analyzed before we start playing the instrument. On the contrary, inherent scores display themselves through the creative exploration of the constituents of a musical instrument. And this happens only at the moment of performance. Therefore, inherent scores cannot be formally understood as objects. They are a purely mental activity enacted from the interaction of a performer with an instrument.

Interestingly, when an author or a performer declares that a visual composition should be considered as a graphic score, the original graphic object extends itself towards music and in a non physical way. It acquires a new abstract dimension able to enact musical compositions. The graphic object becomes a new mental category, in a similar way to inherent scores. However, conceiving music from a graphic score demands some rational. Performing a graphic score means giving visually perceived content a musical meaning. As we remarked, graphic scores do not substitute traditional scores. They are a kind of mental provocation. But this translation from the visual to the aural needs an interpreter, being it an individual, a collective or even a technological device. To this rational many external elements can be added: in-situ possibilities, sociocultural influences, etc. All this plethora of information makes the realization of a graphic score an unique musical work, intimately connected to its performers.

On the contrary, there is nothing to translate when performing inherent scores. The decision-making process during performance is normally intimately connected with our creative exploration and the resulting sonic reinforcements. Understanding the specific performative potentials of an instrument is an *a posteriori* process. It happens once we have already started playing. The performer of an inherent score does not need a translation from the physical to the aural. Even more, many times the instrument can sound without our interaction [2] . Therefore, the performer's task is closer to the role of controlling or modifying this continuous flow of sound. Indeed, many times, these sonic affordances are not predictable. They can change or evolve during a performance with the conditions of the room, the situation of the performer or the instrument configuration. All this makes very difficult to prepare a concert plan in advance but allows ample space for experimentation. In contrast, for playing graphic scores, performers usually will require some *aprioristic* thinking. Sounds usually are produced after some cognitive process of interpretation from the graphic elements found in the score.

4. THE FORM OF AN INHERENT SCORE

4.1 Hybrid Arts Forms

Another substantial problem is the artistic form of the musical work that an inherent score affords. And how it functions in relation to other visual or physical elements existing in the instrument.

Acoustic instruments are eminently defined from their physical materiality. Electronic music instruments consist

[2] For example a "voltage controlled oscillator" can sound since the moment it is connected to a power supply

of hardware and software. Both the visual and physical part of an instrument can be specifically designed to infer some kind of limitation or, in other words, to shape the musical work. Inherent scores are the combination of existing forms resulting in a kind of hybridization.

Within the ontology of arts, philosophers have studied the identity or nature of the art object in *physical arts* (painting, sculpture, etc) and in the so called *non-physical arts* (mainly music and literature). In the latter, there is no particular "thing" to be considered the artwork itself. The score of a musical sonata or the printed paper of a novel are not considered the art object itself but its representation. Some authors like Croce [17], have suggested that music and literary works are purely mental.

In philosophy, there is a wide-spread consensus [3] that a musical work is a variety of abstract object, a structural type or kind. If like in classical Western music, composers have not created the sounds to be heard in a performance, where is the actual work? Is it in the score? Scores are the mere representation of the musical work. They are the symbols to concatenate during a performance plan. But scores do not sound per definition so they cannot be the musical work. Introducing the wide debate on the identity of the musical work would require a longer extension but since there are artists claiming that their interfaces and instruments are scores, then we should at least introduce the problem of defining the musical object into our discussion.

An interesting approach for deducing the form of inherent scores can be taken from Jerrol Levinson's theory of hybrid art [18]. Levinson notes that not all kind of arts are pure, some are hybrids. Examples like kinetic sculpture or interactive audiovisual installation show us that independently of their complexity, these forms of art show elements of multiple art forms. A kinetic sculpture would be the encounter of sculpture and dance. An audiovisual installation would consist of multiple media: cinema, music, sculpture, etc. On the contrary, we perceive a traditional figurative painting as an instance of its category. For Levinson the hybrid status is primarily a *historical* thing, in a way, as is being a biological hybrid. An art form is hybrid *"only in virtue of its development and origin, in virtue of its emergence out of a field of previously existing artistic activities and concerns, two or more of which it in some sense combines"*. Inherent scores are good candidates to be considered hybrid forms. At this stage of the explanation, we could describe them intuitively as the mixture of a musical work and performative materiality enacted from physical or virtual objects. For us, the most important feature that this theory exhibits is that if an art form is hybrid then it must be understood in terms of the combination of its original components.

Levinson extends his theory of hybrid art to the combination of existing art forms and technological processes. For example, laser sculpture, computer music. computer graphics, video installation, etc would be a result of this combination. Thus, Levinson features clearly the plausibility of new art species creation from the hybridization with technologies. The resulting possibilities for this process are three: juxtaposition (or addition), synthesis (or fusion) and transformation (or alteration). In all these three cases of process, Levinson explains that the hybrid combination of art form A and B to produce C, will change the properties that A or B exemplifies in the joint context. These properties would be relative to what one of the original forms would exemplify on its own, or at least affect the prominence of what each exemplifies after combination.

As we have discussed before, the object embedding an inherent score stays in an identical physical form after its perception as a score too. The consideration that an instrument is a score is produced at a mental level. Thus, this change of perception does not carry physical changes. The same occurs in the case of graphic scores. Graphic scores seem to be a good example of hybrid art form resulting from painting and music. As well in this case, the consideration that "a painting as a score" does not bring changes to the painting form itself. In graphic scores, this hybridization changes the perception of a very known physical object (the painting) transforming it into a hybrid of a physical and a non-physical entity, an object and a musical work. The same would happen to a sculpture if at some moment we manifest understanding it as a score. This example supports the idea that hybrid art forms are essentially new historical forms. Once the artistic practice adopts a hybrid art form and it becomes general, we will not refer to its hybrid origin anymore.

Coming back to the discussion on our inherent scores, we have gained an adequate theoretical framework for interpreting the combination of both the physical and performative materiality of an instrument as a new type of hybrid form. This hybrid art form, the inherent score, would be the fusion of two existing forms (physical materiality and performative materiality) resulting on the synthesis of a new kind. Like in the case of graphic scores, the existence of this new hybrid form is congruent with the perception of performers of being playing a score when they manipulate the instrument. Performers have the perception of playing a specific form. Therefore, this inherent score would be a new abstract object perceived no longer as only the physical instrument. It is perceived as a performative potential of the instrument shaping every moment during the act of playing. Certainly, the result of this fusion alters the perception of the original forms. Instrument's physical materiality gets augmented and extends itself towards a compositional object, acquiring some abstract attribute. In the same way, performative materiality gets some kind of order. It defines itself for a specific use and a particular performer.

At this stage we can now answer some of our preliminary questions we left open. For example, when an instrument is a score, our question was if one part can be separated from the other. Having concluded that an inherent score is a new abstract object synthesized from the fusion of two already existing, we can now assert that this separation is not possible. There is nothing to separate. The instrument

[3] A complete review on the debate of "what a musical work is" cannot be afforded here, but we can divide between two actual tendencies. First Platonist or realist theorists holding that musical works are collections of concrete particulars e.g. Goodman 1968, Kivy 1983, Levinson 1980, Davies 2001; and those anti-realists who deny there are any such thing as musical works e.g. Rudner 1950, Cameron 2008, Stecker 2009.

still exists but a new abstract musical object appears on stage as an attribute in the system. There is no possibility of explaining the inherent score to others without the instrument itself. Additionally, as we have shown before in this paper, this inherent score is eminently a subjective and mental creative attribute that can be interpreted differently by every performer. Separating the score from the instrument if possible, would still require information on the performer involved to be understood. The importance of performers for shaping an inherent score will be more exhaustively analyzed in this paper when studying their physical embodiment (section 5.2).

Another question formulated was if in the case an instrument is considered as a score, then if it is true that a score would be an instrument too. As we have explained before, every object can have specific performative potentials to be perceived as a score. On the contrary, inherent scores are abstract musical objects and they usually do not embed any specialized feature for music creation except the natural sound their physical constituents produce. Therefore, in general scores are not perceived as instruments [4] .

4.2 Inherent Scores: typology of symbols

For the analysis of this hypothesis of hybridization we are suggesting here a typology of symbols that we can find within our instruments. Due to the important role that technology holds at many of our actual music instruments, we will focus our attention on those instruments incorporating some kind of computational system behind their configuration.

The first type of symbols are purely *extrinsic*. These would be mainly representational. They give us information about the computational status of the musical instrument being played. For example, the visual composition of tokens on a table-top interface like a Reactable [5] [19] (Figure 3) is an example of extrinsic symbols. They represent the status of the algorithms an user is running at every moment and how they are interconnected. In this case, the systems affords less on the materiality of these tokens (e.g. form, color, material, texture, etc). If instead of using these original acrylic tokens we use other ones made of wood, the sound mapping or the overall sonic output will not be affected.

The second type of symbols are *intrinsic*. These would be inherent to the affordances of the physical interface. Furthermore, we talk about intrinsic symbols when their physical affordances determine various features of what is aurally enacted. Using the same example, *Reactable*'s round table form affords to the multi-player or collaborative performance. Through this specific intrinsic property, the instrument's materiality definitely shapes performative materiality and finally the way the instrument can be played.

[4] An exceptional example of "a score that sounds" is the project *Tangible Scores*, a technological hybrid allowing a visual score being the controller of a sound synthesis. This will be showcased later in this paper at section 6.1

[5] The Reactable is a round form electronic music instrument. By placing blocks called tangibles on the table, and interfacing with the visual display via the tangibles or fingertips, a virtual modular synthesizer is operated, creating music or sound effects.

Figure 3. A Reactable. Photo by Daniel Williams

We can find both types of symbols within instruments and their combinations are possible. For example, a modified *Reactable* incorporating the same type of representational tokens but featuring recognition of specific physical properties of these tokens. For example, the stretch force applied to rubber made tokens could control the volume of an associated sound synthesis. In this case the extrinsic composition of tokens can be affected by the properties of the intrinsic physical materiality supporting it.

5. COMPOSING INHERENT SCORES

5.1 Affordances and Constraints of an instrument

For composing the inherent score every instrument embeds we will follow the principles proposed by James Mooney [3]: shaping its affordances and constraints.

Affordances, as psychologist J.J.Gibson defined them, are the properties of the relationship between the environment and the agent. In our case, the environment would be the musical instrument as a reference frame. The agent would be a performer. Between agent and environment, infinite relationships can be created, but the potentials of performing some event are less probable than others. Sometimes even impossible. A violin affords playing sounds, but it does not afford traveling.

A remarkable propery of affordances is that they are highly dependent on the reference frame where they are inscribed. For example, cultural contexts or personal backgrounds. What an object affords to a a person can be different to another, even living in the same sociocultural environment [20]. Therefore, affordances could be essentially subjective perceptions influenced by our social constructs. And this condition can reach the maximum of dependency in the case of performative arts. From classic ethnographic studies we know how performances are central to human understanding [21] and post-modernism have drawn attention to the way performances seek to reinforce and communicate our identities in society [22]. Recent research on socio-situated interface design [23] is coherent with the idea of socio-oriented performing frames. These theories suggest that when using an interface, cognitive scaffolds can only exist in the context of a social setting. Certainly, the capability of performing carries a substantial context and the sociological ecology of acting in front of others.

In parallel we have the notion of constraint. The application of the concepts of affordance and constraint in electronic music instruments has been deeply studied by Thor Magnusson. The author explains [10] how Margaret A. Boden [25] defines constraints as one of the fundamental sources for creativity: *[F]ar from being the antithesis of creativity, constraints on thinking are what make it possible. ...Constraints map out a territory of structural possibilities which can then be explored, and perhaps transformed to give another one.*. This assessment remarks the potential of constraints as a trigger for creative enactments. Constraints would be characters in the performative materiality of an object, being it physical or virtual.

Within the discipline of improvisation with electronics, the instrument's affordances take the important role of shaping the way an interface is played through its different constraints. But more important for our discussion, if they are considered as scores then they would suggest what is played too. As we have seen before in this paper, this inherent score can afford interesting performative enactments.

Mooney [3] supports the idea that a musical instrument can be designed from the perspective of which kind of music relationships it affords. Also, Mooney identifies the possibility of defining the "spectrum of musical affordance" of instruments. This can be achieved by designing the instrument or interface and establishing a range of musical practices the instrument can support. For example, although very complex textures of sounds can be played and controlled with a *Reactable*, it would be rather difficult to play *Mary had a little lamb*. It is then noticeable that the spectrum of affordance is not comparable to difficulty or to complexity of the instrument. Affordances are fully mediated by the embodied relationship between instrument and performer. Even if the performer knows the musical notes of *Mary had a little lamb* it will be impossible to play it correctly on time with a *Reactable*. Therefore, instrument's performative affordance and other types of affordance, like musical affordance, expressiveness affordance, etc cannot necessarily match.

5.2 Physical Embodiment

As we have seen, performing with an electronic music instrument would be the creation of relations and meaningful structures between the inherent score and its enacted audiovisual interpretations. But these are always mediated and shaped by our embodiment. A good instrumentalist is able to create embodied relationships with the instrument, leading to a feeling of intimacy and control. Undoubtedly this will be perceived as a key factor to evaluate the expressiveness of a good performance.

This evidence was used by Mooney [3] to introduce intuitively the "performer" as another shaping parameter of the musical work. If the instrument is the score, then many of the decisions taken, even shaped by the instrument, will be result of performer's acts in freedom. Although Mooney did not develop further this argument, he introduced another variable in the equation: performer's reference frame and performer's embodiment. First, the sociocultural context of the performer, even the actual mental conditions at the moment of approaching a performance will shape its result. For example, the expressiveness of a first musical approach with a *Reactable* depends highly on knowing the elements of computer music in advance (what is a synthesizer, a sequencer, etc). Second, more objective factors connected with the embodiment can conduct the musical outcome. For example, if the electronic instrument depends highly on a physical ability that cannot be achieved by a specific performer i.e. *through some disability*, all the performative affordances designed can appear hidden or invisible.

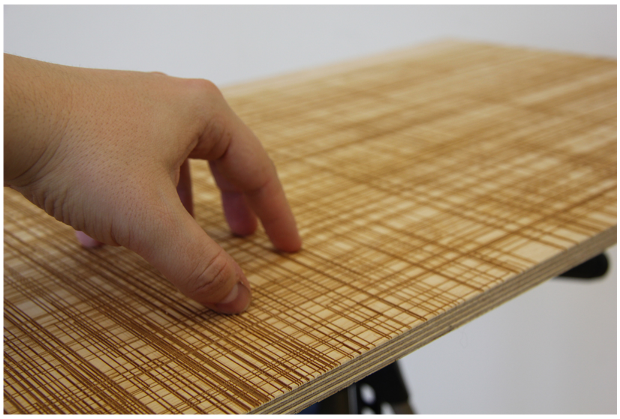


Figure 4. A Tangible Score example.

Thus, we can conclude that design models centered only on defining constraints and affordances must include "the performer" as an influencial parameter. Therefore, we could only speak of inherent scores when connected to a particular performer. In *Rainforest* we would describe Tudor's version, John Cage's realization, etc. Probably all the instances of the same musical work will be very different.

6. TWO EXAMPLES

6.1 Tangible Scores

In 2014, together with Martin Kaltenbrunner, I presented the paper *Tangible Scores: Shaping the Instrument Inherent Score* [1] at the New Interfaces for Musical Expression conference (NIME). *Tangible Scores* are a new paradigm for musical instrument design with a physical configuration inspired by graphic scores (figure 4). This instrument implements practically many of the concepts and ideas of the so called instrument-scores, and it has been reviewed by Maestri and Vlagopoulos a the first TENOR conference in 2015 [2]. Many aspects of the theories explained here were achieved during the practical development of *Tangibles Scores*. Therefore, my intention is now contextualizing the instrument *Tangible Scores* within the framework previously explained.

A Tangible Score [6] is a tactile interface for musical expression that incorporates a score in its physical shape,

[6] Full information on the project and videos can be found online at the following URL: http://interface.ufg.ac.at/tmg/projects/tangible-scores/

surface structure or spatial configuration. Creating an intuitive, modular and expressive instrument for textural music was the primary driving force. Following these criteria, we literally incorporated a musical score onto the surface of the instrument as a way of continuously controlling several parameters of the sound synthesis. Tangible Scores are played with both hands and they can adopt multiple physical forms. Complex and expressive sound textures can be easily played over a variety of timbres, enabling precise control in a natural manner. Using sound as a continuous input signal, both synthesis and control are available simultaneously through direct manipulation on the engraved patterns of the physical score.

Every *tangible score* is conceived from a different graphical score (Figure 5) that still represents a musical idea but it has been also specially designed for providing a diverse palette of acoustic signals when touched. But more important, the tactile scores define and propose specific gestural behaviors due to the different affordances and constraints of the object in front. Sound is generated through a polyphonic concatenative synthesis driven by a real-time analysis and classification of input signal spectra. Each of the scores is loaded with a specific sound corpus that defines its sonic identity. Thus, a *tangible score* provides a implicit visual and haptic feedback in addition to its sonic core functionality, making it intuitive and learnable but as well suitable as an interface for musical improvisation and sonic exploration.

At the moment of designing our paradigm, we were quite influenced by Lucier's quote: *"there were no scores to follow; the scores were inherent in the circuitry"*. We accepted it. It matched our instrumentalist intuition as long time electroacoustic music improvisers. Additionally, it was possible for us to contextualize the instrument within the field of tangible interfaces and human computer interaction (where we have been working for a long time too).

We first understood that, not only musical instruments, but any physical or virtual object loaded with performative materiality affords being played. And second, that any physical or virtual object has the potential of instigate actions so it can potentially afford being interpreted as a score. Then, our direction had to follow the direction of trying to shape those potentials. By intentionally limiting or constraining the infinite possible interpretations of a specific object within its reference frame, we are shaping the inherent score it contains. And we do it in a deliberate act of musical composition.

In order to shape those potentials, we decided conducting or inspiring particular gestures by incorporating contrasting visual and tangible patterns on the surface of the instrument. For the first series of tangible scores, we first composed a set of generative patterns that we later engraved on wooden surfaces. For finding the adequate patterns, not only an attractive a visual or gestural idea was searched. We had to negotiate its form with the adequate sonic result when touched. This relationship mediates radically the sound synthesis of the instrument.

For evaluating the instrument, apart from performing with them at different concerts, we offered and showcased some

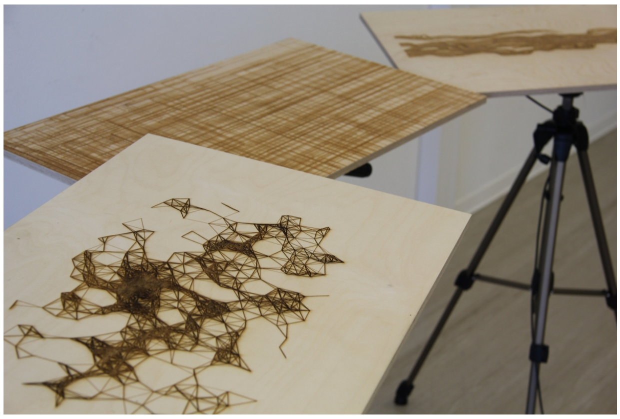


Figure 5. Different designs for Tangible Scores

tangible scores to professional performers and composers (mainly in Linz, Austria). Additionally, we showcased the instrument as a sound installation during two mass audience festivals (*Sonar* and *Ars Electronica* 2014) were thousands of visitors could play it. From the analysis of these experiences, for us was clearly evaluated and proved that both physical gestures and sound gestures were mainly inspired by the visual and tangible patterns: their direction, size, intention, etc.

One important decision taken in this first series of tangible scores was that our design should rely only on intrinsic elements or symbols. Although a tangible scores is fully a digital instrument, we decided not displaying representational information on the instrument. Due to this decision, the computational status is hidden and the sonic mapping depends intimately on the embodied relationship between player and instrument. Due to this unification of score and instrument, the instrument provides the representation and control within a single musical artifact, fully concentrating the performer's attention on the interaction with the musical composition in a physical way.

6.2 Choreographic Objects

Within the field of contemporary dance, the choreographer William Forsythe created the concept of *choreographic objects* [9]. Physical objects, of various types, are considered choreographic when they are able to enact particular behaviors and movements via ballet dancers. These objects reveal a choreography that is inherent to their physical materiality.

This idea comes from Forsythe's intuition on perceiving every object as a source of enactments. As we described earlier in this paper, for Forsythe a score *"represents the potential of perceptual phenomena to instigate action, the result of which can be perceived by a sense of a different order"*. As well *"a score is by nature open to a full palette of phenomenological instigations because it acknowledges the body as wholly designed to persistently read every signal from its environment"*.

An example of the use of choreographic objects is the work *Nowhere and Everywhere at the Same Time, No.2* (Figure 6) created for a solo dancer and 400 pendulums suspended from automatic grids. When activated they ini-

Figure 6. The *choreographic object* "Nowhere and Everywhere at the Same Time, No.2" by W. Forsythe.

tiate a sweeping 15 part counterpoint of tempi, spacial juxtaposition and gradients of centrifugal force which offers a constantly morphing labyrinth of significant complexity. This setup privileges the unconscious choreographic competence induced by this special choreographic situation. For Erin Manning [24], choreographic objects are *"an affordance that provokes a singular taking-form: the conjunctive force for the activity of relation"*.

In a similar way to tangible scores, choreographic objects have been traditionally considered a constraint. Forsythe develops an active stage, a composed reference frame suitable for performative enactments. If stage scenographies are usually representational, choreographic objects do not represent anything else than a potential to move.

In the case of the work *Nowhere and Everywhere at the Same Time, No.2*, Forsythe gives a fundamental form to the space. Through this morphophoric affordance, the danceable space and all the possible movements are discretized by the inertial materiality of the pendulums. In this work, the dancing score can be found in the performative affordance of a myriad pendulums defining a composed space around them.

Undoubtedly, choreographic objects have the ability of inducting creative movements and gestures in its performers. Especially in large spaces and multi-user contexts, we are convinced that the notion of choreographic objects can be useful to inspire the creation of novel interpretation of scores as well as new phenomenological enactments.

7. CONCLUSIONS

Along this paper, we have proved the validity of using the concept of inherent scores for describing the mediated relationship between performer, score and instruments. Especially in electronic music instruments. We have explained how the theory of performative materiality serves well to explain the fact that any object can be understood as a score. We have defined the notational scheme of inherent scores as non-notational and we have described the remarkable differences in instrumentality between inherent and graphic scores. We have elucidated the nature of inherent scores particularizing them as a hybrid forms resulting from the fusion of performative and physical materiality. Finally, we have proposed a framework for composing or shaping musical works, demonstrating its possibilities with two examples: tangible scores and choreographic objects.

8. REFERENCES

[1] E. Tomás, M. Kaltenbrunner, Tangible Scores: Shaping the Inherent Instrument Score, Proceedings of the International Conference for New Interfaces for Musical Expression, 2014

[2] E. Maestri, P- Antoniadis, "Notation as Instrument: From Representation to Enaction", First International Conference on Technologies for Music Notation and Representation, 2015

[3] J. Mooney, "The instrument is the Score": Relationships between Instrument, Performer and Score. Symposium for Performance of Electronic and Experimental Music (SPEEC) Oxford University, 2012

[4] A. Lucier, "Origins of a Form: Acoustic Exploration, Science and Incessancy". Leonardo Music Journal 8, pp. 5-11, 1998.

[5] R. Schnell, M. Battier, "Introducing composed instruments, technical and musicological implications", Proceedings of the 2002 conference on New interfaces for musical expression, 2002

[6] R. Kuivila, "Open Sources: Words, Circuits, and the Notation/Realization Relation in the Music of David Tudor", Presented at the Getty Research Institute Symposium, "The Art of David Tudor", 2001

[7] G. Jacucci, I. Wagner, "Performative Roles of Materiality for Collective Creativity", Proceedings of the 6th ACM SIGCHI conference on Creativity and cognition, 2007

[8] J. Drucker, "Performative Materiality and Theoretical Approaches to Interface," DHQ, forthcoming Winter, 2012

[9] W. Forsythe, "Choreographic Objects", Internet Essay: http://www.williamforsythe.de/essay.html

[10] T. Magnusson (2010) Designing constraints: composing and performing with digital musical systems Computer Music Journal, 34 (4). pp. 62-73, 2010.

[11] N. Goodman, "Languages of Art", 1968.

[12] E. Karkoschka, "Notation in New Music", 1965.

[13] P. Rebelo, "Composing with Graphics: Revealing the Compositional Process through Performance", First International Conference on Technologies for Music Notation and Representation (TENOR) , 2015.

[14] P. Vlagopoulos, "Is It All Autographic? Samples from the Musical Avant-Garde of the 60s", 12th International Conference on Music Perception and Cognition (ICMPC), 2012.

[15] H. Miyashita. Graphic Score Animation for Two Pianos, Feb. 28th, 2003. Toyama shimin plaza, Japan.

[16] M. Berghaus, ”Some Ideas for Three-Dimensional Musical Scores”, Leonardo Music Journal Volume 21, pp. 7-8, 2011

[17] B. Croce, ”The Aesthetic as the Science of Expression and of the Linguistic in General”, Cambridge: Cambridge University Press, 1992.

[18] J. Levinson, ”What a Musical Work Is, Again”, in Music, Art, and Metaphysics, Ithaca, NY: Cornell University Press, 215263, 1990.

[19] S. Jordá, M. Kaltenbrunner, G. Geiger, R. Bencina, ”The reacTable”, Proceedings of the International Computer Music Conference, 2005.

[20] D. A. Norman, Affordances, Conventions, and Design. Interactions 6(3), 1999

[21] J. G. Sheridan, N. Bryan-Kinns, A. Bayliss, ”Encouraging witting participation and performance in digital live art”. Proceedings of the 21st British HCI Group Annual Conference on People and Computers, 2007

[22] G. Butler, ”Bodies that matter: on the discursive limits of ”sex” ”. New York, London. Routledge, 1993

[23] J. Dijk, R. Lugt, C. Hummels,”Beyond distributed representation: embodied cognition design supporting socio-sensorimotor couplings”, Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction, 2014

[24] E. Manning, ”Relationscapes: Movement, Art, Philosophy”. Cambridge, Mass. MIT Press, 2009

[25] A. Tanaka, Sensor -based Instruments and Interactive Music pp. 233-255 in R. Dean, The Oxford Handbook of Computer Music. Oxford University Press, 2009

[26] M. A. Boden, ”The Creative Mind: Myths and Mechanisms”. London:WiedenfieldandNicholson, 1990.

[27] J. S. Brown, P. Duguid, ”Bordeline Issues: Social and Material Aspects of Design”. Human-Computer Interaction Journal 9:3-36, 1994.

MUSICKING THE BODY ELECTRIC
THE "BODY:SUIT:SCORE" AS A POLYVALENT SCORE INTERFACE FOR SITUATIONAL SCORES

Sandeep Bhagwati
Concordia University Montréal
sandeep.bhagwati@concordia.ca

Isabelle Cossette
McGill University Montréal
Isabelle.Cossette1@mcgill.ca

Joanna Berzowska
Concordia University
joanna.berzowska@concordia.ca

Adam Basanta
Julian Stein
Joseph Browne
Alexandra Bachmayer
Felix Del Tredici
Sarah Albu
Concordia University

Marcello Giordano
John Sullivan
Deborah Egloff
Marcelo Wanderley
McGill University

Julian Klein
Institut für Künstlerische Forschung Berlin

ABSTRACT

Situational scores, in this paper, are defined as scores that deliver time- and context-sensitive score information to musicians at the moment when it becomes relevant. Mnemonic (rule/style-based) scores are the oldest score models of this type. Lately, reactive, interactive, locative scores have added new options to situative scoring. The body:suit:score is an interface currently developed in collaboration of four labs at Concordia and McGill Universities in Montréal - an interface that will allow the musical use of all four types of situational score. Musicians are clad in a body-hugging suit with embedded technology - this suit becomes their score interface. Ultimately intended to enable ensembles to move through performance spaces unencumbered by visual scores and their specific locations, the project currently enters its second year of research-creation. The paper discusses the closely intertwined technological, ergonomic, performance-psychology-based and artistic decisions that have led to a first bodysuit prototype - a vibrotactile suit for a solo musician. It will also discuss three etude compositions by Sandeep Bhagwati and Julian Klein for this prototype, and their conceptual approaches to an artistic use of the body:suit:score interface. Finally, the paper discusses next steps and emergent problems and opportunities, both technological and artistic.

1. TOWARDS THE body:suit:score

The practical need for a non-visual score interface such as a body:suit:score arose in performances of comprovisation scores that rely on musicians freely moving in space.

In Bhagwati's scores "Racines Éphemères" (2008) for eight musicians, obbligato conductor and sonic trace amplifier and "Nexus" (2010) for five networked musicians moving through urban indoor and outdoor spaces with obbligato improvisation software, the musicians realize complex spatialisations and sound trajectories by walking while interacting with other musicians and/or the audience. They also integrate auditory and visual cues from the performance space and environment and, at certain moments also conductor's signals, into their realization of a comprovisation score.[3],[4]

Figure 1 "Nexus" at Concordia University (May 2010). Guy Pelletier (flute) and Lori Freedman (bass clarinet)

In these comprovisations, not only the delivery of score information to the musican, but its very meaning crucially depend on the situations the musicians are in: their position in the space, their physical closeness to and their musical relationship with the other musicians of the ensemble.

At the time, different scoring strategies were employed: In "Racines Ephémères", about 8 music stands per performer were placed in strategic positions. Musicians could not move at will or continuously, they were constrained to follow 'their' trajectory, with frequent stops. In 'Nexus', as musicians roamed freely and largely unpredictably throughout a city block, the score became a web of rules that had to be learned by heart. These rules consisted mainly of reaction protocols to either the music from other players sent to their backpack loudspeaker through the network - or to contextual cues, such as imitating the rhythms of conversations, or signaling the crossing of an indoor/outdoor threshold by a pre-defined phrase. Musicians also had to memorize different pitch sets, each corresponding to one of the other four instruments.

Though the performances worked and were received well, it became clear that both approaches to scoring for moving musicians had serious flaws: both burdened performers with unnecessarily distracting non-musical choices and mental constraints: in 'Racines', the use of space could not spontaneously be adapted to the music arising from comprovisation, musicians could not translate musical affinity into spatial proximity. And in 'Nexus', musicians were worried about potential memory lapses that could destroy the web of musical interactions. Moreover, learning such non-traditional constraints and rules well enough to recall them quasi intuitively during performance proved to be quite daunting, especially for improvising musicians. The learning curve for these pieces turned out to be quite steep.

"Musicking the Body Electric" is a four year research-creation project funded by the Canadian Social Sciences and Humanities Council (SSHRC). In its envisaged final incarnation, the body:suit:score we work towards is conceived as an exemplary instance of a polyvalent interface for situational scores that would address and provide solutions for most of these concerns.

2. SITUATIONAL SCORES

When we follow a linear score – whether on paper or on screen – the passage of time reveals context-invariant information structures that predate performance. Information in such scores is accessible at all times, at least in principle.

Situational scores, as defined here, are scores that do not build on such linear, pre-existing information structures. Information in these scores is only available ephemerally, i.e. while it is displayed or accessed in a particular context.[4], [14]

Four principal kinds of situational score can be discerned:

1) ***Rule-Based Scores***: such scores best serve context-oriented musicking that does not assume any inherent temporal dramaturgy. Musicians have memorized a database of rules and sub-compositions, together with instructions about their appropriate contextual use. These are the oldest variety of situational scores, used for example in Indian raags or Arabian maqams, but also e.g. in John Zorn's game-pieces.[10],[23] Their use is most prevalent within oral musicking traditions, for obvious reasons. As the richness and flexibility (i.e. the sensitivity to the sonic and aesthetic situation) of the music grows with the number of specific contextual rules, the number of possible relationships between these rules grows exponentially – as does the time needed to not only learn them, but also to understand how and when each new rule can be artistically and appropriately applied.
2) ***Reactional Scores***: For the purposes of this paper, a reactional score is defined as a score that displays score information based on underlying processes, e.g. algorithms or data mappings (e.g. the current weather conditions), in a manner that cannot be influenced (nor studied beforehand) by the player. The player thus plays the score largely 'prima vista' and must always react to new input. Most animated scores fall into this category. [17]
3) ***Interactive Scores***, then, are similar to reactional scores, with the decisive difference that either intentional input by the performer reading the score (buttons, switches, pedals etc.) or the music played by this performer or, even non-intentional information lifted from the performer (i.e. eye movements or electrical skin activity) is allowed to be a factor in the generation of the displayed score.[1], [5],[22]
4) ***Locative Scores***: The previous score types assume nothing about the actual performance situation, the musician's body and its relationship to other people, the space s/he plays in etc. The spatial relationship between musician and score is conceived as being purely functional. Indeed, most musicians would probably claim that it has no aesthetic or artistic significance in the context of their performance.

   This kind of abstraction from the performance context is impossible to maintain when the score is locative. Locative scores distribute score information in actual or virtual space: the musician thus moves within the information display, accessing the information available at a certain location. They thus add an aesthetic dimension to spatial musicking: sound production and meaning in locative scores arises not only from the decision when to play a sound, but also from the decision where to play it.

The paradigmatic interface for situational scores, too, seems to be the visual score, at least in the last three categories. Writing surfaces and/or display screens dominate the practice of scored music. [6], [12],[16],

The drawbacks of visual displays for musicians wandering through a space is obvious: they cannot see - not only where they will set their feet, but also what goes on around them. Indeed, the advent of screen displays has served to capture the musicians' gaze more intensely than ever before. Whereas a written score always allows the performer some leeway, most reactional and interactive visual scores want the musician's eyes to be on them every split-second. It seems reasonable to assume that the necessity for such intense visual attention distracts the musician from the sounds s/he is shaping.

This consideration also is the main reason why we, after some discussion, decided to not pursue visual head displays (i.e. augmented reality scores) as a viable interface for walking musicians. Instead, we opted to develop a body-suit as our score interface for situational scores - hoping that it would allow for more intuitive and centered musicking. [7], [9], [11],

3. THE body:suit:score

The project we designed has three main stages, spread over 48 months.

Stage 1 [*monody*]: testing and design for a single suit with only vibrotactile elements; composition of min. two "etude compositions" for this solo performer with this suit.

Stage 2 [*counterpoint*]: equipping the suit with various kinds of sensors, two etudes for two or three intercommunicating musicians
Stage 3 [*multiplicity*]: designing bodysize- and instrument-adaptable bodysuits that can be manufactured in small quantities. 2 Compositions for an ensemble equipped with bodysuits.

At the point of writing, 15 months into the project, we have completed Stage 1, and have embarked on Stage 2.

3.1. Ergonomics

Disturbing the musician's reflexes and concentration is a major concern with vibrotactile elements. [19] Great care was taken to not place elements near or in performance-sensitive areas (these obviously vary for each instrument). [15] Detailed experiments determined basic data sets such as body resolution (how near can two elements be placed while still being perceived as discrete? One answer: closer on the arm, wider apart on the back) [18], body image (where can sets of elements be perceived as one coherent group?), and, of course, the influence of vibrotactile intensity ('dynamic variation') on the perception of the elements.

3.2. Intuitive Or Symbolic

In our discussions about the musical functionality of the bodysuit interface, two schools of thought emerged: the bodysuit as a kind of vibrotactile 'screen' with dense placement of elements that can produce intuitive seamless sensations - or the bodysuit as a message interface with sparse element placement that can signal symbolic content in great clarity. For reasons discussed below, we chose to not decide between these two approaches at this early stage. The first suit prototype offered characteristics of both: while a back interface was entrusted with 'symbolic' messages, leg interfaces displayed more 'intuitive' informations.

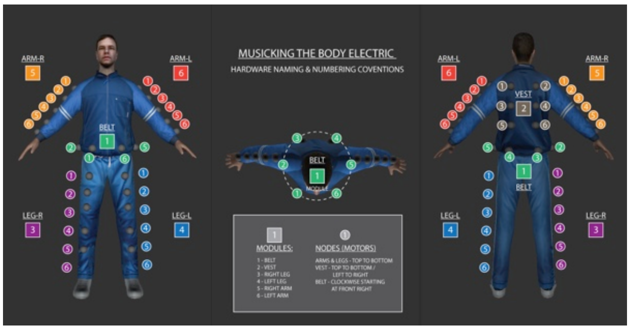


Figure 2 Distribution of vibrotactile elements on the body, with body zones differentiated by colours (see section 5.2.)

3.3. Look

As everything perceived during a performance contributes to its aesthetic meaning, especially when it is deemed to deviate from convention, we were conscious of the fact that a performance using heavily technological bodysuits could evoke all kinds of cultural references, from movie cyborg depictions to the body-alterations of Stelarc. Such (sub)-cultural connotations, while sometimes helpful for a stage director, can also be artistically annoying to those whose medium is sound.

From the the start, therefore, we aimed at integrating the technological elements of the suit (motherboards, vibrotactile elements, cables etc.) into the textile design – for example, all vibrotactile elements were sewn into the suit, connections inside the suit were stitched – and look like embroidery.

Figure 3 The back zone and the belt zone. The vibrotactile elements lie underneath the area between two connector endings emanating from each of the central boards.

The resulting suit prototype largely resembles normal concert attire. This 'neutral artist' look will permit composers, stage directors and musicians from other traditions or genres to add a costume layer suited to their artistic message or stage convention, while also enabling concerts where music is expected to be the only focus.

4. SCORE INFORMATION

The information displayed by a situational score interface can be of three basic types: analog, symbolic, and relational.

Analog score information is iconic (or sometimes indexical): it mimics (or echoes) the type of sonic performance it refers to. Some sonic parameters are best accessible through analog information: pulse (speed), dynamics (intensity), timbral evolution, even sometimes even pitch (especially with non-common uses of microtonality) etc. In conventional written paper scores, such analog informations are often represented by icons that extend over several notes, such as crescendi or slurs - but the structural limitation of written scores for the display of analog information has always been a major motivation for research into animated and interactive scores.

Most of the other information in a notated score is ***symbolic*** - signs by virtue of convention. Symbols are the main elements of the written paper score, as well as a major area in mid- to late 20th century score research, covering both extended instrumental techniques and extended scoring techniques.[20]

The third category, ***relational*** score information, has, despite many different attempts, [1],[3],[6] so far not been systematically explored or codified in written and visual scoring. The 'relations' referred to are those between different streams of musicking, between musicians. Relational score information is implicit in every ensemble arrangement, in every social setting involving music, and even in how musicians and sound sources (or scores) are placed on a stage or non-stage. Largely because they usually deal with conventional and mostly static arrangements, written scores have only rarely integrated such relational information. However, given the *raison d'être* for the body-suit-score interface – i.e. musicians moving while playing - such information becomes a vital, aesthetically highly relevant parameter: "Who do you play with?", "Whereto do you direct your playing?", "What/who do you listen to?", "Whence do you get your next cue?" etc.

Finally, locative information also is relational: by letting the musicians experience where they are in the room, where they are in spatial or musical relation to the other musicians, by augmenting certain physical locations with embedded score information, the locative score affords the performers many additional types of insight into how ensemble playing can become aesthetically relevant - beyond the purely sonic.

We decided fairly soon that we needed the body:suit:score to be able to display and transmit all three types of score information. At Stage 1 of the project, relational information still was largely unexplorable, as we only worked with stationary solo performers, but the other two were exhaustively tested.

4.1. Tactons

One consideration in every new score display or score design is the learning curve for the musicians. Analog information is fairly easy to absorb and follow, whereas new symbols must painstakingly be learned. At a later stage, we plan to develop game-like learning software for the musicians, replacing a score manual with interactive learning processes.

At this stage, however, we debated how we could at all create meaningful and easily retainable symbols for the bodysuit interface. [13] One member of the team, Marcello Giordano, already had - in another project with vibrotactile displays - developed a type of higher-order signal patterns he had named 'tactons' (in analogy to the word 'icons').[7]

'Tactons' combine semiotic properties of both symbol and icon. In a tacton, a few vibrotactile elements are arranged into a short 'firing' sequence which typically is repeated a few times: A tacton behaves more like an animated .gif than like a still image. Moreover, such short sequences can also be 'phrased' in ways that musicians already are familiar with: *staccato, tenuto* and *legato* in precise arrangements.

Tactons thus can carry a modicum of analog information - and this fact can be exploited to make a new vibrotactile symbol both easier to learn and easier to recognize in performance. Tactons also allow us to reduce the number of vibrotactile elements needed. Their potential for versatile recombination of few elements allows us to approach tacton creation and tacton learning with high-level concepts that borrow from language: word formation, syntax, 'style'.

Such concepts have proven to be crucial to their utility in performance: not all mathematically possible combinations of vibrotactile elements become easily recognizable tactons – only those that 'make sense' to the player, i.e. those that seem well-defined, unique in relation to others and can be understood as icons for the information they carry. Thus a tacton encoding e.g. the information "jump to the next section" will be better retained and recognized if the firing sequence in a line of six elements is **1➔2➔6** (where traversing the physical distance between 2 and 6 will be perceived as a jump) rather than, say, **3➔2➔3**.

5. ETUDE COMPOSITIONS

At this point in time, the primary artistic question of any new score design or score interface must always be: Which music or kind of musicking could not be imagined, let alone be performed, without it? It seems to make no practical nor aesthetic sense to develop a new interface in order to perform existing music - or to perform music in a familiar way. To this end, the perspectives, needs and demands of composers and musicians should shape the design and evaluation of a new score interface.

In the body:suit:project, three embedded composers steer and influence the evolution of the interface. We chose to work with three very different composer-musicians to ensure a large variety of approaches towards musicking and composition, also to ensure that the resulting suit would not only serve musicking in one particular style, tradition or genre: Adam Basanta, coming from electroacoustics and installative art, approaches performers as sonic and installative elements in space; Julian Klein approaches musicians as if they were theatre actors and music as if it were their stage; and Sandeep Bhagwati represents both conventional written composition and inter-traditional practices of comprovisation where musicians are artistic interpreters of the score.

This variety is evident in the first etudes they conceived for this project. While Julian Klein imagined the bodysuit as a means of virtually representing the real body of the musician, and was interested in how manipulations of this represented body would influence the live improvisation by the musician, Sandeep Bhagwati composed game-like, ritualized conceptual music spaces: an improvising musician exploring them would be guided, challenged and conducted by the score, which in turn was jointly controlled by a de-centralized conducting team. Adam Basanta's etude was not realized due to other commitments, but it would have involved much less improvisation, using the score as a complex signaling device for composed sonic explorations of bass clarinet multiphonics.

5.1. Klein's Mannequin

Klein was interested in how people treat a person's represented body (in the form of a dressmaker's mannequin covered with pressure sensors and vibrotactile elements) if they are alone in a room, and can interact with it as they wish.

Figure 4 Sarah Albu performing Julian Klein's etude composition (Nov 4, 2015, Montréal)

Whatever they physically do to the doll elicits direct vibratory feedback, but is also transmitted to the singer in an adjacent concert room (Sarah Albu) who, in her own body:suit:score, feels an intuitive representation of this interaction. During rehearsals, Klein and Albu developed what he calls a "mise-en-musique": an aesthetic and behavioural stance enabling quasi intuitive musical reactions that shape her improvisatory response to the unforeseeable signals coming from the audience-manipulated mannequin. It is immediately obvious why such an idea would not be possible to realize with a visual score.

Figure 6 Felix Del Tredici performing Bhagwati's "Fragile Disequilibria" with Jen Reimer, Joseph Browne, Max Stein and Adam Basanta on iPad controllers (matralab Montréal, Nov 4, 2015)

5.2. Fragile Disequilibria (Bhagwati)

While Klein used the entire suit as one contiguous score surface, Bhagwati divided it into four distinct score zones: back, belt, left and right leg. Each of these zones controls another parameter of improvisatory musicking: timbre, dynamics, interval structure, musical lingo (divided into three main lingo groups: bird-like, machine-like, and fluid-like).

Each bodysuit zone is separately controlled by a 'audi-ductor' who, while listening to the performance, can issue change commands by sending specific tactons (The number of tactons used in any given piece is arbitrary. This score uses 16 tactons.)

A brokering software eliminates command overload to the performer by negotiating the current precedence of change commands. It also calculates overall commands such as tempo changes, silences and the end of the performance from the input by the four 'audi-ductors'.

The performer (trombonist Felix del Tredici) thus needs to navigate a landscape of precise musical commands. These are unforeseeable, but not random – after all, he can non-verbally communicate with the four audi-ductors, and they, too, are instructed to issue their change commands 'musically', i.e. as an artistic commentary or guidance.

The negotiations between the 5 musickers (one acoustic and 4 conceptual) are the aesthetic core of this piece – how they change and challenge the improviser to invent a music that fulfils continually changing layerings and combinations of the four parametric zones palpably shapes the "Fragile Disequilibria" of the title. The performer's audible but also visible mental juggling and his musical navigations could theoretically also be achieved via a visual screen score - but they would probably not offer the same intense concert experience for player and audience alike. As del Tredici described it once, "it feels different if the command seems to come from your own skin".

6. CONCLUSIONS

After a little more than one year of research and creation with and around the body:suit:score, several basic, but crucial problem zones around the representation of score information have successfully been addressed: 1) skin resolution for vibrotactile sensors; 2) a good understanding of instrument-specific performance-sensitive zones; 3) a basic prototype suit, tested in performance: both the distribution of technological elements and the necessary properties and constraints for materials and costume design have become clear; 4) various aesthetic approaches and three etude-compositions for the score have prompted a versatile and stylistically agnostic approach to our suit interface design. 5) artistic feedback from both musicians and audiences at workshop and conference performances largely encouraging.

The next years will see further developments as outlined above: while the basic functionality and a promising artistic uses have been established and successfully tested, the next steps involving contrapuntal interactions between multiple players and the technological and conceptual integration of sensors into the suit will pose a new category of research-creation challenges. While reliable wireless communication remains one of the major technical challenges, the new streams of sensor data emanating from the performers will pose new challenges to the composers: the two principal questions of all real-time data analyses, namely pattern recognition and pattern correlation must be addressed in a poetical manner.

REFERENCES

[1] Alcorn, Michael, and Christopher McClelland. “Exploring New Composer/Performer Interactions Using Real-Time Notation.” *ICMC 2008.* Belfast: Sonic Arts Research Center, 2008. n.a.

[2] Berzowska, Joanna, Alexandra Bachmayer, and Sandeep Bhagwati. "Electronic Garments at the Periphery of Awareness." *(unpublished).* 2015.

[3] Bhagwati, Sandeep. "Comprovisation – Concepts and Techniques." In *(Re)Thinking Improvisation*, by Henrik Frisk and Stefan Östersjö. Malmö: Lund University Press, 2013.

[4] Bhagwati, Sandeep. "Notational Perspective and Comprovisation." In *Sound & Score. Essays on Sound, Score and Notation*, by Paulo de Assis, William Brooks and Kathleen Coessens. Leuwen: Leuwen University Press.

[5] Bhagwati, Sandeep. “Towards Interactive Onscreen Notations for Comprovisation in Large Ensembles.” Edited by George E. Lasker, M. Ana Luz and John Dack. *Systems Research in the Arts and Humanities.* Windsor, ON, Ca: International Institute for Advanced Studies in Systems Research and Cybernetics, 2008. 43-49.

[6] Freeman, Jason, and Andrew Colella. “Tools for Real-Time Music Notation.” *Contemporary Music Review* (Routledge) 29, no. 1 (09 2010): 101-113.

[7] Giordano, Marcello, Ian Hattwick, Ivan Franco, Deborah Egloff, Christopher Salter, and Marcelo Wanderley. ""Design and Implementation of a Whole-Body Haptic Suit for “Ilinx”, a Multisensory Art Installation"." *Proceedings of the 12th Sound and Music Computing Conference.* Maynooth: Maynooth University, 2015.

[8] Gunther, Eric, and Sile O'Modhrain. “Cutaneous Grooves: Composing for the Sense of Touch.” *Journal of New Music Research* (Routledge) 32, no. 4 (2003): 369-381.

[9] Hattwick, Ian, et al. "Compositional Approaches to the Design and Use of A Vibrotactile Garment." *Proceedings of the International Computer Music Conference.* Denton: ICMC, 2015.

[10] Hisama, Ellie M. “John Zorn and the Postmodern Condition.” In *Locating East Asia in Western Art Music*, edited by Yayoi Uno Everett?Frederick Lau, 72-84. Middletown, 2004.

[11] Karam, Maria, Gabe Nespoli, Frank A. Russo, and Deborah I. Fels. "Modelling perceptual elements of music in a vibrotactile display for deaf users: A field study." *Proceedings of the 2nd International Conference on Advances in Computer Human Interaction,.* Cancun: ACHI, 2009. 249–254.

[12] Karkoschka, Erhard. *Das Schriftbild der neuen Musik.* Celle: Moeck, 1965.

[13] Lee, Jaebong, Jonghyun Ryu, and Seungmoon Choi. “Vibrotactile score: A score metaphor for designing vibrotactile patterns.” *Proceedings of the World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.* Washington DC: IEEE Computer Society, 2009. 302-307 .

[14] Litke, David, and Keith Hamel. “A score-based interface for interactive computer music.” *Proceedings of the 2007 International Computer Music Conference.* Heidelberg: Springer, 2007.

[15] Miyashita, Homei, and Kazushi Nichimoto. “Thermoscore: A new-Type Musical Score with Temperature Sensation.” *Proceedings of the 2004 Conference on New interfaces for Musical Expression (NIME04).* Hamamatsu, Japan, 2004.

[16] Sauer, Theresa. *Notations 21.* New York, 2009.

[17] Smith, Ryan Ross. *Animated Notation.* http://www.animatednotation.com/index.html (accessed 04 01, 2016).

[18] van Erp, J. "Vibrotactile spatial resolution on the torso : Effects of location and timing parameters." *Proceedings of First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference.* Pisa: IEEE, 2005. 80–85.

[19] Verplanck, William. “Haptic Music Exercises.” *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05).* Vancouver, 2005.

[20] Vickery, Lindsay. "The Evolution of Notational Innovations from the Mobile Score to the Screen Score." *Organised Sound* (Cambridge UP) 17, no. 02 (2012): 128-136.

[21] Waisvisz, Michel. “'Composing the now' On engagement with sonic time through sensors, electronica, loudspeakers and ears.” *crackle.org.* 2003. http://www.crackle.org/composingthenow.htm (accessed 2009-08-30).

[22] Winkler, Gerhard E. “The Real-Time Score. A Missing-Link In Computer-Music Performance.” *Proceedings of Sound and Music Computing 2004.* Paris: Ircam-Centre Pompidou, 2004.

[23] Zorn, John. “The Game Pieces.” In *Audio Culture: Readings in Modern Music*, edited by Christoph Cox and Daniel Warner, 196-200. New York: Continuum , 2006.

PROCESSING OF SYMBOLIC MUSIC NOTATION VIA MULTIMODAL PERFORMANCE DATA: BRIAN FERNEYHOUGH'S *LEMMA-ICON-EPIGRAM* FOR SOLO PIANO, PHASE 1

Pavlos Antoniadis

GREAM, Université de Strasbourg /
STMS Ircam-CNRS UPMC
katapataptwsi@yahoo.gr

Frédéric Bevilacqua

STMS Ircam-CNRS UPMC
frederic.bevilacqua@ircam.fr

ABSTRACT

In the "Performance Notes" to his formidable solo piano work *Lemma-Icon-Epigram*, British composer Brian Ferneyhough proposes a top-down learning strategy. Its first phase would consist in an "overview of gestural patterning", before delving into the notorious rhythmic intricacies of this most complex notation. In the current paper, we propose a methodology for inferring such patterning from multimodal performance data. In particular, we have a) conducted qualitative analysis of the correlations between the performance data (an audio recording, 12-axis acceleration and gyroscope signals captured by inertial sensors, kinect video and MIDI) and the implicit annotation of pitch during a sight-reading performance; b) observed and documented the correspondence between patterns in the gestural signals and patterns in the score annotations and c) produced joint tablature-like representations, which inscribe the gestural patterning back into the notation, while reducing the pitch material by 70-80% of the original. In addition, we have incorporated this representation in videos and interactive multimodal tablatures via the use of INScore. Our work draws from recent studies in the fields of gesture modelling and interaction. It extends the authors' previous work on an embodied model of navigation of complex notation and on an application for offline and real-time gestural control of complex notation by the name GesTCom (Gesture Cutting through Textual Complexity). Future prospects include the probabilistic modelling of gesture-to-notation mappings, towards the design of interactive systems which learn along with the performer while cutting through textual complexity.

1. INTRODUCTION

In the "Performance Notes" of the published musical score of *Lemma-Icon-Epigram* for solo piano, Brian Ferneyhough states:

"An adequate interpretation of this work presupposes three distinct learning processes: (1) an overview of the (deliberately relatively direct) gestural patterning wthout regard to exactitude of detail in respect of rhythm; (2) a 'de-learning' in which the global structures are abandoned in favour of a concentration upon the rhythmic and expressive import of each individual note (as if the composition were an example of 'punctualistic' music); (3) the progressive reconstruction of the various gestural units established at the outset on the basis of experience gained during the above two stages of preparation" [1].

The proposed top-down approach to learning is neither unique to this particular work, nor uncommon in similar repertoire: Both the composer's earlier remarks in his *Collected Writings* [2] concerning prioritisation in learning, as well as reports in [3], [4] of performers specialising in complex contemporary piano music, privilege a pragmatic grasping of global structures of the work in the beginning of the learning trajectory, before navigating the fine detail and stratifying it in a personalised manner.

Setting aside for the moment the question of whether Ferneyhough's "gestural patterning" refers to physical or musical gestures, we make two hypotheses:
a) That his tripartite learning scheme can be externalised and represented as a processing of the symbolic notation on the basis of and by means of mutimodal data and their correlations. This hypothesis is based on findings in the field of *embodied and extended cognition* [5], [6].
b) That the representation of pitch information in the first phase of learning can be modelled in relation to the horizontal movement of the hands along the keyboard space and particularly correlated to gestural signals captured by inertial sensors.

For the rest of this paper, we will review relevant work in the fields of gesture modelling and interaction; we will present our methodology and findings; we will propose derivative representations and interactive tablatures, as well as future prospects for the probabilistic modelling of gesture-to-notation mappings.

2. RELATED WORK

Our work on the creation of gesture-to-notation mappings and interactive systems derives at large from previous research on gesture modelling and gesture-to-sound mappings employing machine learning techniques. Bevilacqua *et al*. proposed in [7] a Hidden Markov Models (HMM) methodology defined as *gesture following*: Incoming gestural features, modelled as multidimensional temporal profiles, are compared in real time to templates stored during a *learning phase.* This is the first step towards implicit or explicit mappings to sound, during a subsequent phase of *following*. Caramiaux has further proposed in [8] a segmental approach to this HMM methodology for the segmentation and parsing of clarinetist ancillary gestures. In this instance, gestural features are considered as temporal and gestural segments, compared to dictionaries of *primitive shapes*, constituting prior knowledge and opening-up the way for higher-order, syntax-like modelling. Françoise has adressed the problem of temporal multidimensionality and computational limitations of the previous models through the employment of Hierarchical HMM and Dynamic Bayesian Networks [9], while addressing also multimodal modelling (simultaneous modelling of movement and sound as opposed to modelling of movement alone) and Mapping-by-Demonstration (MbD) techniques [10] (whereby the end-user controls the process of machine learning interactively). He has also proposed a lower-order syntactical paradigm for gesture-to-sound mapping: a "gesture envelope" of Preparation-Attack-Sustain-Release (referred to as PASR from now on, after the classic ADSR sound envelope paradigm). [11]

Basic ideas from this corpus of work that proved influential, as shown in detail later, are: a) Template alignment (that is alignment between a stored template / dictionary of primitive shapes and an incoming dataflow): In our case, as will be explicated in 3.4, implicit annotation constitutes the template to which gestural features are compared; b) low- and high-order segmentation and syntax (from a PASR model to Attack-Displacement envelopes and to the gradual reduction of pitch material in 3.3 and 3.5 respectively); c) performance-oriented learning, as influenced by MbD; and d) hierarchical and segmental layering, evident in the concept of "embodied layers" (3.2).

These models are currently being employed in a variety of applications, including the performing arts, audio industry, sound design, gaming and rehabilitation with auditory feedback. For an overview of those, please visit http://ismm.ircam.fr/.A notable application was the "augmented violin" project [12], where those models were employed in conjuction with composed music and notation. Nevertheless, many more studies are required to fully understand how a musician's movement can be modelled in a learning situation, as well as the complex relationships between gesture and notations.

Exhibiting the potential for gesture-to-notation mappings, following up from the paradigms of gesture-to-sound ones exhibited above, is one of the objectives of this paper. The other objective is to lay the foundation for the probabilistic modelling of notation according to gesture.

3. METHODOLOGY

Our methodology for the current study of the correlation between multimodal performance data and an implicit annotation of the score of *Lemma-Icon-Epigram* can be summarised as follows:

1. Sight-reading performance of the first page of *Lemma-Icon-Epigram* (fig.2) and recording of multimodal data {audio, 12-axis gestural signals, kinect video, MIDI} as in fig. 3.
2. Representation of the implicit performative annotation of symbolic notation during the sight-reading performance: Embodied layers {fingers, grasps, arms} as in fig.4a. This representation constitutes prior knowledge.
3. Comparison of recorded gestural signals to the recorded audio and video and annotation accordingly, as in fig. 5. At a later stage, information is extracted from the gestural signals alone (and just confirmed from the video and audio).
4. Comparison of the annotation of gestural signals in 3. to the implicit annotation in 2., as transferred in the MIDI piano-roll: fig. 6.
5. Return to the symbolic notation: Transcription of the MIDI piano-roll representation into a reduced proportional representation of pitch in space: fig.7. Annotation of fig.7 according to the annotation of the gestural signals: Gradual reduction of the amount of pitch information and inscription of gestural patterning as in fig. 8.
6. Comparison of 5. to the original symbolic notation: fig. 9.

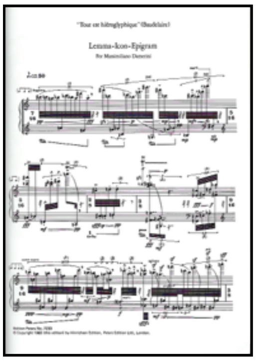



Figure 2.
Brian Ferneyhough, *Lemma-Icon-Epigram*, p.1, original score. Reproduced with kind permission by Peters Edition.

The block diagram in fig.1 presents this methodology. A purple horizontal line represents the transparent border between the traditional approach to learning and its extension into our current approach via the use of recorded multimodal data. We remind that both strategies are tailored after Ferneyhough's top-down learning model and refer to the first phase of "global gestural patterning".
Let us now elaborate on each of these steps.

3.1 Sight-reading and recording of multimodal data

The term 'sight-reading' should not be confused with the literal use of the term, as in the classical music world - especially in the fields of opera coaching or chamber music, whereby training and ability ensure a sufficiently satisfying performance of all notated parameters without prior knowledge of the text. In our case, 'sight-reading' signifies a performance in the beginning of the learning trajectory, which prioritises an "overview of gestural

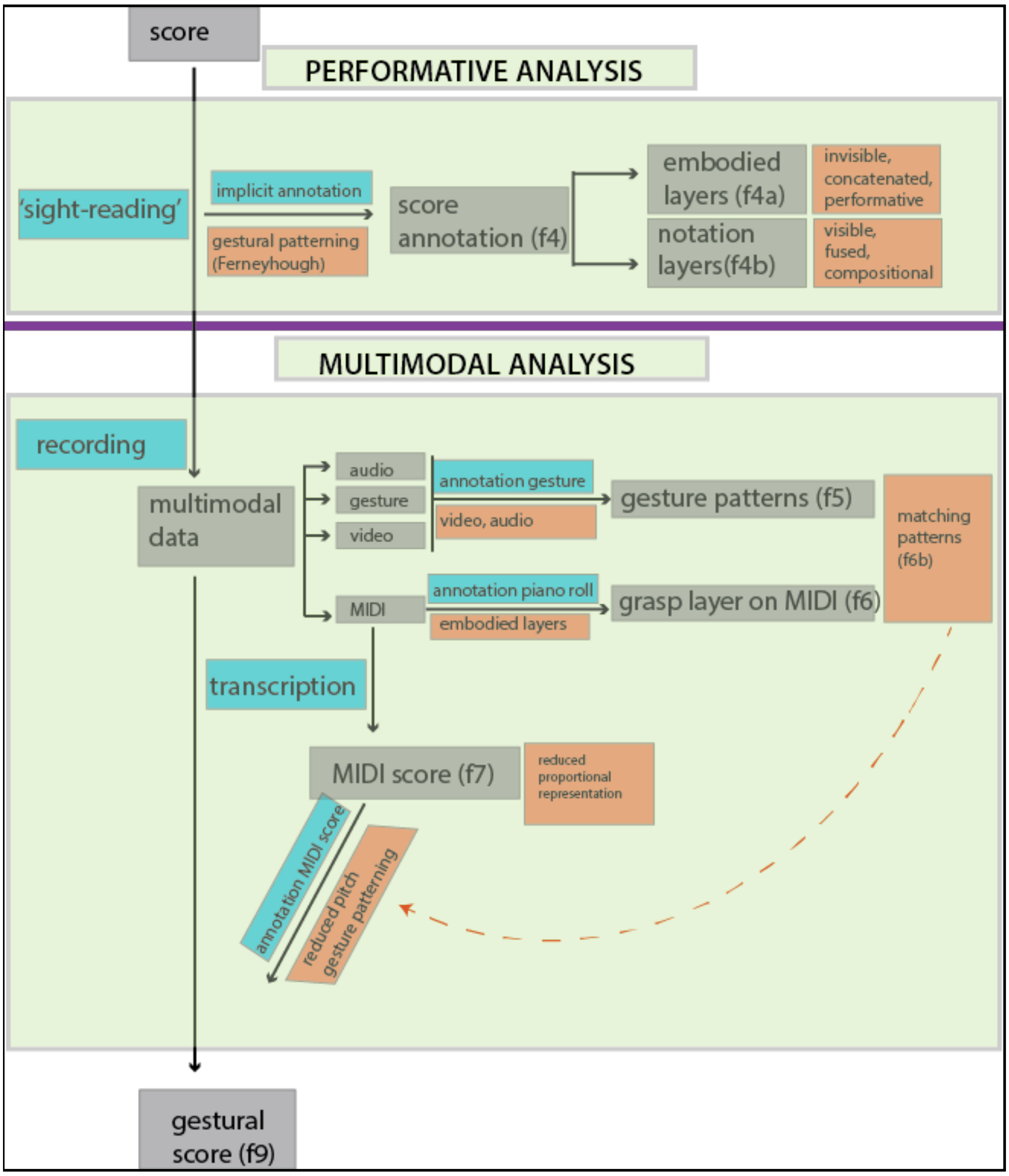


Figure 1. Methodology for the processing of symbolic notation through performance data. The input is the original score, output is a "gestural score".

patterning" (Ferneyhough) rather than precise rhythmic and other detail. Furthermore, as already stated in the Introduction, we hypothesise that the sight-reading equals an implicit annotation of the musical score, representable as explicit annotations detailed in 3.2.

The first author's performance for this case study took place on 18.04.2014 in the context of his Musical Research Residency at IRCAM. He performed and recorded three takes for each page of *Lemma-Icon-Epigram* in one day. His sight-reading prioritised ergonomic hand and arm movement in the keyboard space as well as pitch accuracy, while allowing only sporadic and spontaneous response to the parameters of rhythm, articulation and dynamics. A Yamaha upright Disklavier was used for the recording of MIDI information, while multimodal information included audio captured by two microphones, video captured by Kinect and movement data captured by 3D accelerometers and 3-axis gyroscopes worn on both wrists. Fig. 2 shows the first page of the original score and fig. 3 the MAX/MSP patch used for the synchronisation of the data.

3.2 Representation of implicit annotation

Given the ambiguity of the term *gesture* in musical contexts, referring to both musical and physical, compositional and performative properties, the annotation may include two types of information:

- Notated "gestural patterning" elements such as pitch, articulation, rests, dynamics, pedal, beaming (fig. 4b). This information is visible, but also heterogeneous, multi-layered and fused. As an example, we have here only indicated the most salient gesture boundaries, reserving the rest of the gesture patterning elements for

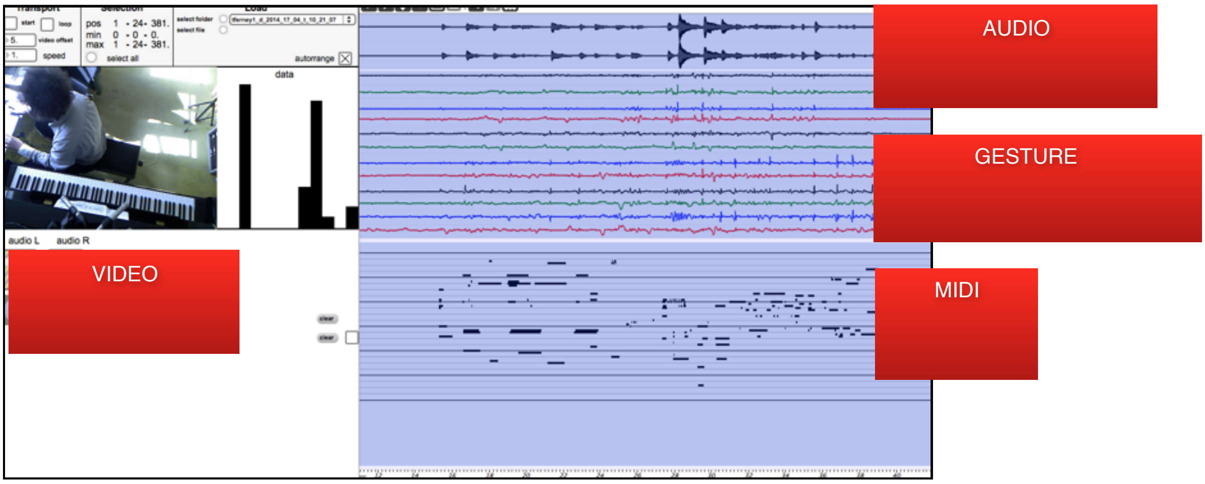


Figure 3. MAX/MSP patch for the synchronisation of multimodal data, created by ISMM team / IRCAM

the second phase of refinement in the learning process.

- Physical gestural elements, such as fingerings, changes of hand position, arm movements, technical patterns. This sort of information is invisible, concatenated and embodied: it constitutes a *hidden layer* of the notation, albeit representable as in fig. 4a. In previous work [13] we have suggested a typology of physical gestural elements in relation to pitch, following up from ideas by the pianist György Sándor [14]. We have proposed a hierarchical ordering of notated pitch information in three embodied layers: fingers, hand-grasps and arm movements. The *finger layer* corresponds to traditional fingering and includes all notated pitch indexed with a number from one to five. Hand-grasps are by default defined as concatenations of pitch contained between fingers one and five. Depending on individual hand span, those pitch sets can be played simultaneously as chords or in succession as melody, potentially involving upper-arm participation and horizontal displacement. Consequently, the *grasp layer* can be effectively represented by the pitches assigned to figers one and five, omitting the pitches corresponding to inner fingers. Similarly, hand displacement takes us to the *arm layer*, which can be defined as a concatenation of grasps. Its boundaries are defined by the succession of fingers one and five (in the case of outwards movements, that is displacement from the centre to the extremes of the keyboard for both hands) or by the succession of fingers five and one for movements from the extreme to the centre. As a result, the trajectories of hand transpositions or *arm layer* can be defined as a series of segments defined by digits one and five, depending on their directionality.

Please note that both the grasp and the arm layers may be defined as a succesion of two-bit units of information: pairs of fingers one and five. Also: The segmental and hierarchical nature of those layers point directly to the gesture probabilistic models reviewed in section 2.

In fig. 4a the grasp layer is represented for both hands in the form of blue ellipses. There are no hand crossings, thus we keep the same colour for both hands. The highlighted noteheads indicate grasp boundaries: red noteheads are employed for finger 5 and blue ones for finger 1 in both hands.

3.3 Comparison of gestural signals to recorded audio and video

The qualitative analysis of the multimodal data followed two phases: First, we observed the 12-axis gestural signals in relation to the audio signals and the kinect video. The results of our observations for page 1 are presented in fig. 5 and are detailed as follows.

- Accelerations related to attacks (clearly visible as amplitude peaks in the audio signals) are unequivocally discernible from accelerations related to the horizontal displacement of the hands. The first are marked with red ellipses, the latter with blue ellipses in the gestural signals of fig.5. Attack accelerations appear as instantaneous high amplitude peaks of the accelerometers and often the gyroscopes signals, while displacement accelerations are mainly captured by the gyroscopes as low amplitude and frequency peaks. Close comparison to the video reveals patterns related to the direction of the displacement, clearly marked also in fig. 5 ("values reversed").
- Next to those two distinct types of events, attacks and displacements, we discern two hybrid events: trills (excitation visible in all six axis of the signal) and displacement with simultaneous attacks. Those events are more complex and more equally distributed between the accelaromeers and the gyroscopes, and are indicated with purple ellipses.
- Observation of the sequence of the above-mentioned four types of events reveals two types of patterns: i) attacks / trills followed by displacements / displacements & attacks and ii) succession of attacks without intermediate displacements. The pattern ii) indicates that the events take place inside the boundaries of a single hand-grasp, while the pattern i) indicates changes of hand position and thus moving on to the arm layer.

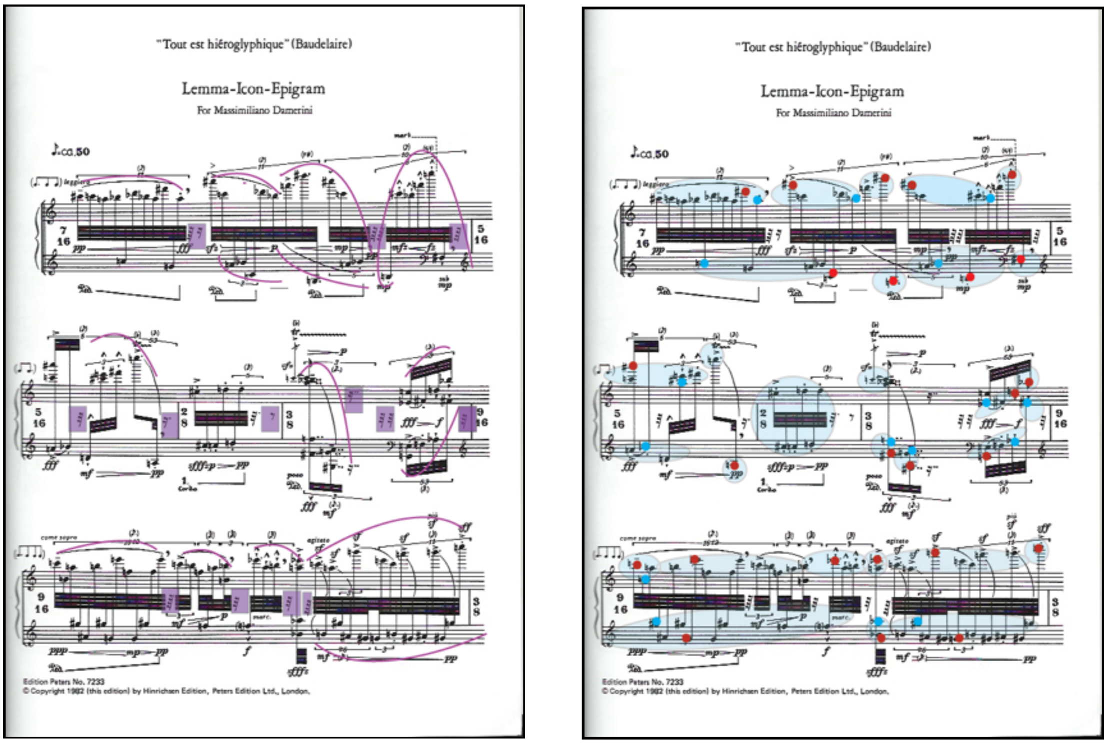


Figure 4. Gestural patterning after articulation and rests (left, 4b); grasp layer indicated by blue ellipses for both hands, boundaries indicated with red blobs for fingers 5 and blue blobs for fingers 1 (right, 4a)

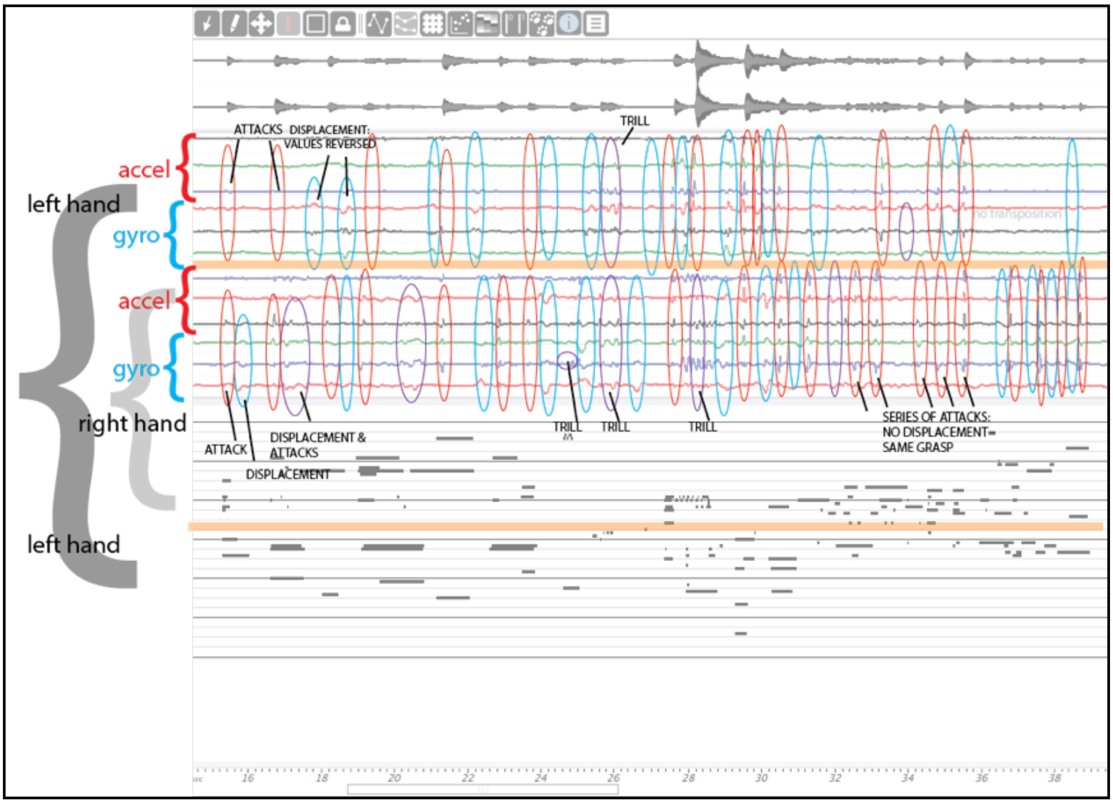


Figure 5. Annotation of 12-axis gestural signals according to video and audio

In short, the gestural signals offer us information about: The horizontal displacement of the hand (or not), its direction, its intensity and the possible presence of intermediate attacks. Higher-order segmentation and parsing will become clear in 3.5.

An interesting finding in the course of this annotation process was the gradual elimination of the need to confirm the information conveyed by the gestural signals through video and audio, the implications of which will be exposed later.

3.4 Comparison of annotated MIDI to annotated gesture

In the next phase, we transferred the implicit annotation of the score as in fig.4a to the MIDI piano-roll representation of our recording patch and compared it to the annotation of the gestural signals as in fig. 5. Our comparative study reveals an one-to-one correspondance between the two annotations: Attack gestures align perfectly with grasps and displacement gestures align with changes of position. The correspondence becomes clear in the matching patterns of blue arrows in fig. 6.The significance of this alignment is that the pianist's implicit knowledge is reflected in the objective gestural, audio and video data. The implication of this alignment is that the gestural data can be used for the modelling of incoming MIDI pitch information, without the need for implicit knowledge.

3.5 Return to symbolic notation

The next step was the automatic transcription of the MIDI piano-roll in symbolic notation, aiming at a new output score describing gesture. For this purpose, we used specially designed command-line tools developed by Dominique Fober and based on the Guido Engine[1].The result is a reduced proportional representation of pitch in space as in fig. 7.

Further on, this representation is gradually annotated after the annotation of the gestural signals, in the form of a gradual reduction of the pitch material according to embodied layers' boundaries, that is fingers one and five, as follows in fig. 8a, b, c.

By keeping only the grasp boundaries (2-bit definition of grasp), we get a reduction in the amount of pitch as in fig. 8b.

The leap to the *arm layer*, defined as concatenation of grasps in a certain direction of movement, allows for a further elimination of one of the two grasp boundaries, depending on the direction of the movement. Grasps now are defined by only one note (upper note for movements outward, lower note for movements inward) and the patterns of hand transposition have an one-to-one correspondence to the gestural signals. The amount of pitch is further reduced.

A final reduction of the pitch information is possible, if we consider only the peaks of the arm trajectories, that is the boundaries of the horizontal arm movement. This representation does not fully coincide with the gestural signal, but can become visible at a high speed play-back of the video.This representation corresponds to an exact 20% of the initial pitch content in fig. 7.

Eventually, the segmentation and parsing of gestures in higher-order syntactic units is possible as shown in Fig. 8e.

Interestingly enough, as shown in fig. 8, from the gestural signals' annotations and given a MIDI score we can infer a) the reduced amount of pitch material needed to describe gesture and b) the fingering of it. A consistent mapping between gestural signals and embodied MIDI representations is possible. Such a mapping would reduce the amount of pitch information by 70-80% for the first stage of the learning process.

3.6 Comparison of reduced pitch representation to the original score

A comparison of the latter reduced proportional representation (fig. 8c) of the grasp layer of the original score to the original (fig. 2) yields the following observations, as presented in fig. 9:

- Information concerning rhythm, articulation, dynamics, pedaling and expression has been removed. Our attempt is to relieve the fusion of those parameters in notation, searching to represent Ferneyhough's proposed "gestural patterning" in the first phase of the learning process only in terms of horizontal displacement of hands over the keyboard and pitch reduction to the boundaries of this gesture.We present pitch information which is definitive for the horizontal displacement of the hands. We have showed that this information constitutes implicit knowledge for the performer, but it may also be inferred from the gestural signals alone.

- Pitch information is re-arranged as follows: It is renotated in four staves instead of the original two. This representation of pitch-space in a continuum, i.e higher and lower pitch visible as such in the notation, differs from the original, where clef changes, ledger lines and additional octave displacement brackets often conceal the distribution of pitch in the notational space.
  - It is reduced in only the amount of pitch which is necessary for the representation of the hand displacement. This amounts to 20% of the original pitch content in this particular instance.
  - Blue arrows indicate change of position in full accordance to the gestural signal.
  - Higher-order segmentation and parsing of the output score clarifies patterns which are not readily visible in the original score.

- Ontologically, the output score is generated from a MIDI stream during performance and offers augmented multimodal feedback to the performer during learning and performance. It reflects on performance at different temporal scales, in the sense of its past, present and future manifestations. The latter correspond to: prior knowledge and prioritisations (as the

[1] An open source rendering engine dedicated to symbolic music notation, see at http://guidolib.sf.net

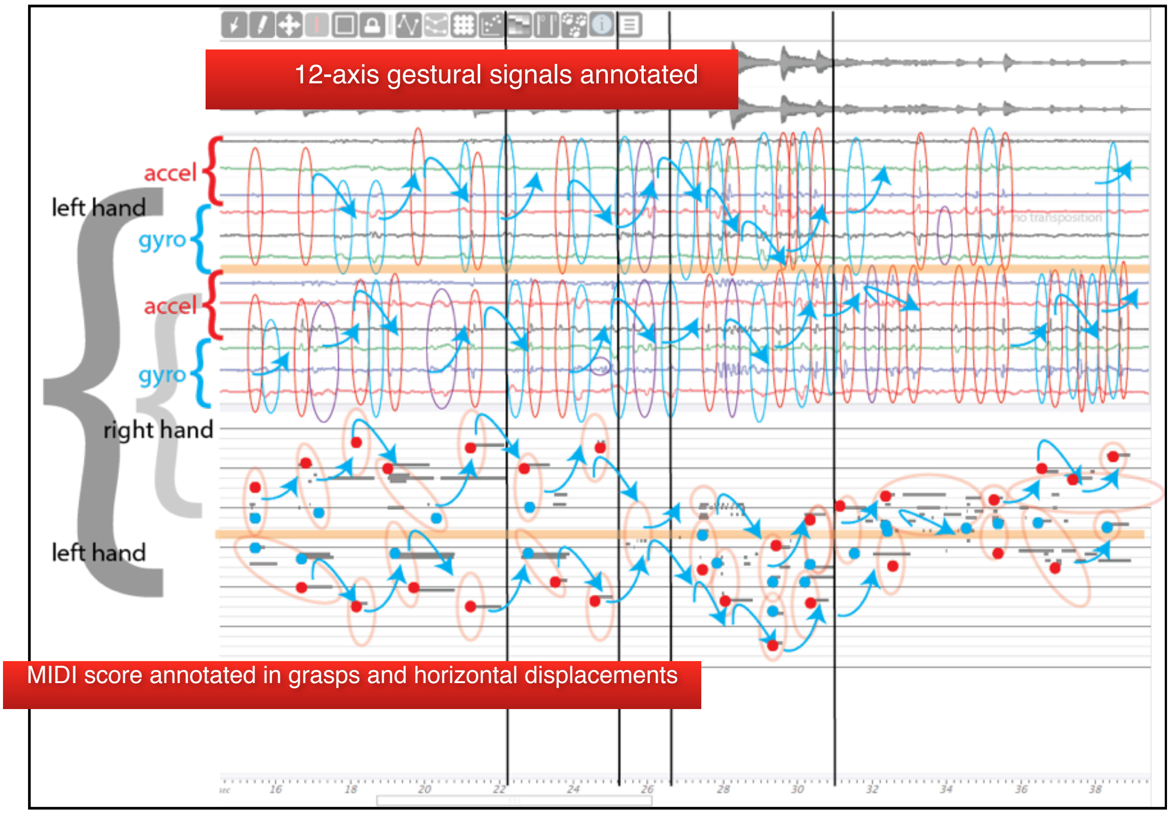


Figure 6. Comparison of two annotations: the annotation of the original score in grasps transposed on the MIDI score and the annotation of the gestural signal in attacks and displacements. Watch the matching blue arrowed patterns.

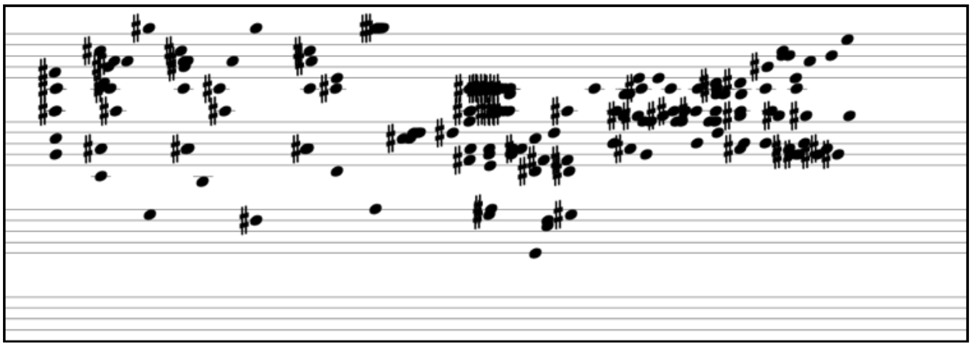



Figure 7. Automatic transcription of the MIDI piano-roll: Reduced proportional representation of pitch in four staves.

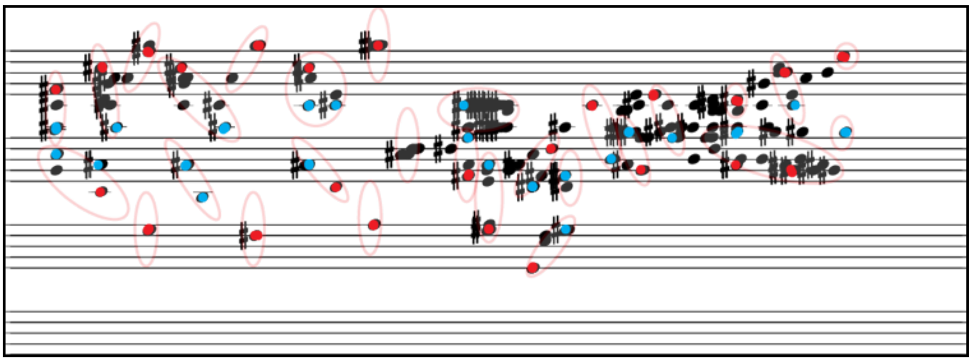


Figure 8a. Annotation of fig. 7 in grasps

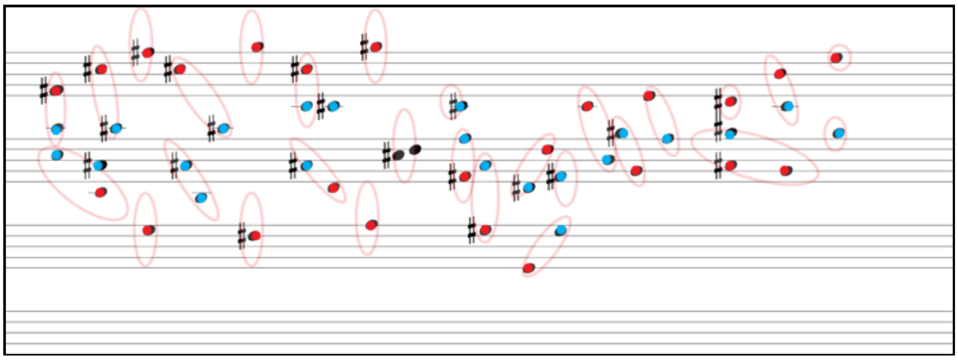


Figure 8b. Annotation of fig. 7 keeping only grasp boundaries

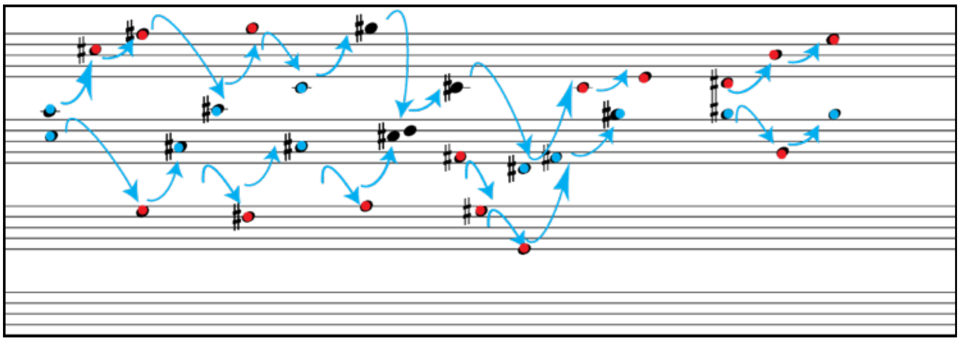


Figure 8c. Annotation of fig. 7 keeping one grasp boundary and indicating the gestural pattern

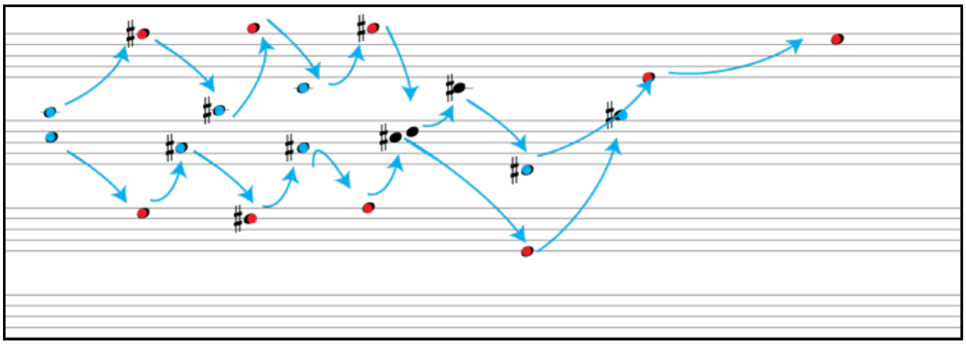


Figure 8d. Arm layer**:** Annotation of fig. 7 keeping one the maxima and minima of arm trajectories, without intermediate position changes

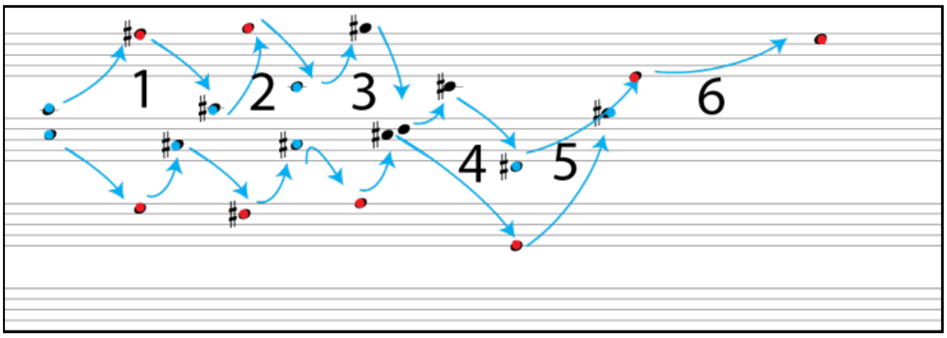


Figure 8e. Segmentation and parsing: The relationship of the arm movement is heterodirectional (opposite motion) in the first 3 symmetrical units and homodirectional (parallel motion) in the units 4 to 6.

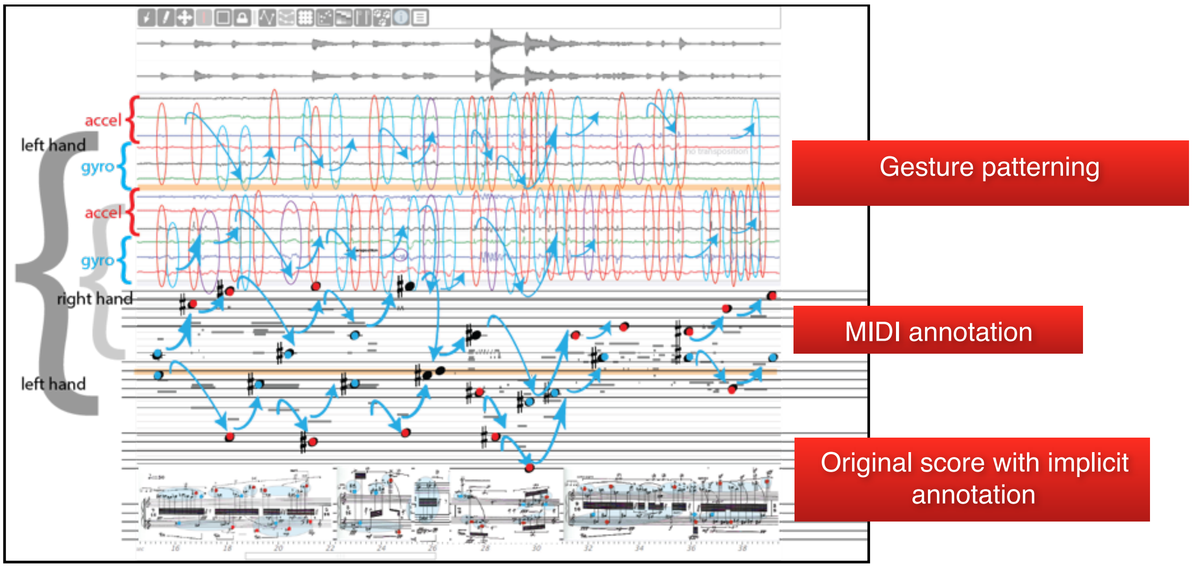


Figure 9. From top to bottom: The alignment of gesture patterning, MIDI patterning with pitch reduction and the original score annotation.

implicit annotation); observed realisation (gestural patterning); anticipated further notational transformations (when the output score enters in the learning cycle and is itself being processed and refined during the second and third stages of learning).

- From an embodied cognition point of view, output notations are embodied and extended: They are produced through performative actions, they represent multimodal data, they can be interactively controlled through gesture and they can dynamically generate new varied performances. They can be considered as the visualisation and medial extension of the player's embodied navigation[2] in the score-space, creating an interactive feedback loop between learning and performance.

4. CURRENT APPLICATIONS

We currently use the output gesturally annotated score in synchronisation with videos and integrated in INScore [15] dynamic representations, to be presented in TENOR 2016. Following previous work on the GesTCom (*gesture cutting through textual complexity*) [16], a system combining the INScore and the *motionfollower* architecture, we plan to integrate and dynamically interact with the output representation of fig. 9 in real-time. For a review of GesTCom, please look at the video linked in [17].

5. FUTURE PERSPECTIVES

Future projections of this work include:

- The comparison of differently prioritised performances corresponding to the second and third phase of learning as defined by Ferneyhough.
- The probabilistic inference of the annotated MIDI score as a hidden layer emitting the gestural signal in a hierarchical Hidden Markov Model.
- Applications in learning and performance documentation, interaction design, score-following and pedagogy.

6. CONCLUSION

We have presented a methodology for the processing of complex piano notation by means of multimodal performance data. Our case study is *Lemma-Icon-Epigram* for solo piano by Brian Ferneyhough. Ferneyhough's notion of global gestural patterning manifests as subjective score annotation observable in objective performance data. We have employed this patterning in output embodied representations, which sample the original symbolic notation after the observed gestural patterning. This work is promising for the probabilistic inference of the patterning and the notation from multimodal data. Applications range from performance documentation and pedagogy to interactive systems design and score-following.

Acknowledgments

This work used recordings conducted in the context of the IRCAM Musical Research Residency 2014. It wouldn't have been possible without the doctoral funding of the LabEx GREAM, Université de Strasbourg and the IRCAM. Special acknowledgments to Dominique Fober for specially developing the command lines for the

[2]For a review of the concept of the embodied navigation of complex notation and its foundation on the field of embodied and extended cognition, please review [18].

automatic transcription of the MIDI piano-roll into reduced proportional representations.

7. REFERENCES

1. B. Ferneyhough, *Lemma-Icon-Epigram* for solo piano, "Performance Notes", Edition Peters No. 7233, 1982
2. B.Ferneyhough,"Aspects of Notational and Compositional Practice", in J. Boros & R. Toop (eds.) *Brian Ferneyhough-Collected Writings*, pp. 2-13, Routledge, 1995
3. S. Kanach (ed.), *Performing Xenakis,* pp. 65-129, in *The Iannis Xenakis Series* vol. 2. Hillsdale, Pendragon Press, 2010
4. I. Pace, "Making Possible the Irrational: Strategies and aesthetics in the music of Stockhausen, Cage, Ligeti, Xenakis, Ferneyhough, Barrett" in *Collected Writings of the Orpheus Institute*. Leuven: University Press, 2008
5. M.Leman, *Embodied Music Cognition and Mediation Technology*, MIT Press, 2008
6. M.Rowlands, *The New Science of the Mind: from extended mind to embodied phenomenology*, MIT Press, 2010
7. F. Bevilacqua, N.Schnell, N.Rasamimanana, B. Zamborlin, F. Guedy: "Online Gesture Analysis and Control of Audio Processing", in: J.Solis & K. Ng (eds.) *Musical Robots and Interactive Multimodal Systems*, pp. 127-142, Springer-Verlag, 2011
8. B.Caramiaux, M. Wanderley, F. Bevilacqua, "Segmenting and Parsing Instrumentalists' Gestures", in J. New Music Research vol. 41, no 1, pp. 13-29, 2012
9. J. Françoise, *Realtime Segmentation and Recognition of Gestures using Hierarchical Markov Models*, Master's Thesis, ATIAM 2010-2011, Université Pierre et Marie Curie, Ircam, Telecom Paristech, 2011
10. J.Françoise, *Motion-Sound Mapping by Demonstration*, PhD Thesis, Université Pierre et Marie Curie, Ircam, 2015
11. J.Françoise, B.Caramiaux, F. Bevilacqua, "A Hierarchical Approach for the Design of Gesture - to - Sound Mappings", in *Proceedings of the 9th Sound and Music Conference (SMC),* Copenhague, Denmark, 2012
12. F.Bevilacqua, N. Rasamimanana, E. Fléty, S. Lemouton, F. Baschet, "The augmented violin project: research, composition and performance report", *6th International Conference on New Interfaces for Musical Expression (NIME 06)*, Paris, 2006
13. P. Antoniadis,"Physicality as a performer-specific perspectival point to I. Xenakis's piano work. Case-study Mists", in *Proceedings of the I.Xenakis International Symposium 2011*, Goldsmiths University, London
14. G. Sandor, *On Piano Playing. Motion, Sound and Expression*, New York: Schirmer, 1981
15. D.Fober, Y. Orlarey, S. Letz: "INScore: An Environment for the Design of Live Music Scores", *Proceedings of the Linux Audio Conference*-LAC 2012
16. P.Antoniadis, D. Fober, F. Bevilacqua, "Gesture cutting through textual complexity: Towards a tool for online gestural analysis and control of complex piano notation processing", in *CIM 14 Proceedings,* Berlin, 2014
17. P. Antoniadis, F. Bevilacqua, D. Fober, "Gesture cutting through textual complexity: Towards a tool for online gestural analysis and control of complex piano notation processing" https://www.youtube.com/watch?v=KV9nQUhhyuI
18. P. Antoniadis: "Corporeal Navigation: Embodied and Extended Cognition as a Model for Discourses and Tools for Complex Piano Music After 1945" in Pedro Alvarez (ed.) *CeReNeM Journal*, Issue 4, pages 6-29, March 2014 http://issuu.com/cerenem/docs/cerenem_journal_issue_4

INSCORE EXPRESSIONS TO COMPOSE SYMBOLIC SCORES

G. Lepetit-Aimon **D. Fober** **Y. Orlarey** **S. Letz**
Grame
Centre national de création musicale
Lyon - France
gabriel.lepetit.aimon@grame.fr

ABSTRACT

INScore is an environment for the design of augmented interactive music scores turned to non-conventional use of music notation. The environment allows arbitrary graphic resources to be used and composed for the music representation. It supports symbolic music notation, described using Guido Music Notation or MusicXML formats. The environment has been extended to provided score level composition using a set of operators that consistently take scores as arguments to compute new scores as output. INScore API supports now *score expressions* both at OSC and at scripting levels. The work is based on a previous research that solved the issues of the notation consistency across scores composition. This paper focuses on the language level and explains the different strategies to evaluate score expressions.

1. INTRODUCTION

Contemporary music creation poses numerous challenges to the music notation. Spatialized music, new instruments, gesture based interactions, real-time and interactive scores, are among the new domains that are now commonly explored by artists. Classical music notation doesn't cover the needs of these new musical forms and numerous research and approaches have recently emerged, testifying to the maturity of the music notation domain, in the light of computer tools for music notation and representation. Issues like writing spatialized music [1], addressing new instruments [2] or new interfaces [3] (to cite just a few), are now subject of active research and proposals.

Interactive music and real-time scores are also representative of an expanding domain in the music creation field. The advent of the digital score and the maturation of the computer tools for music notation and representation constitute the basement for the development of this musical form, which is often grounded on non-traditional music representation [4] [5] but may also use the common music notation [6, 7].

In order to address the notation challenges mentioned above, INScore [8, 9] has been designed as an environment opened to non-conventional music representation (although it supports symbolic notation), and turned to real-time and interactive use [10, 11]. It is clearly focused on music representation only and in this way, differs from tools integrated into programming environments like Bach [12] or MaxScore [13].

INScore has been extended with *score expressions* that provide symbolic scores composition features (e.g., putting scores in sequence or in parallel). Building new scores from existing scores at symbolic level is not new. Haskell is providing such features [14]. Freeman and Lee proposed score composition operations in a real-time and interactive notation context [15]. Regarding the score operations used by INScore, they are imported from a previous work [16] that was focusing on the music notation consistency through arbitrary scores composition.

The novelty of the proposed approach relies on the dynamic aspects of the scores composition operations, as well as on the persistence of the score expressions. A score may be composed as an arbitrary graph of score expressions and equipped with a fine control over the changes propagation.

The paper introduces first the score composition expressions. Next, the different evaluation strategies are explained and illustrated with examples. The articulation with the INScore environment is presented in detail and followed by concrete use cases. An extension of the primary scores composition design to *score expressions* composition is next introduced. A generalisation of this approach to the whole set of INScore graphic objects is finally considered in the concluding section.

2. LANGUAGE SPECIFICATION

The main idea behind the project is to design a relevant language that provides easy to use tools to compose and to manipulate symbolic scores. Indeed, as all the operators have already been defined in a previous work [16], the point is to imagine a handy way to use them from INScore but above all, to benefit of the dynamic aspects of the INScore environment.

2.1 The operators

All the operators have a common interface: regardless their actual definition, they always take two scores as input to produce a score as output. The scores are expressed using the Guido Music Notation format [GMN][17]. A few low-level score manipulation operations are defined (which ap-

operation	arguments	description
seq	$s1$ $s2$	puts the scores $s1$ and $s2$ in sequence
par	$s1$ $s2$	puts the scores $s1$ and $s2$ in parallel
rpar	$s1$ $s2$	puts the scores $s1$ and $s2$ in parallel, right aligned
top	$s1$ $s2$	takes the n first voices of $s1$, where n is the number of voices of $s2$
bottom	$s1$ $s2$	cuts the n first voices of $s1$, where n is the number of voices of $s2$
head	$s1$ $s2$	takes the head of $s1$ on $s2$ duration
evhead	$s1$ $s2$	takes the n first events of $s1$, where n is $s2$ events count
tail	$s1$ $s2$	cuts the head of $s1$ on $s2$ duration
evtail	$s1$ $s2$	cuts the n first events of $s1$, where n is $s2$ events count
transpose	$s1$ $s2$	transposes $s1$ so its first note of its first voice match $s2$ one
duration	$s1$ $s2$	stretches $s1$ to the duration of $s2$
		(note that this operation may produce durations that are not displayable
pitch	$s1$ $s2$	applies the pitches of $s1$ to $s2$ in a loop
rhythm	$s1$ $s2$	applies the rhythm of $s1$ to $s2$ in a loop

Table 1. INScore operators

ply perfectly to INScore language's philosophy) with a deterministic behaviour (none of the operators implement random operations). Basically, these operations apply to the time domain (putting scores in sequence, in parallel, cutting parts of a score, time stretching), or to the score structure (extracting voices). A few additional operations are provided: transposition and application of a score's rhythm or pitch to another score. The small set of operators is not a real limitation, as the uniformity between their inputs and output make them easy to combine into pipeline designs, creating more high-level operations. The selected basic operators are not intended to cover the composition process (a real programming language like Open Music [18] would be required) but to provide tools for dynamic symbolic score computation, especially in the context of music performance.

See Table 1 for the definition of all operators. Note that there is no constraint on the input scores. For the `par` and `rpar` operations, the shortest score (if any) is suffixed or prefixed with the necessary duration to obtain the same length. These extensions appear as empty staves, which is easily expressed using the GMN language.

2.2 Designing a creative language

In the context of software used for artistic creation like INScore, designing a language is not trivial. Like any other creative tools, the *score expressions* language shall inevitably frame the creation process through which the artist must go. To that extent, conceiving a language is actually designing a creative "work-flow" that the users shall then adopt.

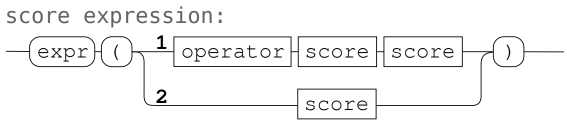
The continuity between inputs and outputs through Guido operators allows to compose a music by successively transform and aggregate scores fragments. This process (applying transformations on various materials and combining them into a whole creation) is similar to electro-acoustic creative processes where, after choosing sound material, the composer applies effects... and mixes them until this raw musical materials become unrecognisable.

Adapting this approach to the symbolic music notation would not only make the language easy to learn for composer but could offer great tools for composition: carving and assembling score samples using structural operators, placing the musical structure in the center of the creative process. In some ways, the art wouldn't emerge from the quality of the raw score fragments but from the process that transforms, shapes, and links them together.

It's with this perspective and emphasis of the structure that the *score expressions* syntax has been defined. In particular, these expressions should make use of various heterogeneous materials including *score expressions* or existing score objects.

2.3 *Score expressions* syntax

Score expressions can be defined using two syntaxes:

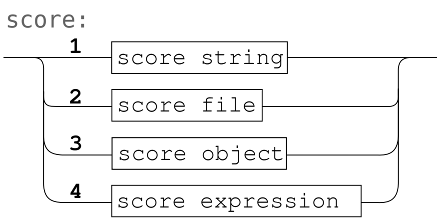


1. The classic syntax reflects the way Guido operators actually work: two scores are combined into one, according to the operator.

2. The alternate syntax defines an expression using a single score, which can be useful to duplicates objects e.g. to provide different views (see section 6.2).

Note that the leading `expr` token is present to disambiguate parenthesis that are already used in INScore scripts with messages lists.

Both of the syntaxes make use of `score` arguments. *Score expressions* are quite permissive regarding to their type:



1. score string: are GMN or MusicXML strings.

2. score file: refers to a score file that should contain GMN or MusicXML data. File path complies to INScore file handling and could indicate an absolute, a relative path or a URL.

3. score object: refers to an existing INScore object using a relative or absolute OSC address. The object must be a guido, musicxml or piano-roll object, as well as guido and piano-roll streams.

4. score expression: *score expressions* can be used as arguments of *score expressions* (in this case the expr token is optional).

Here is an example of a *score expression* that puts a score in parallel with 2 scores in sequence:

```
expr( par score.gmn (seq "[c]" score) )
```

Note that some operations could take more than 2 scores as arguments. For example, the sequence (seq) or parallel (par) operations could apply to arbitrary lists of arguments and higher-order operations could be defined, similarly to the functional programming *fold* (or *reduce*) high-order function [19]. The current syntax doesn't support *folding* but this may be considered in the future. For example, that would allow to write (seq a b c) instead of (seq a (seq b c)).

3. EVALUATION SPECIFICATION

The *score expressions* language is first transformed into an internal memory representation. In a second step, this representation is evaluated to produce Guido Music Notation [GMN] [17] strings as output, that are finally passed to the INScore object as specific data.

3.1 Internal representation of *score expressions*

When encountering an *score expressions*, the INScore parser creates a tree representation of it: arguments are stored as leaves and operators as nodes (Figure 1). This tree form allows to easily store, manipulate, assemble and evaluate *score expressions*.

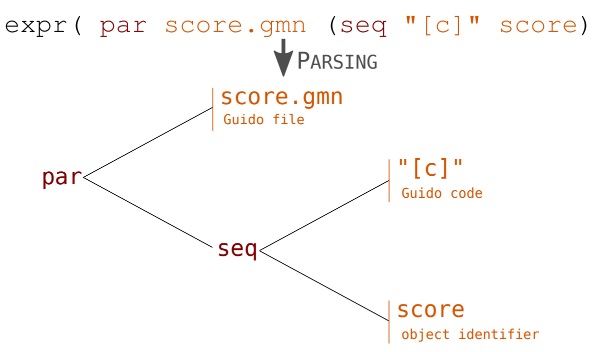


Figure 1. Parsing *score expressions* into tree form.

The tree representation is strictly matching the expression string. Type specification of arguments is the only difference, whereas types are implicit in *score expressions*, arguments are explicitly stored as GMN code or file or identifier... in the tree form.

Once the internal representation has been constructed by the parser, it is stored with the newly defined object, ready for evaluation.

3.2 *Score expressions* evaluation process

The evaluation process goes through every nodes of the expression tree using a depth first post-order traversal, reducing all of them into GMN code. A node evaluation is type dependent (Figure 2).
Evaluation of:

- a GMN file gives its content,
- a GMN string returns the string,
- a MusicXML file returns its content converted to GMN code,
- a MusicXML string returns the string converted to GMN code,
- an object identifier gives its GMN code,
- an operator node returns the application of the operator to the GMN code given as parameters.

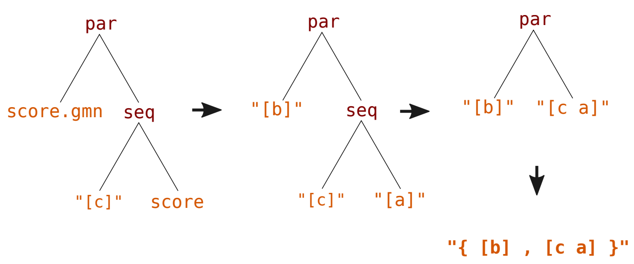


Figure 2. Simple evaluation of an expression tree, where score is defined as [a] and score.gmn contains [b].

This evaluation scheme avoid recursion issues (e.g., a score that modifies itself using an expression based on its own content) since the caller object is modified only at the end of the evaluation process. All arguments are referentially transparent by default: each argument is evaluated once and its value is then considered constant.

3.3 Dynamic evaluation of *score expressions*

Referential transparency (i.e., static evaluation) can be a huge limitation. For example, working with guido stream, one could want to maintain the result of a *score expression* up to date to the stream's actual state. Thus *variable arguments* have been introduced using a & prefix: a variable argument is always evaluated regardless of previous values (Figure 3). Only arguments subject to changes (score object or score file) can be declared *variable*.

A tree that contains a variable argument is a *dynamic tree*. When a variable argument is encountered on a tree branch,

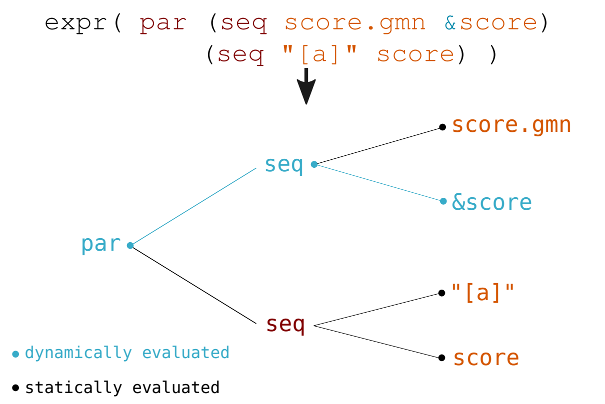


Figure 3. Propagation of dynamic evaluation. `&score` is updated to the actual value of `score` when re-evaluating, while `score` keeps the value computed on the first evaluation. Thus, on re-evaluation the lower `seq` operation will not be computed again.

the dynamic tree attribute is propagated up to the tree root. During the evaluation process, only the dynamic parts of a tree is recomputed. For optimisation, INScore checks if a variable argument value has changed and recomputes the corresponding operator only when needed.

Using variable arguments, an expression tree with arbitrary variable parts can be described: that may be viewed as building a symbolic score with arbitrary aggregation of static and variable parts.

4. SCORE EXPRESSIONS API IN INSCORE

In order to fully integrate score operators, the implementation relies on INScore existing features. As a result, *score expressions* support URLs as file arguments, interaction events and benefit of web features. Interaction events have been extended notably for the purpose of dynamic evaluation (see section 4.3).

4.1 Declaring *score expressions*

Both `gmn` and `pianoroll` objects can be defined with *score expressions* using an extension of the `set` message. The evaluation of the expression is actually triggered by the target object when the `set` message is processed.

```
/ITL/scene/score set gmn expr(score.gmn);
/ITL/scene/pr set pianoroll expr(&score);
```

The previous example creates two objects: `score` is a symbolic representation of the GMN file `score.gmn`, and `pr` is a piano roll representation of `score` (here dynamically evaluated due to the `&` prefix).

4.2 *Score expressions* specific messages

Objects that are based on *score expressions* support additional messages:

- `reeval`: triggers the re-evaluation of the expression tree taking account of the static and dynamic parts.
- `renew`: triggers the re-evaluation of the expression tree regardless of existing constant values.

All these messages are available through the `expr` message:

```
/ITL/scene/score expr reeval;
/ITL/scene/score expr renew;
```

Finally, the *score expression* of an object can be retrieved with the `get expr` message:

```
/ITL/scene/score get expr;
```

4.3 Events typology extension

INScore interaction features are based on the association between an event and arbitrary set of OSC messages [10]. These messages are sent when the event occurs (e.g., a mouse down). INScore events typology has been extended with a `newData` event, that is triggered when the value of the target object changes, either due to a `set` or `reeval` message, or because data has been written in a stream object.

Using the `expr reeval` message in conjunction with the `newData` event, may trigger the automatic reevaluation of an expression when an object changes. With the example below, changing the content of `score` will fire the `newData` event and the associated `expr reeval` message is automatically sent to `copy` that updates its content accordingly.

```
/ITL/scene/score set gmn "[a]";
/ITL/scene/copy  set gmn expr(&score);
/ITL/scene/score watch newData
    (/ITL/scene/copy expr reeval);
```

In order to catch infinite loop issues, `newData` event handling is postponed to the next INScore time slot. As a result, updating the whole scene after changing the value of an object can take several event loop (if one object is referring to another object, itself referring to another one...) and during this process the INScore's graphic scene could go through transitory states. However, if objects are defined with recursive references and are auto-updated using this mechanism, INScore will still be able to update the score (without freezing).

5. COMPOSING SCORE EXPRESSIONS

While the expressions already presented allow to compose symbolic scores, it is also possible to compose *score expressions* which are stored in the referred objects using the prefix `~`. Indeed, whereas `score` and `&score` refer to the object's value, `~score` refers to the *score expression* used to define `score`. In practical, before the first evaluation, all arguments prefixed by `~` are replaced by a copy of the expression tree from the corresponding objects. It allows to easily make use of previously defined *score expressions* to create more complex ones.

Figure 4 illustrates how the expression tree is expanded with the example below.

```
/ITL/scene/score set gmn
  expr(seq "[a]" &sample);
/ITL/scene/score set gmn
  expr(seq (seq ~score "[b]") ~score);
```

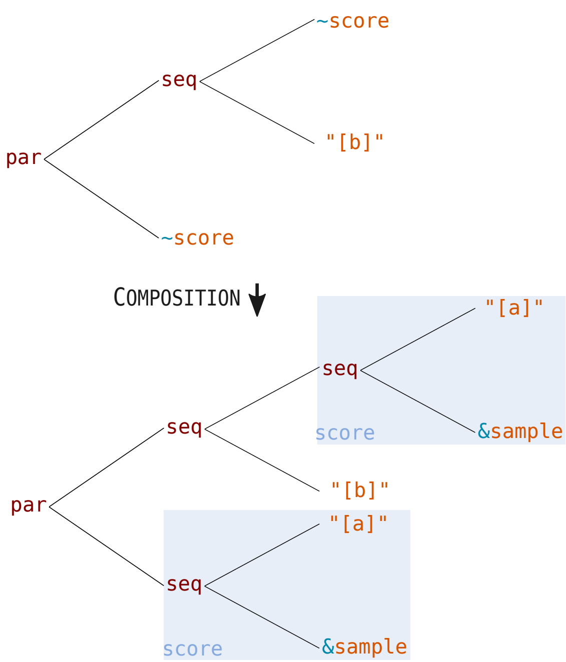


Figure 4. Composing *score expressions*

6. EXAMPLES

6.1 Canon structure

A simple but still well-known music structure is of course the canon. Creating such structure from a `score` is quite easy using *score expressions*.

With the example below, the first line creates a `score` object based on a GMN file. It is then transposed to a fifth and a second voice is added, delayed of a half note. Because transposing according to a specific interval is not a basic guido operator (the transposition interval is computed from the 2 scores arguments), one should combine `transpose` with `seq` and `evtail` to prepend the score with a note, transpose the whole score using this note and finally remove it.

```
/ITL/scene/score set gmnf score.gmn;

# Transposing score
/ITL/scene/canon set gmn
  expr(evtail
    (transpose (seq "[c]" score) "[g]")
    "[a]"
  );

# Putting score in sequence with it
/ITL/scene/canon set gmn
  expr(seq score canon);

# Adding a second voice delayed
/ITL/scene/canon set gmn
  expr(par canon (seq "[_/2]" canon));
```

The result is a simple canon:

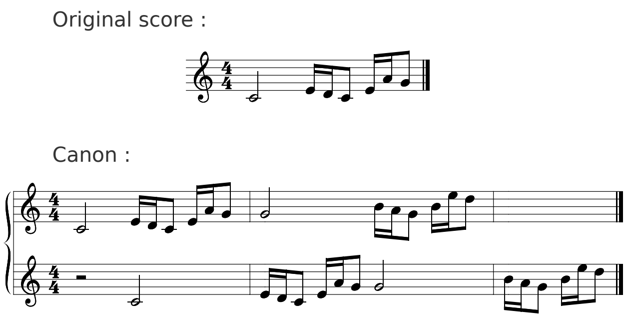



Figure 5. Canon result

6.2 Multiform synced scores

Score expressions is a great tools to duplicate and dynamically transform scores, keeping every copies synced to the original.

```
/ITL/scene/stream set gmnstream
  '[\meter<"4/4">]';

/ITL/saxo/score set gmn
  expr( evtail
    (transpose
      (seq
        "[e&1]"
        &/ITL/scene/stream )
      "[c2]" )
    "[a]"
  );

/ITL/audience/score set pianoroll
  expr( &/ITL/scene/stream);

/ITL/scene/stream watch newData
  (/ITL/*/score expr reeval);
```

The previous example creates 2 copies of the GMN stream object `stream`, one transposed for the saxophone and one displayed as a piano roll, intended as a visual support for the audience. Both are displayed in different scenes. The last line ensure the update of the copies when `stream` is modified. The `/ITL/scene/stream` argument is re-evaluated due to the & prefix. The result is illustrated in figure 6.

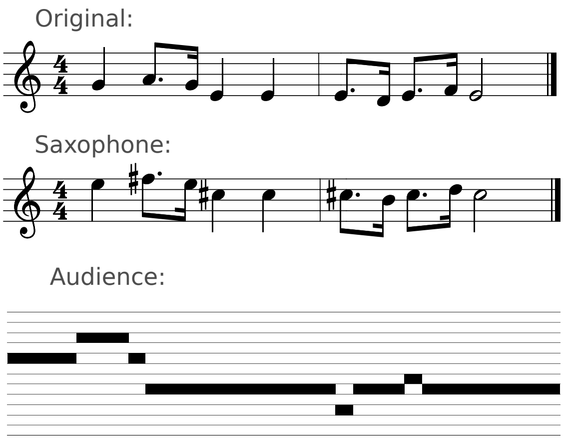


Figure 6. Multiform scores result

6.3 Mixing dynamic and static scores

This example illustrates how dynamic and static symbolic scores can be mixed and transformed in real-time. In a first step, we create a stream (named `stream`) intended to be written in real-time and a static score (named `static`).

```
/ITL/scene/stream set gmnstream
    '[\meter<"4/4">]';
/ITL/scene/static set gmn
    '[\meter<"4/4"> g e f a f d c/2]';
```

In a second step, the last two bars of the stream are extracted and store in a new object named `tail`. Since the 'tail' operation cuts the head of the score using the second argument, the duration of this argument is expressed as the tail of the stream using the desired duration (2 whole notes). Note that `tail` expression is using a reference to the stream in order to be updated each time data is written to the stream.

```
/ITL/scene/tail set gmn
    expr(tail &stream
        (tail &stream '[a*2]')));
```

The final result is simply obtained using the 'par' and 'transpose' operations. It makes use of references to `tail` but the `static` object is embedded statically. Note that `tail` is used as an intermediate object intended to optimise the computation and to facilitate reading of the expression. It can be hidden from the overall score without affecting the result.

```
/ITL/scene/score set gmn
    expr(par &tail
        (transpose static &tail));
```

Activation of the score dynamic computation makes use of the `newData` event watched by the `stream` object, that inform `tail` and `score` that their expressions need to be re-evaluated.

```
/ITL/scene/stream watch newData
     (/ITL/scene/part  expr reeval,
      /ITL/scene/score expr reeval
     );
```

7. CONCLUSIONS

Combining a simple set of operators with the powerful features of INScore (like URL support, full OSC compatibility, interaction support...), *score expressions* fully integrate symbolic score composition into an interactive and augmented music score environment. They suggest a creative process based upon musical structures and scores aggregation by giving the possibility to compose various score materials including score objects. Above all, *score expressions* provide a handy way to manipulate scores regardless to their origin (files, URL, streams...) or their representation (traditional music notation or piano roll) and to design dynamic scores based on arbitrary score composition.

In future work, we're considering extending the *score expressions* to all the INScore objects. Such an approach - composing arbitrary graphic resources using a musical semantic - raises issues that are non-trivial to solve. Indeed, if the operations on the time domain could be applied to any object due to their common time dimension, transformations in the pitch domain or based on structured time (like rhythm) implies to extend the musical semantic of the graphics space.

8. REFERENCES

[1] E. Ellberger, G. Toro-Perez, J. Schuett, L. Cavaliero, and G. Zoia, "A paradigm for scoring spatialization notation," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation - TENOR2015*, M. Battier, J. Bresson, P. Couprie, C. Davy-Rigaux, D. Fober, Y. Geslin, H. Genevois, F. Picard, and A. Tacaille, Eds. Paris, France: Institut de Recherche en Musicologie, 2015, pp. 98–102.

[2] T. Mays and F. Faber, "A notation system for the karlax controller," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. London, United Kingdom: Goldsmiths, University of London, June 2014, pp. 553–556. [Online]. Available: http://www.nime.org/proceedings/2014/nime2014_509.pdf

[3] W. Enstr"om, J. Dennis, B. Lynch, and K. Schlei, "Musical notation for multi-touch interfaces," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, E. Berdahl and J. Allison, Eds. Baton Rouge, Louisiana, USA: Louisiana State University, May 31 – June 3 2015, pp. 83–86. [Online]. Available: http://www.nime.org/proceedings/2015/nime2015_289.pdf

[4] R. R. Smith, "An atomic approach to animated music notation," in *Proceedings of the First International Conference on Technologies for Music Notation and*

Representation - TENOR2015, M. Battier, J. Bresson, P. Couprie, C. Davy-Rigaux, D. Fober, Y. Geslin, H. Genevois, F. Picard, and A. Tacaille, Eds. Paris, France: Institut de Recherche en Musicologie, 2015, pp. 39–47.

[5] C. Hope, L. Vickery, A. Wyatt, and S. James, "The decibel scoreplayer - a digital tool for reading graphic notation," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation - TENOR2015*, M. Battier, J. Bresson, P. Couprie, C. Davy-Rigaux, D. Fober, Y. Geslin, H. Genevois, F. Picard, and A. Tacaille, Eds. Paris, France: Institut de Recherche en Musicologie, 2015, pp. 58–69.

[6] R. Hoadley, "Calder's violin: Real-time notation and performance through musically expressive algorithms," in *Proceedings of International Computer Music Conference*, ICMA, Ed., 2012, pp. 188–193.

[7] ——, "December variation (on a theme by earle brown)," in *Proceedings of the ICMC/SMC 2014*, 2014, pp. 115–120.

[8] D. Fober, Y. Orlarey, and S. Letz, "Inscore – an environment for the design of live music scores," in *Proceedings of the Linux Audio Conference – LAC 2012*, 2012, pp. 47–54.

[9] ——, "Augmented interactive scores for music creation," in *Proceedings of Korean Electro-Acoustic Music Society's 2014 Annual Conference [KEAMSAC2014]*, 2014, pp. 85–91.

[10] D. Fober, S. Letz, Y. Orlarey, and F. Bevilacqua, "Programming interactive music scores with inscore," in *Proceedings of the Sound and Music Computing conference – SMC'13*, 2013, pp. 185–190. [Online]. Available: fober-smc2013-final.pdf

[11] D. Fober, Y. Orlarey, and S. Letz, "Representation of musical computer processes," in *Proceedings of the ICMC/SMC 2014*, 2014, pp. 1604–1609. [Online]. Available: inscore-processes-final.pdf

[12] A. Agostini and D. Ghisi, "Bach: An environment for computer-aided composition in max," in *Proceedings of International Computer Music Conference*, ICMA, Ed., 2012, pp. 373–378.

[13] N. Didkovsky and G. Hajdu, "Maxscore: Music notation in max/msp," in *Proceedings of International Computer Music Conference*, ICMA, Ed., 2008.

[14] D. Quick and P. Hudak, "Grammar-based automated music composition in haskell," in *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling and design*, ser. FARM '13. New York, NY, USA: ACM, 2013, pp. 59–70. [Online]. Available: http://doi.acm.org/10.1145/2505341.2505345

[15] S. W. Lee and J. Freeman, "Real-time music notation in mixed laptop-acoustic ensembles," *Computer Music Journal*, vol. 37, no. 4, pp. 24–36, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00202

[16] D. Fober, Y. Orlarey, and S. Letz, "Scores level composition based on the guido music notation," in *Proceedings of the International Computer Music Conference*, ICMA, Ed., 2012, pp. 383–386. [Online]. Available: icmc12-fober.pdf

[17] H. Hoos, K. Hamel, K. Renz, and J. Kilian, "The GUIDO Music Notation Format - a Novel Approach for Adequately Representing Score-level Music." in *Proceedings of the International Computer Music Conference*. ICMA, 1998, pp. 451–454.

[18] J. Bresson, C. Agon, and G. Assayag, "OpenMusic – Visual Programming Environment for Music Composition, Analysis and Research," in *ACM MultiMedia (MM'11)*, Scottsdale, United States, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01182394

[19] G. Hutton, "A Tutorial on the Universality and Expressiveness of Fold," *Journal of Functional Programming*, vol. 9, no. 4, pp. 355–372, Jul. 1999.

OMLILY: FILLING THE NOTATIONAL GAP BETWEEN COMPOSITION AND PERFORMANCE

Karim Haddad
IRCAM
karim.haddad@ircam.fr

Carlos Agon
IRCAM - UPMC
carlos.agon@ircam.fr

ABSTRACT

This paper describes the design, development, usage, limitations and prospects for future development of Omlily, an OpenMusic library for editing scores with Lilypond, using OpenMusic musical editors [1] .

1. INTRODUCTION

Using a Computer Assisted Composition (CAC) environment such as OpenMusic (OM) [1], rich in functions, macros, and algorithms for composition, we assemble a huge amount of musical material, such as pitch, rhythm, and other musical structures contained in OpenMusic musical classes and editors. OpenMusic editors are powerful objects, they can deal with the most complex musical structures. However, they also have certain limitations regarding display and typesetting capabilities. Furthermore, editing scores directly in these editors seems to be very laborious when particulary if the pieces are of long duration. This is due to two important factors that could be considered as flaws (or weaknesses) in OpenMusic: lack of efficient editing tools and slowness in the display time. Another essential element not available to the composer in the OpenMusic environment, but also related to display, is the music sheet layout view. Although this option is present, the score displayed has very few (if any) options for layout. In order to have a presentable draft for the composer to work with, we can benefit greatly from Lilypond's extraordinary typesetting layout features [2] .

Writing a fully featured typesetter and viewer from scratch in OpenMusic involves coding it in CommonLisp [2]. This does not seem to be a bright approach for this issue at all. In the past we have developed internal Lisp code in OpenMusic to export OpenMusic musical objects to commercial typesetting programs based on their own private SDK standard format. We have also written code for MusicXML standard format export. However the validation of this standard format does not seem stable as a standard due to private parsers of each commercial typesetting software. Therefore, we decided to use a pre-existing typesetter, Lilypond is a GNU opensource typesetting software [3], that is maintained up to date continually. We have based our exchange code on Lilypond's own syntax. Moreover, Lilypond as MusicXML, are widely used in other CAC environment such as PWGL, FOMUS, Orchids, Abjad, etc. . . Lilypond seems be the best choice and solution regarding rendering, efficiency and notational possibilities. OpenMusic's architecture, conceived as a modular environment, handling and loading only essential features, and following the user's needs and requirements, will welcome such a library, typesetting being the most requested and necessary feature for OpenMusic's end user.

[1] https://github.com/karimhaddad/omlily/
[2] such as paper, tuplet, even color display

2. USAGE

2.1 General usage

The purpose of this library is to combine both of the potentialities of a CAC environment by devising complex forms of compositions with the editing efficiency of a powerful typesetter in a dynamic form of interaction. Moreover, another aim will be to require minimal effort for producing huge and complex input content such as rhythm, pitch and other musical material in the form of a typesetting document. In another sense, the composer, most of the time, will need to go back and forth from a CAC environment to typesetting and vice versa most of the time managing huge amounts of musical data, particularly in the case of big ensemble or orchestral compositions.

If we schematize the work-flow of the different steps of a typical compostional process using CAC tools, we can describe it as follows:

- The pre-compositional stage, which involves automatic or algorithmic computation, sound analysis, combinatory computations, etc., in other words, the first draft of musical material production;
- The first stage of typesetting: pre-editing, previsualization, in order to rearrange and revisit the material as a score sheet. This might lead to adjustments, pre-instrumentation, voice redistribution, form editing, etc.;
- The intermediate feedback phase in CAC for re-computation, corrections, arrangements, modifications, form re-injections, segmentation,etc.; this step is the most dynamic one;

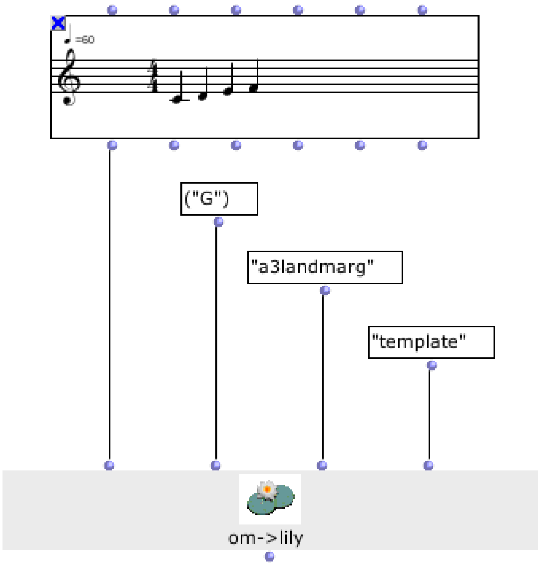


Figure 1. Exporting from OpenMusic to Lilypond

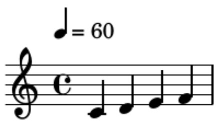



Figure 2. Lilypond rendering

- Finalization of typesetting by Lilypond (final score engraving).

2.1.1 Exporting Lilypond files from OpenMusic

This is done using the *om → lily* method (see Fig.1). It handles VOICE, POLY, and CHORD-SEQ OpenMusic objects. Four arguments are given to this method :

- *self*: the OpenMusic object to translate into Lilypond;
- *clef*: the clef needed (it could be a list of clefs in the case of a POLY object);
- *paper*: the paper default or user's template;
- *layout*: the default or user's score context (equivalent to notational preferences).

Once the .ly file is written, OpenMusic will redirect the file to Lilypond present binary and compile the file (Fig.2) opening it with the user's preferred PDF reader [3] .

2.1.2 Importing Lilypond files to OpenMusic

Each exported score from OpenMusic will generate a file where a commented code will be included. This code (cf. Fig 3), once uncommented, and recompiled with Lilypond binary, will create a Scheme translation file of the score. By default it will be named *temp.lisp*.

```
% #(with-output-to-file "temp.lisp"
% (lambda () #{ \displayMusic {
<<
\new StaffGroup
 << non unecessary
\new Staff  {
\one
}
 >>
 >>
% } #}))
}
```

Figure 3. Lilypond's compilation instructions for a Scheme transcript of a .ly file

Lilypond will then generate a Scheme code where all of the musical and layout elements are translated by the *make-music* method. Again, once this file is evaluated with the *lily→om* method in OpenMusic, the necessary data will be translated and instantiated into an OpenMusic editor (Fig. 4)

2.2 Particular usage

2.2.1 Polymetrics and polytempi notation

OpenMusic has the ability to display and perform the most complicated and sophisticated rhythmical expression due to the implementation of Rhythm Trees (RT) [4]. This includes embedded tuplets, polymetric music, polytempi, and irrational measures [4] such as 3/21 or 4/10 for example (Fig. 5). Such cases are handled poorly or at all, by most commercial typesetters since these features are seen as completely experimental and related specifically to contemporary compositional practice, and therefore not popular for most users and musicians; as a result, they are not supported in these environments. Fortunately, Lilypond has the ability to deal with most, if not all, of these issues, with a standard approach. However in cases such as polymetric music, one should rely upon a slightly different syntax in order to display the complex polyphony correctly.

When using polymetrics in standard notation, i.e with binary time signatures, Lilypond will automatically display the correct score. Hence, the *om→lily-gen* method will be used. If the score uses non-standard, non-binary time signatures, the user should choose the *om→lily-spec* method. Both methods are included in the generic method *om→lily-gen*, which will automatically depict the presence of polymetric time signatures.

[3] The Lilypond binary and version number are generated automatically once the library is loaded in OpenMusic. However, the user may search for the desired Lilypond and PDF reader binaries in OpenMusic's "External" preferences tab.

[4] cf. Livre Premier de Motets: The Time-Block Concept in OpenMusic[5]

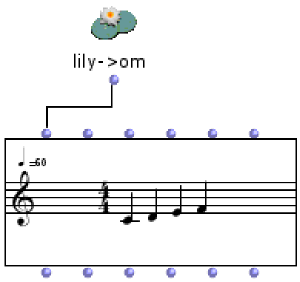


Figure 4. Importing form Lilypond to OpenMusic

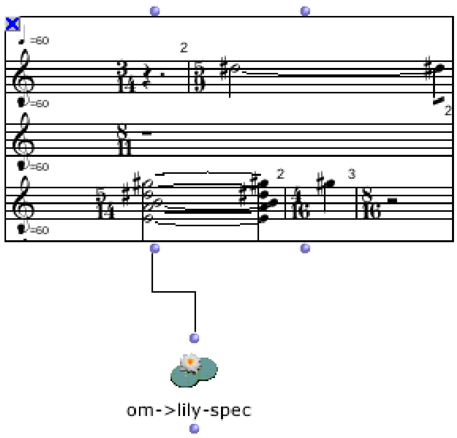


Figure 5. Polymetric score exportation

Although polyphonic polymetric notation is possible in Lilypond, it is not fully documented for the case of "irrational" time signatures. The scaling factor of duration is calculated as follows :

For instance, if we consider a measure with a time signature of 3/20 (this will be a measure of three sixteenth note of a quintuplet), we will multiply all sixteenth notes figures by 1/5 (note 16*1/5). For binary time signatures, the multiplication factor would invariably be 1/4.

The calculation is made using the `calc-scale-fact` function (Fig.7). This will first calculate the beat-symbol using the `find-beat-symbol` function (the note figure, e.g. quarter note, eighth note, etc. . .) that the denominator of the time signature is related to. For instance, if we have a time signature of 4/12, twelve will refer to a twelfth of a whole note, equaling an eighth note of a triplet [5] . The factor will be then calculated with this formula: $\frac{(beat-symb/4)}{denom}$ = $\frac{8/4}{12}$. The scaling factor in our example will therefore be equal to 1/6.

2.2.2 *Discrete spanner notation*

One immediately see that this library is addressed mainly for written instrumental compositions. However, in prospect, there is also to be some audio/graphical symbolic notational outputs that can be extracted from break-point functions (BPF), or representation of audio sources. Indeed, some OpenMusic classes have the ability to enclose both representations, symbolic and graphical outputs of musical objects. Fig.8 shows an example of a Lilypond score rendering exported from OpenMusic's BPF objects. `make-music` Scheme function

[5] 1/3 of a whole note = a half note figure. 1/6 = a quarter note, etc. . .

3. IMPLEMENTATION

OpenMusic Lisp code and Lilypond Scheme [6] syntax handle most of the communication between the two environments. We can describe two different levels for both environments: the first level is proprietary and typical for each (OpenMusic patches, as a visual programming language, or VPL); and a latex-like scripting language, a syntax for editing Lilypond .ly files. Consequently, each of these levels remains opaque to each others. It is on an internal level that the communication occurs: the CommonLisp language used by OpenMusic's kernel, and the Scheme expressions which are part of Lilypond's Guile interpreter. This is where the link between both environments lies, due to the fact that both Scheme and Commonlisp are dialects of the Lisp language.

We can summarize this communication schematically with a straightforward process :

OM patch (VPL)→(CommonLisp) RT→Lilypond (ly syntax) →Scheme interpreter→OM musical class object editor

An example of this translation : A simple voice bearing a single note (Fig.9), such as a middle C on a G staff [6] in a quarter note figure in a 1/4 time signature will be written [7] in OpenMusic as :

```
(make-instance 'voice
:tree '(1/4 (((1 4) (1))))
:chords '((6000))
:tempo '(60)
)
```

Skipping the paper settings, page layout and contexts, the translation of this simple object will render this Lilypond code :

```
...

"one"=
{
\tempo 4 = 60
\time 1/4
c'4
|
}

\score {
 {
% #(with-output-to-file "temp.lisp"
% (lambda () #{ \displayMusic {
<<
```

[6] In OpenMusic, the staff keys are not explicitly formulated.

[7] We may notice that in OpenMusic, the rhythmical inforamtion is separated from pitch information.

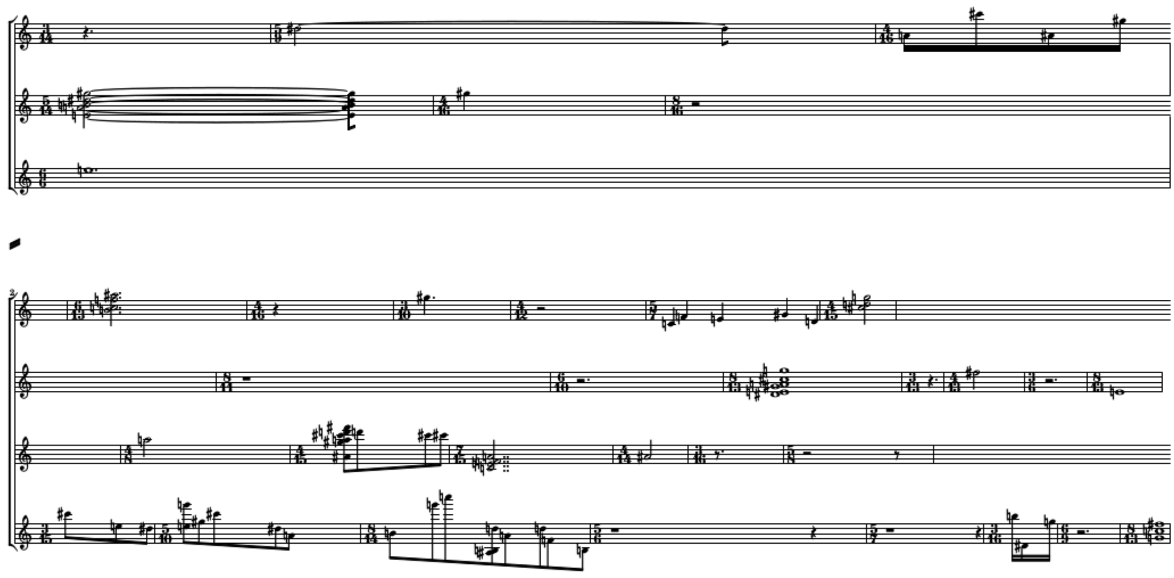


Figure 6. Polymetric score rendering

```
(defun calc-scale-fact (time-signature)
(let*
((denom (second time-signature))
     (beat-symb (find-beat-symbol denom))
     (fact (/ (/ beat-symb 4) denom)))
 (if (= 1 fact) 1/4 fact)))
```

Figure 7. Calculating scaling factor of each note figure according to its time signature.

```
\new StaffGroup
 <<

\new Staff  {
\one
}
 >>
 >>
% } #}))

}

...
```

We can already observe what new data has been produced in this translation:

- Paper settings, layout and contexts (omitted here).
- Alteration display rules.
- Staff grouping layout (not shown here, since it is a single voice).

The generated data will grow more and more as we progress toward typesetting. This is due to the required paper settings, staff layout, line breaks, etc. Inversely, from Lilypond to OpenMusic, most of the typesetting content will be omitted and filtered to the strict minimum since these will not be necessary for the instantiation of an OM object, which editor is a set of linear display of graphical notation using fonts and line drawings without page layout.

The intermediary Scheme translation code generated from OM using the *lily→om* method will look like this:

```
(make-music
...


(make-music
 'TimeSignatureMusic
   'beat-structure
   '()
   'denominator
    4
   'numerator
    1)

 (make-music
  'NoteEvent
      'duration
     (ly:make-duration 2)
      'pitch
      (ly:make-pitch 0 0))

 (make-music
   'BarCheck)
...
)
```

After retrieving pagination, layout, and some unneeded information [8] , if we carefully examine the reduced Scheme

[8] We have omitted some headings in order to save some place in our paper. However, we have displayed intentionally most of the required essential notational data.

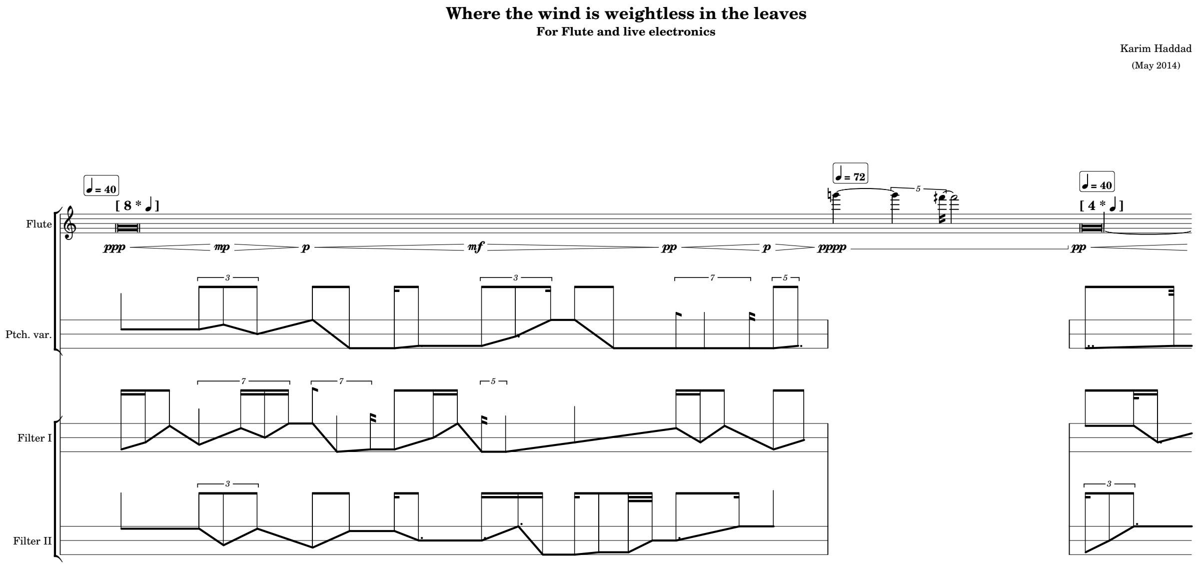


Figure 8. Control spanners

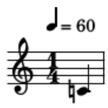



Figure 9. Single note

code above, carefully we will distinguish three redundant and distinct calls of the `make-music` Scheme function. This function is "for internal use, which is the preferred interface for creating music objects"[7]. These objects are created in Lilypond by C++ code, and represent a hierarchy of instances of musical notation. Thus in our example, our instances will be the three arguments of `make-music` function 'TimeSignatureMusic, 'NoteEvent, and 'BarCheck.

In order to translate these instances into OpenMusic compliant objects, we have transformed the `make-music` Scheme function into a CommonLisp (CLOS) method. This method will in turn, instantiate the different classes, such as staff, pitch, duration, etc. needed for OpenMusic to construct a compliant object according to each given type. Inspecting the previous example, we will again find our initial data unaltered.

Here is an example of the `make-music` Scheme function translation in CommonLisp regarding pitch and rhythm transcription [9] :

```
(defmethod make-music
((type (eql 'NoteEvent))
         &rest other-args)
(if *lil-imp-pitch*
  (let* ((art (car
            (find-value-in-lily-args
            other-args
        'articulations)))
              (tie
               (if (equal 'tieevent art)
               1 0))
              (durs
               (find-value-in-lily-args
               other-args 'duration))
              (fig (car durs))
              (dot (second durs))
              (fact (third durs)))
         (make-instance 'lily-dur
               :figure fig :dot dot
               :fact fact  :tieevent tie
                           :restevent 0 ))
     (if (not (member 'tieevent
      (flat (find-value-in-lily-args
      other-args 'articulations))))
         (remove nil (list
          (find-value-in-lily-args
          other-args 'pitch)))  )))
```

However, we should point out that only data necessary for OpenMusic's objects instantiation will be taken into account. If the Lilypond file has been modified by including extra notations such as dynamics or text markings, the import procedure will ignore these. Round tripping is something which will be included in the future as a standard

[9] In Lilypond, time signature is a separate piece of information (as we may have seen in the intermediate Scheme translation given above), which is not the case with OpenMusic's RT structure[4], where it is completely integrated into rhythm information. We are not displaying the method concerning it per se.

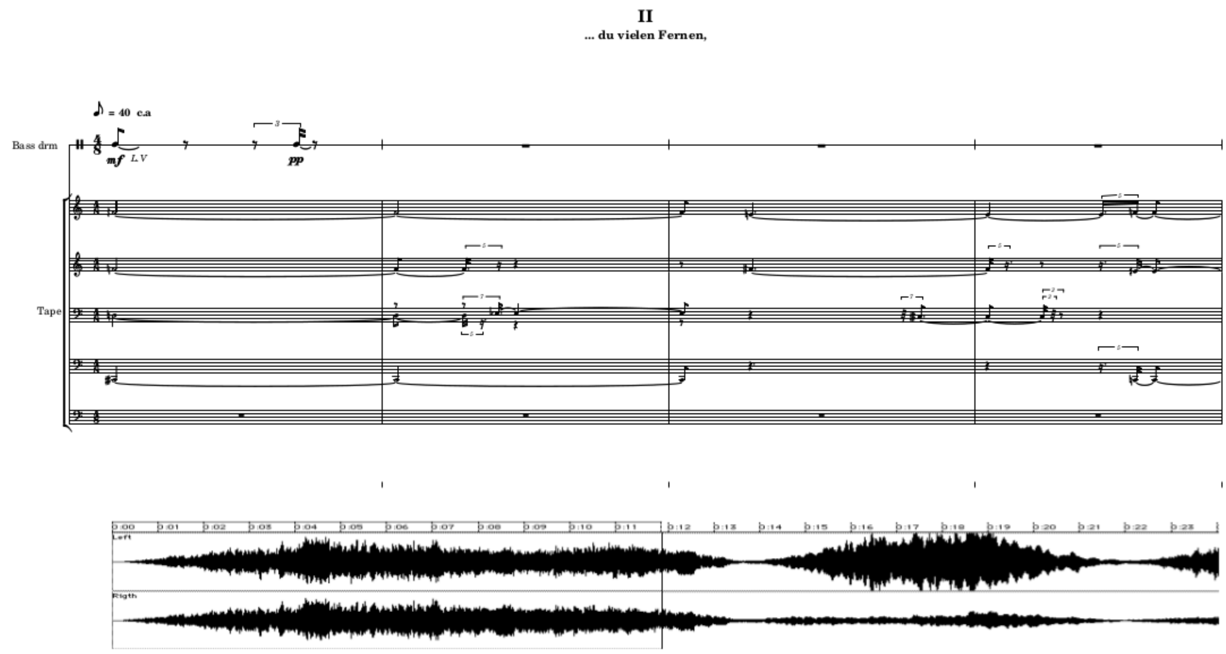


Figure 10. Score with audio wave shape

procedure (cf. future development section).

4. LIMITATIONS

Modern score setting is a field that embraces rich figuration and symbolic notational representation. It would likely be an extremely difficult task to encompass the majority of the musical symbolic representations necessary to render them in such a varied context. As indicated above, most of the essential hierarchical musical classes are represented with the inclusion of independent features such as polytempo; embedded and recursive rhythm structures; dynamics; linear spanners with the exclusion of musical elements not yet supported by OpenMusic, such as grace notes [10]; lyrics; crescendos/diminuendos; and other continuous symbolic extra notation features.

In Lilypond, page breaking process is an implemented algorithm. This performs well with strictly measured music. However, exporting page turns from OpenMusic is not supported, since in this environment, no such concept exists. It is based on graphical display edits rather than on rational musical ones. The main issue will therefore be the page setup regarding the segmentation of a printed output. In the case of graphical representation such as audio amplitude profile embedded in the score, as in Fig.10 for instance, the page layout will be determined by the graphical representation itself. This task normally left to the typesetter's discretion (weight of notes by page, performer breaks, etc ...) cannot be automated in any way, since it is based on a performer-composer appreciation mostly by typesetting rules with regard to a graphical layout that will constrain the page settings.

For instance, if we examine Fig.6 closely, we will see that the first system has a greater line span than the second one which is more dense than the former. Here, the line breaks were calculated by Lilypond without any explicit instructions on the part of the user. For the time being with the absence of a graphical interface within OpenMusic, for such a task we would have to, if needed, edit the breaks in Lilypond manually.

5. FUTURE DEVELOPMENT

Certain immediate issues are under imminent implementation. We can list some of these here:

- CHORD-SEQ structure import from Lilypond to OpenMusic; [11]
- PianoStaff support, most particularly cross stemming and automatic voice splitting;
- Graphical interface for page layout and other typesetting preferences.

Apart form these peripheral additions, our goal for an innovative implementation of this inter-exchange program would ideally be to achieve complete interaction between the two environments, such as musical computation interacting directly with the resulting typesetting. This could

[10] Grace notes are not yet supported graphically. However they are integrated as objects. There is an ongoing effort to implement them in our recent development of a notational viewer which will be hopefully integrated into OpenMusic.

[11] The Lilypond-to-OpenMusic import feature works only for a simple general usage for the time being. It does not yet support the CHORD-SEQ export, since it is not metered music and requires a different approach.

be achieved by automatic computation, i.e. building routines which will deal with the given data (musical material/typesetting material) automatically in order to have direct rendering and transformation on both sides of the platform, e.g. changing a section in the editor (here, Lilypond) and feeding this back in an OpenMusic patch or vice-versa. Ideally, the goal is to build an embedded graphical editor in OpenMusic as stated before, containing Lilypond typesetting rendering with a Lilypond-syntax-embedded editor.

With this in perspective, we are planning to implement an intermediate inter-exchange format for round tripping. This will allow for the safe keeping of all incoming and outgoing data to and from OpenMusic and Lilypond. In order to achieve this objective, we should survey most of the existing classes and internal properties of Lilypond making them available in a registry that will be compliant with both platforms.

Such a thing is possible, even recommended, since the SHEET object[8] (see Fig.11), still in beta state, should be finalized in such a way.

SHEET[8] is a graphical OpenMusic editor whose purpose is to have abilities in editing and ”throwing” a computational operation such as transposition or any other serial operation on symbolic objects such as measures, groups, chords, rhythm, etc. It has the capacity to display symbolic notation (scores), audio wave forms, and Break Point functions (BPF) along with OpenMusic patches.

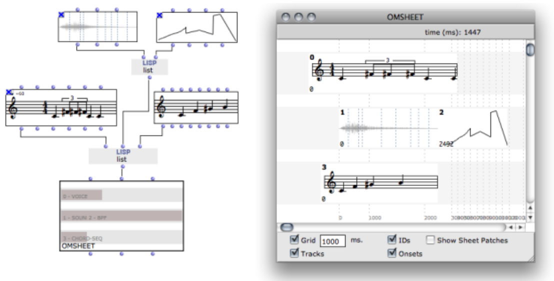


Figure 11. OpenMusic SHEET editor object

6. CONCLUSIONS

Omlily OpenMusic’s library seems for us to be, for the time being, a very good solution for musical score interchange between OpenMusic and Lilypond. It is also a powerful research tool for experimentation for a new scope of musical ideas due to its unique potentialities in exploring complex musical structures. We have been using it for sometime now and have conceived many compositional scores with it, from solo to ensemble music. Extending its potentialities in order to enclose more notational data seems promising. This interchange can lead to expanding both environments, having, on one side, the ability to integrate CAC functions and computations in the Lilypond environment, and on the other, one of the best typesetters for rendering scores in OpenMusic.

Acknowledgments

We would like to thank Jean Bresson for his cooperation, Jeremy Coffman and Deborah Lopatin for their proofreading and last but not least, Gerard Assayag for his valuable expertise and advice.

7. REFERENCES

[1] C. Agon, “Openmusic : Un langage visuel pour la composition musicale assiste par ordinateur,” Ph.D. dissertation, IRCAM - Univ. Paris 6, dcembre 1998.

[2] G. Steele, *Common Lisp the Language, 2nd edition*. Digital Press, 1990.

[3] *LilyPond, a system for automated music engraving*. Firenze, Italy: Colloquium on Musical Informatics, 2003.

[4] C. Agon, K. Haddad, and G. Assayag, “Representation and rendering of rhythmic structures,” *WedelMusic Darmstadt*, vol. 25, pp. 109–113, 2002.

[5] K. Haddad, *Livre Premier de Motets: The Time-Block Concept in OpenMusic*, 1st ed., ser. The OM composer’s book. Edition Delatour, 2008, vol. II.

[6] G. J. Sussman and G. L. S. Jr., “Scheme: An interpreter for extended lambda calculus,” in *MEMO 349, MIT AI LAB*, 1975.

[7] *Lilypond - Internals Reference*, 2000-2012.

[8] J. Bresson and C. Agon, “Scores, programs and time representations:the sheet object in openmusic,” *Computer Music Journal*, vol. 32, no. 4, 2008.

NETSCORE: AN IMAGE SERVER/CLIENT PACKAGE FOR TRANSMITTING NOTATED MUSIC TO BROWSER AND VIRTUAL REALITY INTERFACES

Benedict Carey
Hochschule für Musik und Theater Hamburg
benedict_carey@icloud.com

Georg Hajdu
Hochschule für Musik und Theater Hamburg
georghajdu@mac.com

ABSTRACT

NetScore is an extension of the existing *MaxScore* package (Hajdu, Didkovsky) which adds new functionality for the rapid transmission and display of music notation on remote devices through standard modern browsers with WebSocket support. This was seen as a necessary development for *MaxScore* due to the ubiquity of tablets and other mobile devices, among other advantages for the user, and future applications of the software. We chose a server based solution executed in *Java* using the *Jetty* library for both portability between different platforms, and scalability. Novel applications facilitated by NetScore include transmitting scores generated in Max/MSP into virtual reality interfaces and more convenient performance/ rehearsal of real-time generated music, whereby devices commonly on hand such as smartphones, tablets and laptops are used as e-scores without requiring the installation of additional software.

1. INTRODUCTION

At the time of the emergence of Real-time composition as a trend in the 1980s, proponents of the approach had to have extensive knowledge of computer systems, or have access to a technician that could assist them realise their compositional ideas. Of the systems that did not require knowledge of command line scripting and other advanced techniques, few of them were designed for notation generation and the concept of real-time composition was to be a difficult to realise goal until much later with the advent of *OpenMusic* and *PWGL* [2]. In contrast, with recent developments such as the *Bach* library of Max/MSP/Jitter objects [1], *MaxScore* [2] and *Abjad* [3] and interfaces that can work with web browsers to display score data such as *INScore* [4] and *Scribe JS* (https://cruncher.ch/blog/scribe/), real-time musicians have experienced a veritable explosion of devices with which to perform from and produce complex symbolic notation in real-time. These tools are certainly far more accessible today to the average computer literate composer than those that came before them. Yet this technique remains in part highly technical, and while not prohibitively so, the associated configuration, installation and customisation of new compositions in addition to reconfiguring those compositions for ever changing platforms consumes much time for the composer. It may be the downfall of rehearsals and in worst case scenarios, actual concert performances should performers feel uncomfortable with the technologies they are asked to use.

A desirable outcome of the NetScore project was to create a score viewer that required no advanced knowledge on behalf of the person expected to operate and read the score. When considering developing a solution for MaxScore that supports mobile devices, we therefore considered it essential to develop a solution that did not require any extra installation or configuration on the users part, was platform independent and did not consume significant amounts of system resources. MaxScore, long associated with one of the authors' other similar projects *Quintet.net* [6], is at the heart of *NetScore*. MaxScore is a Max object, which accepts messages that can create a musical score, add notes to it, transform the notes, perform, save, and load the score, and export the score to popular formats for professional publishable results (for further info see www.computermusicnotation.com). In addition we chose a browser-based solution due to the ubiquity of mobile devices such as tablets and mobile phones, since many mobile devices available today contain a web browser of some sort. According to gsmaintelligence.com at the time of writing this paper there were as many as 3,763,381,520 unique mobile subscribers using a browser of some kind to interact with web content. It is fair to say then that these devices have indeed become ubiquitous; this number will no doubt continue to grow. The advantage of using mobile devices to display real-time notation is not just to do with how wide spread these devices have become.

The WebSocket Protocol (Fette and Melnikov, 2011) allows two-way communication in browser-based applications across a single HTTP connection. This protocol allows for the smooth interactions between server and client required for musical performance and reduces network overhead when compared with prior methods such as polling. According to StatCounter the most used browsers between October 2014 and October 2015 were Chrome, Safari, Internet Explorer and Firefox respectively (Figure 1), interestingly all of these browsers offer support for WebSockets making this choice of protocol a relatively safe one in terms of future developments of NetScore.

The music community is not without existing solutions to the problem of interacting with real-time notation through browser interfaces however. *Flight*, *Melodius*, *Scorio and Flat,* are all commercial solutions that offer in browser score viewing functionality. The GUIDO HTTP server exposes much of the GUIDO API

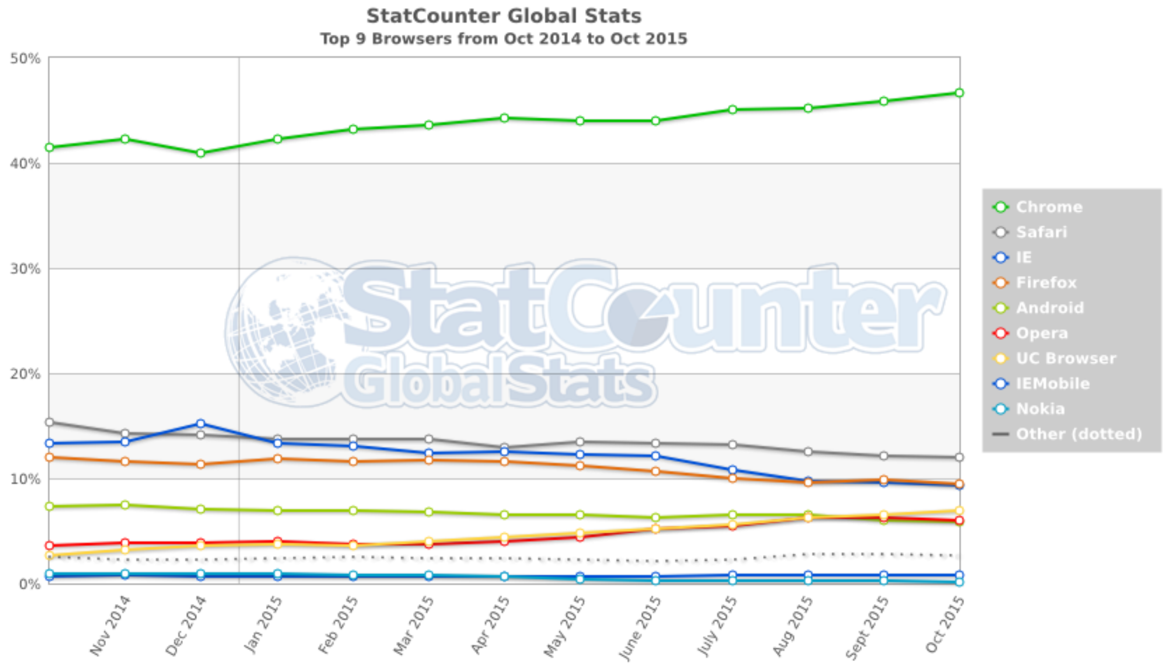


Figure 1. Browser usage statistics between October 2014 and October 2015, these popular browsers all support WebSockets a useful development for composers of real-time generated scores.

making it possible to create scores online using only open source software [7]. *Scribe* (https://cruncher.ch/blog/scribe/) a JS Library that allows the user to render music notation directly within web pages. *LeadSheet JS* performs a similar function but leans towards Jazz Leadsheet production [8]. INscore now offers WebSocket support and even in-browser interaction directly with the score material itself [9]. These solutions are highly customisable but they remain specific to the software packages that they are a part of and are therefore restricted to their own intended scope. The MaxScore package was already

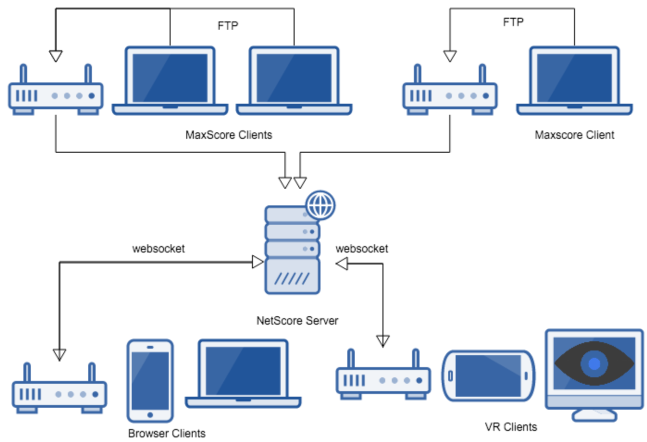


Figure 2. An example setup with NetScore, making use of its modular architecture to enable multiple users to interact remotely across a range of software and hardware platforms with minimal setup

designed to work over networks due to its close relationship to the development of Quintet.net. In a technical sense, its separation of the graphical and symbolic representations of scores, meaning that the score files can be efficiently rendered to disk as PNG files, is very useful for network music applications. As the scores can be quickly rendered as graphics files directly from the LCD object in Max/MSP, adding this functionality to the MaxScore package does not increase processor overheads significantly on the host machine.

2. NETSCORE IMPLEMENTATION

2.1 Four Separate Modules

NetScore operates using four separate components: a MaxScore Client, NetScore Server, MaxScore NetCanvas, and MaxScore VR-Client. Communication between these components is facilitated via FTP and WebSocket connections (Figure 2). With scalability a key focus of this project already, NetScore is based around a similar modular architecture, which allows for a high level of customisability. This lets users to explore a wide variety of composition and performance scenarios found in real-time and non-real-time composition such as live solo or group performance with e-scores, network music rehearsal and performance and VR collaboration. In fact a dedicated server terminal is not required, and one can even run all four components on the same workstation simultaneously if desired. This kind of setup is useful for testing a composition environment in its early development phase for example, as one can easily transport the various score data to wherever it is required using the server.

As platform independence is an important consideration in Max/MSP, MaxScore and Quintet.net, the NetScore Server application itself is executed in Java, much like JMSL [10], which MaxScore relies on for data processing and storage. The NetScore helpfile is a typical Max/MSP helpfile. It provides an example of how to generate a MaxScore NetCanvas file though automated editing of HTML and JavaScript files, creating a small webpage. This webpage is customised with the websocket address of the host server. The resulting html file can be transferred to the potential client via whichever method they choose. Users are encouraged to create their own implementation, so as to take advantage of the flexibility offered within the Max/MSP and MaxScore environment. The VR client was created using Unity 5 and currently supports the Oculus Rift DK2.

(https://www.oculus.com/en-us/) with front mounted or tabletop mounted Leap Motion camera on OSX, and Google Cardboard on Android, with a view to create a Windows and iOS version down the track. It relies on the C# WebSocket implementation *WebSocket Sharp*, which provides bi-directional communication over a single TCP Port as is needed for use with the NetScore server. (https://github.com/sta/WebSocket-sharp). The user must manually select the IP of the WebSocket server in order to achieve a connection.

2.2 WebSocket Image Server Implementation with Jetty 7

The advantage of running a separate application as our WebSocket server is that multiple remote connections are possible, allowing for scenarios where different MaxScore users are concurrently creating scores and uploading them to the appropriate locations on the server, which are being continuously scanned for changes. The user is still free to run the app locally of course, which could be useful as in a situation where local performance is to be the outcome and the networking functionality is less important. We explored the potential for using node.js from within Max, as well as other Java WebSocket libraries, which were supported with the older JVM. In the end we adapted source code developed by Desmond Shaw as the basis for the server application (http://www.codepool.biz/how-to-implement-a-java-WebSocket-server-for-image-transmission-with-jetty.html), which we then modified to suit our purposes adding functionality for handling the folder scanning operations and our own filename referencing system for requesting scores from the client side. After the handshake is completed the client transmits a part number request (user definable through the client interface) and the desired part is delivered. The server understands any integer as an instruction starting a part request, where a "0" stops the server from scanning the folder for changes for that particular client. Each WebSocket connection is handled in a separate worker thread. In the console window messages are displayed indicating the network activity of all connected clients, and notifies the user of any file changes. Only one copy of the server application may be running locally at any one time per machine.

2.3 Image Handling

The PNG files generated by MaxScore can be forwarded via FTP or locally copied to a specific folder hierarchy on the terminal running NetScore Server with a filename corresponding to which part they represent. The server in turn creates WebSocket connections to clients who have successfully completed the handshake and reports their IP to a console window, and sends the requested score whenever it changes. These PNG scores are "pushed" to the clients as byte arrays from the server, where they are converted into images, and are either applied as a texture in VR via a C# Unity script or displayed in a browser window. Graphic scores, with their larger file sizes are better supported by this method than if the page was to constantly refresh itself in the off chance that the score file on the server had changed.

Currently PNG and JPEG formats are supported by the Server application, and the user can choose to manually transmit these file types to all connected clients using the *Load* and *Send* buttons in the server GUI, for testing purposes (Figure 3). Images are translated into byte arrays, transmitted and then converted back on the other end by the client into graphics as with other connections. A separate thread is run per connected NetCanvas Client or VR client instance instead of it restricting an IP address to a single connection, meaning that a client can open multiple browser windows and simultaneously stream different parts. On the client side, in the MaxScore NetCanvas for example, the byte arrays are reloaded into the page as images.

2.4 Client and Server Graphical User Interfaces

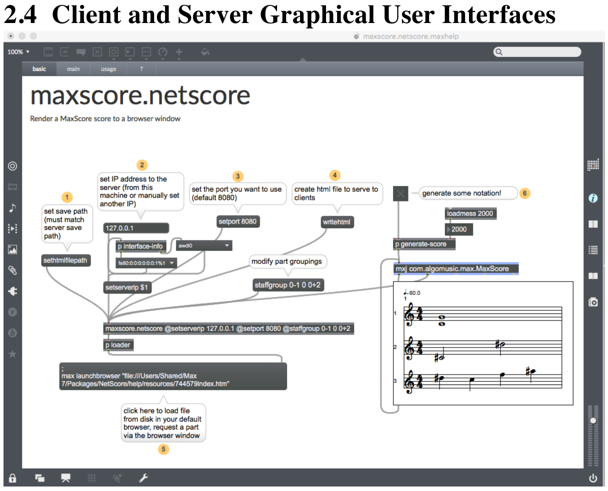


Figure 3. Running multiple browser windows displaying different parts.

2.4.1 Max/MSP Help File

The Max/MSP help file illustrates the execution of a real-time composition using MaxScore and NetScore to customise a website to send to clients, and generate a composition. The user can follow the steps outlined to obtain the current external or local network IP address, which can be used to build a small webpage that is sent to users via email or other methods such as Apple's *AirDrop* or local file transfer. The file path where the score files are to be saved must be specified here, but alternatively the user can opt to send their image files to the MaxScore Server via FTP. An example is given whereby the user can generate a random real-time piece of music with the help file.

2.4.2 Server Side

The MaxScore Server has a simple interface consisting of a load and send button mainly for testing if a connection has been achieved. On launch, the user is prompted to select a location for the score files. This is the folder that will be scanned for changes in files with the appropriate file names and extensions. The main window displays messages regarding the connection status of clients and folder activity. It also reports every time it forwards an image to a client and when a text message or file request is received from a client. Any errors associated with the connection are also reported here.

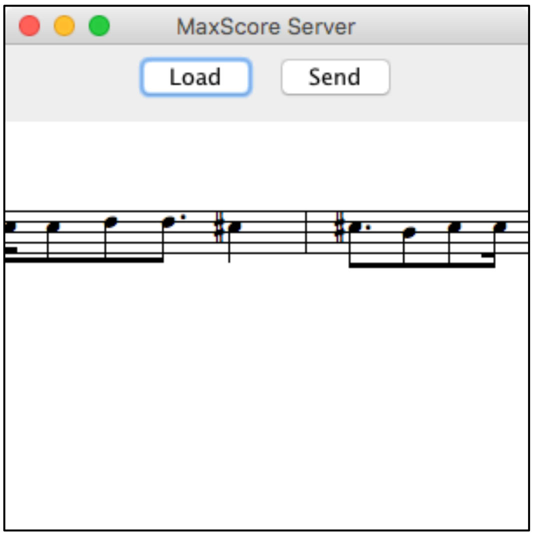


Figure 3. The Server interface, with its manual load and send buttons, displaying a score file about to be sent across a network.

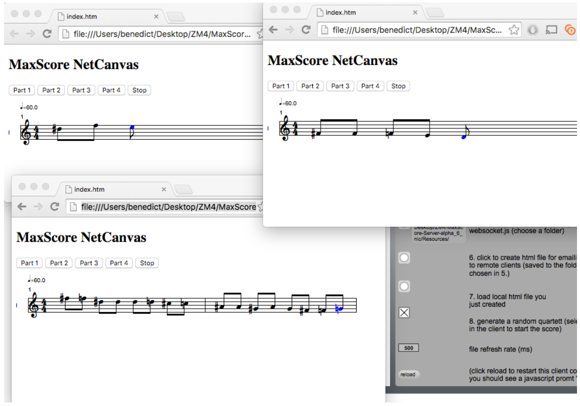


Figure 4. Running multiple browser windows displaying different parts.

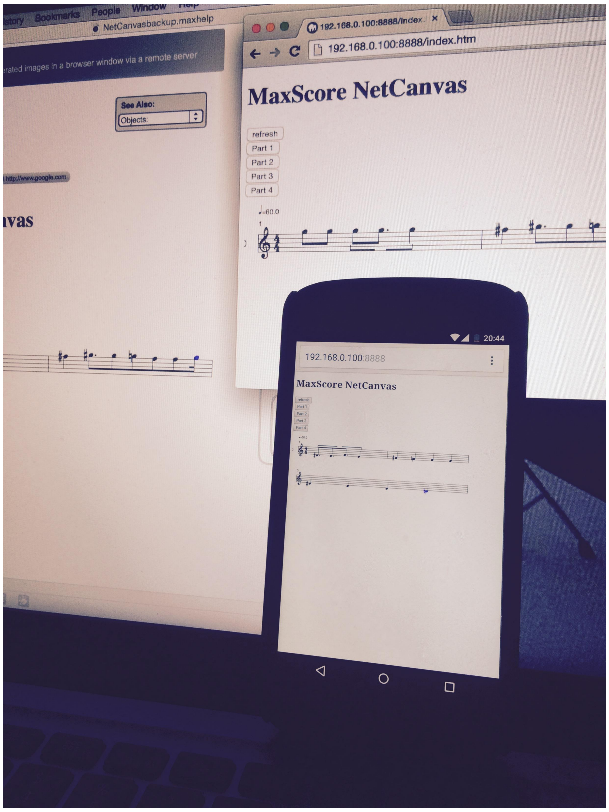


Figure 5. MaxScore NetCanvas. This browser based image client makes it possible to view realtime generated scores (or any picture file for that matter) on iPads, iPhones, Android mobiles, Laptops - basically anything that can run a modern browser.

2.4.3 MaxScore NetCanvas

The MaxScore NetCanvas (Figure 5) consists of a number of buttons for requesting scores from the server, a score title and the contents of the most recently delivered PNG file. Since it is created based on a simple HTML and JS webpage, the user is free to customise this interface through text editing of the original resources, located in the installation folder. Currently the demo interface supports selection of separate parts and a stop refresh button (figure 4). These can easily be pinched to enlarge or flipped to rotate the orientation on iOS and Android devices as is typical of a normal webpage, if it is supported by the device.

2.4.4 Virtual Reality Client

The Virtual Reality client is built in Unity 5 as a collaboartive environment reminiscent of the stage layout of past Quintet.net performances (Figure 5). Although the MaxScore Server is required to serve scores to the environment, (in the same fashion as scores are served to browser clients), the actual multiplayer functionality is handled by Photon server (https://www.photonengine.com/en/PUN). This free service supports up to 20 simultaneous player connections and delivers a high performance low latency interaction. Upon launching the application, each player spawns into the next available spawn spot, set up in front of a virtual score. It is currently possible to track the users hands if they are using a leap motion controller to facilitate conducting or performance of virtual reality instruments designed in Max/MSP. Each VR score in the environment has an attached C# script which handles both the websocket connection and applies the incoming byte array as a texture to the object to which it is attached (Figure 5). This means that for each connected player, the scores in their own local version of the client will change in time, even if latency has affected the synchronization of a fellow player's avatar. A Leap motion controller can be connected in the case of the OS X client, whose movements are mapped directly onto the avatars skeleton, creating convincing interactions appropriate for a convincing VR performance. Players are able to select virtual instruments running in Max/MSP, communication is facilitated with a customised version of the *Max/Unity Interoperability Toolkit* [10].

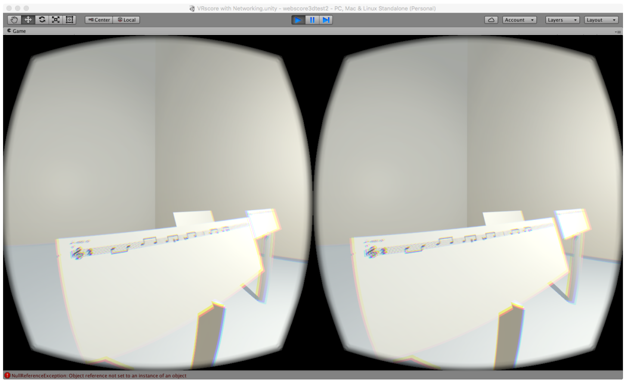


Figure 5. – Above, C# script to receive texture data via WebSocket and apply to surface, below, Virtual reality client with notation texture applied to music stand.

```
using UnityEngine;
using System.Collections;
using System;
using WebSocketSharp;

public class socketTexture : MonoBehaviour {
    // Use this for initialization

    public int partNumber;

    IEnumerator Start () {
        WebSocket toJava = new WebSocket(new Uri("ws://192.168.0.10
        yield return StartCoroutine(toJava.Connect());
        int i=0;
        Debug.Log ("Connected");
        yield return new WaitForSeconds(partNumber);
    while (true)
    {
        if (i<100) {
            {
            String fromSt = ("0");
            toJava.SendString (fromSt);
            //Debug.Log ("Sent:" + partNumber);
            yield return new WaitForSeconds(1);
            string toTex = toJava.RecvString();
            Debug.Log ("Received: "+ toTex);
            var tex = new Texture2D(2, 2);
            byte [] outTex = Convert.FromBase64String(toTex);
            tex.LoadImage(outTex);
            // Assign texture to renderer's material.
            GetComponent<Renderer>().material.mainTexture = tex;
            yield return 0;
        }
        }
        if (toJava.Error != null)
        {
            Debug.LogError ("Error: "+toJava.Error);
            break;
        }
    }
    toJava.Close ();
    }
}
```

3. FUTURE DIRECTIONS

Although the NetScore package itself offers some new possibilities for MaxScore users, it is still in its earliest phases of development. Firstly, there is a great deal of functionality we could still implement. Interactive elements from within the browser based canvas as well as the VR interface could take the project even further in the direction of the Quintet.net package. Even during the development phase we have already seen changes to how iOS handles HTML files via email for security reasons and have had to alter the way the customised MaxScore NetCanvas files are served to them. Keeping abreast with developments across multiple platforms presents additional challenges that will hopefully be made easier through our deliberate reliance on Java and Unity, which facilitate fairly rapid development across platforms. The VR client also has potential in the future to be used as a basis for further experiments and artistic work, and could hopefully become better integrated with the existing *Quintet.net* software, where it could act as an input device in addition to handling score display.

On the other hand using *libwebsockets* (https://libwebsockets.org/index.html) it may be possible to create a MaxObject that acts as a WebSocket Server but so far no implementation has been attempted. Another version of the server relying on MassMobile may be another way to achieve this [12]. If such a solution were possible, the NetScore package could be significantly simplified, without a loss of functionality.

It remains to be seen how NetScore will perform in performance scenarios, and benchmarking is planned for the future. The primary purpose of this proposed research would be to reduce latency to the point where scores can be delivered at a much higher rate than is possible with the current system, which utilises a 500 ms refresh rate, far to slow to facilitate animation for example, but certainly suffient for the puposes of extreme sight reading [11]. It is without a doubt a very exciting prospect to be able to control Ableton Live in LiveScore, a package of Live devices built with MaxScore, with our VR interface, so as to free the real-time composer from the distraction of a laptop screen, mouse and keyboard interaction system. Additional efforts to embed the fonts used for symbolic representation within MaxScore in SVG files using the Apache Batik SVG toolkit (https://xmlgraphics.apache.org/batik/) could potentially introduce client-side control over the resolution of scores. Using the ol.wsserver (http://olilarkin.blogspot.de/2014/01/olwsserver.html) object from Oliver Larkin or similar could handle resolution or other rendering specific requests to Max/MSP using the current system however and would be somewhat easier to implement.

4. CONCLUSION

By extending MaxScore to mobile devices and a virtual reality interface, the authors have achieved a welcome upgrade to an already feature rich software package with over a decade of previous development behind it. The ability to integrate notation into virtual reality environments is a novel development, and opens up a plethora of creative and technical possibilities to the Max/MSP/Jitter and Ableton Live communities.

Acknowledgments

The authors would like to thank he city of Hamburg for its financial assistance via its Landesforschungsförderung program and Desmond Shaw from Dynamsoft for contributing his source code.

5. REFERENCES

[1] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal,* 2015.

[2] N. Didkovsky and G. Hajdu, "Maxscore: Music notation in max/msp," in *Proceedings of the International Computer Music Conference*, 2008, pp. 483-486.

[3] J. Freeman and A. Colella, "Tools for real-time music notation," *Contemporary Music Review,* vol. 29, pp. 101-113, 2010.

[4] D. Fober, G. Gouilloux, Y. Orlarey, and S. Letz, "Distributing Music Scores to Mobile Platforms and to the Internet using INScore," 2015.

[5] J. Freeman and A. Colella, "Tools for real-time music notation," *Contemporary Music Review,* vol. 29, pp. 101-113, 2010.

[6] G. Hajdu, "Quintet. net: An environment for composing and performing music on the Internet," *Leonardo,* vol. 38, pp. 23-30, 2005.

[7] K. Renz and H. H. Hoos, "A WEB-based Approach to Music Notation using GUIDO," in *Proc. of*, 1998, p. 98.

[8] D. Martín, T. Neullas, and F. Pachet, "LEADSHEETJS: A JAVASCRIPT LIBRARY FOR ONLINE LEAD SHEET EDITING." (https://csl.sony.fr/downloads/papers/2015/dani-15a.pdf)

[9] N. Didkovsky and P. Burk, "Java Music Specification Language, an introduction and overview," in *Proceedings of the International Computer Music Conference*, 2001, pp. 123-126.

[10] I. I. Bukvic, *M Max-Unity3D Interoperability Toolkit*: Ann Arbor, MI: MPublishing, University of Michigan Library, 2009.

[11] J. Freeman, "Extreme sight-reading, mediated expression, and audience participation: Real-time music notation in live performance," *Computer Music Journal,* vol. 32, pp. 25-41, 2008.

[12] N. Weitzner, J. Freeman, S. Garrett, and Y. Chen, "massMobile–an audience participation framework," in *Proceedings of the New Interfaces For Musical Expression Conference*, 2012.

FEATUR.UX: An Approach to Leveraging Multitrack Information for Artistic Music Visualization

Ireti Olowe
Center for Digital Music[†]
Queen Mary
University of London
i.o.olowe@qmul.ac.uk

Mathieu Barthet[†]
m.barthet

Mick Grierson
EAVI Group, Depart. of Computing
Goldsmiths College
University of London
m.grierson@gold.ac.uk

Nick Bryan-Kinns[†]
n.bryan-kinns

ABSTRACT

FEATUR.UX (Feature - ous) is an audio visualisation tool, currently in the process of development, which proposes to introduce a new approach to sound visualisation using pre-mixed, independent multitracks and audio feature extraction. Sound visualisation is usually performed using a mixed mono or stereo track of audio. Audio feature extraction is commonly used in the field of music information retrieval to create search and recommendation systems for large music databases rather than generating live visualisations. Visualizing multitrack audio circumvents problems related to the source separation of mixed audio signals and presents an opportunity to examine interdependent relationships within and between separate streams of music. This novel approach to sound visualisation aims to provide an enhanced listening experience in a use case that employs non-tonal, non-notated forms of electronic music. Findings from prior research studies focused on live performance and preliminary quantitative results from a user survey have provided the basis from which to develop a prototype for an iterative design study that examines the impact of using multitrack audio and audio feature extraction within sound visualisation practice.

1. INTRODUCTION

Sound visualisation is primarily performed using mixed tracks of mono or stereo audio. One type of sound visualisation created for live performance features representations of the score, which exhibit temporal and tonal structures of music that complement the listening experience using an archetype of representation that elucidates the relationship between music perception and the musical staff. These visualisations commemorate the traditional notated form of written music and are not derived from characteristics of inherent signal properties, capturing performers' expression. Other examples of sound visualisation, which utilise the signal properties of audio signals to explore synchronization between sound and image, rely on databases of pre- and post- processed video clips and loops, and use a limited selection of common high-level audio feature extractors: e.g. tempo expressed in beats per minute (bpm), sound intensity, pitch, and timbre.

FEATURE.UX is a visualisation tool that allows users to create generative visualisations by connecting directed paths between nodes signifiers that represent an audio signal or audio feature, graphic methods of drawing and, color and threshold parameters within the primary workspace of a graphic user interface (GUI). Each completed path creates a separate, layered visual composition, which is exported to its own screen buffer to be displayed on a monitor or projected. Modular panels afford the user the flexibility to organise the interface within the workspace. Data visualisations monitor the behaviour of audio features in real time to indicate how mapped visuals respond to an audio signal. The ability to work with many instances of an audio signal or feature object provides users the opportunity to have greater and finer access to the sonic material, determine the complexity of layered visual compositions and to experiment with mapping strategies between audio features and visual properties.

2. BACKGROUND

2.1 Related Work

Applications of multitrack sound visualisation in literature are documented within the fields of data sonification, audiovisuals and sound visualisation.

Diniz, Demey and Leman modeled complex event structures in electronic music to develop a system for multilevel sonification of data [1]. Song and Beilharz explored perceptual musical characteristics such as timbre to identify aesthetic considerations during the sonification of multiple data streams through visual spatialisation [2].

Audiovisual systems such as the Reactable [3] and AV Clash [4] can be considered multitrack visualisation tools. Both visualize interdependent relationships between multiple, simultaneous streams of sound within a performance context. In contrast to visualisations that produce a visual representation of sonic information from an audio signal, these projects feature individual sound objects, which generate visuals that also synthesize the accompanying audio component of the performance. Relevant comparative models to our proposed FEATURE.UX system are those whose graphics and aesthetics are generated from the signal analysis of audio but whose visual components are not also tasked with providing a simultaneous audio accompaniment.

Throughout multitrack audio research, audio streams have been visualized to provide analysis and control. Hiraga, Mizaki and Fujishiro developed a system to review live performance as a method for performers to share an ensemble experience between co-players, compare different degrees of expression between performances and to understand the intent and mood of each performer [5].

Dahyot, Kelly and Kearney also explored the use of multitrack stream visualisation for live performance. Their objective was to utilise separate streams of data to control the environment in which artists performed. An individual stream of audio output from each band member is used to trigger lighting events and enable animations [6].

Soma is an applicable visualisation tool designed to offer real-time multitrack visualisations. The system renders visuals from separate streams of MIDI data for live performance. Bergstrom developed Soma to challenge typical mapping conventions of limited high-level features used to define the sound-image relationship. Bergstrom wanted to exceed the limited conventions of visualisation practice that mapped visuals to the beat or amplitude of music. He proposed to gain deeper access to lower level audio features to explore the elements of expectation: tension and surprise [7]. His system enables a group or an individual performer to display visual music generated from the output of performed instruments [8]. The visuals in Soma map to MIDI parameters: i.e. scales, notes, chords, tempo, volume and force.

Another aim of Soma was to create intuitive control interfaces to replace the ubiquitous use of knobs, buttons and sliders. The system is composed of a graphics rendering module, a module to monitor gestures and control input and a module to manage mapping between the renderer and control information. Real-time graphics are produced through the interpretation of MIDI sent via Open Sound Control (OSC) or multi-channel musical data processed using visual synths that produced unique effects [9].

The decision to forgo hardwired mapping between audition and visual domains allows a performer to generate dynamic graphics throughout the visual performance. Soma separates the role of the musician from the role of the visual mix engineer. While information is generated with played instruments, the visual engineer performs by deciding how to create links between the data and the visual synths [9].

2.2 Multitrack Audio

It is our hypothesis that a multitrack approach allows the generated visualisations to feed upon a richer, more abundant source of data to produce a more complete representation of its characteristics and expression.

A major difference between using a single track of mixed or multiple, pre-mixed tracks of audio as the data source from which to generate visualisations lies in the amount of available, accessible and employable information. Mixing and mastering individual stems to produce a final mix may significantly alter the fidelity of audio features, depending on the feature extracted from the pre-mixed tracks. In cases where the processes fail to preserve the sonic distinction between audio tracks, particular traits of the sonic information within individual stems may be lost. The benefit of multitrack visualisation offers a richer pool of data from which to extract musical features, map parameters and exhibit their behaviours and relationships in the visual domain. Fazekas demonstrates that tracks analyzed independently impart information that would be nullified by the mixing process [10]. As stated by Hargreaves, independently analyzed tracks avoid occlusion within salient portions of audio in individual tracks that become difficult to isolate and analyze in mixed signals [11].

Software tools for audio and composition analysis have employed multitrack audio. TaCEM was developed to study the influences of technology on electroacoustic music composition. [1] Coupries EAnalysis framework sought to introduce new composition tools through the exploration of graphic representation. [2] Providing support for multitrack audio in creative software is gaining popularity. VDMX [3] routes audio signals over OSC from Ableton Live using Soundflower. Magic Music Visuals [4] also supports multitrack audio. Other popular software packages such as Quartz Composer, [5] Pure Datas Gem, [6] Max/MSPs Jitter, [7] Touch Designer, [8] and VVVV, [9] which allow users to fabricate their own tools by patching modular objects together can also support multitrack audio if assembled by their user.

Native Instruments, a manufacturer of hardware and software audio production and performance instruments, has developed and introduced the Stem, [10] an open sourced, audio file format built upon the MP4 framework. The Stem file format incorporates five separate audio stems. A stem is an independent track of audio that may be mixed with additional stems during the production of music to compose a mixed and mastered mono, stereo or multichannel audio file. Each of four stems within the Stem format file holds one dedicated stream of audio, e.g. drums, vocals, bass, harmony. The fifth stem holds the original stereo master of the mixed composition. This file format lets one independently interact, modify, and, isolate or combine playback of any one or more streams of audio in real time. We plan to support the Stem format in future iterations of FEATURE.UX.

2.3 Feature Extraction

Feature extraction can operate on a time-varying or steady audio signal. A signal is partitioned into shorter segments during which a signal can be considered to be locally invariant. The representation of an audio signal can be extracted from within the time domain directly from the wave-

[1] http://www.hud.ac.uk/research/researchcentres/tacem
[2] http://logiciels.pierrecouprie.fr
[3] https://vidvox.net/
[4] https://magicmusicvisuals.com/
[5] http://quartzcomposer.com/
[6] https://puredata.info/
[7] https://cycling74.com/
[8] http://www.derivative.ca/
[9] https://vvvv.org/
[10] http://www.stems-music.com/

form or after the signal has been transformed into the frequency domain to disclose its spectral characteristics [12]. An audio signal, analyzed either locally by frame or globally over longer durations of time, reveals structural or semantic properties descriptive keywords defined from familiar language used to describe their sonic characteristics [13] by which they can be classified and understood [14].

Common audio features, derived from music composed of pitched sound objects of short duration and fixed timbre organised into larger structures are identifiable and quantifiable [15]. Wisharts quote categorises properties of classical, contemporary and popular music (the current emergence of electronically-tinged popular music, notwithstanding) whose formulaic, melodic and harmonious arrangements are constructed from phrase structures that constitute sonic events [16] [17]. The task of creating a representational visualisation of audio from music of this specification despite its complexity can be accomplished directly from the data, where known values can be extracted from the notation.

Music whose characteristics are exhibited through complex textures and shifting, evolving transformations rather than notes and chords are encompassed in what Edgard Varse coined organized sound in the 1920s [18]. This description has since evolved into a class of music that consists of many forms, structures and styles. [11] Electronic music is a variegated signifier that endeavors to describe the diverse methods of composition and aesthetics of all encompassed sub-genres it aims to define [17] [19]. The type of music with which this research is concerned is non-notated spectral music, which fails to provide neat numerical data by which to showcase its attributes no note or MIDI information may exist to appraise the contour of an envelope or detect the discrete distinct grains within layered scales of sonic or temporal structures [20]. Music Information Retrieval and audio content-based processing can help close the gap towards extracting musical and perceptually-relevant information from a non-notated, electronic music audio signal [21].

Dahan outlines problems with using computer analysis on electroacoustic music signals; tools are primarily developed to evaluate Western, tonal and pitched music. He suggests three MIR techniques that can be used to analyze electronic music signals. Firstly, using Mel-Frequency Cepstral Coefficients (MFCCs) produces perceptually-relevant impressions of timbre. Secondly, signal analysis of electronic music can take advantage of MIR pattern recognition techniques used to access audio repetition, which can be viewed as a trademark of certain electronic music genres, e.g. chiptune, dubstep, breakbeat and glitch. Also present in traditional music, repetition, presents itself in irregular patterns that can be appear within various time scales [20]. Finally, the segregation of sound may not be an obstacle as multitrack audio sources remain distinct [22].

2.4 Audio Visual Practice, Performance and Tools

Baker et al. collated writings from blog posts that contributed to an online community of 13 writers and 19 commenters [23]. Built around real-time live performance and media, the community discussed topics associated with real-time media from the perspective of the performer, performance and audience over a period of three months. Posts from the project expressed that its writers want to be introduced to new methods of performance, i.e. shared performances that break the limitations of a solo VJ presenter [23]. Comments suggested that VJs should have increased prominence when working with DJs who benefit from more notoriety and visibility [24]. Views expressed that the experience of the viewer should be less dependent upon their interpretation of the VJs artistic intent and that the causal relationship between sound and image should be easily discernible. The experience of live performance should be presented with new representations of time and innovative aural, visual and spatial aesthetic experiences [24].

Hook et al. analyzed the expressive interactions of VJs. The research team filmed a documentary with four VJ collectives as they transitioned between practice, preparation and performance. They hosted focus groups with the VJs and asked them to re-edit the documentary according to specific topics. The findings of the experiment were categorised by themes: the aspirational category focused on artistic intent, goals and desired outcomes for their performances; the interaction theme concentrated on the impact of interaction upon the VJs practice; and the live category addressed the significance of liveness that the VJs placed on their practice. Results indicated that the VJs want live visual performance to evolve and become an integral part of musical performance. They want software that facilitates visual improvisation, mutability and that is less dependent on rendering assets. The artists sought to obtain finer control of the audio, interact with the data and receive immediate feedback from their actions. The VJs expressed the need for flexible, reconfigurable GUIs and tools that influence creativity by affording fewer options. While the VJs would like to engage the audience by revealing the causal link between sound and image, they also expressed that they would like to constrain audience interaction [25].

Correia and Tanaka surveyed the landscape of existing computer-generative tools for audiovisual performance and composition. Taking a user-centered approach, they conducted interviews and hosted workshops focused on the expression and usability of tools, and audience engagement. Participants called for modular GUIs; integration of tools across a variety of software; and tools that afford greater expressibility, generative capabilities, flexible timelines, and the ability to expose the performers effort to the audience [26].

In VJ: Audio Visual Art + VJ Culture, author and artist, Faulkner, also known as D-Fuse, provided a passport to VJing by providing a thorough survey of audiovisual cultures, artists and resources from around the world. In his remarks, the artist calls for new methods of visualisation, perceptible expressions of structural relationships between the aural and visual domains in addition to aesthetics, the ability for the user to influence the audio portion of the VJ performance, and format agnostic software. He proposes a VJ practice that is less technical and reliant on amassed

[11] http://ears.pierrecouprie.fr/spip.php?rubrique3

collections of video assets, decreases the pre-production required to manage the assets, and that breaks away from looped-video presentations [27].

3. METHOD

3.1 User Survey

A combination of iterative design and thematic analysis will be used to evaluate FEATURE.UX as a tool to control parameterized graphics using audio features extracted from multitracks. Other areas for investigation include expressibility of the user, mapping between audio features and visual parameters, and usability.

The attributes of FEATURE.UX were selected after performing a survey of research literature focused around VJ tools, practice and theory and, live performance and interfaces. Also, a comprehensive survey was designed to collect information from live visualists, sound visualisation and audio-visual artists, and VJs. 31 open-ended questions were focused along six categories: experience, preparation, performance, mapping between audio and visual domains, multitrack audio and technological enhancement. Thirty opinion scale/Likert questions focused on multitrack sound visualisation, audio feature extraction and applied mapping strategies used to link sound with image. Demographic questions included queries about the subjects work samples for context. The online survey was hosted on the BOS online survey platform run by the University of Bristol and distributed via email, Twitter, Facebook and throughout several audiovisual communities, digital artist networks and commercial software forums. Quantitative results were evaluated from the responses of 22 participants (three women and 19 men aged from 21 to 57 years submitted surveys. The median age is 36.5 years.) Fifteen (68.2%) participants are professionals, six (27.3%) are amateur performers and one (4.5%) is a hobbyist.

3.1.1 Audio Stem Results

Overall, the findings show that the practice of using multitrack audio to create visuals is not prevalent enough to assess. More artists (27.2%) would use audio stems to create their visuals if given access than the 18.1% who stated they would not. But, 49.5% of the participants are at least satisfied with only having access to the stereo or mono mix to create visuals, 22.7% either felt indifferent or disinterested, and the same percentage, 22.7%, were not satisfied with having access to only a stereo or mono mix.

For 45.5% of the respondents, access to stems would provide greater control of the audio data with which to create visuals as shown in Fig. 1. But, the same 45.5% acknowledged that including stems within their workflow would make the process more complicated, some noting that the added complexity may not be worth the effort.

The availability of stems is not directly linked to the assessed quality of the finished composition. 50.0% felt that working with stems wouldnt necessarily make their visuals better, however, 18.2% agreed and 18.2% disagreed that employing them would make their visuals more meaningful. Also, 31.8% believed that having access to stems would create a stronger link between sound and image.

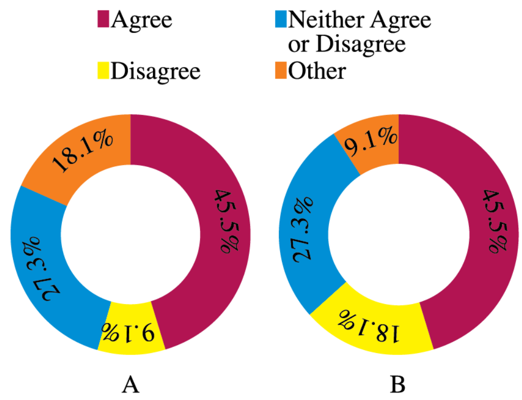


Figure 1. Chart A shows the percentage of participants who felt that having access to audio stems to create visuals would afford more control over the musical data. Chart B shows how they responded to adding complexity to their workflow by employing stems to create visuals.

Statements expressed concern that utilizing stems would affect audience engagement. 36.4% participants agreed and 36.4% disagreed that using stems to create visuals may render them too complicated for an audience to follow. Additionally, one artist (4.5%) offered the opinion that the assessment of complexity for particular visuals depends on the application.

3.1.2 Audio Feature Results

45.4% felt that additional audio features beyond what is already available are not needed to create visuals and 40.9% responded that the beat and the pitch are the most meaningful features to use. However, 59.1% neither agreed or disagreed, which may signify a deficiency of awareness about audio feature extraction as a tool for visualisation. Written responses stated that the lyrics, style of music and the extracted spectrum can be equally important.

Only 9.1% of the participants felt strongly uninterested in using additional features to create visuals. Although 45.4% of participants deemed access to more features unnecessary, 50.0% of the artists were at least interested in having access to more features as shown in Fig. 2.

Even though the results suggest that using multitracks would adversely affect audience engagement, 81.8% of the artists are not concerned that viewers may not understand the visuals created using additional audio features. The artists (40.9%) are more concerned with the synchronization between audio and visuals. 22.7% agreed and 22.7% disagreed that additional features will help create a tighter sync between sound and visuals.

3.1.3 Mapping Results

The results imply that the practice of mapping is more of an intuitive exercise than based on an exact system as shown in Fig. 3. 22.7% of artists felt that creating arbitrary pairings between audio features and visual parameters is an adequate method to create a link between the two domains. 36.4% disagreed. Furthermore, there is no consensus that

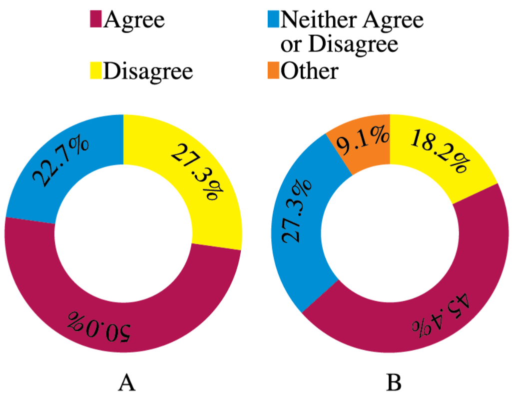


Figure 2. Artists expressed interest in having access to additional audio features with which to create their visuals as shown in Chart A. Although, as shown in Chart B, the participants overwhelmingly agreed that using additional audio features to create visuals is not essential to their practice.

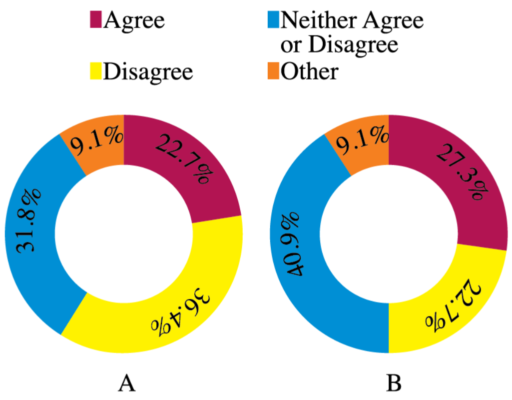


Figure 3. In Chart A, the artists assess mapping between audio parameters and visual attributes as a random exercise within their practice. Whether there exist established conventions between audio parameters and visual attributes is measured in Chart B.

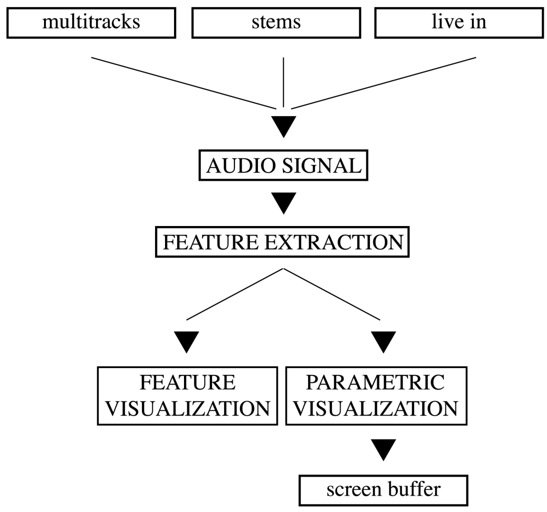


Figure 4. Schematic of the FEATUR.UX system.

there either are or arent established conventions between pairings, 27.3% agreed, 22.7% disagreed. Most (40.9%) responded neutrally. It follows that a majority of artists (72.7%) create their own rules when mapping between audio features and visual parameters.

Establishing a causal relationship between sound and image is a concern for 45.4% of the artists. 100.0% of the participants disagreed that an audience could only determine links between sound and image if the audio and visuals were mapped using a 1-to-1 mapping strategy, in which parameters in one domain are linked to one and only one feature in the other [28].

The degree of expression or meaning that an artist seeks to achieve within the visuals does not seem to be directly related to the implemented mapping strategies. Either a 1-to-many or many-to-1 relationship between sound and visuals allows for one parameter in one domain to be represented by more than one parameter in the other. Implementing one of these mapping strategies is more likely to increase the complexity of a composition since links between associated parameters are superimposed [28]. However, there is no consensus about the visible influence of executing these strategies. 68.2% neither agreed or disagreed that the link between sound and visuals can be expressed or distorted due to the employed mapping between the two and, 13.6% equally agreed and disagreed.

Despite the mapping strategy employed, 68.1% of the participants disagreed that audiences should be able to relate every sound event in the music with an accompanying visual. In addition, the viewer's interpretation, which may depend on the type and location of a performance [24], is not expected. 36.6% felt that the audience does not have to understand the visuals, even though 40.9% are interested in conveying meaning through their performances to those who experience them.

3.1.4 Quantitative Results Discussion

Most artists neither agreed or disagreed with 26 of the 30 Likert statements. Utilizing stems from multitrack audio and additional, uncommon audio features to create live visuals is not yet popular enough to build opinions about their impact on practice and performance. In addition, there is no established common language to distinguish between disciplines within live audiovisual practice and performance. Carvalho and Lund sampled the live audiovisual community to learn how practitioners define their own practice. The results of the 2014 international survey found that the boundaries and language used to define practices within visual music, expanded cinema, live cinema, VJing and audiovisual performance are continuously debated, fluid and ambiguous. Finding consensus about the practices involved within live audiovisual performance using terms like visuals and visualisation is difficult when their meanings are malleable and depend on their application, usage and context [24].

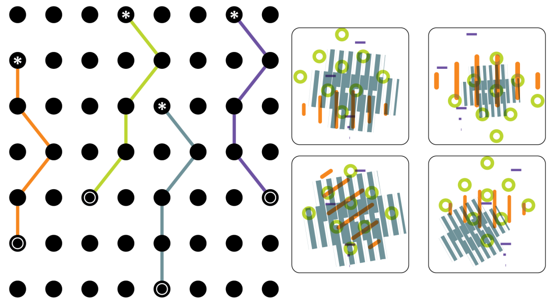


Figure 5. A representation of sound visualization within FEATUR.UX. (A) Directed paths are drawn within the workspace. The paths, which flow from top to bottom, begin with an audio signal pin (noted by an asterisk) and end with an output window pin (noted by a white circle). (B) The illustrations to the right of the directed paths represent consecutive frames of a visualization.

3.2 FEATUR.UX Prototype

The FEATUR.UX prototype is programmed using openFrameworks. [12] The ofxGui addon was used to build the user interface. Maximillian is an audio synthesis and signal processing library written in C++, whose addon, ofxMaxim, was chosen for its native real-time audio analysis and feature extraction capabilities. It is an easy-to-use framework with a collection of fundamental feature extractors commonly used for music information retrieval [29].

A tabulated list of desires and needs articulated by participants to evolve the practice and performance of live visuals, as mentioned in previous research discussed in Section 2.4, was used to choose objectives for FEATUR.UX. Given access to multitrack music and audio feature extractors, we hypothesize that the prototype affords users greater control of audio data and new methods to present visualisations. The technical load, reliance on amassed libraries of video assets, and pre- and post- processing requirements of traditional live visual practice and performance are eliminated by the use of computer generated methods of drawing. A modular GUI provides a flexible, adaptable workspace.

As shown in Fig. 5(A), users draw directed paths between nodes in a grid to create visualisations. This mapping model is inspired by the design and interactive interfaces of visual synths and offers a space for improvisation and creative spontaneity. The layered screen buffers to which the visuals are exported allow greater preferences for creating composites (synchronous and layered graphic visualisations). The limited palette of windows and menu options inspire the user to create with less.

3.2.1 UI Design of FEATUR.UX

The modular user interface, shown in Fig. 6, is composed of separate panels from which the user can, (A) start and stop audio playback, (B) view selected paths, (C) create directed paths within the workspace grid, (D) manually control playback of individual or groups of stems, (E) monitor the live waveform and spectrum, (F) view visualizations of

[12] http://openframeworks.cc/

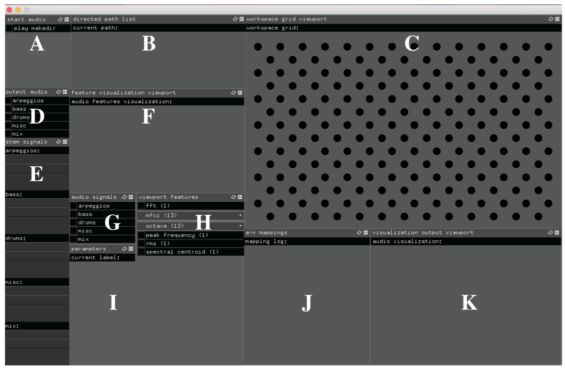


Figure 6. The modular GUI in FEATUR.UX.

the audio feature response to audio, (G) control which stem or group of stems is visualized in (F), (H) control which feature is visualized in (F), (I) manipulate parameters of graphic attributes, (J) monitor a log of which audio features are mapped to which visual attributes, and (K) view the composite visualizations created in the application.

3.2.2 Workspace Grid

The main workspace, shown in Fig. 6(C), is a grid within which the user can draw paths between pins. As shown in Fig. 7, a completed path starts with an input audio stream pin and ends with an output buffer window pin. The paths in between the input and output pins can include different combinations of pins that control color, thresholds and parameters for graphic methods of drawing.

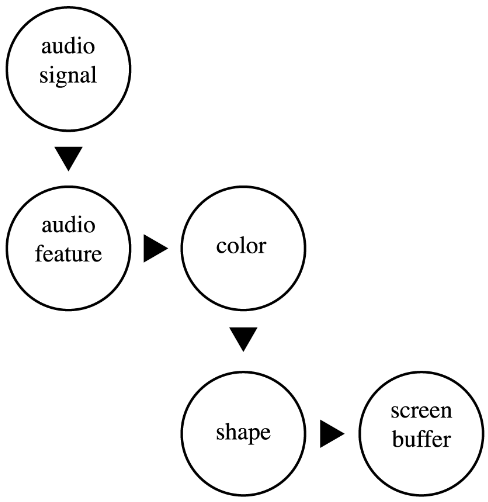


Figure 7. Data flow between pins.

One instance of an audio feature can be used to control the behavior of several visual parameters as shown in Fig. 8. One stem can be connected to multiple audio features, whose visualizations can be combined into a single layer in a shared screen buffer as shown in Fig. 9. And, many instances of any type can be used within a directed path as seen in Fig. 10. Also, if more than one audio stream is connected at the input pin of a path, the audio data used to generate the visualization is the mixed audio signal.

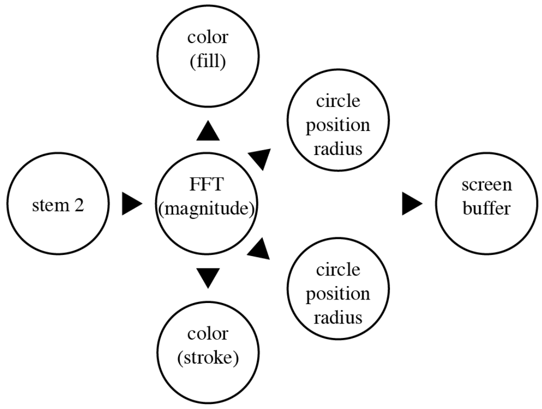


Figure 8. One stem and one audio feature is used in this path to control separate visual parameters as shown. A parameter of the FFT is used to manipulate the color, fill, stroke and position of circles as shown in the resulting layer composition in Fig. 11(A) and (B).

3.2.3 Audio Input Panel

There are three proposed cases in which multitrack audio can be imported into FEATUR.UX as seen in Fig. 4. In the case that the mixed audio track is not a sum of individual stems, the audio for the mixed track can be routed to the output channels while the user uses the data from the constituent stems to create visualisations. If the imported audio file is a Stem format file or the final mix is the sum of the separate stems (and the imported file is not a Stem format file), the user can control which audio streams are routed to the output channels while creating visualisations with the respective streams of data. Lastly, for live audio inputs, the audio for each stem is routed to the output channels while the user visualizes the live data.

3.2.4 Feature Visualisation Panel

FEATUR.UX lets the user monitor the response of audio features. The user selects a feature to visualize, as shown in Fig. 6(H), by choosing one or more audio stems as shown in Fig. 6(G). If more than one audio input is selected, the mixed audio is used to create the feature visualisation for the chosen extractor.

3.2.5 Dynamic Parameter Panel

Sections of this panel as shown in Fig. 6(I) appear only after a user selects a visual parameter pin along a closed path in the workspace grid. The FEATUR.UX interface is designed to offer access only to UI elements that are required to complete the task being considered.

3.2.6 Output from Screen Buffer Window

Each directed path in the workspace grid ends with a screen buffer as shown in Fig. 7. For every completed path, there exists a separate, layered visualisation in order of user preference, as shown in Fig. 11(A) through (F). Access to parameters to manipulate the appearance of the screen buffer are dynamically accessible as mentioned above. A composite image of a layered visualization is shown in Fig. 11(G).

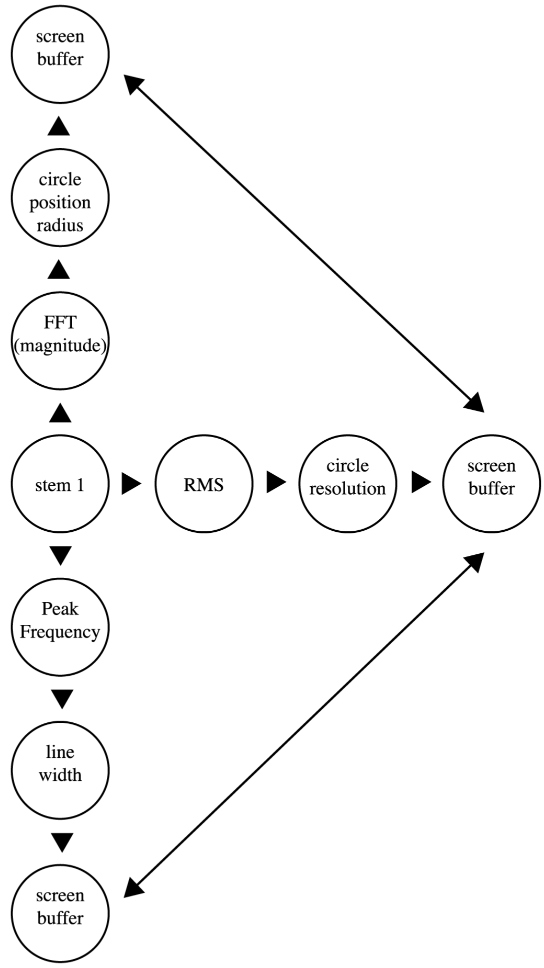


Figure 9. One stem and three audio features are used in this path to control separate visual parameters as shown in the resulting layer composition in Fig. 11(C) and (D).

3.2.7 Feature Extraction in FEATUR.UX

The following are the current audio features available in the FEATUR.UX prototype:

1. The Fast Fourier Transform extracts spectral information from an audio signal. The resulting complex signal is composed of a real and an imaginary part, which are used to calculate the magnitude and phase of the signal [30]. The FFT calculation performs as an auditory filter that mirrors, to some extent, the physiology within the human ear [31]. This perceptually-salient audio feature is a standard extractor used for sound visualisation.
2. MFCCs are a perceptual feature used to represent the timbral characteristics of an audio signal [32]. The representation of the short-term power spectrum is usually depicted using 8 - 20 of the first coefficients. The number of coefficients used can be adjusted based on the complexity of the signal [12]. Each of the coefficients can be isolated separately to monitor its behaviour and visualise.
3. The Chromagram, referred to as the Octave Analyzer in ofxMaxim, reveals the distribution of energy in an audio

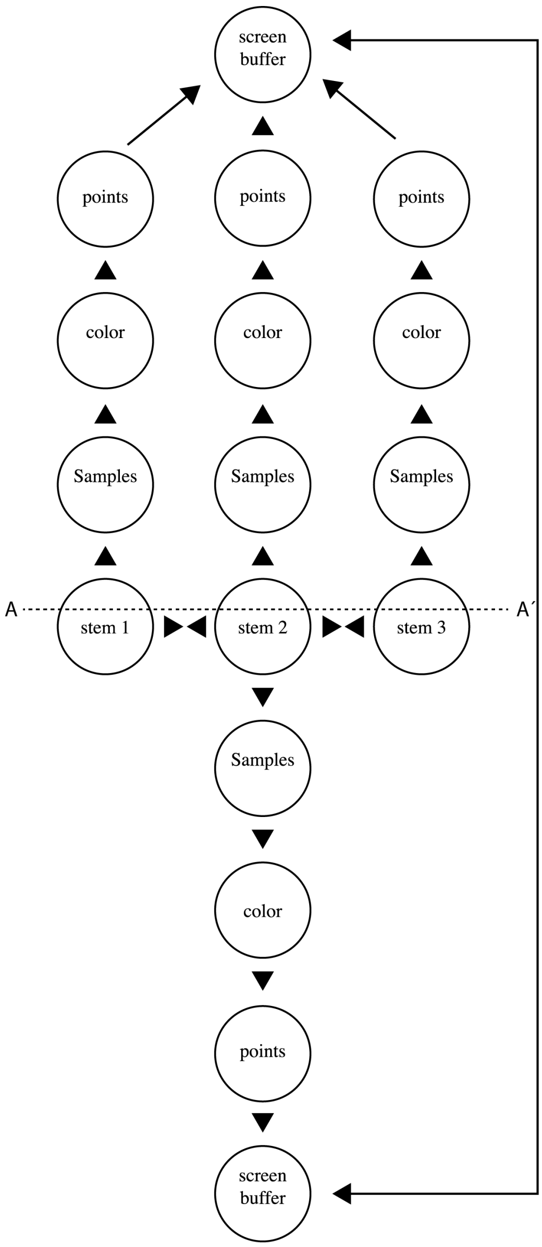


Figure 10. From the cross section A-A, downwards, the sample data is created with a mixed audio stream. Three stems are combined to influence the color and position of points. Upwards from the cross section A-A, shows how sample data from each stem can also be used to control the individual behavior of visual attributes. The visualisations that result from the featured path are shown in Fig. 11(E) and (F).

signal along a range of pitches. The dimension of tone height, where the range is segmented into octaves rather than pitch classes from traditional music scales [12] [21]. Each of the 12 pitch classes can be isolated separately to monitor its behaviour and visualise.

4. The waveform is an aggregate of compound sinusoidal waves and makes up the raw audio signal [32]. The waveform itself is not considered an audio feature, but is a ubiquitous method used to create sound visualisations.

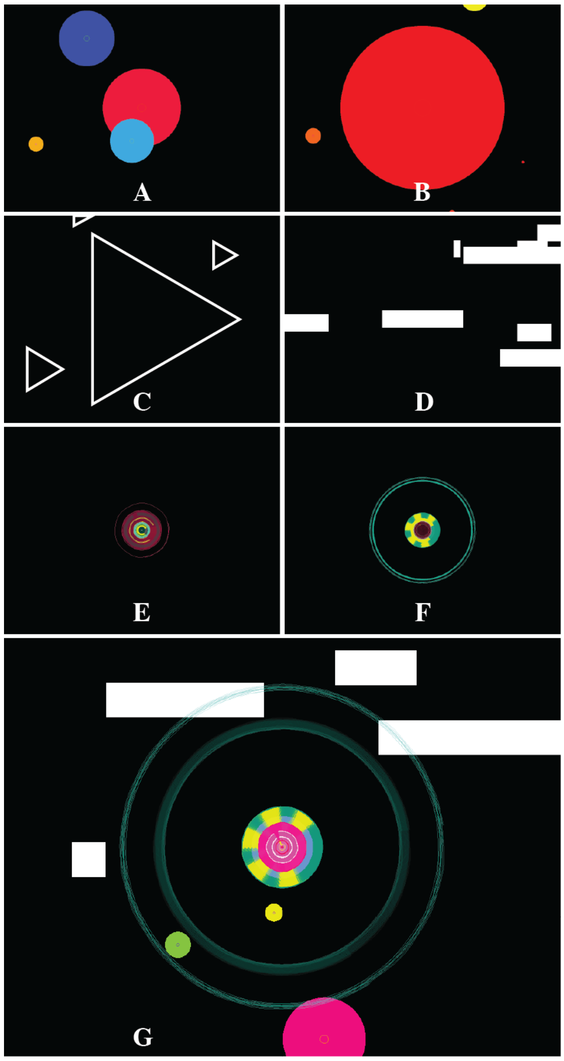


Figure 11. Images (A) and (B) display visualizations generated from the directed path shown in Fig. 8. The directed path shown in Fig. 9 generated the forms seen in images (C) and (D). The circle resolution is controlled by the RMS value, an indication of loudness. In image (C) where it is valued between 3.0 and 4.0 (The upper bound is exclusive), triangles are generated. In image (D), the RMS is at least 2.0 and at most 2.99, therefore a line is drawn. Larger RMS readings generate shapes that closer represent the circular form. Peak Frequency readings are expressed in the resulting line widths. The compound path shown in Fig. 10 is used to generate the meshes drawn in images (E) and (F) in which the graphic depicted in yellow represents the mixed audio stream. The meshes that represent the behaviour of the individual stems are rendered separately. A frame of a multi-layered visualization composed with the paths described in Fig. 8, Fig. 9 and Fig. 10 is shown in image (G). Wombatman6581 was used to generate this visualization. Musician Goto80 produced the song using a Commodore 64 with a 6581 SID-chip.

5. The peak frequency is the measure of the frequency bin with the highest magnitude within the spectrum of an audio signal. In some cases, it is an indicator of pitch, which may reveal the position of the fundamental frequency [33].

6. The spectral centroid is the frequency corresponding to the center of gravity of the energy spectrum. It is the threshold at which half of the energy is above or below that frequency. The measure of the spectral centroid relates to the perception of brightness or sharpness and quality of timbre that increases with the presence of high frequencies [12] [32].

7. The Root Mean Square (RMS) is a measure of signal intensity that evaluates the envelope of an audio signal and can be seen as a basic model of loudness of an audio signal [12] [32].

4. CONCLUSIONS AND FUTURE WORK

The adoption of multitrack audio for creative applications is still new and the use of stem technology is growing. [13] FEATURE.UX proposes to introduce a platform to apply multitrack audio towards live audio visual performance.

The quantitative results from the preliminary survey of a limited sample of participants reveal that introducing additional audio features and multitrack audio into the pipeline for developing live visuals is in its infancy. It is our assumption that the responses to the survey lacked clear motivations to use multitrack audio because there exists a lack of tools and opportunity to do so. With FEATURE.UX we aim to provide a framework to be able to test this hypothesis. The lack of awareness about audio features beyond the commonly exploited extractors and of multitrack stems is significant. Furthermore, the evaluation of the qualitative results thus far supports the earlier findings of Carvalho and Lund [24]. The qualitative results from the survey reveal that few participants use a common language to discuss topics related to live audiovisual practice and performance and, the departmentalization of the various disciplines within the audiovisual space creates a barrier that inhibits communication. Although at least 2 participants noted that they currently use stems in their audiovisual practice, the utility of multitrack audio visualization will remain unknown until it is experienced by more users. Additional studies will be conducted to learn how the community considers and implements mapping between sound and image and to further explore the use of audio features and stems to control parameters for generative computer visuals.

5. REFERENCES

[1] N. Correia Da Silva Diniz, M. Demey, and M. Leman, "An interactive framework for multilevel sonification," 2010.

[2] H. J. Song and K. Beilharz, "Aesthetic and auditory enhancements for multi-stream information sonification," in *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*. ACM, 2008, pp. 224–231.

[3] S. Jordà, "The reactable: tangible and tabletop music performance," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2010, pp. 2989–2994.

[4] N. Corriea, *Interactive Audiovisual Objects*. Aalto ARTS Books, 2013.

[5] R. Hiraga, R. Mizaki, and I. Fujishiro, "Performance visualization: a new challenge to music through visualization," in *Proceedings of the tenth ACM international conference on Multimedia*. ACM, 2002, pp. 239–242.

[6] R. Dahyot, G. Kearney, and C. Kelly, "Visual enhancement using multiple audio streams in live music performance," in *Audio Engineering Society Conference: 31st International Conference: New Directions in High Resolution Audio*. Audio Engineering Society, 2007.

[7] E. H. Margulis, "A model of melodic expectation," *Music Perception: An Interdisciplinary Journal*, vol. 22, no. 4, pp. 663–714, 2005.

[8] K. Brougher and O. Mattis, *Visual Music: Synaesthesia in Art and Music Since 1900*. Thames & Hudson, 2005.

[9] I. Bergstrom and R. B. Lotto, "Harnessing the enactive knowledge of musicians to allow the real-time performance of correlated music and computer graphics," *Leonardo*, vol. 42, no. 1, pp. 92–93, 2009.

[10] G. Fazekas and M. Sandler, "Structural decomposition of recorded vocal performances and it's application to intelligent audio editing," in *Audio Engineering Society Convention 123*. Audio Engineering Society, 2007.

[11] S. Hargreaves, A. Klapuri, and M. Sandler, "Structural segmentation of multitrack audio," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 10, pp. 2637–2647, 2012.

[12] D. Mitrović, M. Zeppelzauer, and C. Breiteneder, "Features for content-based audio retrieval," *Advances in computers*, vol. 78, pp. 71–150, 2010.

[13] R. Stables, S. Enderby, B. De Man, G. Fazekas, and J. Reiss, "Safe: A system for the extraction and retrieval of semantic audio descriptors," in *15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014.

[14] D. Smalley, "Spectromorphology: explaining sound-shapes," *Organised sound*, vol. 2, no. 02, pp. 107–126, 1997.

[15] T. Wishart, *On Sound Art*. Harwood Academic Publishers, 1996.

[16] N. Adams and M. Simoni, "Time-frequency visualization of electro-acoustic music," in *Enriching Scholarship Workshop, University of Michigan*, 2005.

[17] J. Humiecka-Jakubowska, "Electronic music in the perspective of semiotics," *Interdisciplinary Studies in Musicology*, no. 14, p. 258 273, 2014.

[13] http://www.stems-music.com/stems-partners/

[18] N. Valsamakis, "Aesthetics and techniques in the electroacoustic music of iannis xenakis," 2000.

[19] A. Cameron, *Instrumental Visions: Electronica, Music Video, and the Environmental Interface*. Oxford University Press, 2013.

[20] C. Roads, *Microsound*. MIT Press, 2001.

[21] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[22] K. Dahan, *Electroacoustic Music: Overcoming Analysis Paralysis*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2011.

[23] C. Baker, *VJam Theory*. Realtime Books, 2008.

[24] A. Carvalho and C. Lund, *The Audiovisual Breakthrough*. Fluctuating Images, 2015.

[25] J. Hook, D. Green, J. McCarthy, S. Taylor, P. Wright, and P. Olivier, "A vj centered exploration of expressive interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 1265–1274.

[26] N. N. Correia, A. Tanaka *et al.*, "User-centered design of a tool for interactive computer-generated audiovisuals," in *Proc. of 2nd Int. Conf. on Live Interfaces*, 2014.

[27] M. Faulkner, *VJ: Audio-Visual Art and VJ Culture*. Laurence King Publishing, Ltd., 2006.

[28] S. Callear, *Audiovisual Particles: Parameter Mapping as a Framework for Audiovisual Composition*. Bath Spa University, 2012. [Online]. Available: https://books.google.co.uk/books?id=TTvYoAEACAAJ

[29] M. Grierson, "Maximilian: A cross platform c++ audio synthesis library for artists learning to program," in *Proceedings of the International Computer Music Conference, New York*, 2010.

[30] S. W. Smith, *The Scientist and Engineers Guide to Digital Signal Processing*. California Technical Publishing, Ltd., 1997.

[31] S. Ravindran, K. Schlemmer, and D. V. Anderson, "A physiologically inspired method for audio classification," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 9, pp. 1–8, 2005.

[32] A. Lerch, *An Introduction to Audio Content Analysis*. John Wiley and Sons, Inc., 2012.

[33] O. Tchernichovski, F. Nottebohm, C. E. Ho, B. Pesaran, and P. P. Mitra, "A procedure for an automated measurement of song similarity," *Animal Behaviour*, vol. 59, no. 6, pp. 1167–1176, 2000.

A ROBUST ALGEBRAIC FRAMEWORK FOR HIGH-LEVEL MUSIC WRITING AND PROGRAMMING

David Janin
LaBRI CNRS & INRIA Bordeaux Sud-Ouest &
Bordeaux INP & Université de Bordeaux
351, cours de la libération
33 405 Talence cedex
janin@labri.fr

ABSTRACT

In this paper, we present a new algebraic model for music writing and programming. It is based on separating music object contents: what music they defined, and music object usage: how they can be combined. These are two orthogonal aspects of music representation/programming that should be kept separate although handled in a combined way.

From a mathematical point of view, music objects are modeled by means of some notion of tiled music graphs that can be combined by a single operator: the tiled sum. This operator is neither sequential nor parallel but both. The resulting algebraic structure is well studied in algebra: it is an inverse monoid.

From a programming point of view, our approach provides a high level domain specific language (DSL), the T-calculus, that is both reactive, hierarchical and modular. It is currently under implementation in the functional programming language *Haskell*.

From a representation point of view, various music examples are provided to show how notes, chords, melodies, musical meters and various kind of interpretation aspects can easily and robustly be modeled in this new formalism.

1. INTRODUCTION

1.1 From music programming languages

In the field of computer music, several music programming languages are available such as, to name but a few, the functional programming languages *Faust* [1] or *Euterpea* [2], the data flow programming languages *Max/MSP* [3] or *OpenMusic* [4], or a more imperative programming languages such as *CSound* [5] or *Super-Colider* [6].

As a matter of fact, all these music programming languages *are* music representation formalisms. Indeed, every music program can be seen as sort of a music score that describe the music that can be played. Moreover, using music programing language necessarily induces a mental representation of the music pieces that can be defined and the way they can be defined. This implicit representation *does* influence the perception of the user [7]. Thus one may question the adequacy of a given programming language as a representation formalism.

More precisely, when seen as music representation formalisms, music programming languages must be abstract enough to allow the transcription of the composer's creative intentions. Music-oriented program constructs must be available and implementation details must be made, as much as possible, implicit. For instance, adequate user interface as in *OpenMusic* may hide programming language syntax, replacing it by higher level, graphic-based editors. However, representing the music that is encoded by a given program is not, in general, an easy task.

Somehow on the contrary, when seen as programming languages, these formalisms must be effective. The written music must be playable. For such a purpose, some implementation details need to be provided. Standard requirements of software engineering must also be satisfied. For instance, every complex program should result from the composition, duplication and transformation of simpler programs.

1.2 From music representations to programs

To some extent, every music representation formalism can also be seen as music programming languages. Classical western music notation is a perfect example of such a fact. Numerous music editors allows to define music scores that, in turn, can be encoded into computer objects that can be played.

However, even though musical objects are rather well defined in music sheets, western notations do not provide a rich set of composition and transformation methods. Dedicated to written music, these notations suffice to write music, but not necessarily to create it. Even worse, the more or less adhoc functionalities offered by music editors may be musically ill-defined.

For instance, the standard cut, paste and insert transformations, essentially inherited from text editors, are inadequate because they often break the global consistency of the music. Indeed, inserting a sequence of notes in a score necessarily pushes the rest of the score regardless of the underlying musical meter.

Somehow paradoxically, music composition operators are well defined in music programming language. However, in a programing language, relevant music concepts may be lost in non musical technicalities. Specialized technician may be required to bridge the gap between composer ideas and their programming realisation.

1.3 Model based approach

One way to handle both music representation necessities and software engineering requirements is by deriving both of them from a unified model of musical objects with well understood algebraic properties.

Indeed, every music programming language, every music design software, as well as every music representation formalism induces a more or less implicit *music algebra* that defines both the basic objects that can be used and the combinators that allow to build complex music objects from simple ones. In the absence of a unified model, it is very likely that these algebras will be incoherent. In a model based this cannot happen as illustrated in Figure 1.

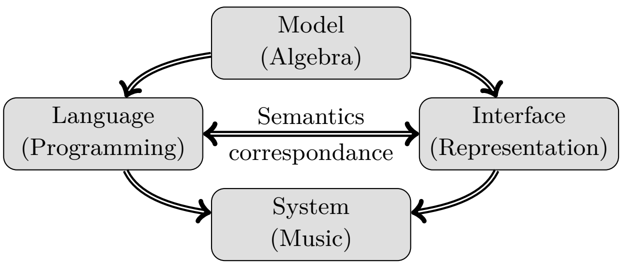


Figure 1: Model based approach.

Indeed, every functionality, be it defined at the programing level or at the representation level, necessarily derives from the algebraic model. Any (mental or explicit) representation induced by the usage of these functions converges to a single and coherent representation of the elements of this algebra. A correspondance between music programs and music representations becomes possible.

In the long term, this may even leads to the definition of a well founded graphic-based programming interafce, henceforth offering an easier access to programming techniques.

1.4 Known approaches

Technically, following such an approach leads to the design of a Domain Specific Languages (DSL). These languages are high-level programming languages dedicated to a specific application domain. They provide no more and no less than the necessary high-level constructs relevant to the underlying application domain [8]. In music programming, languages such as *Faust* [1] or *Euterpea* [2] are examples of DSLs.

In most of the proposed languages, musical objects are mainly combined by means of two operators: a sequential composition that allows to play two musical objects one after the other, and a parallel product that allows to start in parallel two musical objects. Hudak's notion of polymorphic temporal media [9] makes this algebra explicit. It allows to reason about programs and provides a better understanding of program's semantics.

However, it has already been argued [10] that the algebra induced by sequential and parallel operators, though rather natural at the DSP level provided by a language such as *Faust*, induces implementation oriented point that may not be that convenient in the case of high-level musical design.

Using the parallel (or rather fork) operator induces a "forward vision" of music writing, from the past to the future. It merely amounts to decide at any time *what comes next*. Musical metrics, synchronization points, visualized by bars in classical western music notations, quite disappear under such a unidirectional view of music writing [11]. As a matter of fact, typical music constructs necessitates a "backward vision", from the future to the past, that allows to decide at a given time *what comes before*.

Cadences such as $II^m/V^7/I$ or $II^m/I^7_\sharp/I$ are typical exemples of such phenomena. Resolving on a first degree constitutes a goal in the future. A cadence is a way to reach such a goal. It is thus implicitly understood as a construction from the future to the past. Another exemple is the anacrusis. Aiming at introducing a given note on a strong beat, an anacrusis is positioned in a backward way, depending on its length. It can be argued that such a phenomenon cannot be modeled properly with a forward point of view [10, 12].

1.5 Contribution and structure of the paper

Our goal is thus to define such a music algebra from which we derive both a programming language and a representation formalism.

For this purpose, aiming at relaxing the "forward" point of view induced by the sequential/parallel music algebra we have developed the notion of tiled modeling [13, 14] and tiled programming [15]. Both sequential and parallel composition operators are eventually merged into a single one : the tiled composition operator [12, 11]. This offers a higher-level point of view over the usual sequential/parallel music algebra.

In this paper, we present the latest development of this new music algebra and we eventually provide various and explicit music modeling examples. Doing so, aside the mathematical robustness of our approach that is already detailed in former presentations, we aim at illustrating the relevance of this approach for music modeling.

In Section 2 we describe a modeling example that essentially details what representation models we aim at achieving. How these models can generated from basic elements is detailed in the remaining sections.

In Section 3 we describe the formal model of tiled music graphs that are equipped with a single operator: the tiled composition. Properties of the resulting algebra that are relevant for music modeling are presented.

Then, in section 4, we show how these music graphs can simply be generated by means of music graph expressions. Music modeling examples are provided throughout all sections.

2. A MODELING EXAMPLE

Before getting into a detailed description of tiled models and programs, we provide in this section a first modeling example that illustrates most of the musical features we aim at capturing. The way such a model may actually be built, step by step, is described throughout the next sections.

2.1 Debussy's first arabesque

Debussy's first arabesque is a typical example of a music structure which is difficult to model by means of tree-based (or syntactic) modeling approach [16]. Indeed, as illustrated in the 91th and 92th bars of Debussy's first arabesque (see Figure 2), four voices, with distinct rhythmic structures, are interleaved. They induce rhythmic trees that have incompatible shapes and thus cannot be shared properly [17].

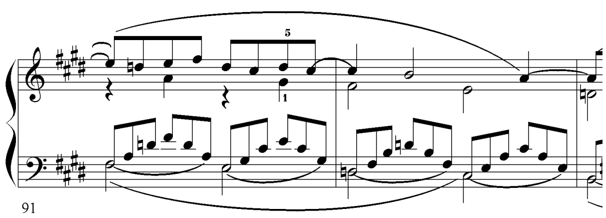


Figure 2: Two bars of Debussy's first arabesque.

Our modeling approach amounts to describe this piece of music by means of a music graph: a directed labeled graph vertices modeling instants in time and with edges modeling musical elements such as rests, notes or chords.

More precisely, the first half of Debussy's 91th bar is depicted in Figure 3. The various features that are appearing in this figure can be be detailled as follows :

- *vertices*: depicted by bullets (•); vertices are modeling instant in time that are called synchronization points,
- *directed edges*: depicted by arrows (• ⟶ •); edges are modeling basic musical such as rests, notes or chords that last for some duration : the time that elapse between the two synchronization points they relate,
- *upper or left edge labels*: depicted by (blue) positive rational numbers that appear either on the left or above the middle of each edge; these edge labels are modeling durations,
- *lower or right edge labels*: depicted by (red) strings that appear either on the right or below the middle of each edge; these edge symbols are giving the nature of rests, notes or chords.

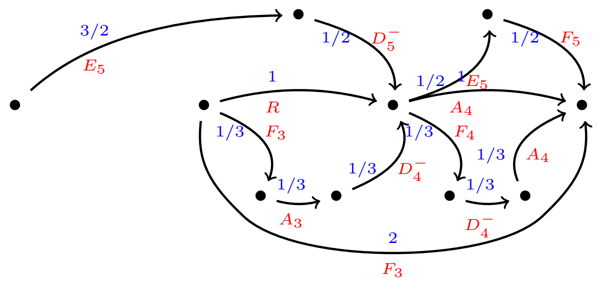


Figure 3: A graph model of part of the 91th bar.

Observe that the first E_5 is of duration 3/2 since the initial eight note E_5 is actually linked with a quarter note in the 90th bar, much in the same way the first A_4 quarter note in the 93th bar is linked with the quarter note A_4 at the end of the 92th bar.

On this graph, R indicates no notes, that is, a rest. One can also observe that all D have been written D^-. This models the fact that, as indicated in the key signature, all these notes must be lowered by one semitone, from $D^\sharp$ down to natural D.

A remarkable property of this graph model is that, if one counts a positive duration when traversing an edge forward and if one counts a negative duration when traversing an edge backward, then, the resulting sum of durations along any cycle always equals zero. Indeed, this follows from the fact that vertices are modeling instants in time henceforth every cyclic traversal amounts to go back and forth in time until the initial point is eventually reached.

Remark. One may observe that some musical elements are missing in this modeling. Indeed, neither the musical meter, nor the key, nor even the links indicating musical phrases are described.

However, as opposed no note duration or names, these attributes apply to groups of musical objects. We thus need a way to specify these groups or, as another way to say it, we need the algebraic tools that allow to generate these groups.

It follows that the modeling of group attributes is postponed to the last section when the music algebra will be defined.

2.2 Alternative graph representations

Of course the graph depicted in Figure 3 is not easily readable by humans. Another visualisation, more oriented towards human, is depicted in Figure 4.

In this figure, every vertex has been scattered along a vertical dashed line. These lines have also been labeled by their distance, expressed in quarter note durations, from the beginning of the bar. Clearly, such an illustrative vertex labeling can easily be computed from the duration labeling the edges, as soon as one vertex is chosen as the origin.

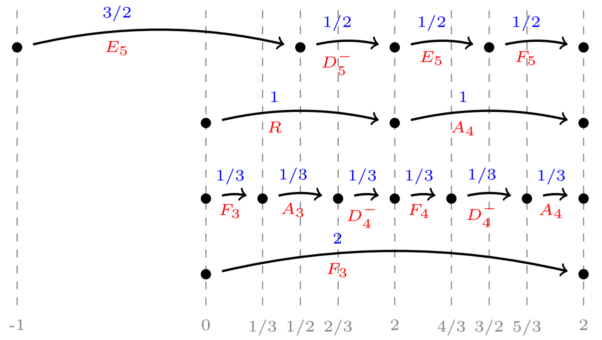


Figure 4: Scattered view of the same model.

Remark. Of course, other representations are possible such as, for instance, presentations based on cyclic or spiral shaped timelines. The reader should not make the confusion between the underlying mathematical models that are graphs, possibly unreadable, and their possible representations that may take various forms more readable.

3. TILED MUSIC GRAPHS

The models we aimed at building, music graphs, have been depicted in the previous section. As such, they almost form a music representation formalism [1] . The question we adress from now and throughout the remaining sections is how to generate such representations.

The resulting algebra is presented in two step. The first step is only concerned with generating timed graphs. It is presented in this section.

Then, it can be shown that such a modeling can simply be extended to music graphs by associating values to timed graph edges. As a result, we obtain an algebraic language for defining musical objects. Such a point of view is presented in the next section.

3.1 Timed graphs

We first aim are generating timed graphs, that is, a *directed acyclic graphs with labeled edges* with vertices representing synchronization points and edges labeled by the duration representing yet unspecified musical objects or rests. Examples of basic timed graph are depicted in Figure 5. They can be detailled as follows. In (5a), two musical objects of respective duration a and b are launched in parallel starting at the same time. In (5b), similar musical objects are played independently. There is no temporal dependency between

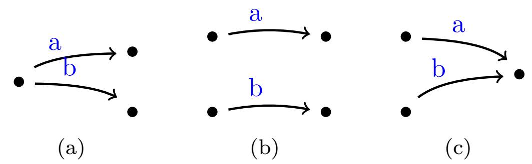


Figure 5: Basic timed graphs.

them. In (5c), two musical objects are finishing at the same time. In all these figures, there is no knowledge of the respective values of a and b that are presumably distinct.

3.2 Local unambiguity

By assumption, vertices are synchronization points in time. It follows that two musical objects with the same durations that are launched at the same time eventually are reaching the same synchronization point. Borrowing the vocabulary from automata theory, timed graphs are both *deterministic*.

Symmetrically, two musical objects with the same duration that end at the same time necessarily start from the same syncrhonization point. Timed graphs are also *co-deterministic*. In other words, timed graphs are bi-deterministic graphs or, as another way to say it [18], *locally unambiguous*.

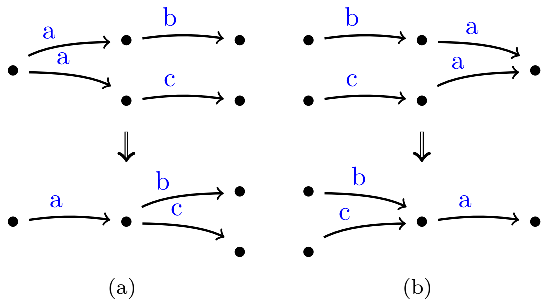


Figure 6: From ambiguous to unambiguous timed graphs.

It is shown in [18], there is a simple way to transform arbitrary directed graphs into its greatest locally unambiguous image. Indeed, it just amount to merge every pair of edges with the same duration label that share a common origin (see (6a)) or target (see (6b)).

Of course, this merging process needs to be repeated until the resulting graph is locally non ambiguous. Doing so, it may happend that acyclicity is lost; a directed cycle may appear. In this case, the timed graph is considered to be erroneous. In a derived programming tool, a design error is raised.

3.3 Synchronization attributes

Timed graphs are extended by two synchronization points: two distinguished vertices respectively called the input and the output root of the timed graph.

[1] up to the modeling of group attributes that we have postponed.

Resulting graphs are simply called birooted graphs or, to fit our application perspectives, birooted timed graphs.

Examples of birooted timed graphs are depicted in Figure 7 where input roots are depicted by (Υ) and output roots are depicted by (↓).

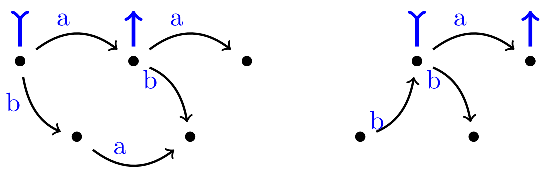


Figure 7: Two birooted timed graphs.

3.4 Tiled composition

These graphs can then be combined by means of the tiled composition. More precisely, these specified input and output roots allows to combine two musical objects [2] by gluing the input root of the first one with the input root of the second: this is the *synchronization step*. The local ambiguity that may results from these gluing is then removed following the bi-determinization process depicted above: this is the *fusion step*. The input (resp. output) root of the first object (resp. the second) is kept as the input root (resp. the output root) of the second object.

The birooted graph resulting from the composition of the two graphs depicted in Figure 7 is depicted in Figure 8 below.

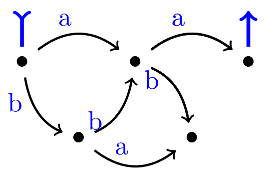


Figure 8: The result of a tiled composition.

It is know from inverse semigroup theory [19] that the resulting composition is associative. Since the single vertex graph with equal input and output root is clearly a neutral element for this composition, the resulting graph is known in algebra as a monoid. Moreover, it can also be shown that this monoid is an inverse monoid [20] (see also[18]).

From now on, such a composition is denoted additively, that is, the tiled composition of two tiled timed graphs t_1 and t_2 is denoted by $t_1 + t_2$. The single vertex graph with equal input is denoted by 0 and we clearly have $t + 0 = t = 0 + t$ for every tiled timed graph t.

[2] yet just birooted timed graphs, but the composition of general musical objects defined later in the text just follows the same rules,

3.5 Inverse, reset and coreset

The "inverse" arising from the underlying monoid is also denoted additively. In other words, for every tiled timed structure t, it is denoted by $-t$. It is just obtained from t by permuting the input and output roots, without changing the direction of the music.

This allows to define the difference $t_1 - t_2$ between two music graphs as the sum $t_1 + (-t_2)$. Then, we define the *reset* of t by $re(t) = t - t$ and the *coreset* of t by $co(t) = -t + t$. These remarkable elements are depicted in Figure 9 below.

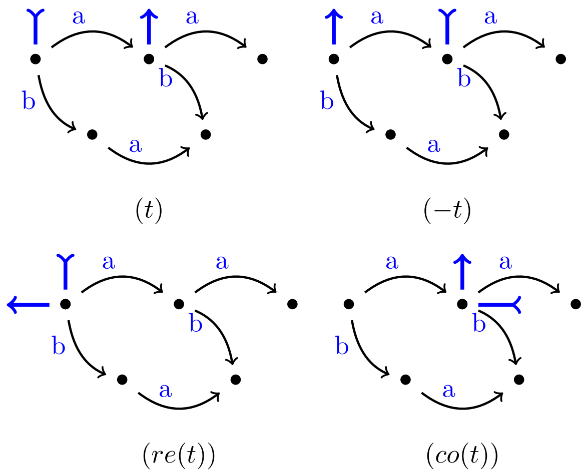


Figure 9: Resets and coresets.

Observe that, unless the music graph t is the graph of the zero duration rest 0, none of these expressions equals zero. However, as detailed in inverse semigroup theory, the reset and coreset defines *local zeros* as made explicit in the following equation that are always satisfied:

$$t = re(t) + t \text{ and } t = t + co(t)$$

3.6 Induced parallel composition

When building complex time structures (or later musical structures), a typical usage exemple of reset and coreset primitives are parallel insertions.

Indeed, given three tiles t_1, t_2 and, resp., t_3, simply denoting single edges of duration a, b and, resp., c as depicted in Figure 10.

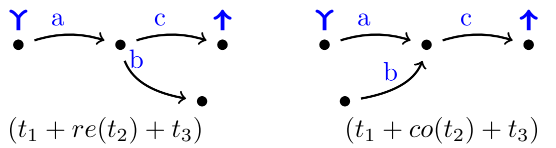


Figure 10: Parallel insertions.

A construction of the form $t_1 + re(t_2) + t_3$ inserts a copy of t_2 between t_1 and t_3 without altering the synchronization of t_1 and t_3. More precisely, it amounts to start t_2 at the end of t_1 in parallel with t_3, with all possible overlaps and merges allowed by tiles.

Symmetrically, a construction of the form $t_1+co(t_2)+t_3$ still inserts a copy of t_2 between t_1 and t_3. However, in this case, both t_1 and t_2 ends at the same time, before t_3 starts.

In both cases, synchronization between t_1 and t_3 are just the same as they would be in the sum $t_1 + t_3$ so both cases describe sort a parallel insertion.

These constructions offered by the reset and the coreset primitives constrast with the standard, string-based, insertion primitives that are available in most softwares for graphical music editions; classical insertions that would "push" appart t_1 and t_3.

3.7 Additional time equations

A carefull reading of the example depicted in Figure 8 above shows that more vertices gluing have been performed than those that were strictly necessary. This comes from the fact that we have additionally applied the equation

$$a = bb$$

that says that the duration of two bs just equals the duration of one a.

In other words, under the equation $a = bb$ two succesive b-labeled edges starting from the same starting synchronization point than an a-labeled edge eventually reach the same ending synchronization point. With a view towards application in music, such a feature allows to define standard timed graphs with length measured in hole notes, half notes, quarter notes, etc..., these relative lengths being easily defined by such kind of equations.

Observe that the graph depicted in Figure 3 just follows these rules. The semantics of duration labels such as 1, 1/2, 1/3, etc., ... has implicitly been defined by such a kind of equations. For instance, two successive edges of duration 1/2 eventually reach the same synchronization point than a single edge of duration 1.

4. THE RESULTING MUSIC ALGEBRA

Tiled timed graphs are turned into musical graphs by adding additional attributes to edges. In the proposed frameworks, these edge attributes can be sets of notes, possibly with some more attributes denoting instruments, tracks, velocity, etc.... Additionally, global (or group) attributes can also be defined over tiled musical graphs in order to describe some expressive features such as legato.

In this presentation, we restrict ourselves to the simpler case edge attributes are sets that are simply combined by union. Observe that such a set based modeling of edge attributes is already implicitly used in Figure 3 where rests are modeled by the attribute 0 that denotes the empty set of notes.

It can be shown that adding edge labels from a lattice does preserve the inverse monoid structure. In view of our modeling perpectives, we show in this section how this theory can be put in practise.

The resulting labeled tiled timed are from now on simply called music graphs.

4.1 Elementary music graphs

Elementary music graphs are either rests, denoted by R or a single note, of the form X_i^e where X is a pitch class (e.g. A, B, C, etc...), i is an octave, and e is a possible alteration. For instance $C_4^\sharp$ denotes C sharp on the fourth octave.

By default, all notes or rests duration equals one quarter. In other words, $C_4^\sharp$ as above actually denotes a quarter note. However, every note or rest (and later every score) can be stretch by some rational factor. Such a stretch is denoted by a left multiplication. For instance, the expressions $2 * D_5^\flat$ denotes the half note D flat on the fifth octave. Similarly, the expressions $1/2 * E_3$ denotes the eight note E on the third octave.

Of course, choosing quarter notes as unit duration is arbitrary. Our choice follows from the fact that, quite often, quarter notes represents one beats. Following english duration naming rules, it would certainly makes sense to use the duration of whole notes as duration unit. But this would just amounts to multiply all written melodies by 1/4 so this can be done in the last moment. In other words, the naming of duration can be left to the user.

By convention, single fractions are also seen as rests. For instance, the notation 2 is equivalent to $2 * R$. As a particular case, the notation 0 stands for the rest of duration zero. This implies that $d * 0 = 0$ for any stretch factor d.

4.2 Synchronized sums of music graphs

The tiled composition arising from the underlying monoid is written additively not to be confused with the stretch. In other words, given two music graphs t_1 and t_2, we denote by $t_1 + t_2$ the synchronization of the first music graph t_1 with the second one t_2.

Somehow as expected, for every music graph t, we have $t + 0 = 0 + t = t$.

For instance the following expressions denotes the beginning of a little waltz.

$$1/2 * (2 * C_4 + D_4 + 2 * E_4 + G_4 + 2 * E_4 + D_4 + 3 * E_4)$$

when played in 3/8. The resulting graph is depicted in Figure 11 below.

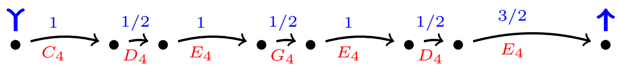


Figure 11: A little waltz.

Observe that the stretch operation distribute over the sum. For instance, the expression $1/2*(2*C_4+D_4)$ may as well be denoted by the equivalent expression $(C_4 + 1/2 * D_4)$.

4.3 Generalized product

The stretch of a music graph t by a factor d is thus expressed as $d * t$. Viewing d as a rest of length d, this means we allow certain product of musical graphs. This construction may be generalized. In fact, for every music graph t_1 and t_2, we put

$$t_1 * t_2 = re(|t_2| * t_1) + |t_1| * t_2$$

where, for every music graph t, we write $|t|$ for the elapsed time from the input root of t to the output root of t.

Using such a generalized product, we can thus add a bass line to our little waltz, by putting

$$C_3 * (2 * C_4 + D_4) + G_2 * (2 * E_4 + G_4) + \dots$$

The resulting music graph is depicted Figure 12.

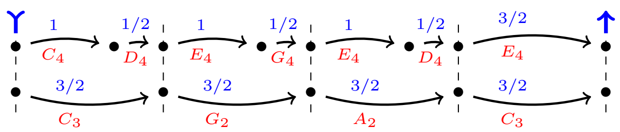


Figure 12: Adding a bass line.

4.4 Debussy example: the return

As an illustration of the tiled music algebra, we can build step by steps the musical graphs depicted in Figure 3. As already detailed above, the basic elements are notes with associated durations. Then these elementary tiles can be combined via sums, stretches, possibly taking inverses, resets or coresets.

For instance, the soprano voice of the 91th bar of the arabesque can then be defined by

$$v_1 = 1/2 * \left(2 + co\left(3 * E_5 + D_5^-\right)\right) + E_5 + F_5 + D_5^- + C_5 + re\left(D_5^- + 3 * C_5\right) + 2)$$

In this construction, a coreset (resp. a reset) is used to model the initial E_5 (resp. the final C_5) that starts before the beginning of the bar (resp after the end of the bar).

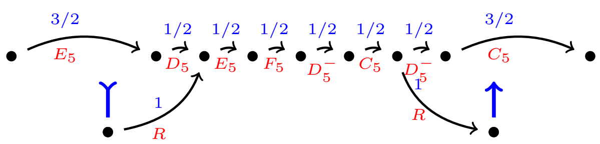


Figure 13: The soprano voice v_1 of the 91th bar.

The resulting music graph is depicted Figure 13.

The second voice, an alternation of rests and quarter notes, can simply be defined by $v_2 = 1 + A_4 + 1 + G_4$. Then, the two combined voices can be defined by $re(v_1) + v_2$ or, equivalently, $v_1 + co(v_2)$, among many other equivalent expressions since the distance between their input and output roots are both equal to 4.

The result of the combined first and second voices is depicted in Figure 14. Again, distances from the input root are labeling the vertical lines.

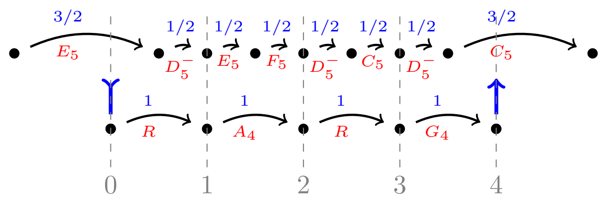


Figure 14: The combined first and second voices of the 91th bar in scattered representation.

Clearly, the entire 91th bar can be modeled this way, as well as the combination of all bars, till the complete Arabesque is modeled.

Remark. Compared to the tree-shaped model approach proposed in [16], one may observed that the terms defining voices are somehow similar : as terms there are indeed tree shaped. However, our graph based approach allows easily to combine time structures that have different shapes. Moreover, the parallel operators just follows from the inverse semigroup structures that provide inverses, resets and coresets.

Somehow implicitly, the tree-shaped terms defining birooted musical graphs are factorized by the relevant notion of musical equivalence (aka two terms are equivalent when they defined the same birooted graphs). The fact this equivalence is a congruence, that is, it remains stable under sum, is by no mean a trivial property.

4.5 Group attributes

We are now ready to address the question of group attributes. The main idea is that a group attribute such as links or time signature can be seen as a special edge attribute that is distributed over a group of edge. Since edge attributes are sets of values, it makes no difficulty to extend the possible values on edge by any additionel set of marks. The resulting algebra is essentially the same.

At the programmatic level it is convenient to specify only once any group attribute. This is simply done by adding another functions that add the attribute values to all edges of its tile argument. Assume that such a function is $att(t, v)$. Then, given two musical tile t_1 and t_2, we may specify that t_1 will be played with a 3/4 meter and, afterward, t_2 will be played with 4/4 meter just by the sum $att(t_1, 3/4) + att(t_2, 4/4)$.

At the representation level, a construction of the form $att(t, v)$ can be visualized by adding some background color on the underlying group of edges making explicit the "region" that have been affected by the group attribute value v.

Of course, in a sum of the form $att(t_1, v) + att(t_2, v)$ one may argue that the two underlying groups are implicitly merged since they seem to share the same group value. However, at the implementation level, the syntactic distinction between the first and the second group can simply be achieved by extending the value v by a single *id*, automatically generated, that refers uniquely to a given call to the function *att*. Such an encoding, invisible to the user, prevents any possible confusion.

Remark. It can be the case that some group attribute values cannot overlap. For instance, an edge labeled by both a 3/4 and a 4/4 meters would make no sense.

This amounts to say that we must forbid edge attribute that contains two incompatible group attribute. It occurs that such a property is stable under composition hence it forms a monoid ideal. At the algebraic level, all these tile can be merged [3] into a forbidden tile $\perp$. Forbidding music tile with cyclic underlying graphs is just handled the same way since the tile property "having a cycle" is also preserved by composition.

5. CONCLUSIONS

Starting from a fairly simple notion of music model, defined by means of instants in (symbolic) time that are related with elementary music objects, we have shown that adding synchronization points leads to a fairly robust algebra: an inverse monoid. This algebraic modeling provides, thanks to its richness, most of the music construct a composer may need, either as primitive constructs such as the tiled composition, or as a derived constructs such as the reset and the coreset functions.

All algebraic expressions and their corresponding graphical views that appear throughout our presentation proves that, based on the underling algebra, a robust correspondance between music programs and music representations can be developed. To some extent, the tasks of playing music or representing music have many common features that are made explicit in this approach.

The DSL induced by such an approach is currently under developpement. Embedded into a high level functional programming language such as *Haskell* and the various libraries available such as Euterpea [2] and UISF [21], it allows to inherit from its elegance and expressive power [12, 11, 22]. Design mistakes can be controlled both at the static level via the underlying type system and, at the dynamic level, thanks to the lazy evaluation mechanism of *Haskell*, via the evaluation of time or shape constraints.

Of course, this project is still at an experimental stage. Designing a new model and developing the related methods and tools necessarily take time [23], especially when we aim at revealing and exploiting the robustness of the underlying mathematical framework that may appear.

[3] technically, this amounts the take the quotient of music tiles by this ideal; such a quotient, well studied in semigroup theory, is called a Rees' quotient.

6. REFERENCES

[1] Y. Orlarey, D. Fober, and S. Letz, "Faust: an efficient functional approach to DSP programming," in *New Computationals Paradigms for Computer Music*. Editions Delatour France, 2009.

[2] P. Hudak, *The Haskell School of Music : From signals to Symphonies*, Yale University, Department of Computer Science, 2013.

[3] A. Cipriani and M. Giri, *Electronic Music and Sound Design - Theory and Practice with Max/Msp*. Contemponet, 2010.

[4] J. Bresson, C. Agon, and G. Assayag, "Visual Lisp / CLOS programming in OpenMusic," *Higher-Order and Symbolic Computation*, vol. 22, no. 1, 2009.

[5] B. Vercoe, "Csound: A manual for the audio processing system and supporting programs," MIT Media Lab, Tech. Rep., 1986.

[6] J. McCartney, "SuperCollider: a new real time synthesis language," in *Int. Computer Music Conference (ICMC)*, I. C. M. Association, Ed., 1996.

[7] M. Malt, *La représentation dans le cadre de la composition et de la musicologie assisteées par ordinateur*. Université de Strasbourg, 2105, habilitation à Diriger les Recherches.

[8] P. Hudak, "Keynote address - the promise of domain-specific languages," in *Proceedings of the Conference on Domain-Specific Languages (DSL)*, 1997.

[9] ——, "A sound and complete axiomatization of polymorphic temporal media," Department of Computer Science, Yale University, Tech. Rep. RR-1259, 2008.

[10] D. Janin, "Vers une modélisation combinatoire des structures rythmiques simples de la musique," *Revue Francophone d'Informatique Musicale (RFIM)*, vol. 2, 2012.

[11] P. Hudak and D. Janin, "From out-of-time design to in-time production of temporal media," LaBRI, Université de Bordeaux, Research report, 2015.

[12] ——, "Tiled polymorphic temporal media," in *Work. on Functional Art, Music, Modeling and Design (FARM)*. ACM Press, 2014, pp. 49–60.

[13] F. Berthaut, D. Janin, and B. Martin, "Advanced synchronization of audio or symbolic musical patterns: an algebraic approach," *International Journal of Semantic Computing*, vol. 6, no. 4, pp. 409–427, 2012.

[14] D. Janin, F. Berthaut, and M. Desainte-Catherine, "Multi-scale design of interactive music systems : the libTuiles experiment," in *Sound and Music Comp. (SMC)*, 2013.

[15] D. Janin, F. Berthaut, M. DeSainte-Catherine, Y. Orlarey, and S. Salvati, "The T-calculus : towards a structured programming of (musical) time and space," in *Work. on Functional Art, Music, Modeling and Design (FARM)*. ACM Press, 2013, pp. 23–34.

[16] F. Lerdahl and R. Jackendoff, *A generative theory of tonal music*, ser. series on cognitive theory and mental representation. MIT Press, 1983.

[17] D. Rizo, "Symbolic music comparison with tree data structures," Ph.D. dissertation, Universidad de Alicante, November 2010.

[18] D. Janin, "Inverse monoids of higher-dimensional strings," in *Int. Col. on Theor. Aspects of Comp. (ICTAC)*, ser. LNCS, vol. 9399, 2015.

[19] M. V. Lawson, *Inverse Semigroups : The theory of partial symmetries.* World Scientific, 1998.

[20] J. Stephen, "Presentations of inverse monoids," *Journal of Pure and Applied Algebra*, vol. 63, pp. 81–112, 1990.

[21] P. Hudak, D. Quick, M. Santolucito, and D. Winograd-Cort, "Real-time interactive music in haskell," in *Work. on Functional Art, Music, Modeling and Design (FARM)*. ACM, 2015, pp. 15–16.

[22] S. Archipoff, "An efficient implementation of tiled polymorphic temporal media," in *Work. on Functional Art, Music, Modeling and Design (FARM)*. ACM, 2015, pp. 25–34.

[23] W. Thomas, "Logic for computer science: The engineering challenge," in *Informatics - 10 Years Back, 10 Years Ahead.*, ser. LNCS, vol. 2000. Springer, 2001, pp. 257–267.

THE POSSIBILITIES OF A LINE: MARKING THE GLISSANDO IN WESTERN ART MUSIC

Cat Hope
Western Australian Academy of Performing Arts
c.hope@ecu.edu.au

Michael Terren
Western Australian Academy of Performing Arts
m.terren@ecu.edu.au

ABSTRACT

The glissando as it is deployed in Western art music notation carries with it a number of challenges to the hegemony of traditional harmony, rhythm, and notation. The glissando embodies the smooth line, unlike the striated pitch-time space of traditional Western music, which aligns the glissando to many philosophical concepts, as well as mathematical, scientific, and architectural disciplines. Select works by Iannis Xenakis, James Tenney and Giacinto Scelsi are discussed for their development of glissandi as integral formal components, especially around the glissando's tendency to encourage stasis. Compositional attempts to combine the nature of glissandi with drone in the author's own work are described, providing an examination of examples of the way glissandi and related concepts can be notated formally, rather than decoratively, in musical works.

1. INTRODUCTION

The glissando is unusual in music notation. Unlike almost every other sign for a sound and its relationship to time, the glissando seems to indicate a unique interpretation. It can be thought of as the ultimate analogue musical symbol, as opposed to the digital symbols used to represent music in most traditional notation. It visually describes a trajectory, a direction, a path - whereas a musical note represents the moment of attack, followed by 'implied' time. The glissando looks very much like it sounds, notwithstanding complications arising from instrument design, such as the difficulty for some woodwind instruments and the piano to slide smoothly between chromatic notes. Traditional music notes, tied together, don't represent time as clearly.

This idea of the line, and its relationship to time, has been contemplated by numerous philosophical movements over many years. Kant and Husserl held a "Newtonian view of time as linear succession … [as] unified, uninterrupted unfolding" [1, p. 81]. Derrida's concept of the trace conceives of lines as being inferred from a "series of arrests," instants in time, which demarcates and defines space and time in a linear way [2]. Deleuze, in his *Difference and Repetition*, writes that all repetitions are necessarily ordered in a "pure form [of time], or straight line," despite repetitions threatening to destabilise this linearity [3, p. 294]. Deleuze and Guattari bring their understanding of the line closer to music in their analysis of Pierre Boulez, in which they describe "smooth space" as embodying "continuous variation, continuous development of form… the pure act of the drawing of a diagonal across the vertical and the horizontal." [4, p. 480]

In music, it was the Greek composer Iannis Xenakis that really began a committed interrogation of lines and their relationship to music. For Xenakis, the line was something that united music, architecture and mathematics. He compared the straight line or curve of mathematics to a wave in physics, to a glissando or sine tone in music [2]. He often spoke of sonic shapes when discussing architecture, and his design of the iconic Philips Pavilion at the 1958 Brussels World Fair was also the initial composition plan for his work *Metastasis* (1955). "Composing music amounts to lay a series of points on a line," he suggested when discussing the relationship of his composition practice to that of his architectural one [5]. Yet for all his discussion of line and shape, and the existence of many drawings and plans for his musical compositions, Xenakis never embraced graphic notation as a system of communication to the performers; finding it imprecise, and giving away too much of his compositional responsibility to the performer [6, p 2]. Xenakis instead meticulously notated the contents of his long, overreaching glissandi using traditional notations. He often drew a line describing his glissandi above the staff, calling this a "Cartesian notation – graphical representations of sounds in a pitch-time space" [6, p 3]. In this way, he was indicating the intention in a way traditional notation could not.

The glissando can be thought of as a type of graphic notation that often appeared within traditional notations, making it a development of that notation rather than a separate movement altogether. A glissando extends what conventional musical notation is capable of depicting. Originating as an ornament attached to traditional notation, it developed into a more significant component of music fabric in the music of the postwar avant-garde. Electronic music, with its ability to make endless sound, and the use of sine tones to represent them, provided an interesting area for experimentation. The use of sirens in Varèse's *Hyperprism* (1922–23) provides an important early example of extended glissandi. The texturalist composers such as Krzysztof Penderecki and György Ligeti used the massed, plural glissandi as a way to emphasise attention on timbre. Others, such as George Crumb, who pioneered the now often despised 'seagull' effects in his work *Vox Balanae* (1971), extended the use of the glissando beyond a colouring technique. Composers such as Beat Furrer and Gloria Coates use glissandi to connect different pitches together over different tempi.

2. XENAKIS AND THE LINE

It could be argued that Xenakis uses glissandi to imply motion in his music, replacing the harmonic impetus provided in music dependent on traditional tonal systems. This can take the form of a direct trajectory between one note and the other, as exemplified in works such as *Metastasis*, or 'wandering' as found in the solo violin work *Mikka* (1971), where the music meanders between quarter tones or darts across large leaps. In these works, the glissandi provide a mechanism to make time audible in a way traditional notations cannot – they do not provide steps to make out time or rhythms. They are images that are heard – made up of time, rather than existing in it. They take the idea of drone, and put it into motion in a different pitch-time space than other kinds of notation.

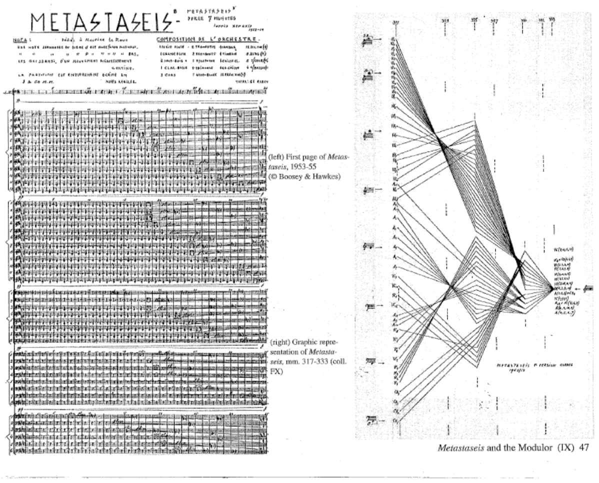


Figure 1: Iannis Xenakis, *Metastaseis* (1955) score and early sketch.

Despite the prevalence of the term "line" in musicology to suggest a trajectory or contour of a melody, these do not embody the line in the Euclidean sense of the word, due to the striated, stepwise nature of pitches in the chromatic scale in traditional harmonic music [7]. Glissandi have been described as having a more direct association with movement and motion than the traditional, striated notes of the European art-music scale [8]. As such, the relationship between the glissando and arguably more empirical fields such as physics, mathematics and architecture, becomes clearer in Xenakis' work.

In his electroacoustic work, glissandi have been used for decidedly less arithmetic means. In his work *La Légende d'Eer* (1977–78), glissandi take on a mimetic role, in its evocation of unhuman life-forms and environments. Allen S. Weiss also interprets the dense and unnerving glissandi of this work as stylising "those very same war sounds earlier valorized and sublimated by Filippo Marinetti and Luigi Russolo, most notably the Doppler effect of enharmonically changing pitch as shells pass overhead." [9]

Glissandi, when employed in this way, takes on a more diffuse role, more informed by metaphors of ancient Greek mythology than the non-ambiguity of Cartesian pitch-time space. They establish what Francis Bayer called a "relation of incertitude at the heart of sonorous matter, opposed to the somewhat artificial precision of articulated systems: one can even claim that in the *glissando* we are no longer dealing with precise tones, but with a sonorous ensemble movement where, on the spatial plane, only the general direction is really determinable" [10]. Isabella van Elferen writes that because the glissando marks a "continual transgression of harmonic, melodic, and often rhythmic boundaries," all of which constitute an understanding of time as linear, the glissando " destroys linear temporality and therefore necessitates the consideration of the impossible possibility of Being-outside-time" [11], a concept pertinent to *La Légende d'Eer*.

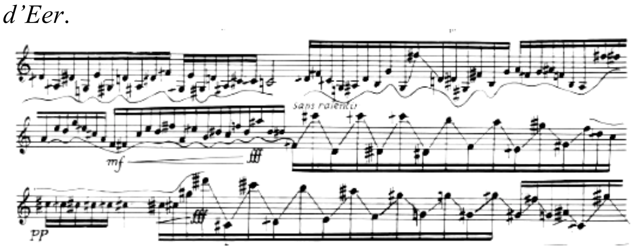



Figure 2: An excerpt from Iannis Xenakis' *Mikka*, showing 'wandering glissandi'.

3. GLISSANDI AND STASIS

The impact of electronic music on the employment of glissandi in music cannot be overstated. American composer James Tenney's *Postal Pieces* (1965-71) study three important musical elements; intonation, 'the swell' and perceptual states, with the swell being thought of as a series of interconnected glissandi. Tenney points out, "what we take to be the substance or content of some sound – say, a string quartet – is really the result of forms - formal shapes and structures at a microscopic, or 'microphonic level" [12].

For Tenney, musical form and the nature of sound are the same thing, and this is demonstrated in his notations. In *Postal Pieces* no. 9 '*Cellogram*' (1971), the movement of a sound wave is employed in a similar way to Xenakis's use of glissandi, but described very differently. He draws the sine tone into the stave, as glissandi. The choice of bass instruments ensures a clearly articulated and smooth, lengthy period movement of the sound wave, leaving the slow undulations to enable enough space to delicately weave between small differences in frequency.

Despite the ability of the glissando to depict movement, it also retains a close relationship to stasis, as it holds the possibility to indicate long form, slow change over time. An examination of drone music, where long musical forms hold small and gradual changes, provides useful tools to describe glissandi. Joanne Demers suggests drone and noise music create an immanent, rather than transcendent relationship with time [13, p. 93]. Building on contributions from electronic music composers and the texturalists, Demers suggests that long form drone music appears to defy transcendence by avoiding the development and arrival of conventional harmonic and melodic goals expected in most Western music [13, p. 193]. Rather than illustrating a passage through time, it obscures its passing – which is no more than a perceived effect, since all sound is constantly in motion by its very physical makeup. Drone music provides what Kramer calls 'vertical time' [14], where small events become very significant and the idea of a work 'becoming' is annulled. A very slow moving glissando, or very microscopic glissando-like movements are likely to feature as some of the detail that features in a piece of drone music.

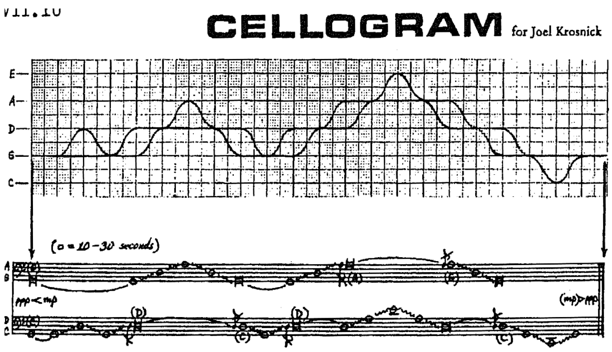


Figure 3: James Tenney, *Cellogram*, 1971.

A composer working with microscopic glissandi to create harmonic stasis is Giacinto Scelsi, whose later string works often feature a single tone replicated over octaves in different instruments and enveloped only by vibrato, pulsations, glissandi and microtones [15]. In *String Trio* (1958), a perception of stasis replaces any sense of harmonic or formal development. Yet there are small details that elaborate this stasis - the very opening note of the first movement is a small upward glissando, and from thereon in, any change to the initial pitches are not iterated by pulse or harmony, but rather by inflection – varying speeds of vibrato, quarter tone movements, dynamic variation and the occasional pizzicato. The single movement *String Quartet no. 5* written between 1974 and 1985 takes the idea further; the whole work is based on a single F - which is slid to and from, attacked and sustained in different timbre, shapes and dynamics. These later works of Scelsi employ glissandi as a way to sustain interest a kind of harmonic stasis, colouring it and shaping it using the very nature of the sound itself, as Tenney did in *Celllogram.* Scelsi used an early synthesiser, the Ondiola, to create these works - using the small glissandi knobs on the machine to create these sounds which were then notated for acoustic instruments [16].

4. GLISSANDO AS STRUCTURE

To meld this idea of drone with that of glissandi in music notation became a focus of the authors own work. The first of these experiments was *In The Cut* (2009), a small ensemble work which examined the idea of 'descent'. This did not rely on a long, slippery glissando alone, but also used small descending steps and bends against sliding tones. A trio of acoustic instruments slowly descend in pitch until they reach their lowest note, when they are asked to detune even further as to distort the timbre and make pitch identification difficult, as happens with very low sounds. They are accompanied by bass guitar providing a long descending tone, semi-articulated and effected with heavy delay and reverb, as well as a vinyl record that plays a descending sine tone coloured only by the vinyl noise from playback. In this way, a range of inflections colour the singular, long glissando the instruments play, and which is the basis for the form of the work. This descending glissando is made up of interlocking parts, and culminates in the absolute lowest sound possible on each instrument - detuned on the bass, cello and viola, extended using pipe on the bass clarinet, and even the sine tone which goes beyond the frequency capacity of the bass amplifier that sounds it. *In The Cut* employs a glissando as singular form, as well as trajectory. Unlike the seemingly static works of Scelsi or Tenney, it has a movement, a place to go.

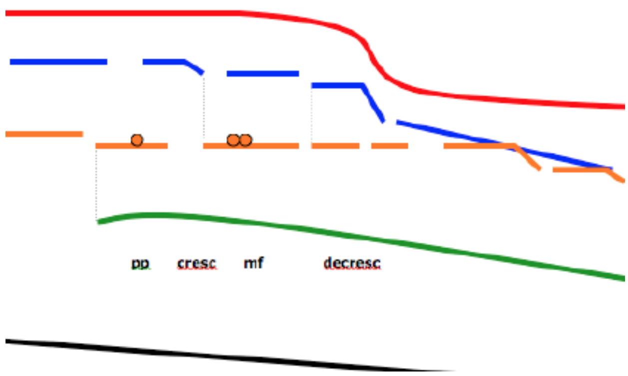


Figure 4: An excerpt of *In The Cut* (2009) by Cat Hope.

The score to this and the following works is presented as animated notation, to enable the reading of the long form lines in a smooth and coordinated way. The image passes from left to right, past a line that signifies the moment of performance. The rate of movement is smooth - without pulse, and obscuring any sense of tempo. It provides a perfect representation of the score that enables the players to focus on the point of performance, whilst predicting the direction of any change in pitch without steps or counting.

5. GLISSANDI INTERFERING WITH THE DRONE

Longing (2011) is a work that examines the glissando from a different perspective. It focuses around a single note for each of five performers, which is sounded at the start., and constantly referenced by way of a coloured 'reference' line that serves as a reminder of the original pitch as the instruments wander away from it. The note is not specified - the performers may choose any note to start, bearing in mind their capacity to smoothly journey away and back to it. They must also try to relate their activity to those of the other performers, creating peaks and troughs that are proportional in relation to other performers. The work is interrupted by upward moving staccato/pizzicato notes at the half way point, as axis to highlight the lack of 'forward' motion before or after its appearance.

Unlike *In the Cut*, the form of the work is flat, it has no end point, no trajectory. Each instrument constantly refers to their original pitch, after wandering away from it by way of curved, almost circular glissandi, with the intention of creating a perception of time turning in on itself. Like Scelsi's work, the drone is coloured by the glissandi, but in a more lugubrious way. These glissandi create structure to the work, rather than decoration to the line, due to their large pitch range and rapid trajectories.

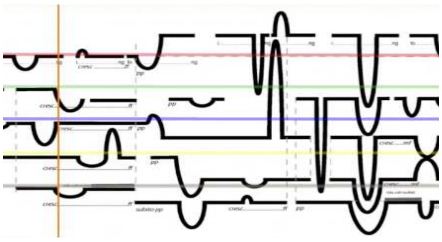


Figure 5: An excerpt of *Longing* (2011) by Cat Hope

6. MICROPHONIC POINTS ON A LINE

Cruel and Usual (2011) uses Tenney's idea of microphonic points as Xenakis's points on a line. In this work small points create a static electronic sound sampled from a very small point of the acoustic activity. The work is for string quartet, and uses similar concepts to *Longing* in its formal construction. The performers choose their own first pitch. A computer samples single microsecond moments in each instruments line notated on the score, using individual microphones. The computer then transcribes the samples down in pitch within a much less precise predicated range, and extends that pitch for ascribed durations, with timbral and dynamics variations indicated in the score. The electronics are reproduced through four bass amplifiers, one for each string instrument. The static acoustic lines are punctured by these bass interjections that initially seem quite foreign in timbre, but then melt into the line as they fade away, coalescing their difference into the ensemble, returning to the drone, and escaping in the glissandi..

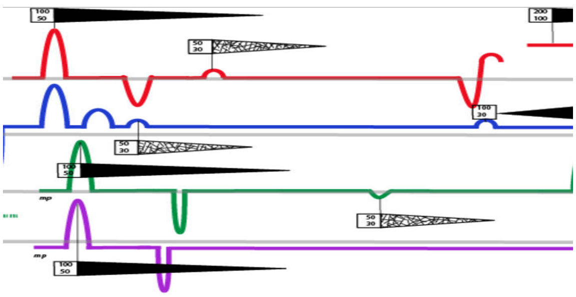


Figure 6: An excerpt of *Cruel and Usual* (2011) by Cat Hope.

7. CONCLUSION

These three works demonstrate how ideas of drone and glissandi in compositions can be used to inform formal and structural cornerstones in notated works. Informed by key works of the twentieth century, these compositions attempt to challenge the idea of the glissando as decoration and reframe the potential of the technique to have formal and structural applications. Using simple but focused notations facilitated by animated notation, these works can be thought of as a step toward to more complex and asynchronous examples. Glissandi provide the potential to reconfigure the teleological conventions of musical structure and open up new ways of listening to music through time that is not driven by tempo or beat. A principal difficulty to engage glissandi in large forms has been the coordination of performance, as the line without rhythmic markings offers no points for performers to reference. This has largely been overcome through the innovations of digital scoring and animated notation facilities, opening the way for a richer ground of exploration.

8. REFERENCES

[1] D. C. Hoy, *The Time of Our Lives: A Critical History of Temporality*. Cambridge, MA: MIT Press, 2009.

[2] M. Iliescu, "Glissandi and Traces: A study of the relationship between musical and extra-musical fields," presented at the International Symposium Iannis Xenakis, Athens, 2005.

[3] G. Deleuze, *Difference and Repetition*. New York: Columbia University Press, 1994.

[4] Deleuze, G., & Guattari, F. (1980). *A Thousand Plateaus*. (B. Massumi, Trans.). Minneapolis: University of Minnesota Press.

[5] I. Xenakis. "Lettre de Iannis Xenakis à Jean Baudot (Centre de Calcul de l'Université de Montréal) du 1er novembre 1972," *Centre Iannis Xenakis*, Accessed 24 November 2015. http://www.centre-iannis-xenakis.org/items/show/780

[6] F. Delalande. Interview with Xenakis, Radio France, program broadcast on 14 December 1981.

[7] I. Khannanov, "Line, Surface, Speed: Nomadic Features of Melody," in *Sounding the Virtual: Gilles Deleuze and the Theory and Philosophy of Music*, B. Hulse and N. Nesbitt, Eds. Ashgate, 2010, pp. 249–267.

[8] J. Shepherd and P. Wicke, *Music and Cultural Theory*. London: Polity, 1997. Pp. 159–160.

[9] A. S. Weiss, *Varieties of Audio Mimesis: Musical Evocations of Landscape*. Berlin: Errant Bodies Press, 2008. P. 78

[10] Francis Bayer, *De Schönberg à Cage*, cited in A. S. Weiss, *Varieties of Audio Mimesis*. Berlin: Errant Bodies Press, 2008. P. 74

[11] I. van Elferen, "Music of other spheres: Diagonal time and metaphysics in Lost," *Science Fiction Film & Television*, vol. 3, no. 2, pp. 253–270, Feb. 2010.

[12] L. Polansky, The Early Works of James Tenney. Soundings NO. 13, Ed Peter Garland, p. 200.

[13] J. T. Demers, *Listening Through the Noise: The Aesthetics of Experimental Electronic Music*. New York: Oxford University Press, 2010.

[14] Kramer, J. D. (1988). *The Time of Music: New meanings, new temporalities, new listening strategies.* London: Collier MacMillan Publishers.

[15] F.-M. Uitti, “Preserving the Scelsi Improvisations,” *Tempo*, no. 194, Oct. 1995, pp. 12–14,

RESURRECTING A DINOSAUR - THE ADAPTATION OF CLARENCE BARLOW'S LEGACY SOFTWARE AUTOBUSK

Georg Hajdu
Center for Microtonal Music and Multimedia
Hamburg University of Music and Theater (HfMT)
Harvestehuder Weg 12
20148 Hamburg

georg.hajdu@hfmt-hamburg.de

ABSTRACT

This paper aims at describing efforts to conserve and further develop the legacy real-time generative music program *AUTOBUSK* by Clarence Barlow. We present a case study demonstrating that a simple port of 30+ year old code may not suffice to infuse new life into a project that suffered from the abandonment of the hardware it was developed on/for. In the process of resurrecting this "dinosaur," *AUTOBUSK* was entirely redesigned for the popular music software environments Max and Ableton Live (via Max for Live) and renamed *DJster*. It comes in several incarnations, the most recent ones being *DJster Autobus* for Ableton Live, a device for real-time event generation and *DJster Autobus Scorepion*, a plugin for the MaxScore Editor. These incarnations take advantage of being embedded in current environments running on modern operating systems and have since acquired some new and useful features. As *AUTOBUSK/DJster* is based on universal musical principles, which Barlow formalized during the 1970's while working on his generative piano piece *Çoğluotobüsişletmesi*, its algorithms are of general applicability for composers and performers working in diverse fields such as microtonality, interactive installations and/or film music. It has therefore inspired me to lay the foundations of a shorthand notation, which we will discuss in the last section.

1. INTRODUCTION

AUTOBUSK is a real-time generative program developed by Clarence Barlow, which "took 272 days to write, spread between 18 August 1986 and 30 October 2000." [1] It is one of the first ones of its genre, which includes applications such as *M* by David Zicarelli, Joel Chadabe, John Offenhartz, and Antony Widoff (launched in 1987) [2], the *Lexikon Sonate* by Karlheinz Essl (development starting in 1992) [3], George Lewis' *Voyager* system (development starting in 1993) [4] as well as *Koan* (released by SSEYO in 1995), which Brian Eno used extensively for his generative pieces [5]. AUTOBUSK uses a probabilistic approach and is based on universal[1] musical principles which Barlow formalized while working on *Çoğluotobüsişletmesi* (completed in 1979). It was written in Pascal for the Atari ST computer platform, which became immensely popular amongst European musicians in the mid to late 1980's—mainly due to its low price and built-in MIDI ports. The first piece realized with this system was Barlow's performance art piece *Variazioni e un pianoforte meccanico* in which a performer plays the opening bars of the *Arietta* from Beethoven's piano sonata op. 111 on a Disklavier, with the performance gradually being taken over by *AUTOBUSK* controlling the piano via MIDI. The performer eventually leaves the piano to play by itself, only to return at the very end in order to conclude the piece. Another piece of his, *Pandora* (1989) took advantage of *AUTOBUSK's* ability to save generated scores as MIDI files. Cologne-based American pianist Kristi Becker took the 3-part Finale printout and arranged it for piano. Barlow composed a few more pieces on the Atari but was eventually forced to resort to a Windows emulator called Steem after Atari stopped building computers in 1993.

2. AUTOBUSK - A CASE STUDY FOR THE PRESERVATION OF A DIGITAL MEDIA ART WORK

The issue *AUTOBUSK* users are facing are typical for legacy software products. These are (1) decay of physical

[1] I always appreciated the fact that Barlow's algorithms aimed at capturing universal musical principles such as tonal and metric hierarchies, independent from personal style and bias. While the existence of universals in music is still under much debate, we are inclined to assume that distinct cultural efforts manifest themselves as distortions of universals (which would be much easier to formalize) rather than as unrelated particulars. Hence, using *AUTOBUSK/DJster* as a universal music machine justifies, the continued effort I have poured into further developing this environment and exposing my students to it, who repeatedly have raised, using someone else's brainchild to compose music, interesting questions about ownership and intellectual property.

media and/or (2) abandonment of media types, formats software, OS's and hardware.

Efforts to save digital works and thus to contribute to their longevity require pro-active preservation which comes in the form of migrating files to current media, using emulators to run old software or porting the code to more recent platforms. The most radical approach is to re-design the software, either by transcribing the algorithms or re-creating them entirely. Fortunately, in case of DJster I was able to fall back on several of Barlow's publications as well as his personal input. The most useful resource was the website maintained by the University of Mainz department of music informatics. It contains links to a zip archive with the Atari files and a 54-page user manual called "AUTOBUSK: A real-time pitch and rhythm generator" [1].
The *AUTOBUSK* GUI has several panes (Figure 1) of which the one on the left is most relevant. It features, for three individual "streams", a number of GUI items which serve to control the generative process. Its algorithms have been described by Barlow in his book *Bus Journey to Parametron* [6]. An excellent introduction to the underlying theory including an explanation of the terms *indispensability*, *indigestibility* and *harmonicity* is given in Barlow's textbook On Musiquantics [7].

The panes on the right pertain to real-time MIDI control as well as the creation and editing of parameter scores via additional "helper" programs.

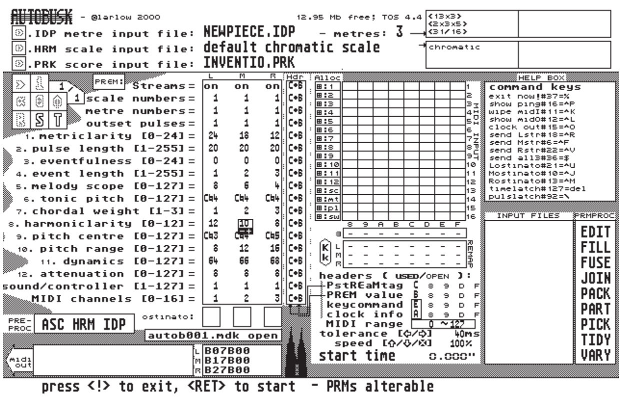


Figure 1: The GUI of Barlow's legacy program *AUTOBUSK*, written in Pascal for the Atari ST platform.

For its input, AUTOBUSK reads three different file types—some of which in binary format to save valuable hard-disk space (mind that a floppy disk in the early 1990's could only hold 1.4 MB). All three file types rely on the conversion of human-readable precursor files in text format:

- .mtr files, holding the stratifications of meters such as "2 2 3" (for a multiplicative meter with 12 pulses), to be converted into .idp files, spelling out the *indispensability* values for the given meters.
- .cts files holding the cent values for the chosen scales such as "0 200 400 500 700 900 1100 1200." Upon conversion into .hrm files, these values are expanded to the range from -9600 to 9600 cents, 0 cents being tonic pitch. In the process, for each scale step, various tuning alternatives are taken into consideration of which the one contributing to the best overall *harmonicity* (Barlow's term for tonal consonance) is chosen and prioritized according to its individual harmonicity value. This rationalization of the interval set is a lengthy process, as large numbers of combinations may need to be assessed.
- .prm files, to be converted into binary .prk files, consist of the temporal sequence of control messages such "5 900 L 9 40," denoting "set *pitch centre* for player 1 (left) to 40 at 5.9 seconds."

3. DJSTER - A RESURRECTION?

In 1992 while working as graduate student instructor at CNMAT/UC Berkeley under David Wessel, I oversaw a student project aiming at implementing the concept of *indispensability* in Max 2.0. The resulting abstraction was called *dispenser* and was used by me (and others) in various projects. It later added support for additive meters as well as user-defined arbitrary meters, which deliberately work against the grain of Barlow's "natural metric order" which conceptualizes syncopations in terms of low "metriclarity" or phase-shifted *outset pulses* (see Figure 1) in respect to a reference meter. It also became the basis for the first version of DJster, completed in early 2008. Its name is a reference to DJing as well as Barlow's notion of the *indigestibility* of a number, a concept central to his notion of tonality [8].

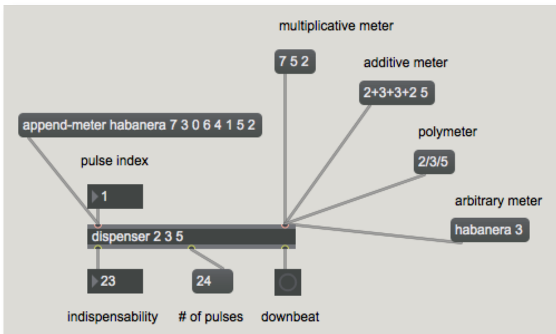


Figure 2: The Max *dispenser* abstraction accepts various meters in its right input. An arbitrary meter such as Habanera requires the meter to first be appended to dispenser's internal table with a message consisting of the message name *append-meter*, a symbolic value and the corresponding list of *indipensabilities*.

3.1 The Original Max Implementation

DJster was programmed in MaxMSP 4.6, preserving as much as possible the original layout of the left pane, while forgoing the implementation of MIDI control and helper applications (Figure 3). Instead, all parameters are exposed via the Max *pattr* system and controllable via OSC messages. For backwards compatibility it can also read and play *AUTOBUSK* input files. Translation of .cts into .hrm files is achieved on the fly by table look-up of "harmonic energy values," thus no longer requiring the time-consuming rationalization of the scales' interval set.

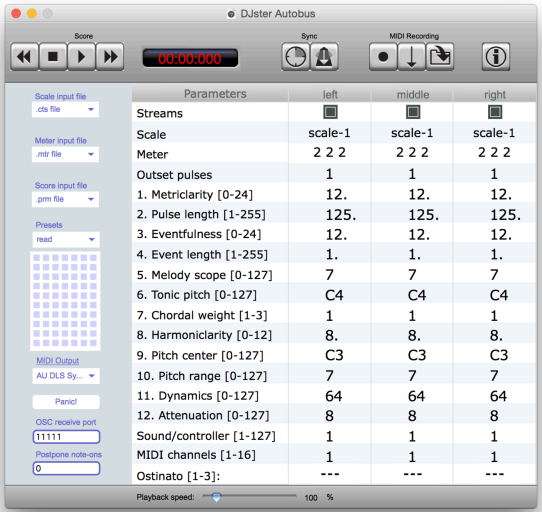


Figure 3 shows the GUI of the 2008 version of *DJster*, based on the generative music part of *AUTOBUSK* and implemented in *MaxMSP*.

3.2 Version for Quintet.net

A second Max implementation exists for the networked multimedia performance environment *Quintet.net* (Figure 4) [9]. It separates the streams into individual instances and is capable of dealing with microtonality. Scale files use key and value pairs, of which the key (a symbol such *Pentatonic*) will appears in the scale menu (instead of just an index). Since *indispensability* values are (as with *harmonicity*) calculated on the fly, the conversion of .mtr to .idp files is no longer necessary either. *Pulse length* and *event length* are no longer limited to the original ranges and all parameters dealing with pitch use cent resolution for display and microtonal playback. Real-time control can be exerted by sending OSC messages between various *Quintet.net* components. In my classes, it has been used as the target of Brain-Computer Music Interfaces or gestural interfaces such as the LeapMotion (via a few-to-many mapping performed by an artificial neural network).

3.3 Ableton Live Device

The third incarnation represents the biggest leap (Figure 5): By creating a Ableton Live device via the Max for Live API, DJster needed to be adapted to the philosophy of its host, a DAW driven by beat and loop-oriented *electronica*. Pulse length and meter had to be reconsidered and reworked into two parameters owned by the host (*song tempo* and *time signature*) in addition to *subdivision* of the beat—settable in DJster device.

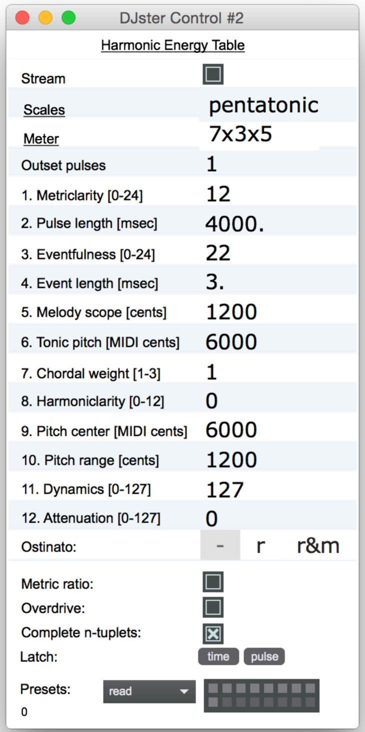


Figure 4: The GUI of the *Quintet.net* version of *DJster*. It contains a few additional parameters on the bottom added for the performance of *Just Her – Jester – Gesture.*

Internally, in the DJster Autobus device, *pulse length* is thus re-calculated by dividing 240000 by the product of tempo, time signature denominator and the number of pulses subdividing a beat:

$$Pulse\_length[msec] = \frac{240000}{tempo \cdot ts\_denom \cdot num\_psub} \quad (1)$$

In turn, *meter* is derived by concatenating the prime factorization of the time signature numerator with the *stratification divisors* of the subdivision of the beat. E.g. for a 12/4 time signature with a quintuple subdivision of the beat we get 2x2x3x5 as meter.

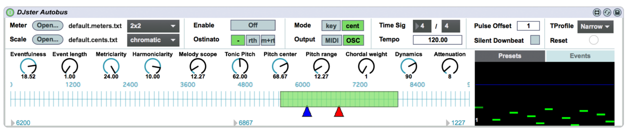


Figure 5: The GUI for the *Ableton Live* incarnation of *DJster*.

As all of DJster's parameters are automatable, .prm/.prk score files are no longer supported. Interpolations between parameter values will now be performed automatically by the host. Ableton's arrangement view allows continuous tempo changes as well as time signature changes from one measure to another. The repertoire of scales can easily be expanded simply by dropping files from Manuel op de Coul's Scala archive [10] onto the Scale menu.

3.4 MaxScore Plugin

The last incarnation of DJster is a MaxScore Editor Scorepion plugin, which shares Ableton's tempo/time signature/subdivision concept and is able to exchange presets with the latter (Figure 7). MaxScore is a Max Java notation object programmed by Nick Didkovsky [11]. A visual editor for it was created by the author. It also possesses a plugin structure called Scorepion, which extends MaxScore core functionality through the use of Max patches invoked from a particular folder in the MaxScore folder hierarchy. The DJster Autobus Scorepion fills selected measures of a MaxScore staff with notes (Figure 6) and—with its non-real-time approach—brings back to life *AUTOBUSK's* ability to write its output to a MIDI file readable by Finale, Sibelius and co. In DJster, though, this intermediary step is no longer necessary as this all happens in one environment. Therefore, work is more intuitive and combines the manual approach of traditional composition with the generative approach of computer composers who prefer to tweak code rather than music[2].

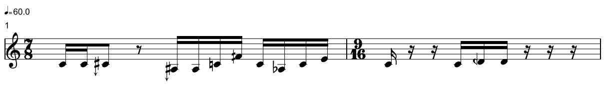



Figure 6: Two measures of music in Carlos Alpha tuning created after adding a scale file from the Scala archive to Djster's scale repertoire. See Figure 7 for the settings used in this example.

In order to fill the selected measures in MaxScore with music, the user first requests information about their time signature and tempo attributes by clicking on the "Gather Info" button (Figure 7).

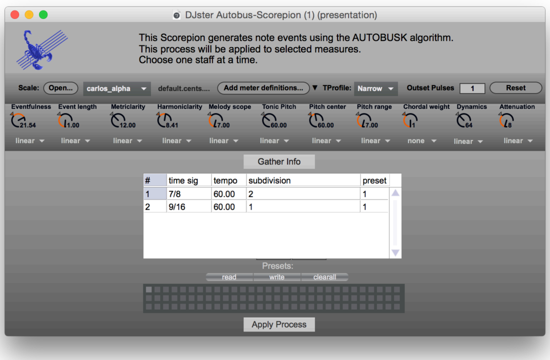


Figure 7: The GUI of the *DJster Autobus Scorepion* plugin for the *MaxScore* Editor. Note the three bottom rows of presets, allowing users to exchange settings with the Ableton Live device.

After creating any number of presets consisting of DJster parameters and interpolation modes, he/she can use the info pane to change the default subdivisions manually and assign preset numbers to each measure. Depending on the interpolation mode, parameter changes between measures will be either abrupt or gradual. Upon clicking on "Apply Process" the generative process will be triggered and the resulting events transcribed into notation.

4. EXAMPLES OF RECENT WORKS

After using the *dispenser* abstraction for real-time composition in the *Intermezzo* of my opera *Der Sprung – Beschreibung einer Oper* (1996) [12], I have employed DJster in the interactive composition *Just Her – Jester – Gesture* and focused on the non-realtime application of the Scorepion in my works *In ein anderes Blau* (2012) (Figure 8) and *No, I won't* (2014) [13]. *Dispenser* was used for real-time composition/notation in the percussion and multimedia piece *Slices* by Jacob Sello [14] while the Ableton Live version of DJster was employed in the real-time interactive dance performance *Mond in Wogen* by Xiao Fu [15].

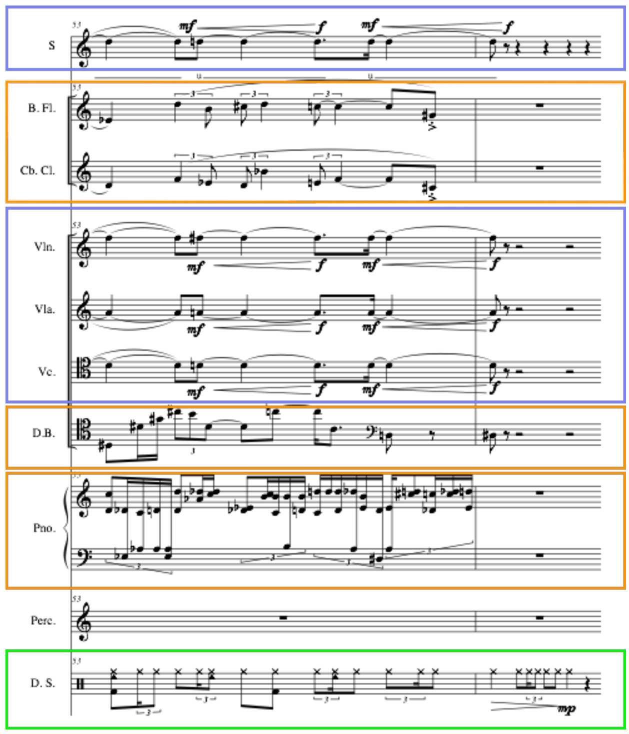



Figure 8: A page from Georg Hajdu's composition *In ein anderes Blau* in which the transcription of improvised music (blue frame) is combined with music generated with the DJster Scorepion. For this, meter and scale definitions were created to match the pitch set of the improvisation as well as the underlying 2/3/5 polymeter.

[2] I have elaborated this relationship in a presentation given at the 2012 Ligeti symposium in which I compared György Ligeti's and Clarence Barlow's approaches to generative music (http://quintetnet.hfmt-hamburg.de/Ligeti-Symposium/?page_id=90)—the former composer characterized by his non-computer PPP approach (paper, pencil, pocket-calculator).

5. TOWARDS A NOTATION OF DJSTER

In *Just Her – Jester – Gesture* MaxScore sends out, in sync with the performer, messages to several instances of the Quintet.net version of DJster. The notation consists of single notes to which lists of parameter values have been attached via the MaxScore Editor note-slot feature (Figure 10). Since there is no way of guessing those values from the appearance of a note, a specific DJster shorthand notation could be handy while serving two purposes: Firstly, it could symbolically represent the parameter constellations to be sent as sequenced message to the real-time version of DJster, or, secondly, the notation could serve as a control track for a non-real-time composition in which the DJster Scorepion actually spells out the notes, which bears some resembles to figured bass.

Figure 9 shows a mockup of this shorthand notation. It consists of a regular note (denoting *tonic pitch*) and two smaller notes denoting *pitch center* (diamond) and *melody scope* (in terms of the interval between the diamond and the circle). On top of the note, there is a slider box with five sliders and a symbol referring to the scale currently in use. Table 1 delineates the relationship between DJster parameters and their symbolic representation in shorthand notation.

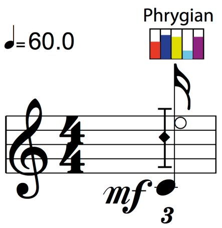


Figure 9. DJster shorthand notation. Refer to Table 1 for a detailed explanation of the parameters involved.

Parameter	Representation
Scale	String above slider box
Meter/Subdivision	Ratio between Denominator and note value (MaxScore duration property)
Eventfulness	1. element of slider box
Event Length	MaxScore hold property
Metriclarity	2. element of slider box
Harmoniclarity	3. element of slider box
Melody Scope	Interval between pitch center and round notehead
Tonic Pitch	Regular notehead (MaxScore pitch property)
Pitch Center	Diamond notehead
Pitch Range	Brackets extending from pitch center
Chordal weight	4. element of slider box
Dynamics	Dynamics symbol (MaxScore amplitude property)
Attenuation	5. element of slider box

Table 1: List of parameters represented by DJster's shorthand notation.

The DJster notation editor will be implemented as a MaxScore slot-editor module (Figure 10). Upon clicking on Slot in the Note Attributes palette, the pitch, duration, amplitude, hold and tuplet attributes of the selected note will set the corresponding DJster parameters as defaults. The other 9 parameters will then have to be specifically set in the GUI. Once all values are set, MaxScore generates two types of data:

- A list of data to be output by the playback engine
- A *Picster* graphic [11] to be embedded in the score

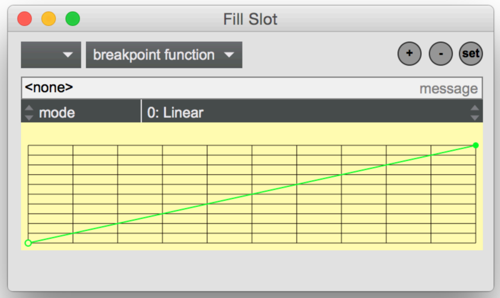


Figure 10. Example for a MaxScore Editor slot module. The DJster slot module will inherit its controls from the DJster Scorepion.

6. CONCLUSIONS

In this paper I gave an account of how a legacy computer music program can be revived by adapting its algorithms to modern environments. Similar attempts have successfully been accomplished by David Zicarelli who in 1997 resumed maintaining his "Intelligent Composing and Performing System" *M* and most recently by Gottfried Michael Koenig whose *Projekt 1* from 1964 was just recently translated into SuperCollider by Rainer Wehinger. In case of DJster we went through a evolutionary process leading to several versions of the original system. The last one, a non-real-time plugin for MaxScore, allows users to intuitively combine traditional and generative approaches to music composition. As the DJster project has implications that touch on the unabating issues of real-time music generation, microtonality, man-machine interaction as well as symbolic data representation and mapping, development will continue with a strong focus on documentation, user friendliness and flexibility. Currently, its usability for music generation within hospital environments is being studied within the Healing Environment project, jointly organized between two department at the HfMT Hamburg departments and the university hospital Hamburg-Eppendorf (UKE).

DJster in its various incarnations can be downloaded from http://djster.georghajdu.de.

Acknowledgment

I would like to thank the *Behörde für Forschung und Wissenschaft Hamburg* for supporting our research in the framework of its *Landesforschungsförderung*.

7. REFERENCES

[1] Clarence Barlow's AUTOBUSK user manual "AUTOBUSK: A real-time pitch and rhythm generator" for the Atari ST version is available as report no. 44 (2001) in the series "Musikinformatik & Medientechnik" or as download from http://www.musikwissenschaft.uni-mainz.de/Autobusk/.

[2] D.Zicarelli, "M and Jam Factory," *Computer Music Journal*, vol. 11, no. 4, pp. 13-29, 1987.

[3] K. Essl: "Lexikon-Sonate. An Interactive Realtime Composition for Computer-Controlled Piano" *Proceedings of the II. Brazilian Symposium on Computer Music,* 1995. Also available online at http://www.essl.at/works/Lexikon-Sonate.html.

[4] G. E. Lewis, "Too Many Notes: Computers, Complexity and Culture in Voyager," *Leonardo Music Journal,* vol. 10, pp. 33-39, 2000.

[5] N. Collins, "The Analysis of Generative Music Programs," *Organised Sound,* vol. 13 , no. 3, pp. 237 248, 2008.

[6] C. Barlow, "Bus journey to parametron," *Feedback Papers*, vol. 21–23, 1980.

[7] C. Barlow, *On Musiquantics – English translation of Von der Musiquantenlehre (2008). Issue 51 of Musikinformatik & Medientechnik*, University of Mainz, 2012.

[8] C. Barlow, "Two Essays on Theory," *Computer Music Journal,* vol. 11, no. 1, pp. 44-60, 1987.

[9] G. Hajdu, "Quintet.net: An Environment for Composing and Performing Music on the Internet," *LEONARDO,* vol. 38, no. 1, pp. 23-30, 2005.

[10] http://www.huygens-fokker.org/scala/

[11] G. Hajdu, and N. Didkovsky, "MaxScore – Current State of the Art," in *Proceedings of the International Computer Music Conference, Ljubljana*, 2012 *pp. 156-162.*

[12] G. Hajdu, "Research and Technology in the Opera Der Sprung". In *Nova Acta Leopoldina*, vol. 92, no. 341, pp. 63-89, 2005.

[13] http://georghajdu.de/compositions/

[14] http://zkm.de/media/video/jacob-sello-slices

[15] http://quintetnet.hfmt-hamburg.de/fuxiao/?page_id=206

Webpages all accessed on November 15, 2015.

HEXAPHONIC GUITAR TRANSCRIPTION AND VISUALISATION

Iñigo Angulo
Music Technology Group
Universitat Pompeu Fabra
angulolof@gmail.com

Sergio Giraldo
Music Technology Group
Universitat Pompeu Fabra
sergio.giraldo@upf.edu

Rafael Ramirez
Music Technology Group
Universitat Pompeu Fabra
rafael.ramirez@upf.edu

ABSTRACT

Music representation has been a widely researched topic through centuries. Transcription of music through the conventional notation system has dominated the field, for the best part of the last centuries. However, this notational system often falls short of communicating the essence of music to the masses, especially to the people with no music training. Advances in signal processing and computer science over the last few decades have bridged this gap to an extent, but conveying the meaning of music remains a challenging research field. Music visualisation is one such bridge, which we explore in this paper. This paper presents an approach to visualize guitar performances, transcribing musical events into visual forms. To achieve this, hexaphonic guitar processing is carried out (i.e. processing each of the six strings as an independent monophonic sound source) to get music descriptors, which reflect the most relevant features of a sound to define/characterise it. Once this information is obtained, our goal is to analyse how different mappings to the visual domain can meaningfully/intuitively represent music. As a final result, a system is proposed to enrich the musical listening experience, by extending the perceived auditory sensations to include visual stimuli.

.

1. INTRODUCTION

Music is one of the most powerful art-expressions. Through history, humans have shared the musical realm as part of their culture, with different instruments and compositional approaches. Often, music can express what words and images cannot and thus remains a vital part of our daily life. Advancements in technology over the last decades have brought us the opportunity to go deeper into developing an understanding of music, in the context of other senses such as sight, which dominates over other senses for representing information. In this work we propose a system to extend music by developing a visualisation/notation approach to map the most important features that characterise musical events into the visual domain.

The idea of providing mechanisms for understanding music using our eyes is not new, as traditional music notation (i.e. scores) may provide us with an idea about the acoustic content of a piece without the need of previously listening to it. However, our approach is not intended to design a performance instructor, but a visual extension of the musical events that compose a performance/piece. Our goal is to develop a system that is able to visually represent the musical features that best characterise the music produced by a guitar, that is, to develop a real-time visual representation system for guitar performances. One challenge for the development of such a system is the polyphonic nature of the guitar. The complexity of polyphonic sound transcription is well known, so to solve this issue we opted to use a hexaphonic guitar, in which each of the strings is processed as an independent monophonic sound source, simplifying the transcription of the sounds. We propose a way of transforming a conventional classical guitar into a hexaphonic one. Once the desired musical features are obtained, different ways to represent them are studied, analysing the mappings between sound and visual dimensions.

The aim of this work is to offer a tool in which information about the musical events (e.g. pitch, loudness, harmony) of a guitar performance is visualized through a graphical user interface. Additionally, these visualisations could enrich the experience of listening to music by accompanying musical events with visual stimuli.

2. STATE OF THE ART

A music notation system can be any symbolic notation that transmits sufficient information for interpreting the piece [1]. In our case, we wanted to explore the interconnection of music and visual fields, following the concept of bimodal art expression, the result of joining music and visuals, in which both dimensions are equally relevant since they are not perceived separately, but as a whole [2]. Throughout history, this kind of works has often been discussed in relation to "synaesthesia". This is a neurological condition in which a stimulus in one sense causes an involuntary answer in another one. The disadvantage about synaesthesia, from a scientific point of view, is its idiosyncrasy, which means that two synaesthetes with the same type of synaesthesia will most likely not share the same synaesthetic experiences [3]. This characteristic means we cannot develop an objective basis for the interconnection of visual and musical domains based on synaesthetic esperiences.

2.1 Music Visualisation

Music visualisation is a field that has attracted the attention of many researchers, scientists and artist for centuries. Many attempts have been made to create a machine that joins music and visuals, as for example the "Color Organ" (or "Light Organ") created by Louis Bertrand Castle in the 1730's. Further examples of early inventions, and the history of this field can be found in [4][2].

As mentioned previously, synaesthesia seemed to play a very important role in the early works in the field. Despite the subjectivity of synaesthetic cases, many people have proposed different theories to try to accurately relate musical and visual domains. An example of this is the notes´ pitch to colour mapping, a topic that has been addressed by many researchers in the past, see Figure 1 [4].

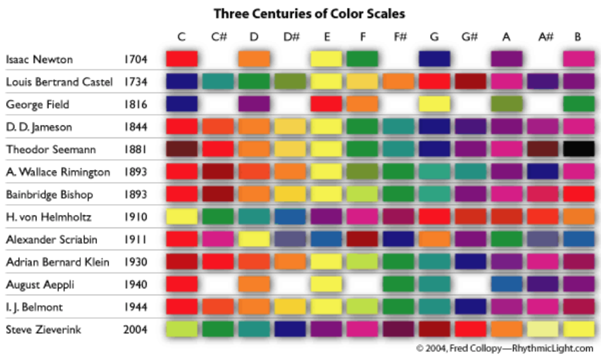


Figure 1. Pitch to colour mappings through history.

Nowadays, the music visualisation field is very active. The idea of joining sound and visuals to enrich the experience of music continues to attract the attention of many researchers, and there are different theories that support this hypothesis. An example of this is Michel Chion´s widely established concept of synchresis: *"(...) the spontaneous and irresistible weld produced between a particular auditory phenomenon and visual phenomenon when they occur at the same time"*. This theory, which asserts that synchronised auditory and visual events provides "added value", is vital knowledge for sound for cinema. These theories, and especially synaesthesia contributed to the argument that there is a strong indication that multimodality perceptions are not processed as separate streams of information, but are fused on the brain into a single percept [2].

When talking about music visualisation, many different approaches are comprised, which lead to different purposes. These may include, the simple representation of a waveform or a spectrum to visualize a signal into the frequency domain; the transcription of sound as accurate as possible, using scores or other notational system ("objective approach"); and the artistic visualisation of sound, which aim to create beautiful imagery to accompany music and create a sensory answer in the listener/viewer ("subjective approach"). In the next sections some examples of the different approaches are presented.

2.2 Music Visualisation Systems

At the present, many systems exist whose aim is combining sound and visual dimensions in music, generally having an artistic approach ("subjective approach"), as the output often consists of abstract imagery that accompany music. As technology has progressed, so have the tools that permit the exploration of the relation between these art modalities. Artist from many disparate fields dedicate themselves to experiment within this field, and this activity has led to the development of different practices through time. One example could be Video Jockeys (VJ´s), who perform by mixing videos and applying effects to them, normally in the context of some background music. Furthermore, concepts such as Visual Music or Audiovisual Composition appear together with these practices [2].

One example of a Music Visualisation system is Soma [2]. It was created in 2011 by Illias Bergstrom, for the live performance of procedural visual music/audiovisual art. The author wanted to improve the established practice of this field´s art form and break through it´s main limitations, which included constrained static mappings between visuals and music, the lack of a user interface for controlling the performance, the limitation of collaborative performances between artists, and the complexity of preparing and improvising in performances. With those ideas in mind, he proposed a system, both in hardware and software, to address these limitations. Figure 2 reflects the architecture of this system.

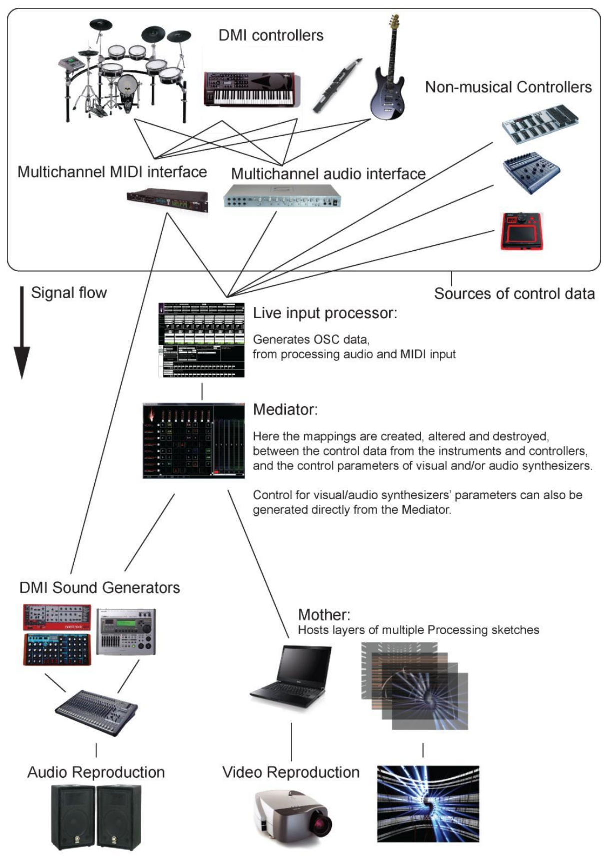


Figure 2. Illustration of the signal flow in Soma system.

Another interesting example of a music visuals system is Magic [5]. Magic is an application that allows one to create dynamic visuals that evolve from audio inputs. It is though as a tool for VJing, music visualisation, live video mixing, music video creation, etc. It allows one to work with simultaneous audio and MIDI inputs, both pre-recorded tracks and live input signals, computing different features in each case, for example: the overall amplitude of the signal, the amplitude per band, pitch, brightness; or MIDI features such as velocity, pitch bend or channel pressure. Figure 3 shows the graphical user interface of the software. The boxes are modules which have different functionalities and are interconnected to create scenes.

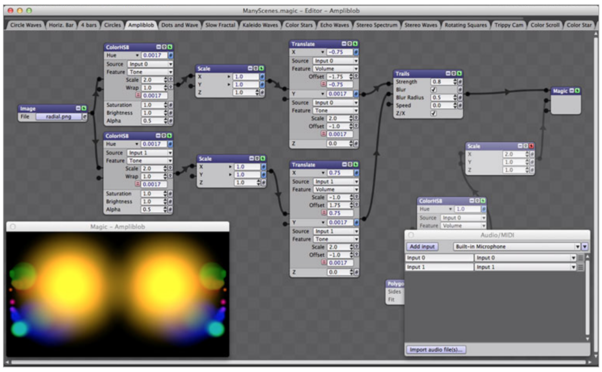


Figure 3. Magic Music Visuals graphical user interface.

The previously presented systems (Soma and Magic) are examples of systems that permit the visualisation of music, by creating mappings between sound and visual features. Thus, the created visualisations are directly controlled by the “changing-along-time” musical features extracted from the audio signals. These features, such as pitch, loudness or rhythm, are normally computed using signal processing techniques.

However, many of this kind of systems are more generic as they are designed for controlling the visuals from a group of DMI (Digital Musical Instruments) controllers, or directly from a stereo audio signal representing the mix of all the instruments that compose the piece. We propose a system specifically designed for the guitar.

2.3 Music Visualisation Systems based on Guitar

The systems analysed in this section approximate to the “objective approach” domain, as their visualisation aim consists of the transcription of music so that it can be reinterpreted. In other words, these visualise music from a transcription/notation perspective, and entail a performance instruction rather than a “creative visualisation” of the music that is being played.

One of the possible reasons that contributed to the popularity of these systems was the release of Guitar Hero [6], a videogame that appeared in 2005, which aimed to recreate the experience of playing music and make it available to everyone as a game. It consisted of a DMI guitar-shaped controller through which music was “interpreted” by the player, who was guided by the instructions that appeared on the screen. These instructions consisted of the notes that compose a particular song, presented over time. So, with the original song´s backing track sounding, the aim of the player was to press buttons on the guitar controller in time with musical notes that scroll on the game screen.

Another game called Rocksmith [7] was released in 2011. This game followed the main idea of music performance instruction (as Guitar Hero), but with an essential difference: a real guitar was used instead of a DMI controller. The idea behind the game was to be able to use any electric guitar, so it was approached as a method of learning guitar playing. The game offered a set of songs, for each of which a performance instruction was presented based on the notes that had to be played along time. Then, some feedback about the performance quality was given to the user.

There are other systems that follow the same approach of guitar music visualisation as notation, to give the user the necessary instructions to reinterpret a particular piece. Some examples of this are GuitarBots and Yousician [8]. These systems provide an easier way of learning to play guitar by helping the user with instructions about what to play.

The system we propose in this paper lie into this last category of “objective” music visualisation system, which aims to transcribe music accurately. Our goal is to provide a guitar performance representation tool, transcribing sounds to visual forms instead of traditional notational systems (i.e. scores), which could result more meaningful for people with no musical training. This system is able to reproduce musical events in real-time, creating thus visual stimuli for both the listener/viewer and the musician. However, although our first intention is to accurately reproduce guitar performances, we also want to explore the artistic representation domain (once we have enough sound and visual features to be mapped) by creating more abstract/impressive visualisations, aiming to analyse if this leads to a stronger sensorial perception of music by the user of the system.

3. MATERIALS

3.1 Essentia

Essentia [9] is an open-source C++ library for audio analysis and audio-based music information retrieval. The library is focused on the robustness of the music descriptors it offers, as well as on the optimisation in terms of the computational cost of the algorithms. Essentia offers a broad collection of algorithms, which compute a variety of low-level, mid-level and high-level descriptors useful in the music information retrieval (MIR) field. This library is cross-platform and provides a powerful tool that collects many state-of-the-art techniques for the extraction of music descriptors and optimized for fast computations on large collections.

3.2 Processing

Processing [10] is a programming language and environment targeted at artists and designers. It has many different uses, from scientific data visualisation to artistic explorations. It is based on the Java programming language. In the context of music visualisation, Processing has available several libraries to deal with audio, such as Minim, which lets one work with different audio formats and perform many different signal-processing techniques to obtain musical features. The visualisations can be controlled with the results of computing these features, and range from raw frame data that correspond to visualizing data as precisely as possible to the perceived music, (for example drawing the waveform of a frame), to generative visualisations that focus on producing beautiful images and impressive effects.

3.3 Hardware

This research project also focuses on the hardware implementation of the hexaphonic guitar. Our approach is based on transduction using piezoelectric sensors. These sensors work on the piezoelectric principle, which states that electric charge is accumulated in certain solid materials in response to applied mechanical stress. Thereby, one of these sensors is able to transform the mechanical force that is exerted on it by the vibration of the string, into an electric current representing this vibration. In addition to these sensors, other materials such as wires to build the circuits and 1⁄4” TS (Jack) connectors to be plugged into the computer´s audio input device are needed.

4. METHODS: DEVELOPMENT

4.1 Hexaphonic Guitar Construction

As explained earlier, some hardware is needed to transform a traditional guitar into a hexaphonic one. It consists of six piezoelectric sensors, six 1⁄4” TS jack connectors and twelve wires to interconnect them. With this material a circuit was built to capture the signal from each string independently and send it through a cable to the computer´s audio input device, via a common audio interface with six channels.

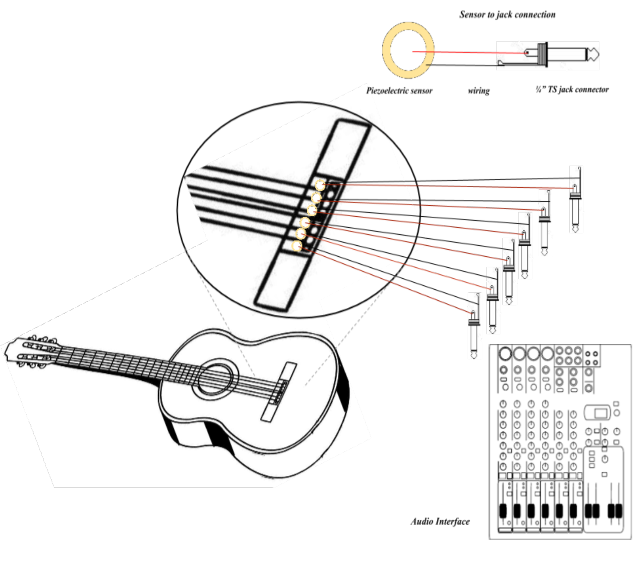


Figure 4. Hexaphonic guitar construction scheme using piezoelectric sensors.

The construction is shown in Figure 4. Each piezoelectric sensor is welded with a jack connector as shown in the scheme. The tip of the jack (the shortest part in the end) is connected to the white inner circle of the piezoelectric, and will transmit the signal. The sleeve of the jack is thus connected to the golden surface of the piezoelectric. Each of the sensors is cut and placed between the string and the wood of the bridge of the guitar, which is where we found the vibration of the string was best captured by the sensor. Once this is done, for each of the strings, the output jack connectors are plugged into different channels of the audio interface, so that the signals can be independently processed. These sensors act like small microphones capturing the sound produced by each of the strings.

4.2 Audio Signal Processing

Once we had our six separate audio signals, corresponding to each of the strings, we processed them using the Essentia library to obtain meaningful musical features. For our purpose, several descriptors were used to extract the desired features from the sound, such as PitchYinFFT, Loudness and HPCP, which extract information about the pitch, energy and chroma of the notes. These were computed in real-time and sent to Processing, where they were used to control the visualisations.

In addition to this, a map of frequencies was created, corresponding to the frequencies of the notes along the fretboard of the guitar, in standard tuning. Hence, obtaining the fundamental frequency of the note played on each of the strings (which is easy as the signal for each string comes on an separate input channel into the computer), tells us exactly which frets (and on which strings) were being played at a given moment.

4.3 Visualisation

To perform the visualisation of the musical features previously extracted, we used Processing. Using this software, a graphical interface was created to visualize the musical features using different visual forms (Figure 5).

At present, we have developed a simple visualisation to test the system. The amount of musical features involved, as well as the quality of the mappings and visualisation approach will be revised and enhanced in the future.

We presented the information in a 2D-plane, in which the X-axis represented the six different strings and the Y-axis the pitch height. The notes were represented using circles. From left to right (on the X-axis) strings were visualized as vertical lines from the 6th to the 1st one. For example, when playing any note on the 4th string, it will be always represented on the same “invisible” vertical line, which crosses the X-axis at a particular point (corresponding to that 4th string).

The loudness of each note was mapped to the size of the circle, which becomes smaller as the note decays. Also, having built the frequency map, it was easy to know which particular note was played, so the name of the corresponding note is plotted in the center of the circle. To easily distinguish one note from others, these were mapped to the range of visible colours. The lowest frequency on the guitar (E in standard tuning) is mapped to the lowest frequency in the visible range (red). The reason for this was more aesthetical than scientific. Figure 6 shows an approximation to this mapping.

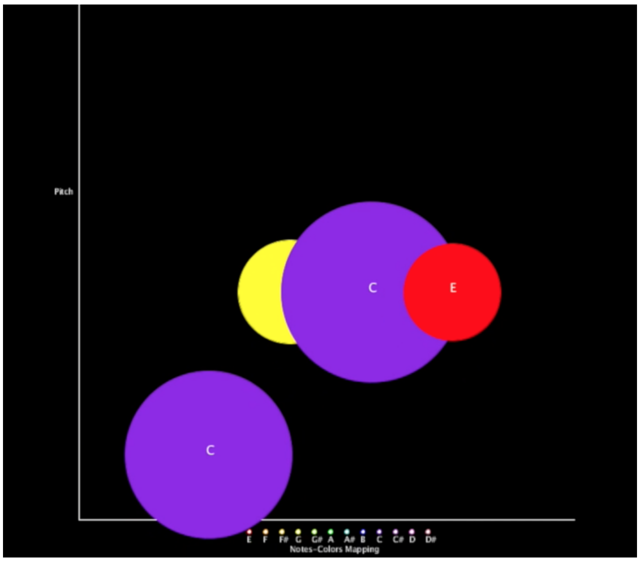


Figure 5. Example of visualisation interface.

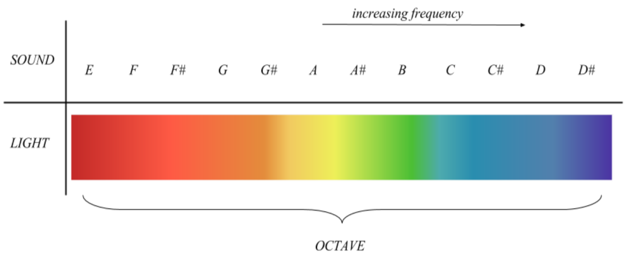


Figure 6. Note to colour mapping.

5. EVALUATION

5.1 Experiments

As this research project is still a work in progress, we prepared a simple evaluation based on some basic guitar “riffs/phrases” visualisations. We focused on four different guitar phrases: two different chord progressions, a melody, an arpeggio, and a solo. The phrases were played in the same key, in order to produce similar visualisations (same colours, localisation of notes, etc).

We proposed three different experiments to the users. In the first one, one of the two different chord progression recordings (Figure 7 and Figure 8) was presented to the user, and then the visualisations of the two chord progressions were shown in silence. The user had to choose the visualisation that matched the audio recording.

The second experiment was the opposite, given one visualisation (presented in silence), the user had to select from two recordings the one that matched that visualisation, in this case, using fragments of the solo and melody (Figure 9) phrases. In addition, the user was asked to indicate the complexity he/she found when doing the first two experiments.

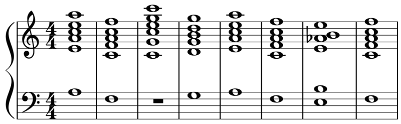


Figure 7. Chord progression 1 score.

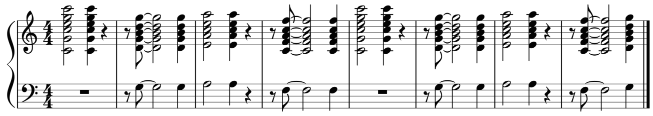



Figure 8. Chord progression 2 score.

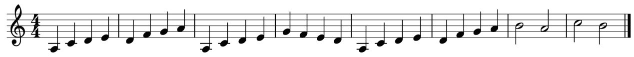



Figure 9. Melody score.

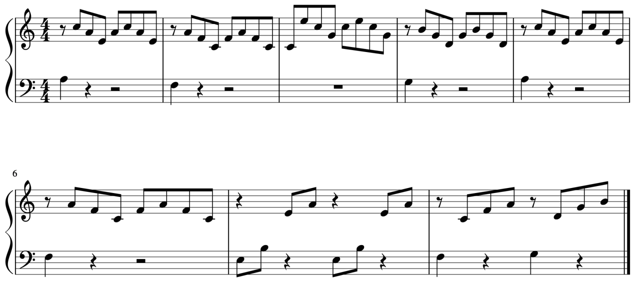



Figure 10. Arpeggio score.

The last test consisted of listening to all the phrases (sequentially, presented as a song) together with their corresponding visualisations, and afterwards, answering some questions to rate the system.

The questions evaluated the system in terms of:

- mapping quality and meaningfulness,
- expressiveness, subjectively evaluated by the user considering if the visualisations led to a stronger experience of music (multimodal perception),
- interest, if the system was considered interesting/promising by the user
- utility, in which context would a system like this one be used by the user.

The answers consisted of a score from 1 to 5 to express agreement, disagreement or neutrality, in addition to a text box in which the users could write their opinion, suggestions, or ideas for improvements.

5.2 Results

The experiment was conducted with 20 participants whose ages ranged from 21 to 55. They had different backgrounds and musical training. Besides, their musical taste was varied, as well as the frequency with which they went to concerts and listened to music. Table 1 shows the summary of the results of the experiments.

	Correct answers	Difficulty (1-5)
Test 1	80%	2.9
Test 2	75%	3.2

Table 1. Experiment results.

80% of the users were able to identify the correct answer to the first experiment, with a difficulty of 2.9 (the mean of the 1 to 5 range, where 1 was easy and 5 difficult); and 75% of the users answered correctly to the second experiment. In particular, participants with musical education and/or guitar players found the task easy, were able to distinguish between the three visualisations, and even imagine how the music would sound before listening to it.

	Score (1-5)
Mapping quality/meaningfulness	4.3
Expressiveness	4.2
Interest	4.8

Table 2. System valoration.

Table 2 shows the users valoration of the system in term of mapping quality and meaningfulness, expresiveness and interest. The score ranges from 1 to 5, in which 1 means disagreement and 5 is strong agreement. Several comments were made about the mappings. Most users found intuitive the proposed connections between the sound and visual domains, but many of them argued about the use of colour to identify notes. Also, most of the users liked the experience of simultaneous music and visuals, but some of them said the visualisations were very basic, and suggested that developing more "artistic" visualisations would work better and transmit more sensations.

All the users found the system very interesting, and suggested different contexts in which it could be used. Most of them proposed using the system in live music performances and concerts, to reinforce the emotions a particular music piece tries to evoke in the listeners; some participants suggested that the system could be used as a didactic tool to help people learn playing guitar, and musical concepts in general. Moreover, some participants said "as a tool to emphasize sensorial experiences for infants in primary education and give support in art classes", or even "for helping disabled people (i.e. people with hearing problems) to perceive and experience music".

6. CONCLUSIONS

Music is one of the most powerful art-expressions, and advancements in technology have opened new paths towards its exploration. Nowadays, many researchers focus their work on accurately representing music, but without forgetting its most emotive dimension of evoking sensations in ourselves. Our interest resides in guitar music representation to offer, through the system described in this paper, a way to visually experience it.

Throughout the experiments that were carried out, we noticed people found the system interesting and promising in many different contexts, and they liked the experience of simultaneous music and visuals stimuli. This research project is a work in progress. However, the process of making an evaluation and obtaining some feedback from the users was really useful to get ideas to improve the system, and also to demonstrate the attractiveness of this system for the public.

The next step we will take is to study how more musical features could contribute to a more useful system for the musician (as more detailed and complete information about musical events would be included), as well as making it more attractive/captivating for the audience, by the design of new approaches to visualizing the artistic dimensions of music.

Acknowledgements. This work has been partly sponsored by the Spanish TIN TIMUL project (TIN2013-48152-C2-2-R) and the H2020-ICT-688268 TELMI project.

7. REFERENCES

[1] A. P. Klapuri, "Automatic Music Transcription as We Know it Today," *J. New Music Res.*, vol. 33, no. 3, pp. 269–282, 2004.

[2] I. Bergstrom, "Soma: live performance where congruent musical, visual, and proprioceptive stimuli fuse to form a combined aesthetic narrative," 2011.

[3] R. Ivry, "A Review of Synesthesia," *Univ. California, Berkeley*, 2000.

[4] M. N. Bain, "Real Time Music Visualization: A Study In The Visual Extension Of Music," The Ohio State University, 2008.

[5] "Magic Music Visuals: VJ Software, Music Visualizer & Beyond." [Online]. Available: https://magicmusicvisuals.com/. [Accessed: 14-Jun-2015].

[6] "Guitar Hero Live Home | Official Site of Guitar Hero." [Online]. Available: https://www.guitarhero.com/. [Accessed: 01-Nov-2015].

[7] "Rocksmith® 2014 | Página oficial de España | Ubisoft®." [Online]. Available: http://rocksmith.ubi.com/rocksmith/es-es/home/. [Accessed: 01-Nov-2015].

[8] "Yousician." [Online]. Available: https://get.yousician.com/. [Accessed: 01-Nov-2015].

[9] 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 sound and music analysis," *Proc. ACM SIGMM Int. Conf. Multimed.*, 2013.

[10] C. Pramerdorfer, "An Introduction to Processing and Music Visualization."

DESIGNING DYNAMIC NETWORKED SCORES TO ENHANCE THE EXPERIENCE OF ENSEMBLE MUSIC MAKING

Alice Eldridge
Department of Music,
University of Sussex
alicee@sussex.ac.uk

Ed Hughes
Department of Music,
University of Sussex
e.d.hughes@sussex.ac.uk

Chris Kiefer
Department of Music,
University of Sussex
c.kiefer@sussex.ac.uk

ABSTRACT

This paper describes the impetus for, and design and evaluation of, a pilot project examining the potential for digital, dynamic networked scores to enhance the experience of ensemble music making. We present a new networked score presentation system, and describe how it has evolved through a participatory design approach with a primary school orchestra and through one-off sessions with several other ensembles. The design process has highlighted key issues concerning synchronisation between conductor, performers and notation, and autonomy and adaptation for performers. These key points are discussed and we show indicative feedback from users of the system along with future plans for the project.

1. INTRODUCTION

It is well recognised that ensemble performance participation raises self-confidence [1], and more recent work indicates that this is only true when the experience is enjoyable and rewarding [2]. We are motivated by the possibilities for dynamic networked digital scores to enhance both access to and the experience of ensemble music making so that the benefits can be more widely shared.

Development of non-standard notation systems has historically been motivated in part by a desire to realise broader social and political ideals of engagement that common practice notation, as a closed system, is unsuited for [3],[4]. We are similarly concerned with promoting inclusivity, but focus on the potential for *mixed and modified* notation systems to increase engagement and enjoyment of notated ensemble music making.

The project impetus came from regular long term observation of a voluntary primary school orchestra activity, in which children showed signs of apparent distress when they got lost while trying to perform arrangements of classical and popular music. Following from this, a significant proportion of the conductor's time was spent helping students to find and keep their place in the score. This raised the question of whether a digital system could be developed to synchronise and highlight the performers' score with the conductor's, mitigating the chances of getting lost.

Such a system could also lower the bar of entry to ensemble playing, opening out the benefits of ensemble performance of pre-composed music to players without musical training. Although motivated by and designed for a specific user group, such a system could have huge potential benefit in supporting ensemble music in the classroom, as well as in therapeutic settings and community and professional creative music making.

In order to design and evaluate the potential for such systems, we have adopted an iterative participatory design methodology [5], with a team consisting of a composer/ arranger, programmer/ researchers and psychologists. In this paper we describe work to date on a pilot project exploring the high level question: *Could digital, dynamic networked scores be transformative to the experience of musical ensemble playing?*

In the remainder of the introduction we give an outline of the established benefits of active participation in group music making. The specific needs of our case study group are then given, before a summary of the aims of the project. Related research and commercial systems of relevance are described in section 2. Our design methodology, software design and repertoire choices are described in section 3; our evaluation strategies are outlined in section 4 and finally formative results and future directions are discussed.

1.1 General benefits of ensemble playing

Participating in orchestral and ensemble performance of notated music has been found to confer multiple benefits including enhanced perceptual, cognitive, creative, social and physical development and well being across all ages. Active engagement with practical music making in young people has been shown to augment a range of widely transferable skills including improved performance in reading, mathematics [6] and verbal memory [7]; enhanced auditory and audiovisual processing of speech and music [8] and increases in general Intelligent Quotient [9] have also been reported.

Collaborative music making is also known to enhance key aspects of social skills in all ages such as co-operation, commitment and mutual support and to increase a positive sense of shared accomplishment. Enhanced self-esteem, self-confidence, and sense of belonging as well as providing an outlet of relaxation are similarly cited as positive benefits [2]. The success of the recent Opera North residency programme, *In Harmony* provides a vivid real-world illustration of the transformative power of ensemble music making across academic, social and personal lives of young people [1].

As a shared experience that can bolster cognitive, creative, personal and emotional capacity, enhance well being and transcend linguistic and cultural boundaries, ensemble music making has tremendous potential as an activity in educational, social as well as therapeutic settings. However, the use of standard notation can exclude those without existing musical literacy skills and seem inaccessible to those who cannot afford or access private music lessons. Further, research shows that the positive effects of engagement on personal and social development are only conferred when the overall experience is *enjoyable and rewarding* [2]. This can be a real issue for inexperienced readers when they first start playing in ensembles: lack of confidence technically and especially with reading musical notation means they struggle to keep their place - this can be off-putting and distressing, especially for beginners, who may conclude incorrectly that they lack ability.

1.2 Southover CE Primary School (case study)

As noted in section 1, the impetus for this project came from observation of exactly this lack of confidence detracting from otherwise positive musical experiences. Whilst the ultimate scope of application is broader (see section 6), the school orchestra provides a case study with which to develop and evaluate a pilot system addressing core pedagogic themes.

In this particular school, rehearsals take place every Thursday morning during term, before school starts, from 8.15 to 8.55am. Each week, simple arrangements of classical or popular music are rehearsed and performed. Although styles vary, difficulty is fairly consistent and arrangements have certain things in common, including: brevity (typically 16 to 24 bars); flexible instrumentation; simpler parts for novices; lead lines normally for the confident 'treble' players (e.g. flute, violin); a basic bass line (e.g. cello); a keyboard part for teacher or other skilled volunteer to 'fill in'.

Even in shorter arrangements, it is quite common for students to get lost and for the performance to break down. The experience for the students is positive enough for them to attend (current membership is just over 20) but observation suggested that they also experience stress when they get lost. This is deleterious to enjoyment and therefore the benefits of the experience. The music teacher and ensemble leader describes the challenges of rehearsing:

> *Running a primary school orchestra is a rewarding but challenging task. ... The challenge of juggling the able and those who need support is always difficult. As the orchestra becomes larger, keeping the children all in the right place on the score becomes harder particularly those whose skills in reading musical notation is weak.*
> – Gill Fenton. Music teacher and ensemble leader.

[1] www.operanorth.co.uk/education/in-harmony

1.3 Summary of aims

The key question we addressed in this pilot programme is: *(How) Could a networked dynamic scoring system be transformative to the experience of musical ensemble playing?*

Software development requirements and evaluation methodology were designed to address the following sub-questions:

1. Could a digital system that promotes/supports ensemble music performance help mixed ability musicians to keep their place and understand the shape of the music?
2. How might this technology impact the student players' experience and enjoyment of music making?

   Given the characteristics of repertoire outlined above, in particular:

   (a) could a system help make longer musical structures more accessible to a wider group?
   (b) could a system facilitate the performance of ensemble music featuring more polyphony?
   (c) could a system support the performance of ensemble music that is initially unfamiliar (i.e. sight reading)?

2. RELATED PROJECTS

2.1 Pedagogic - notation support

Online, digital and off-line materials already exist to support musical ensemble performance but have hitherto not harnessed the potential of networking. For example, *Figure Notes* [2] which provides a paper and label based system for developing notation skills through a progressive approach from graphic notation to full musical notation; *Charanga* [3] also provides an extensive music teaching resource including a combination of digital support and materials to cultivate musical appreciation and performance in schools. A growing number of desktop and mobile apps also support the digitised score management: *Forscore* [4], for example enables creation, download, management and sharing of PDF scores. Similarly, a number of interactive applications are emerging to support music pedagogy more broadly: *Smartmusic* [5], for example, generates interactive scores from Finale files, and deploys real-time machine listening to track performance skills.

2.2 Digital Music Stands

Various commercial digital music stands exist which include progressive features such as remote page turning functionality and enlarging font size (e.g. *musicpad*, [6] textitsame page music [7], *MusicReader* [8]). Some networked solutions also exist: *eStand* [9] supports document sharing

[2] http://www.drakemusicscotland.org/figurenotes
[3] http://charanga.com/site/
[4] http://forscore.co/
[5] http://www.smartmusic.com/
[6] http://www.musicpad.co.uk
[7] http://www.samepagemusic.com
[8] http://www.musicreader.net/
[9] http://www.estand.com

across a collection of stands, for example, but at the time of writing centrally coordinated dynamic update of content is not supported. Products which automatically annotate digital scores, such as *weezic* [10] also exist, but without networking capabilities.

2.3 Score Following and Networked Music

Decades of research in computer music provides inspiration for solutions to some of the technical challenges of the project. Research in score-following and gestural control, networked music and bespoke platforms for generative music are of particular relevance. Pioneered in the 1980s [10] and still an active area in Music Information Retrieval (MIR) today, *score-following* involves the analysis of a live audio input in order to provide real-time tracking of location in a predefined score. Although we do not plan to include any audio analysis features in the current project, technical aspects of this research may prove useful in the case where a conductor is present in order to enable their gestures to control the tempo and so presentation of the dynamic scores. Relevant technical solutions can also be found in research in Networked Music, a rapidly expanding community [11] exploring implications and applications of networking technology on performance practice e.g [11], [12]. Similarly, dynamic creation of musical scores which has been explored in the context of generative music [13] provides inspiration.

2.4 Realtime Scores

As the laptop approached ubiquity as a performance instrument in the early 2000s, an increasing number of practitioners and researchers explored *realtime notation* - traditional or graphic notation "which is created or transformed during an actual musical performance" [11] p.1. See [11] for a good selection of approaches to real-time notation practice, exploring musical, technical and design perspectives.

3. DESIGN

We developed a set of iOS applications to create a unique combination of these existing systems: an open source, networked, active, score display system. Apple's iOS was chosen as the platform for development, as it is currently the predominant choice in UK schools, with tablet use predicted to continue rising over the coming years to levels that could facilitate use by musical ensembles in a significant number of institutions [14].

We adopted a participatory design (PD) approach to the development of the system; an ethos which foregrounds 'designing *with* rather than *for*' stakeholders (see e.g. [5]). PD grounds design in a democratic process where designers work in-situ with a community; by encouraging participation from stakeholders the design process is more likely to reflect their needs. PD is an emergent process, with an artefact taking shape through regular engagements with the stakeholder community in which the design is iteratively refined. To this end we had regular meetings and workshops with the key stakeholders for this project: hands-on workshops with young people representative of the players in the orchestra; advisory group meetings with other music educators and performers; interviews with the classroom music teacher of our case study, sessions with the school orchestra and a series of engagements with a variety of musical ensembles to evaluate broader uses of the system. Themes arising from these meetings are outlined below in section 5.

3.1 Mobile Applications

The software takes the form of two iOS applications: *NETEM Conductor* (NC) gives the conductor control over instances of the slave application *NETEM Performer* (NP). NC presents the conductor with the full ensemble score, and allows them to control the position in the score. NP shows a section of the score, synchronised to the current position of the conductor's software. To aid performers, the current bar is highlighted in the score. The application can display parts for all instruments in the score; the performer selects the one they would like to view. The software is open-sourced, available on GitHub [12] .

3.2 Score Rendering

The application imports data in the widely used MusicXML format [15]. This enables compatibility with the majority of notation software and online score archives. We have developed our own lightweight rendering engine, using the OpenFrameworks [16] creative coding toolkit. Standard notation is represented using Adobe's *Sonata* font.

3.3 Networking

Our system uses Bluetooth 4 networking to synchronise the performer apps to the conductor. The use of this protocol instead of standard WiFi removes reliance on institutional networking systems or additional routing equipment, and also requires no configuration, making it simple and straightforward to use. Synchronisation data is multicast to all local devices through the advertising data block. By sharing data in this way, there is no need to create a formal connection between devices, and the system can thereby exceed the connection limit of 12 devices on the iOS bluetooth stack. Further to this, messages can be transmitted at a high frequency, every 1-2ms, compared to the 20ms minimum for Bluetooth characteristic notifications in iOS.

3.4 Conductor Control

PD sessions emphasised dynamic synchronisation between conductor, performers and the score as one of the key factors in the usability of the system (further discussed in section 5). The systems offers two options: automatic sequencing at a selectable tempo, or tap tempo control. The former is designed for early stages of learning a new piece when the ensemble leader might need to be 'handsfree' in order to direct sectionals – the system can also provide a

[10] `http://weezic.com`
[11] `http://networkmusicfestival.org`
[12] `https://github.com/NETEMSussex`

click track to assist time keeping. Tap tempo mode mimics the traditional model of conductor control and allows the conductor to advance of score beat by beat. A new motion tracking system is under development which will enable the conductor to control the score gesturally.

3.5 Repertoire

In 2014 the BBC released a film, *Ten Pieces*, aimed at young people to promote the excitement of classical orchestral music through visually engaging performances of music ranging from Handel to Anna Meredith. The BBC also supported this by commissioning static (PDF) parts for mixed ability performance and made them freely available online. All but four of these ten pieces are on IMSLP (because they are out of copyright). Selections of these works as well as new compositions were prepared in MusicXML format editions for the networked music notation system.

4. EVALUATION STRATEGY

The pilot project focused on specific features of the case study orchestra, but we are also concerned with the broader potential impact of networked digital scores in ensemble music making with a view to wider-ranging, longer term research and development. To this end, in addition to considering the effect of the introduction of the system within the case study primary school we trialled the system with three other existing ensembles of different musical styles, and staged a public event to see how *mixed* notations systems can support performers with an even wider range of musical experience.

4.1 Sessions with the Southover CofE Primary School Orchestra

The system was tested and developed throughout five rehearsals with a voluntary primary school orchestra group, culminating in a final performance in the school assembly. See figure 1. We recorded video documentation for all of these sessions, along with interviews with the conductor, performers and the parents who assist the running of the orchestra. Between these sessions, the software was iteratively refined based on feedback from all stakeholders. Along with the collected notes of the research team, these materials were archived for qualitative analysis.

4.2 Student surveys

The impact of the networked scores on the primary school pupil's experience of the orchestra was evaluated through survey-style questionnaires with the students, and a focus group with the students and music teacher who leads the session (both parties, being considered as 'users' in this setting) and analysis of video documentation of the sessions [13]. The Likert-style survey was designed to capture students' thoughts and feelings around their sense of belonging and worth within the orchestra, musical understanding, engagement and enjoyment. Students completed the survey every week for 5 weeks after orchestra practice using their usual paper scores and a further 5 weeks whilst rehearsals were run using the networked system. Familiarity with musical repertoire was controlled for by working with one new and one familiar piece of similar standards during both phases.

4.3 Further Ensemble Sessions

The system was used to facilitate a session combining professional classical players [14] together with rock musician students at the British and Irish Institute of Modern Music (BIMM) [15] – shown in figure 2. The system was also used for a rehearsal combining the same professional classical musicians with students from East Sussex Academy of Music.

4.4 Public performance

A public performance took place as part of Brighton Science Festival 2016. Members of the public with no musical experience were invited to come and take part in the rehearsal and performance of a new work by composer Ed Hughes for Sussex CoMA [16] (pictured in figure 3). Sussex CoMA is a sinfonietta orchestra of adult players with diverse backgrounds; public participants were recruited via the science festival. Scores for orchestral players were presented in standard western music notation; non-notation reading public participants had colour coded scores where simple note events corresponded to coloured pitched chime bars and bells. Two sets of rehearsals and performances (with up to 10 public members in each) were followed by a round table discussion. The event enabled the real-world testing of the potential for dynamic, networked, mixed notation systems to support high quality amateur music making in mixed ability ensembles.

5. DEVELOPMENT AND EVALUATION

Statistical analyses of the questionnaires and qualitative analysis of materials collected during the PD sessions is under-way and will be published in forthcoming journal articles. In this section we report indicative feedback, arising throughout the development and testing sessions, under a number of themes; quotes from players and workshop leaders are used to illustrate the ways in which work to date addresses our core research questions.

1. Presentation of performer parts.

   Ease of adoption versus ease of adaptation. As with all digital design projects, consideration must be paid to the degree to which operational metaphors of the 'analogue' tools should be followed versus introduction of new modes of interaction which take advantage of the unique dynamic, programmable capacities of digital media. The former guarantees rapid

[13] Evaluation methods have been subject to ethical review and approved by University of Sussex Social Sciences and Arts C-REC (ER/EDH20/1 and ER/EDH20/2)

[14] The Orchestra of Sound and Light, http://www.orchsoundlight.org/

[15] http://www.bimm.co.uk/

[16] Contemporary Music for All: http://www.coma.org

Figure 1. Rehearsals at Southover CE Primary School Orchestra Group.

Figure 2. Workshop at BIMM Brighton

Figure 3. Brighton Science Festival Open Rehearsal: Members of COMA play from standard notation displayed on the iPads; members of the public (back row) play simple pitched parts coloured-coded with their instruments.

adoption of the functionally familiar tool; the latter may ultimately confer advantages, but relies on good design to enable users to adapt. Our intuitions as performers were supported by comments from all groups: it was noted that being able to preview the forthcoming bars was crucial. Some individuals expressed an interest in embracing the potential of digital technology and developing a scrolling score; others wished to preserve static pages.

Based on early feedback, the performer software allowed four modes of presentation, which can be set by the conductor: 1) displaying a static page which 'turns' a la paper score; 2) slowly scrolling the notation, giving the player a dynamic 'look ahead' of X bars, 3) a hybrid model in which a new 'page' of staves is displayed after the end of the *penultimate stave*, 4) a mixed view with current location magnified, and small scale 'overview' of the piece. Option 3 proved the most popular as it allowed players the necessary 'look ahead' conferred by a static page of music, whilst mitigating against the urgency of page turning.

Autonomy. The software used in the workshops allowed only the conductor to control the position of the scores for the performers. In early workshops, some of the younger players felt frustrated at the lack of autonomy in viewing the score. This had obvious disadvantages when performing new music too. The final version therefore includes a 'browse mode' which allows players to scroll through their parts during a break in the rehearsal (i.e. X seconds after the conductor stops conducting).

2. Conductor control.

   In the first workshop we conducted, the score was sequenced by the conductor app, at a tempo selected by the conductor. This led to serious synchronisation issues when the musicians played rubato, as the conductor was unable to correct the constant score playback speed accurately. This highlighted the fundamental importance of synchronisation between conductor, performers and the score. In workshop two, we trialled a simple tap tempo device, where the conductor tapped a large button the tablet, to progress the score by one beat. Even without any sophisticated tempo detection algorithms, this was successful in resolving the synchronisation issues but introduced another problem by forcing a change of the conductors' behaviour. This approach evolved with feedback from the conductor of the primary school orchestra into a system that would sequence the score autonomously, but allow the conductor to intervene to adjust tempo. Informal feedback and observations indicate this system provides a sufficient balance between managing synchronisation and allowing expressive tempo variation; for example, it allowed the tempo acceleration in Grieg's *Hall of the Mountain King*, and also allowed the school conductor to walk around and help children while the system played automatically.

Ideally the system would synchronise tempo to the motion of the conductor by observing their natural style of conducting. To this end we are currently designing a wearable wireless tempo estimation system using the Invensense MPU9250 motion sensor for use in the final trial. The test system will afford two modes of control: metronome mode where tempo is rigidly controlled by a clock and click track and expressive conductor mode, where tempo is inferred from the conductor's baton, allowing expressive beat-by-beat tempo variation.

5.1 Indicative results

Observations from sessions with all ensembles suggest that the simple intervention of dynamically highlighting the current beat and bar resulted in an increase in confidence in players of all abilities which led to both increased enjoyment and greater engagement in the music. Several players and attendant parents of the school orchestra also commented how much better the orchestra *sounded* - presumably due to the greater relaxation and enjoyment.

- Beginner students in the Lewes primary school orchestra group commented that in not having to concentrate so hard on keeping their place in the score, there were able to focus more on other aspects of their performance, such as intonation, and listening to the others.

- Similar responses came from players at the BIMM workshop:

  "it really facilitated a [first time] play through and we were able to get creative, all in one session" ;

  "I'm not used to reading gigs at all, I never read, and I thought my biggest fear was that I was going to lose my place and I found this really helpful because it allowed me - especially in the improvising parts - I could really focus on what I was playing and thinking more about how I phrased it and dynamics and stuff without worrying where I was."

- Some less confident members of the school group suggest that the networked scores not only enhanced their experience by supporting them in keeping their place in the music, but scaffolded their longer term musical learning:

  " The iPads has stopped me getting lost in the music and I thought I was letting down the team and they are making me feel happier and Im not letting them down and now I know where Im going. When it stops I get lost but it doesnt usually happen. Its better than sheet music when I got very lost. Other people were down below and I was at the top. I will be more confident going back to sheets it will be better because the iPads are helping me use the music."

  The possibility than enhancing the *experience* of playing (by reducing the stress of getting lost) may have positive long term impact on musical engagement is interesting and warrants further consideration and investigation.

- Comments from a parent-helper at the school group suggest that the young players felt more secure in general, and that this had an perceptible impact on their music making:

  "Ive been watching these sessions and I think its amazing how they all stay together; in the silences of 'In the Hall of the Mountain King' these are all observed; its been a fascinating experience. You lose that sense of panic when you dont know where you are...its really helped them to stay together and stay focused on the music."

- For the school group, the system enabled the orchestra to sight-read through an arrangement of Holst's *Mars* from *The Planets*, the conductor commenting *"that would be unthinkable without the iPads"*.

- In the public Science Festival performances, novice and non-musicians young and old were supported in actively participating in a public orchestral performance in a way which would not have been possible without the system. The potential for such mixed and modified notation systems for supporting mixed experience performances is really encouraging.

6. SUMMARY AND FUTURE DIRECTIONS

We developed a dynamic, networked notation system for iPads linked by a Bluetooth network. The system was designed and evaluated over several months in a series of Participatory Design workshops with a primary school orchestra club, as well as workshops with players with a range of other ages, genres and abilities. Initial results, illustrated by comments from stakeholders, point to the benefits of dynamic networked scores in pedagogic settings. As outlined in sections 1.2 and 1.3 we are addressing some very specific questions about musical engagement and the subjective experience of rehearsing and performing in amateur ensembles and assessing the impact of the introduction of the system with qualitative and quantitative methods. Initial results suggest that the system serves the primary function of supporting players in keeping their place. Individual feedback suggests that this has positive effect not only on immediate experience; the possibilities that this may aide longer term musical engagement, as well as impacting the quality of music making is implied and inspires further research. The Science Festival event in which members of the public played simple pitched parts within a contemporary music ensemble provided further insight into the possibilities for dynamic networked scores using a mix of standard and graphical notation to cohere mixed ability musicians in a single ensemble performance.

In future work we plan to explore how generalisable these observations are by investigating the creative musical, pedagogic and therapeutic possibilities of such technology using further modified (layered, augmented, annotate-able etc.) notation, experimental graphical notation and mixed models (for example in electro-acoustic settings). In consultation with our advisory group, we see scope for adapting and extending this work to professional and wider adult community settings, where ideas about synced parts with richer elements of dynamic information may be productively explored (e.g. in experimental music, or professional classical ensembles), alongside the potential therapeutic benefits of ensemble performance experience amongst adult beginner musicians in different contexts, for example workshops for those recovering from the trauma of Accident and Emergency, arts therapeutic settings and so forth. In order to maximise accessibility we are currently planning ports to Android and ChromeOS to enable access beyond iPads.

The NETEM project brings together many elements of concern in contemporary music making, drawing from established research in experimental notation, networked music, dynamic scores and more recent commercial interest in music apps to develop a networked, dynamic music score presentation system. In a short term pilot we developed the system and are evaluating whether such technology can have a positive impact on the experience of ensemble music making. In future work we plan to explore dynamic networked scores of mixed and modified notation systems in a range of therapeutic, pedagogic and creative music settings.

Acknowledgments

This project was funded by AHRC Digital Transformations, grant number AH/M010120/1. Many thanks to Fidelam Hanrahan, the Southover CE Primary School Orchestra, Lewes and their conductor Gill Fenton, Brighton BIMM, Brighton Science Festival, COMA Brighton and all the public participants who took part in the open rehearsals.

7. REFERENCES

[1] D. Kokotsaki and S. Hallam, “The perceived benefits of participative music making for non-music university students: a comparison with music students,” *Music Education Research*, vol. 13, no. 2, pp. 149–172, 2011.

[2] S. Hallam, “The power of music: Its impact on the intellectual, social and personal development of children and young people,” *International Journal of Music Education*, vol. 28, no. 3, pp. 269–289, 2010.

[3] H. Brün, *When music resists meaning: the major writings of Herbert Brün*. Wesleyan University Press, 2004.

[4] C. Wolff, “Open to whom and to what,” *Journal of New Music Research*, vol. 16, no. 3, pp. 133–141, 1987.

[5] E. B.-N. Sanders and P. J. Stappers, “Co-creation and the new landscapes of design,” *Co-design*, vol. 4, no. 1, pp. 5–18, 2008.

[6] D. E. Southgate and V. J. Roscigno, “The impact of music on childhood and adolescent achievement*,” *Social science quarterly*, vol. 90, no. 1, pp. 4–21, 2009.

[7] Y.-C. Ho, M.-C. Cheung, and A. S. Chan, “Music training improves verbal but not visual memory: cross-sectional and longitudinal explorations in children.” *Neuropsychology*, vol. 17, no. 3, p. 439, 2003.

[8] N. Gaab, P. Tallal, H. Kim, K. Lakshminarayanan, J. Archie, G. Glover, and J. Gabrieli, “Neural correlates of rapid spectrotemporal processing in musicians and nonmusicians,” *Annals of the New York Academy of Sciences*, vol. 1060, no. 1, pp. 82–88, 2005.

[9] E. G. Schellenberg, “Music lessons enhance iq,” *Psychological Science*, vol. 15, no. 8, pp. 511–514, 2004.

[10] R. B. Dannenberg, “Music representation issues, techniques, and systems,” *Computer Music Journal*, pp. 20–30, 1993.

[11] J. Freeman and A. Clay, *Virtual Scores and Real-time Playing*. Routledge, 2010.

[12] A. R. Brown and S. Dillon, “Networked improvisational musical environments: learning through online collaborative music making,” *Music education with digital technology*, pp. 96–106, 2007.

[13] A. R. Brown, “Generative music in live performance,” 2005.

[14] BESA, “Tablets and connectivity 2015: Full report, english schools,” June 2015.

[15] M. Good and L. Recordare, “Lessons from the adoption of musicxml as an interchange standard,” in *Proceedings of XML*. Citeseer, 2006, pp. 5–7.

[16] J. Noble, “Programming interactivity: A designer’s guide to processing, arduino, and openframeworks,” 2009.

Conversion from Standard MIDI Files to Vertical Line Notation Scores and Automatic Decision of Piano Fingering for Beginners

Yasuyuki SAITO
Nat. Inst. of Tech., Kisarazu College
Chiba 292-0041, Japan
saito@j.kisarazu.ac.jp

Eita NAKAMURA
Kyoto University
Kyoto 606-8501, Japan
enakamura@sap.ist.i.kyoto-u.ac.jp

Riku SATO
Nat. Inst. of Tech., Kisarazu College
Chiba 292-0041, Japan
j11424@j.kisarazu.ac.jp

Suguru AGATA
Showa Univ. of Music
Kanagawa 215-0021, Japan
agata-s@snow.ocn.ne.jp

Yuu IGARASHI
Showa Univ. of Music
Kanagawa 215-0021, Japan
y_uu_@yuxmu.com

Shigeki SAGAYAMA
Meiji University
Tokyo 164-8525, Japan
sagayama@meiji.ac.jp

ABSTRACT

This paper introduces a method for converting standard MIDI files to the "vertical line notation" (VLN) and an algorithm for automatic decision of piano fingering for beginners.

Currently, staff notation is widely used for various instruments including piano. However, this notation often appears hard to beginners. On the other hand, VLN is intuitive and easy to understand for piano beginners since it graphically indicates the time order of notes as well as fingering. With the VLN score, piano beginners can make smooth progress with correct fingering, and it is expected to be useful for early education in music.

A problem with VLN is that it is currently created by hand with a spreadsheet software. It would be desirable to automatically produce VLN scores from existing digital scores.

Our proposed method can solve above problem. In addition, this paper also presents some examples of practical and successful use of VLN scores.

1. INTRODUCTION

Among many existing types of music notations including staff notation, tablature notation for guitar and luto, one line notation for percussion, graphic notation, numeral notation, and Japanese traditional notation [1], staff notation is most commonly used for a wide variety of musical instruments.

Notations have evolved to express composer's intentions as accurately as possible, and therefore, they became advanced and complex with the times. For instance, staff notation provides musician with a variety of information, including pitch, accidentals, note length, dynamics, trills, turns, articulations (including staccato, tenuto, accent and attack), etc.

However, to music beginners, it is difficult to read even pitch and the length of a note in staff notation. For this reason, in many cases, people who are not accustomed to play the piano often give up in the process of reading a staff notation scores, even for a short musical piece and simple melody (see section 3.4).

With this background in mind, in 1985, Suguru Agata proposed vertical line notation (VLN) which is designed by piano roll style for the piano beginners [2]. It is very easy to understand the pitch of notes on a VLN score, and it can also show piano finger numbers.

In this paper, we first review how to read VLN scores. Next, we explain a conversion method from standard MIDI files to VLN scores, and propose an automatic decision of piano fingering for the piano beginners. Moreover, we show applied examples in which piano beginners could play the piano easily using VLN scores.

2. VERTICAL LINE NOTATION

Figure 1 illustrates how to read a VLN score. In a VLN score, the horizontal direction expresses note pitch; the left side is for lower pitches and right side is for higher pitches. The vertical direction shows time flow; the direction toward the bottom represents the progression of the music.

The horizontal positions of circles represent the pitch of notes while blue bars designate the temporal order of notes. We call the circle "node" and the bar "link," after the research field on graphical model [3]. Each node contains a finger number. Therefore even piano beginners can play a musical piece with correct fingering very easily. They only need to strike the keys by following the finger numbers in the nodes. For piano players, it is very important to get into the habit of playing with correct fingering. The link shows whether the pitch of the next node is higher or lower than the current node. This is useful in searching for the next node.

Figure 2 shows an example of staff notation score of a children's song "tulip." Figure 3 shows a VLN score of the song "tulip." This VLN score is made for Japanese children or for older persons, so the title is written on the top of the VLN score in hiragana letters. The lyrics are also written on the left side using hiragana letters.

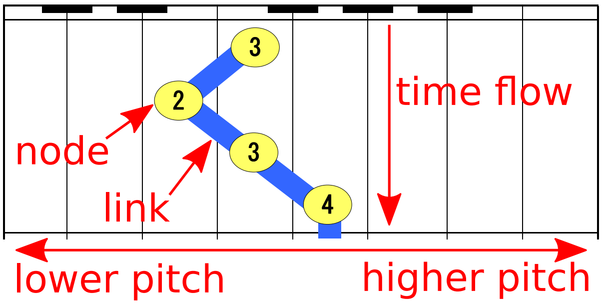


Figure 1. Format of VLN score.

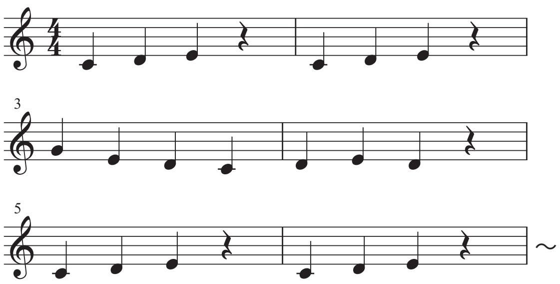


Figure 2. Staff notation score of a children's song "tulip."

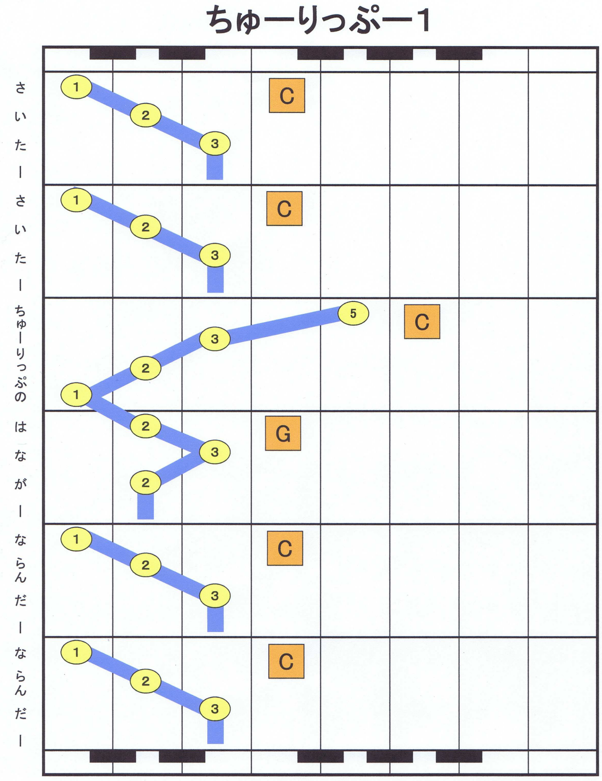


Figure 3. VLN score (handmade) of "tulip" with Japanese lyrics.

If the appropriate adjustment of magnification ratio of a VLN score is achieved, the graph of keys in the VLN score can be made to correspond to the keys of a real piano keyboard. As a result, the piano player can easily find the key to be pressed on the keyboard just below the node. Moreover, because a VLN score always shows real pitches, it is not influenced by key and clef sign in staff notation. In contrast, in the staff notation, pitch is shown as vertical position, and therefore the correspondence of pitch in staff notation to piano keyboard is not intuitive.

At the same time, it would be necessary for the beginners to gradually be capable of reading staff notation. We will discuss the transition from using VLN scores to staff notation scores in Section 8.3.

3. PERFORMANCE BY USING VLN SCORES

3.1 Beginners' piano class for senior persons in Showa Univ. of Music

We began a beginners' piano class for elder people in Showa University of Music in 2010. The age of participants is from 60s to 80s. In this piano class, we have used electronic keyboards. Figure 4 shows the configuration of the piano class.

In the lesson, we teach through three stages instead of playing music from the beginning. First, the participants hit table by finger in accordance with VLN score. In this stage, the participants concentrate on only correct order of the finger without much attention to the position of key on the keyboard.

Second, while turning off the power of the electronic keyboard, they practice key strokes. Even if they make mistakes in the fingering, they need not to worry about it. The important things are to continue until the end, and not to be afraid to fail. This "silent practice" characteristics is unique to the electronic keyboard, and cannot be done on the acoustic piano.*1

Third, participants play the music with sound after turning on the electronic keyboard.

We have found the all the participants can play the piano and have enjoyed the practice.

3.2 Piano class in Rest Villa Ebina

We have held piano classes with VLN scores at Rest Villa Ebina which is a private facility providing long-term care to the elderly. Figure 5 shows the situation. In this photo, two persons use wheelchairs. One of the residents of the facility suffers from Parkinson's disease. The upper half of her body usually leaned to the right side. However, her posture has improved, since she started the keyboard performance using VLN scores. In addition, her speech ability has also changed for the better, and she has been able to evaluate the performance of others. It is considered that muscle and brain are activated by playing the keyboard, and we suggest that VLN score can be effectively utilized in the welfare field and it gives purpose in life for senior persons.

If we had used a complex staff notation score, people might have lost interest. However, by using the VLN score,

*1 The same effect is obtained by using acoustic pianos with "silent mode", for example, YAMAHA Silent Piano series.

Figure 4. Piano class for senior people at Showa Univ. of Music.

Figure 5. Piano class for senior people at an elderly persons' home.

they have been interested in the piano performance, and they have practiced enthusiastically.

3.3 Beginners' piano class for students at NITKC

We tried to apply VLN scores to piano class for young people. We held a beginners' piano class for students at the National Institute of Technology, Kisarazu College (NITKC) in May 2015. Actually, 13 students who were interested in playing the piano volunteered for this class. They experienced the same three stage practice method as the senior persons. Amazingly, after 90 minutes of practice, they could play the piano using both hands.[*2]

3.4 Open campus at NITKC

In open campus at NITKC, various studies are introduced to junior high school students and their parents. We explained the automatic accompaniment system (AAS). The name of the AAS is "Eurydice". This AAS is developed by using C++ language on "Qt" which is a cross-platform development framework for graphical user interface (GUI). Eurydice deals with MIDI signals; the input MIDI signal is given from MIDI piano as performance data of human player, and the output MIDI signal is accompaniment data [5]. A standard MIDI file (SMF) as musical score information is given to Eurydice in advance, and the AAS estimates a musical score position of the performance of the human player. Eurydice allows errors in performances such as insertion of unnecessary notes, deletion of notes, and lack of accidentals. Notably, Eurydice is the world's first system that allows arbitrary jumps; the human player may practice playing the same bars again and again, or go to a later section of the score by skipping over some bars; in either case Eurydice follows the player's performance.

Until 2014, we have used staff notation at the presentation. At first, as a performance example, we played a melody of a short piece of music, and Eurydice played an accompaniment part at the same time. After that, we asked

[*2] Some video clips are shown in the following Website: `http://beam.kisarazu.ac.jp/~saito/research/VLN/`

Figure 6. Piano class for students at Nat. Inst. of Tech., Kisarazu College.

Figure 7. Piano performance by a junior high school student in open campus of NITKC. She could play the VLN score at sight.

junior high school students to play the music. However, most of the students did not try to perform on the piano.

In 2015, we explained VLN and demonstrated performance by using a VLN score with Eurydice. Figure 7 shows the piano performance. Consequently, all the junior high school students who visited our demonstration (about 20 persons) tried playing the piano, and performed to the end at sight without regular practice. We consider that the psychological distance for piano of junior high school students is shortened by using VLN score.

4. AUTOMATIC CONVERSION FROM SMFS TO VLN SCORES

Currently, VLN scores are made by using Microsoft Excel or fully hand written. The work takes a long time and is laborious. Therefore, it would be desirable to automatically produce VLN scores from existing digital scores.

There are several formats of digital score such as standard MIDI file (SMF), MusicXML, and custom formats of music score editors. Among these formats, SMF is the most widely used. It is easy to make SMFs by using commercial or free music software. Fortunately SMFs are distributed on the internet, in that case, people do not need to make SMFs themselves. This is a great advantage. We implemented a conversion system from SMFs to VLN scores [4].

4.1 MIDI standard

Musical instrument digital interface (MIDI) is a universal standard for the digital transmitting of performance data between electronic musical instruments [5]. It consists of certain standards, such as a physical transceiver circuit and interface, a communication protocol, a file format, etc. Transmission and reception of data on the MIDI standard are all carried out in MIDI messages. A MIDI message is constructed of plural bytes (8 bits per one byte). To transmit MIDI messages efficiently, the bytes which express MIDI messages are divided into two types: a "status byte" and a "data byte." The status byte has "1" as the most significant bit (MSB); namely, it contains 128 data types from 80H to FFH in hexadecimal format. In contrast, the data byte includes "0" as the least significant bit (LSB). Thus the range of the data is from 00H to 7FH.

4.2 Formats of SMF

There are three types of formats of SMFs. The format 0 has only one track which includes all note information. It is equivalent to mixed down music data. Therefore, this format is for the purpose of playback only. The format 1 can contain many independent tracks. This format easily deals with decomposed data on a MIDI sequencer. The format 2 has multitracks and multisequence patterns. In the case of playing the SMF format 2 data, one of the sequence patterns is chosen in each track. This format seems to have been originally developed and intended for use in, for example, karaoke. However, it is little used in other practice.

The SMF is composed of several parts. One is the "header chunk," which includes information of the whole SMF. The other is the "track chunk," which contains real performance data.

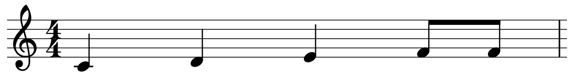


Figure 8. An example score in staff notation which includes eighth notes.

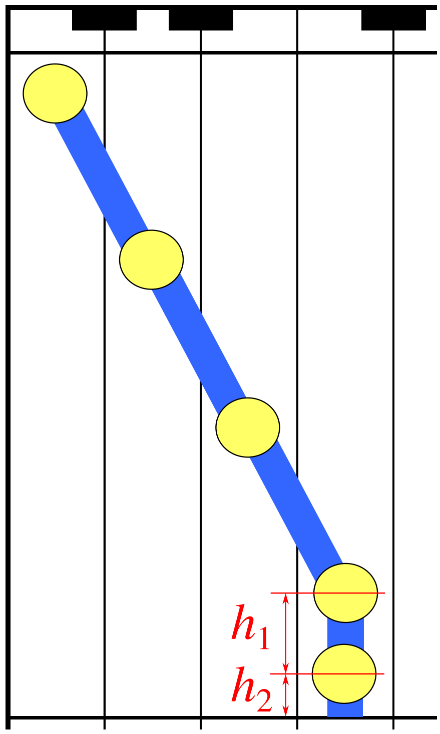


Figure 9. Previous style of VLN score (handmade) for Figure 8.

4.3 New style of VLN

Figure 8 shows an example score in staff notation which includes eighth notes. Figure 9 represents the conventional style of VLN score (handmade) which is equivalent to the score in Figure 8. In this case, in Figure 9, the length of the last eighth note is expressed shorter than the length of the previous one: $h_2 < h_1$.

Therefore we propose a new style of VLN as shown in Figure 10: the first note is put on the bar line. The length of the last eighth note is equal to the previous one: $h_2 = h_1$.

In the next section, we will explain the conversion method for making this style of VLN scores from SMFs.

4.4 Conversion from SMFs to VLN scores

In SMFs, the header chunk and the track chunk begin with a magic number as "MThd" and "MTrk." First, it is necessary to find these bytes in a MIDI file. In the header chunk, a time resolution, which is called "time base," is defined. This shows a capacity to decompose a quarter

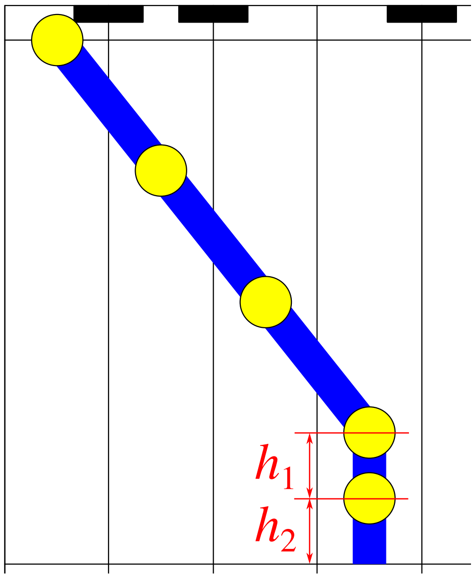


Figure 10. New style of VLN score (automatic converted from SMF) for Figure 8.

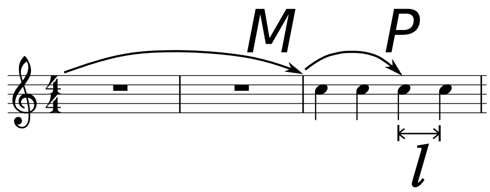


Figure 11. Measure number M, note position P, and note length (duration) l.

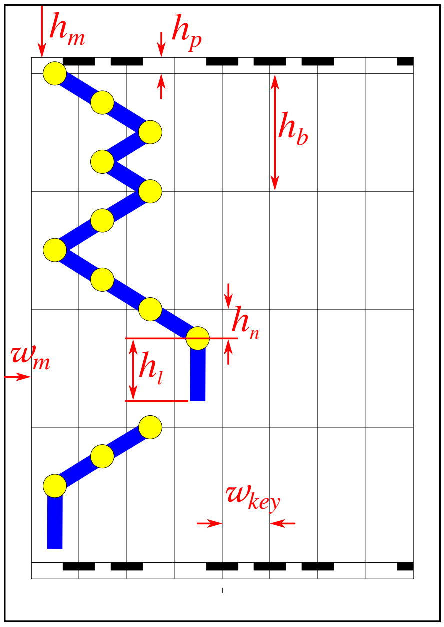


Figure 12. Parameters which are used for calculating y- and x-coordinates of node.

note. For example, if a time base is equal to 480, it is possible to express the quarter note as 480 divided lengths. "Delta time" is also described together in all MIDI events in which the MIDI file is represented. It represents the time until the next event is performed. Both "note-on" (a command for turning the sound on) and "note-off" (turning off the sound) are used to represent a note length. For example, in the case that a time length from note-on to note-off is equal to time base, it is equivalent to a quarter note.[*3] If we let S represent a time for note-on, and E be a time for note-off, then D means a time base. The meter will be four-four time, and a piece of music has no anacrusis. In these conditions, a note length l is calculated by Equation (1):

$$l = (E - S) \;/\; (D \times 4), \tag{1}$$

where $D \times 4$ is equal to the whole time length in a musical bar.

A note position P from the beginning in a measure is obtained by Equation (2), and a measure number M of a note is given as Equation (3):

$$P = S \; mod \; (D \times 4), \tag{2}$$

$$M = [S \;/\; (D \times 4)] + 1, \tag{3}$$

where "mod" means the remaining operations, "[ ]" indicates the floor function, and the multiplier "4" is equal to the denominator of the musical time. By using these three equations, performance data of a MIDI file are converted to data sequences which have only note length, note position, and measure number. Figure 11 shows l, P, and M.

[*3] Ref. [6] describes formalisms of duration-onset of notes which reminds to performance theory.

These parameters are used to indicate the vertical position y_{node} of the center of a node in a VLN score;

$$y_{node} = h_m + h_p + h_b \; M + h_n, \tag{4}$$

where h_m, h_p, and h_b shows the height of the top margin, of the piano area, and of musical bar in the VLN score, and h_n corresponds to the position of the node is the musical bar (see Figure 12):

$$h_n = h_b \; P. \tag{5}$$

Note that P is a common fraction. h_l shows the note length (in a VLN score) which is given as Equation (6):

$$h_l = l \; h_b. \tag{6}$$

The horizontal position of the node is calculated easily because SMF includes pitch information clearly. Let n_{lowest} mean the note number of the lowest pitch in the VLN score. This note number corresponds to a pitch v_{lowest} in Table 1 which is the lowest pitch in the VLN score. Let n_{node} represent the note number of the node, x_{node} which is x-coordinates of the center of the node is

p_{lowest}	v_{lowest}
C, D, E	C
F, G, A, H	F

Table 1. Lowest pitch in VLN score. p_{lowest} means lowest pitch in music piece, and v_{lowest} shows the lowest pitch in VLN score.

p_{range}	$v_{highest}$
within 1 octave	1 octave higher of lowest pitch
over 1 octave	highest pitch

Table 2. Highest pitch in VLN score. p_{range} means pitch range in music piece, and $v_{highest}$ shows highest pitch in VLN score.

given as Equation (7):

$$x_{node} = w_m + \frac{w_{key}}{2} + (n_{node} - n_{lowest})\frac{w_{key}}{2}, \quad (7)$$

where w_m shows the width of the left margin in the VLN score, and w_{key} means the key width on the VLN score. The reason why w_{key} is divided by 2 is that the unit of x_{node} is semitone.

The highest pitch in VLN score is obtained from Table 2. For example, in Figure 12, the lowest pitch C4 is assigned to v_{lowest}. Therefore n_{lowest} is equal to 64 which is MIDI note number of C4. The highest pitch is within the 1 octave from n_{lowest} (C4). Thus C5 is substituted into $v_{highest}$.

5. AUTOMATIC DECISION OF PIANO FINGERING FOR BEGINNERS

As shown in the previous section, VLN scores can be made from SMFs automatically, however, it is necessary to put a finger number into each node. VLN scores have finger numbers which are very important to pianist. On the other hand, it is not easy to decide the piano fingering. Hence, we have studied automatic decision of it.

Models and algorithms for decisions deciding appropriate piano fingerings have been studied previously [7, 8, 9, 10, 11]. To enable the adaptation of fingerings for beginners, we present a method based on the hidden Markov model (HMM) [10].

Piano fingering for one hand, say, the right hand, is indicated by associating a finger number $f_n = 1, \ldots, 5$ ($1 =$ thumb, $2 =$ the index finger, $\ldots$, $5 =$ the little finger) to each note p_n in a score,[*4] where $n = 1, \ldots, N$ indexes notes in the score and N is the number of notes. We consider the probability of a fingering sequence $f_{1:N} = (f_n)_{n=1}^N$ given a score, or a pitch sequence, $p_{1:N} = (p_n)_{n=1}^N$, which is written as $P(f_{1:N}|p_{1:N})$. As explained below, an algorithm for fingering decision can be obtained by estimating the most probable candidate $\hat{f}_{1:N} = \operatorname{argmax}_{f_{1:N}} P(f_{1:N}|p_{1:N})$. The fingering of a particular note is more influenced by neighboring notes than notes that are far away in score position. Dependence between adjacent notes is most important, and it can be incorporated within a Markov model. It also has advantages in efficiency in maximizing probability and setting model parameters. Although the probability of fingering may depend on inter-onset intervals between notes, the dependence is not considered here for simplicity.

[*4] We do not consider the so-called finger substitution in this paper.

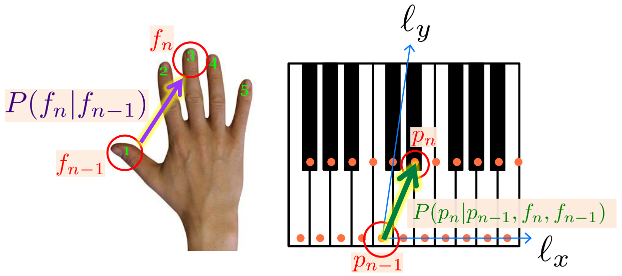


Figure 13. Schematic illustration of the piano fingering model based on HMM.

Supposing that notes in a score are generated by finger movements and the resulting performed pitches, their probability is represented by the probability that one finger would be used after another finger $P(f_n|f_{n-1})$, and the probability that a pitch would result from two consecutively used fingers (Figure 13). The former is called the transition probability, and the latter output probability. The output probability of pitch depends on the previous pitch in addition to the corresponding used fingers, and it is described with a conditional probability $P(p_n|p_{n-1}, f_{n-1}, f_n)$. In terms of these probabilities, the probability of notes and fingerings is given as

$$P(p_{1:N}, f_{1:N}) = \prod_{n=1}^{N} P(p_n|p_{n-1}, f_{n-1}, f_n)P(f_n|f_{n-1}), \quad (8)$$

where the initial probabilities are written as $P(f_1|f_0) \equiv P(f_1)$ and $P(p_1|p_0, f_0, f_1) \equiv P(p_1|f_1)$. The probability $P(f_{1:N}|p_{1:N})$ can also be given accordingly.

To train the model efficiently, we assume some reasonable constraints on the parameters. First we assume that the probability depends on pitches only through their geometrical positions on the keyboard which is represented as a two-dimensional lattice (Figure 13). We also assume the translational symmetry in the x-direction and the time inversion symmetry for the output probability. If the coordinate on the keyboard is written as $\boldsymbol{\ell}(p) = (\ell_x(p), \ell_y(p))$, the assumptions mean that the output probability has a form $P(p'|p, f, f') = F(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f')$, and it satisfies $F(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f') = F(\ell_x(p) - \ell_x(p'), \ell_y(p) - \ell_y(p'); f', f)$. A model for each hand can be obtained in this way, and is written as $F_\eta(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f')$ where η shows left and right hand respectively, and each model exists independent. It is further assumed that these probabilities are related by reflection in the x-direction, which yields $F_L(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f') = F_R(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f')$. The present model can be extended to be applied to pas-

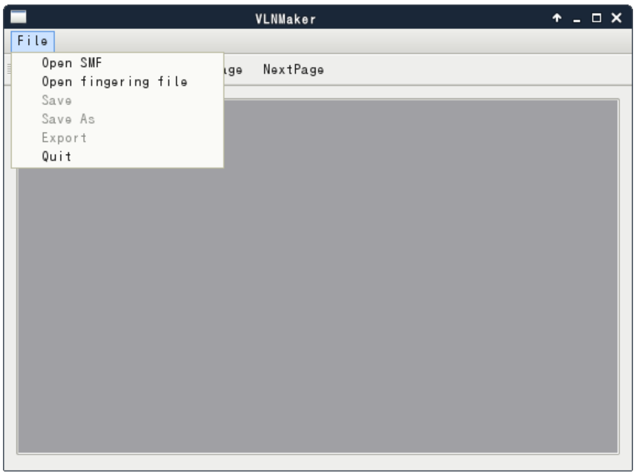


Figure 14. Drop-down menu of "File" on VLNMaker.

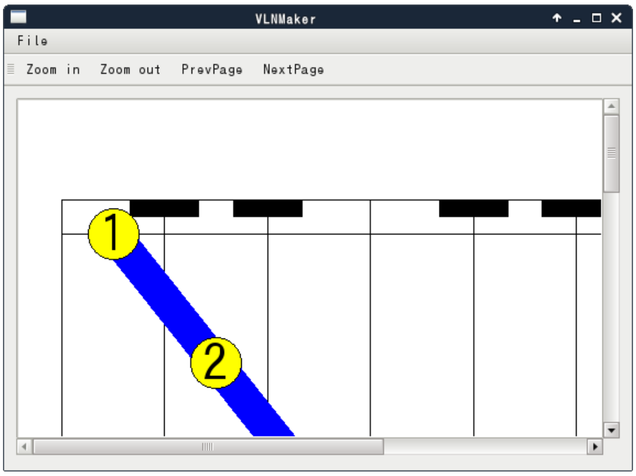


Figure 15. Window state after loaded SMF and piano fingering file.

sages with chords, by converting a polyphonic passage to a monophonic passage by virtually arpeggiating the chords [9]. Here, notes in a chord are ordered from low pitch to high pitch.

To find the optimal fingering for a given piano score using the model, we need to maximize the probability $P(f_{1:N}|p_{1:N})$. This can be computationally efficiently solved with a dynamic programming called Viterbi algorithm [12]. An extension of the above model for both hands is discussed in Ref. [13]. With this extension, it is possible to simultaneously estimate the optimal fingering for both hands from a MIDI file in which notes for the left- and right-hand parts are not separated.

The parameter values for the transition and output probabilities have been obtained in a previous study [13]. Because the used dataset consisted of fingerings for experienced piano players, the obtained fingerings were not fully appropriate for the beginners. For example, using different fingers for pressing successive identical keys is common for the experienced players, but it is not easy to perform for the beginners. To solve this problem, we adapted the trained model parameters to increase the probabilities of self transition for using identical fingers for successive notes.

6. IMPLEMENTATION

We developed a conversion software from SMFs to VLN scores, and we named it "VLNMaker". We use "Qt" which is a cross-platform programming environment for graphical user interface (GUI). Qt works on Windows, Mac OS X, and Linux. Moreover, it can make an executable file for each OS by using C++ compiler respectively from same source code.

Figure 14 shows drop-down menu of "File" on VLNMaker: "Open score file" to load SMF, and "Open fingering file" to read piano fingering information, and "export" to output PDF file.[*5] After loading SMF and piano fingering information, user can choose "zoom in" and "zoom out" for changing scale of view, and "PrevPage" and "NextPage" for moving previous and next page.

[*5] "Save" and "Save as" have not implemented yet. These menus will be used to save the modified piano fingering by hand. The interactive hand modification function of piano fingering also has not implemented yet.

Subject No.	EPI	reading staff N.	reading VLN	play of the piano	play by using VLN
1	none	C	A	C	A
2	none	C	A	C	A
3	none	C	A	C	B
4	none	C	A	C	B
5	Tb	C	B	C	B
6	Drums	C	B	C	B
7	Gt	B	A	C	B
8	E.Gt	B	A	C	A
9	Cla	B	A	B	A
10	Gt	B	B	B	B
11	Euph	B	B	B	A
12	Piano	A	A	B	B
13	A.Sax	A	B	C	B

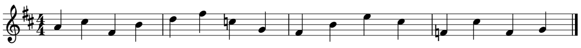
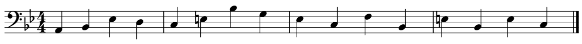
Table 3. An inquiry result of beginners' piano class at Nat. Inst. Tech., Kisarazu College (A: excellent, B: good, C: poor). EPI means experience of playing instrument.

7. EXPERIMENTAL RESULTS

7.1 Inquiry of beginners' piano class at NITKC

Table 3 shows a result of inquiry of beginners' piano class at NITKC (see Section 3.3). 13 students who were interested in playing the piano volunteered for this class which took 90 minutes.

All the subjects could read VLN scores easier than or equal to staff notation scores except one of them. The subject 13 is used to reading staff notation scores and usually plays alto saxophone, and this is the first time seeing VLN scores. Therefore, it seems that prior knowledge of traditional staff notation caused interference in reading VLN scores for this subject. Although most of the subjects have

Test 1

Test 2

Test 3

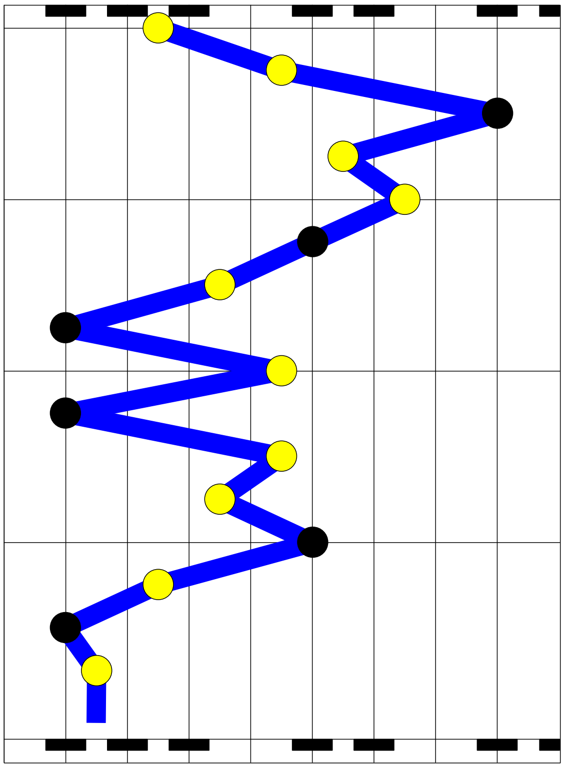
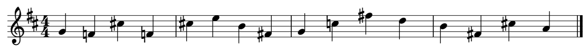
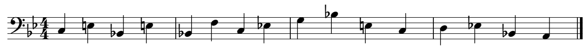
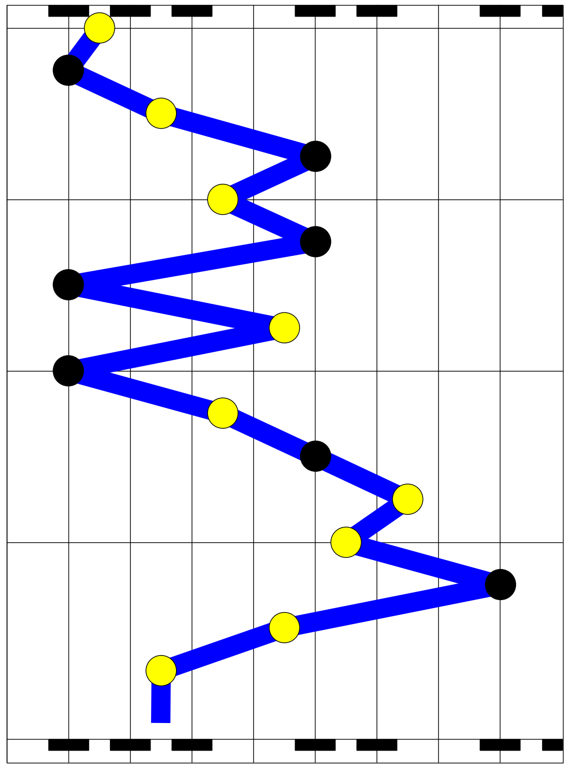


Figure 16. Scores used reading test.

Test 4

Test 5

Test 6



Test 7

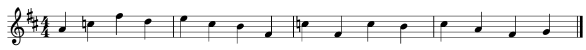



Figure 17. Scores used reading test (continued).

not took any regular piano lesson, all the subjects could play the piano well in the class.

These results suggest that VLN scores provide a greater chance of success when introducing the piano to beginners.

7.2 Comparison of performance with VLN scores and staff notation scores

In order to examine the readability of VLN scores, we compared with piano performances between in case of using VLN scores and using staff notation scores. We used example scores as shown in Figure 16 and Figure 17. The score of Test 3 and Test 6 does not contain finger number because we expected that subject just concentrates on pressing correct key. Therefore all the subjects pushed the key only with index finger. In the score of Test 4 and Test 5, each note has an accidental respectively. Note that the score of Test 3 and Test 7 are the same music piece. Subjects were NITKC students. Among them, each member of group A was not used to read staff notation score, and group B was used to. We taught them the pitch of first note of each score and instructed them to continue playing the piano even if the subject noticed mistake by oneself. The playing tempo was free.

Table 4 shows the result of the number of miss-key hitting. Most subjects A_i made mistakes in the case of using staff notation score. On the other hand, subjects B_i played completely without errors. All subjects performed without or with a few mistakes in the case of using VLN scores. Especially, although the score of Test 3 and Test 7 are the same music piece, each number of mistakes for Test 3 is less than that for Test 7 for subjects A_i.

Table 5 shows the results of playing duration. The durations in the case of using VLN scores are less or equal to the duration in the case of using staff notation scores for most subjects A_i. Although subject A_{10} finished Test 7 faster than Test 3, the number of mistakes in Test 7 is larger than in Test 3. For subject B_i, in contrast, the duration in the case of using VLN score is greater or equal than using staff notation score.

From the above results, we consider that the readability of VLN scores is high for music beginner players. However, it is not necessarily the case for people who are accustomed to read staff notation score. The most important thing is to be able to play the piano for music beginner, namely first consideration is that subject experiences success performances. Therefore, at first, piano beginners may not recognize pitch name clearly, and they only have to under-

Subject No.	Scores / Number of miss-key hitting						
	1	2	3	4	5	6	7
	G	F	VLN	G	F	VLN	G
A_1	8	2	0	6	5	3	1
A_2	6	6	0	11	6	0	9
A_3	9	8	0	7	3	1	8
A_4	10	9	0	13	8	0	8
A_5	8	8	0	9	3	0	8
A_6	6	6	0	1	0	0	6
A_7	8	8	0	3	3	0	8
A_8	6	8	0	9	7	0	6
A_9	6	0	0	1	1	0	6
A_{10}	1	0	0	1	0	0	7
B_1	0	0	0	0	0	0	0
B_2	0	0	0	0	0	0	0

Table 4. The number of miss-key hitting. Subjects A_i are not used to read staff notation score, subjects B_i are used to. G and F mean staff notation score in G clef and F clef, and VLN signifies VLN score. Note that score 3 and score 7 are same music piece.

Subject No.	Scores / Duration [sec]						
	1	2	3	4	5	6	7
	G	F	VLN	G	F	VLN	G
A_1	140	96	40	144	68	52	64
A_2	88	76	68	80	76	44	84
A_3	72	68	48	72	68	32	52
A_4	140	88	88	116	64	76	88
A_5	176	76	44	108	80	36	72
A_6	64	80	48	52	60	52	60
A_7	64	52	36	80	76	40	88
A_8	64	64	48	60	52	36	48
A_9	40	48	32	64	52	28	52
A_{10}	84	84	72	82	60	52	40
B_1	40	52	44	44	40	44	32
B_2	20	16	28	20	20	28	20

Table 5. Playing duration. Subjects A_i are not used to read staff notation score, subjects B_i are used to. G and F mean staff notation score in G clef and F clef, and VLN signifies VLN score. Note that score 3 and score 7 are same music piece.

stand the position of the key which will be pressed. Of course, understanding pitch name is also important, thus we will discuss in Section 8.2 and Section 8.3.

7.3 Production of VLN scores for beginner players and advanced players, and their influence on performances

Figure 18 shows an example of VLN score: the beginning of "March" from "The Nutcracker Suite op.71a" by Peter Ilyich Tchaikovsky. Finger numbers in Figure 19 are for beginners, and the VLN score is produced by VLNMaker. Finger numbers in Figure 20 are for advanced player, and the numbers are given by a professional piano player.[*6]

We examined differences of ease of playing in the case of using VLN scores for beginners versus advanced players. The subjects were students who belonged to NITKC, and their age was from 19 to 20 years old. First, we instructed how to read VLN score to subjects. Second, we asked them to play the song "tulip" to be used to read a VLN score. In this case, if the subjects made mistakes in their piano playing such as missed-key strokes and/or wrong fingering, we ignored those errors. Third, they played the only 2 bars of the beginning of "March" twice by changing the fingering. In this case, we asked them to repeat their practicing of the piece until they could play completely without errors.

Table 6 shows results of inquiry about piano playing using VLN scores. None of the subjects had previously practiced playing the piano in regular lesson. The results show that they could play easily when they used a VLN score for beginners versus one designed for an advanced player.

[*6] VLNMaker can also load a text file which contains piano fingering information which is made by hand. In this case, we prioritized ease of understanding, hence we omitted rests in the VLN scores.

8. DISCUSSION

8.1 Score arrangement

We consider that it is necessary to establish an arrangement method. Although the vertical line notation can show music faithfully, it is difficult to play complex chords and phrases, a sequence of very short notes, or a large jump of pitch for beginners. We will solve these problems by reducing the number of notes, simplifying rhythms, and adjusting pitches to match player's skill while keeping the atmosphere of the original music as much as possible.

8.2 Expansion of VLN

We plan to expand the vertical line notation. A characteristic of the vertical line notation is simplicity compared with staff notation. On the other hand, we are considering giving it a more functional representation. Figure 21 and Figure 22 show examples of musical scores ("For Elise" by Ludwig van Beethoven) in expanded vertical line notation. Both figures contain the left hand part, which is arranged, and repeat marks shown as a green line and dots on the upper portion in Figure 21 and on the lower portion in Figure 22 respectively. Moreover, these VLN scores contain large jump of pitch relatively, in these part, same finger does not always use. Another examples are shown in Figure 23 and Figure 24. These figures include chords which is shown as connected two nodes with a horizontal broken line respectively, and left hand part as orange nodes. Our developed software will be able to deal with information of these kinds.

Furthermore, there is room for improvement for the use of color. For example, the background of the keys of C and F are expressed using pale red and green.

In addition, it is important to express a length of each note. Currently, the vertical line notation shows the length of the note by the vertical distance between nodes. Al-

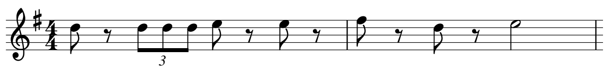


Figure 18. Staff notation score at the beginning of "March" by Tchaikovsky.

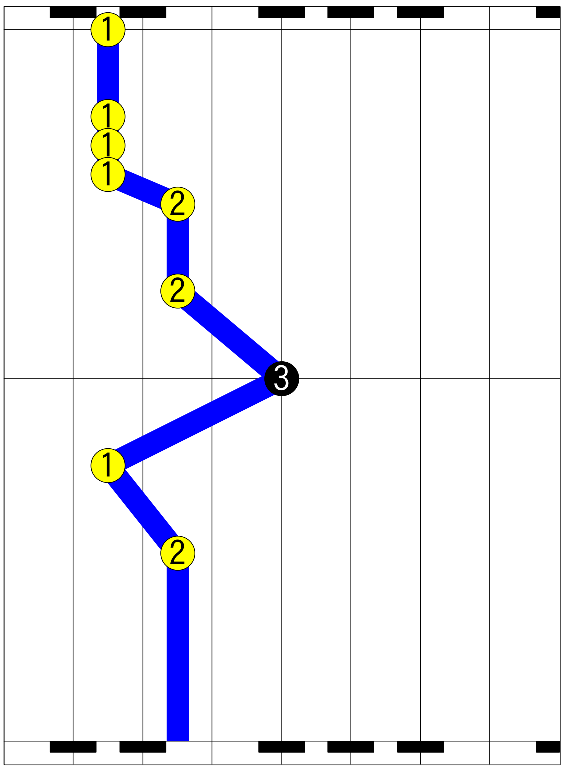


Figure 19. A converted VLN score from SMF for beginners of "March" from "The Nutcracker Suite" by Tchaikovsky.

though it is simple, the player may not understand the length of the note immediately. Therefore, for example, we are considering showing each node by a rectangle instead of a circle, and the height of the rectangle would indicate the length of the note. Furthermore, it may be useful to use different colors for various note lengths.

However we should pay attention to complexity of notation. Although complex notation scores have various and detailed information of music, such scores cause confusion for beginner player. Namely, readability and complexity are a trade-off relationship.

8.3 Transition from VLN to staff notation

At senior keyboard class in Showa University of Music, some of the early joined members tried to play the piano while putting VLN scores in a landscape orientation. It shows as same as staff notation: vertical direction represents the pitch of notes and horizontal direction expresses time flow.

Currently, most of the members proceed to using staff notation scores which are chosen as liking music pieces

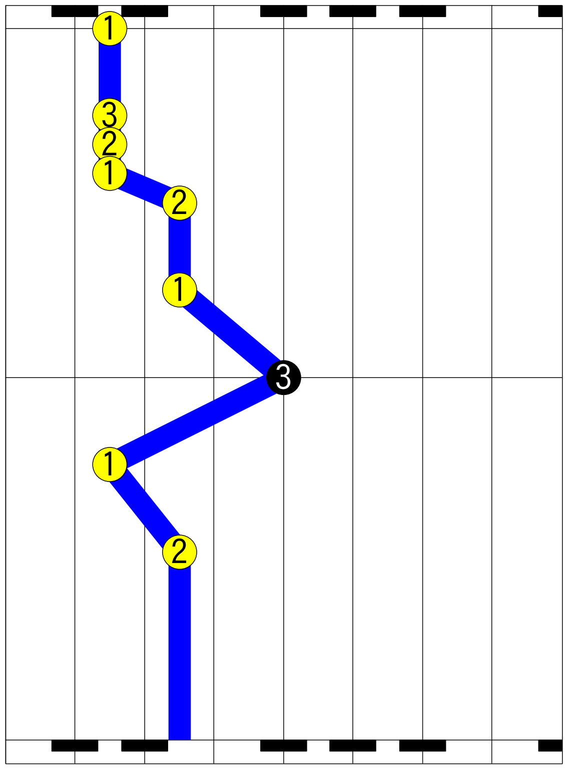


Figure 20. A converted result of VLN score of "March" for advanced players.

subject No.	reading VLN	playing by using VLN		
		T	$\mathrm{M_{bgn}}$	$\mathrm{M_{adv}}$
1	B	B	B	C
2	B	A	A	B
3	B	A	B	B
4	B	B	A	B
5	B	B	B	C
6	B	B	C	C
7	B	B	B	C
8	A	A	A	C
9	A	A	A	C
10	B	B	A	C

Table 6. Results of inquiry about piano playing using VLN scores (A: excellent, B: good, C: poor) and situation of play. Each T, $\mathrm{M_{bgn}}$ and $\mathrm{M_{adv}}$ shows use of a VLN score of the song "Tulip," "March" for beginners and "March" for advanced players. All the subjects had not formerly practiced playing the piano in regular lesson.

themselves. In fact, many piano score books in staff notation are sold, and this is one of advantages of staff notation. It seems that the members are used to play and have confidence in playing the piano gradually, thus they are going to be more aggressive. This case is a success instance of introduction of playing the piano, and is also "graduation" of VLN scores.

We will try to construct a more systematic learning method for transition from VLN score to staff notation

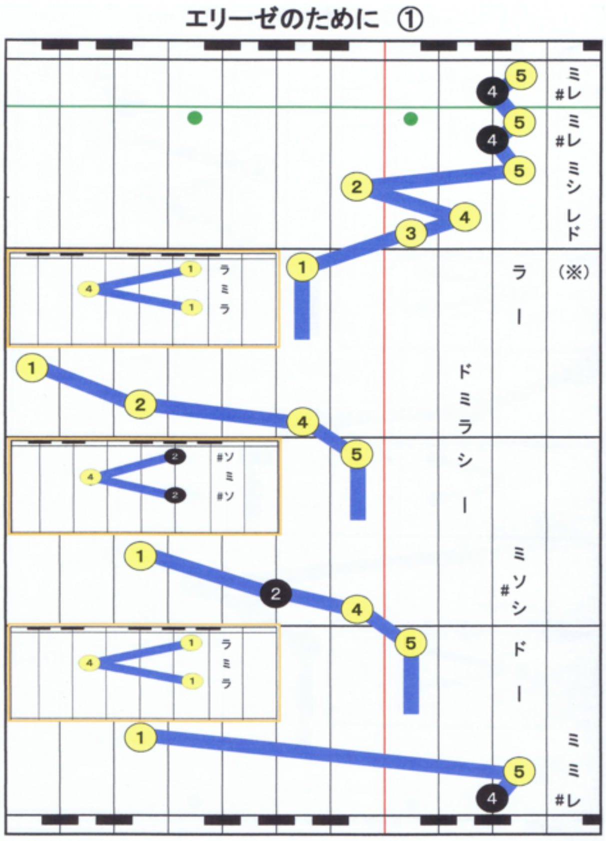


Figure 21. VLN score of the first page of "For Elise" by Beethoven (handmade).

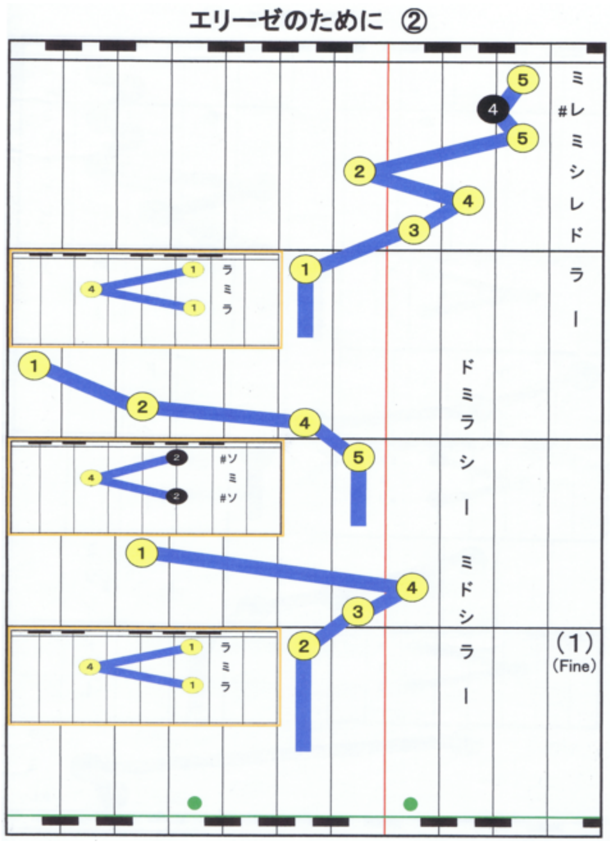


Figure 22. VLN score of the second page of "For Elise" (handmade).

score: it is necessary to learn the various information in staff notation such as pitch names, rule of accidentals, note lengths, expression marks, etc.

8.4 Fingering check system

We plan to construct a fingering check system which obtains images of the moving state of a piano player's hand and fingers by using a movie camera, specifically, an RGB-D camera such as Microsoft Kinect sensor [14], ASUS Xtion Pro Live [15], Leap Motion [16, 17], etc. Using these device, we can implement non-contact type system. It is important to use "natural input" for constructing user friendly system which does not ideally bring any stress for users. The fingering check system can provide a self-learning material for beginner player, and also automatically estimate the playing skill level of a piano player by analyzing the images. Hence, an automatic conversion system for VLN scores from SMFs will be improved to provide VLN scores with appropriate difficulty of piano fingerings which would correspond to the player's piano skill.

8.5 The spread of VLN

We consider spreading VLN score widely. Now, regular piano lessons with VLN scores have held in limited places yet such as Showa Music University, Heisei Music University, Ooizumi Elementary School, etc. One of the reason is that VLN scores are made by hand. With regard to this problem we believe that our study will solve it. Another one is that practice method is developing. Because the number of piano teacher with VLN score is not enough, it is necessary to teach the learning method using VLN score to teachers rather than students in advance.

In the near future, we will apply VLN scores not only for music education but also rehabilitation, music therapy, etc.

9. CONCLUSIONS

This paper described a conversion method from standard MIDI files to the "vertical line notation" (VLN), and an algorithm of automatic decision of piano fingering for piano beginners. Because VLN scores have been made by hand on Microsoft Excel currently, we developed an automatic conversion system from standard MIDI files to VLN scores. Moreover, we proposed an automatic decision of piano fingering for beginners. We confirmed that fingering for beginners is easier to play than fingering designed for the advanced players.

As future work, we plan to solve several problems described in section 8, and improve both VLN itself, and the automatic conversion system of VLN scores from SMFs.

Acknowledgments

This study is supported in part by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science, No. 26240025, 15K16054, 16K00501.

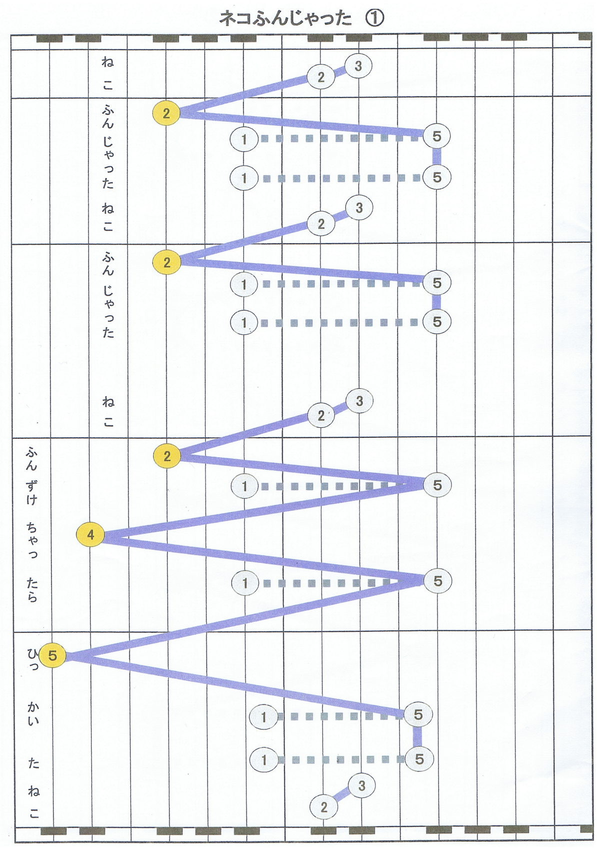


Figure 23. VLN score of first page of “Flea Waltz” (handmade).

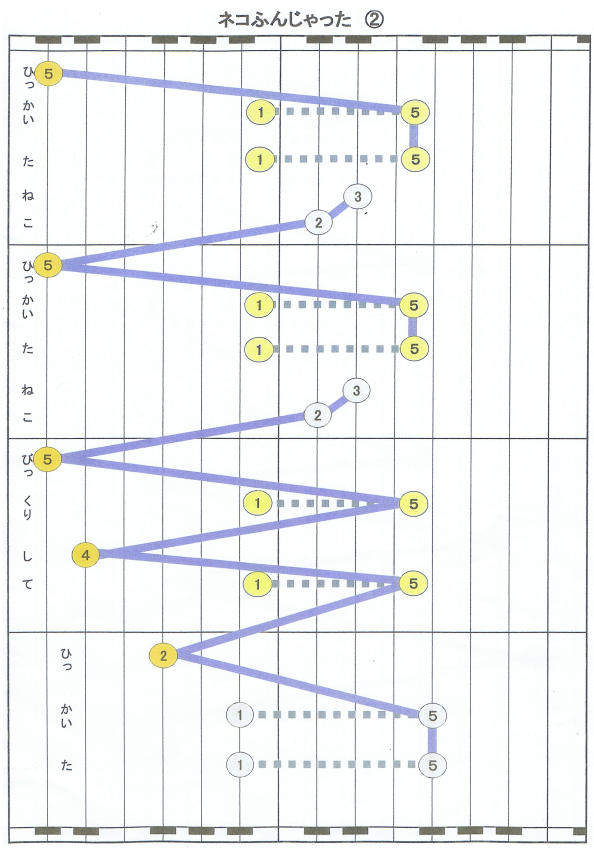


Figure 24. VLN score of second page of “Flea Walz” (handmade).

10. REFERENCES

[1] Musical notation, `http://en.wikipedia.org/wiki/Musical_notation`

[2] S. Agata, “YAMAHA portable keyboard self tutor for people who have never played a keyboard before,” Yamaha Europa G.m.b.H, 1985.

[3] C.M.Bishop, “Pattern Recognition and Machine Learning,” Springer, pp. 359–418, 2010.

[4] R. Sato *et al.*, “Automatic generation of musical score in vertical line notation from MIDI file,” *Proc. ICAI*, T105-04, pp. 559–562, 2015.

[5] MIDI, `http://en.wikipedia.org/wiki/MIDI`

[6] G. Mazzola, “The Topos of Music,” Chapter VIII, Birkhuser, 2002.

[7] R. Parncutt *et al.*, “An ergonomic model of keyboard fingering for melodic fragments,” *Music Perception*, **14(4)**, pp. 341–382, 1997.

[8] M. Hart and E. Tsai, “Finding optimal piano fingerings,” *The UMAP Journal*, **21(1)**, pp. 167–177, 2000.

[9] A. Al Kasimi *et al.*, “A simple algorithm for automatic generation of polyphonic piano fingerings,” *Proc. ISMIR*, pp. 355–356, 2007.

[10] Y. Yonebayashi *et al.*, “Automatic decision of piano fingering based on hidden Markov models,” *Proc. IJCAI*, pp. 2915–2921, 2007.

[11] M. Balliauw *et al.*, “Generating fingerings for polyphonic piano music with a tabu search algorithm,” *Proc. MCM*, pp. 149–160, 2015

[12] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, **77(2)**, pp. 257–286, 1989.

[13] E. Nakamura *et al.*, “Merged-output HMM for piano fingering of both hands,” *Proc. ISMIR*, pp. 531–536, 2014.

[14] Kinect sensor, `http://www.xbox.com/en-US/en/kinect/`

[15] Xtion Pro Live, `https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/`

[16] Leap Motion, `https://www.leapmotion.com/?lang=en`

[17] “Playing the Grand Piano with Leap Motion Gestures and BigBang”, `https://www.youtube.com/watch?v=ytGcKfhzF2Q`

TAXONOMY AND NOTATION OF SPATIALIZATION

Emile Ellberger
Institute for Computer Music and Sound Technology
Zurich University of the Arts
emile.ellberger@zhdk.ch

Germán Toro Pérez
Institute for Computer Music and Sound Technology
Zurich University of the Arts
german.toro-perez@zhdk.ch

Linda Cavaliero
Institute for Computer Music and Sound Technology
Zurich University of the Arts
info@dustandwater.com

Johannes Schütt
Institute for Computer Music and Sound Technology
Zurich University of the Arts
johannes.schuett@zhdk.ch

Giorgio Zoia
Institute for Computer Music and Sound Technology
Zurich University of the Arts
giorgio.zoia@bluewin.ch

Basile Zimmermann
Faculty of Humanities
University of Geneva
basile.zimmermann@unige.ch

ABSTRACT

The SSMN Spatial Taxonomy and its symbols libraries, which are the corner stone of the Spatialization Symbolic Music Notation (SSMN) project, emanates from research into composers' attitudes in this domain. It was conceived as the basis for the development of dedicated notation and rendering tools within the SSMN project.

The taxonomy is a systematic representation of all relevant features necessary to specify sound spatiality: shape and acoustic quality of the space, structure, position and movement of sound sources. It is based on single descriptors that can be combined in order to define complex spatial configurations. Descriptors can be transformed locally and globally and can be the object of structural and behavioral operations. The SSMN Spatial Taxonomy proposes a corresponding graphic symbolic representation of descriptors, operations and other functional elements facilitating the communication of creative ideas to performers and technical assistants.

This paper focuses on the presentation of the taxonomy and the symbols. Additionally it describes the workflow proposed for using symbols inside a notation software prototype developed within the project. Finally, further aspects concerning the actual and future developments of SSMN are mentioned.

1. INTRODUCTION

The field of sound representation has undergone continual development throughout the history of creative arts. The issue of sound motion representation, as concerns composers, has however hardly been studied. Composers have been continually obliged to reinvent strategies for communicating their ideas of spatial movement of sound, performers' displacement, and description of the performance space. In fact, even today's musical software tools that include implementation of spatialization have been limited in their capacity to propose adequate notation possibilities to convey spatial information within musical scores. In spite of the availability of a variety of strategies and tools for spatialization within the context of electroacoustic music composition, decisions about position and movement of sound in space, or the general question of virtual space quality remain often a secondary formal issue; in many cases these decisions are left to a post-production stage instead of being fully integrated throughout the composition process. This situation can marginalize spatialization to an ornamental aspect that can be adapted or reduced without affecting musical substance.

On the other hand, performers engaged in the interpretation of music involving electroacoustic spatialization (and other kinds of signal processing) find mostly a reduced graphic representation of the ongoing processes in the score [1]. According to the experience of the authors during several years of performance practice the notation of electroacoustic events prioritize mostly cue numbers and synchronization events. This limits the possibility of a more intimate interaction within the performance situation. In addition, the lack of a spatialized acoustic feedback while studying prevents performers from preparing a piece taking into account sound motion. This issue becomes especially relevant when considering the usual restrictions of rehearsal time in performance spaces.

The need for a graphical representation of spatialization within the context of sound diffusion of electroacoustic music in concert has been also addressed with arguments pro [2] and contra [3]. Nevertheless a generic and practical way to accurately notate spatialization has not been formulated yet. Even meticulous spatial notation as in Stockhausen's *Oktophonie* [4] using sequences of channel numbers instead of symbols –as in the introductory notes to the score– is difficult to read for performers.

Finally, when audio engineers collaborate with composers preparing compositions within a multi-channel environment, they have to overcome the difficulties of interpreting placement of sound in space as imagined by the composers, who typically invent a personal system of graphical or textual explanations.

The aim of SSNM is thus to open a new approach of substantial integration of spatial relationships and spatial processes in musical thinking as well as in composition, rehearsal and performance practice. For this purpose SSMN has defined a typology of spatial movements and designed a library of symbols to represent them. In order to enable its use in creative processes, an open source software tool that integrates this library within a common western musical notation context is being developed, allowing editing and acoustic feedback through a rendering engine. Composers are thus able to use and edit symbols describing spatialization in a notation program and immediately hear the results. Performers are given full information on spatialization in the score and can hear the results from the beginning of the studying process.

2. SCOPE OF RESEARCH

During the preparatory stages of the SSMN Spatial Taxonomy, research has focused on the following:

- Musical scores containing verbal or graphical annotations of spatial indications, focusing on spatialization and extended notation in contemporary music since 1950 [5]
- Artistic performance practice wherein spatial placement and/or mobility of live performers is relevant to musical compositions as well as composers' means of expressing placement and/or motion in space [6], [7]
- Spatialization in electroacoustic media [8]
- Extended musical notation [9]
- 2D and 3D symbolic notation [1]
- Typologies of spatial qualities of sound [10]
- Spatial perception of sound placement, sound motion and physical space [11]
- Semiotics and epistemology of notation [12]
- Relevant programming languages, audio and graphic design software including Csound, PD, Iannix, SuperCollider, Max/MSP, Illustrator, WFS [13] [14], Ambisonics, IRCAM's OpenMusic [15] & Antescofo [16], *inScore* [17]
- Spatial notation in other fields, especially dance, aeronautics, geographical migration patterns, and theater staging [18]

In a nutshell, the specifications for the SSMN research project are based on a four-pronged study: (a) spatial typo-morphology resulting in the SSMN Spatial Taxonomy, (b) design of symbols, (c) integration of symbols and trajectory editing in notation software, (d) integration of notation software in a rendering engine. So far, an introduction to the project was first presented at ISMC|SMC2014 [19] followed by a poster presentation at TENOR 2015 [20].

Concerning the specific issue of a spatial taxonomy the contributions referred to above present important ideas but are limited in the sense that they were developed in view of specific aspects and purposes different from those of notation. Trochimczyk's [10] classification of spatial designs comes closer to our needs but is consciously limited to certain instrumental setups; Smalley focuses mainly on spectral structure [21] or develops a perceptual approach to the analysis of acoustic scenes [22]; Vandergorne's spatial categories and figures are specifically concerned with sound diffusion [8]; UST (Unités Sémiotiques Temporelles) [23] are obviously focused on temporal meaning. In our opinion, a taxonomy for notation of spatialization should ideally be universal, generic and based on low-level structural features that can be represented through symbols. The terminology should emanate directly from musical practice and be as much as possible self-explaining. The work presented in the next section was developed under these premises.

3. TAXONOMY

3.1 Preliminary considerations

The basic units of the SSMN spatial taxonomy are called *descriptors*. There are two kinds of descriptors: room descriptors and descriptors of sound sources. Sound sources can be physical root sound (RS) like instruments and voices or projected audio signals (PA) like microphone signals, audio files and streamed audio.

Descriptors can be simple or compound and are assumed to be perceptually relevant, but definitive perception depends on the interaction between the actual sound and the actual spatial configuration. Although descriptors are primarily defined in structural (geometrical, mathematical, acoustical) terms, they have been conceived in view of musical practice.

Simple descriptors are the basic atoms of the SSMN spatial taxonomy. They are able to denote all single primary features relevant to sound spatiality and can be represented as symbols. Compound descriptors are arrays of simple descriptors. They are used to represent more complex spatial configurations and processes (e.g. patterns, figures, motives, etc.) and can also be represented as symbols.

Descriptors can have several properties that are finally defined through names, numeric parameters and flags. For instance, the descriptor "Position of loudspeakers" has the entry "labels" to name specific items, the parameters "position" given as Cartesian or spherical coordinates and "inclination" (yaw) defined as angle and the flag "interior" or "exterior" according to their position inside or outside the room. For reasons of clarity, parameter units as well as some parameters and flags will be omitted in this paper.

The third part of the taxonomy is dedicated to *operations,* also called *modifiers*. They can be used to transform elements previously defined using single or compound descriptors or to generate new elements. For instance, the basic structural operation "Scaling" can be used to multiply a given parameter or group of parameters by a certain factor, "Repetition" to repeat a compound trajectory previously made out of single segments as straight lines and curves. *Global operations* can be used to generate relationships between complex unities like sequences and superpositions of existing compound trajectories. *Cross-domain interactions* can be used to rule relationships between spatial audio information and other media like synchronization with visual or choreographic sequences.

Finally, *behavioral relationships* like "co-incidence" or "attraction" inspired by social and biological movement patterns and observed in other contexts (see 3.3) could help to envision a new paradigm of sound spatiality based on processes rather than geometrical or visual structures. This aspect is not fully integrated in the taxonomy yet and suggests a promising research direction.

As mentioned above, the SSMN spatial taxonomy is intended to become universal and generic. At the moment not all descriptors have been defined as symbols and not all symbols have been implemented within the software prototype.

Although the taxonomy describes and classifies sound in a three-dimensional space, some objects and symbols are, for practical reasons (mainly rendering, and dependence on existing standardized formats), represented in two dimensions.

All projected audio can be subjected to speed, acceleration and the Doppler effect. Simple trajectories can be followed in two opposite directions.

After considering the wide number of possible curve types only a small number of them was explicitly addressed in the taxonomy. A detailed evaluation of their perceptual relevance remains to be done.

While sound sources can easily be defined as "points" or "groups", a concept such as "sound plane" is an abstraction of visual forms often used by composers but difficult to define in purely acoustical terms. We have nevertheless integrated it into the taxonomy.

The following section presents the complete work as an almost self-explaining, structured list of descriptors and operations. Explanatory comments are provided in brackets. Behavioral relationships will be discussed separately.

3.2 Descriptors and operations

I. Room descriptors

A. Disposition

1. Shape of the room (generic shapes)
   - a. Cube
     length, width, height
   - b. Hemisphere
     diameter, height
   - c. Church (cross form)
     length, width, height
   - d. Other shapes
     dimensions
2. Placement of performers, objects and audience
   - a. Fixed
     label, position
   - b. Variable
     - i. Line
       start/end, speed
     - ii. Arc
       start/end, curvature, speed
     - iii. Other (e.g. random, choreography)
3. Position of Microphones
   - a. Referred to an instrument
     name of instrument
   - b. Referred to the space
     label, *position*, *inclination*
   - c. Referred to specific movements (e.g. swinging microphone)
4. Position of loudspeakers
   - a. Fixed
     label, position, direction, inclination
   - b. Variable (mechanic or human driven)
     - i. Line
       start/end, speed
     - ii. Arc
       start/end, curvature, speed
     - iii. Pendular motion
       length, initial height, direction
     - iv. Other (e.g. choreography)

B. Spatial quality of the room

1. Space definition
   - a. Open
   - b. Closed
   - c. Virtual
2. Reverberation
   - a. Interaction source-room
     energy of first reflections related to direct sound, energy of late reverberation, decay time of primary reflections
   - b. Room perception (related to late reflections):
     decay time, heaviness (change of decay time of low frequencies), *"liveness"* (change in decay time of high frequencies)

II. Descriptors of sound sources

A. Types of sound sources

1. Sound points
   - a. Physical root sound (RS)
     label
   - b. Projected audio signal (PA)
     label
2. Groups
   => Definition: A group is a collection of sound points with common spatial features. A group is defined by a perimeter wherein the single elements can be found. Position and movement of single elements within the perimeter can be defined in the same way as single points.
   - a. Root sound
     label, number of sources, position of reference point
   - b. Projected audio
     label, number of sources, position of reference point

3. Planes (PA)
=> Definition: a plane is a homogenous sound spread out in space.
label, *shape*

B. Spatial quality of single sources

1. Perceived distance (PA)
presence, brilliance, warmth (equalization)

C. Dimension of single sources

2. Scale
=> PA, perception of «bigger or smaller» than real sound source)
scale factor

D. Localization of sound sources

1. Localization of single points (PA, RS)
position, direction, inclination, aperture (PA)

2. Localization of groups
shape, *geometrical center, position of each element, within the shape*

3. Localization of planes
position, direction, inclination

E. Simple trajectories of sound points, groups or planes

1. Linear

a. Straight
start/end

b. Polyline (open)
segments, start/end

c. Poly_closed (closed polyline)
segments, start/end

2. Circular

a. Circle
center point, radius, start/end angle, direction

b. Slinky (named after the toy invented 1945 by Richard James)
start/end center point, radius, start/end angle, direction

c. Spiral
start/end center point, start/end radius, start/end angle, number of rotations

3. Curve

a. Bézier
start/end, control points, reiterations

b. Bézier_spline
start/end, control points, reiterations

c. Béziergon (closed Bézier curve)
start/end, control points, reiterations

d. Bernoulli (lemniscate)
start/end, control points, reiterations

e. Other (e.g. Lissajoux, etc.)

F. Compound trajectories

1. Compound using simple trajectories

2. Generic polygons (selection of basic shapes)

3. Free hand using interface

III. Operations (transformation or generation of new trajectories from preexistent single or compound trajectories)

A. Structural operations and modifiers

1. Operations on single sound sources, groups and planes (position); on simple or compound trajectories

a. Repetition
number of reiterations

b. Scaling
factor

c. Shift
value

d. Rotation
roll, yaw, pitch

e. Mirror (inversion)
mirror flag

f. Reverse (crab)
reverse flag

g. Palindrome (returns to the starting value)
palindrome flag

h. Random
random parameter

i. Signals as modifiers

i. Sinus
frequency, amplitude, phase

ii. Triangle
frequency, amplitude, phase

iii. Square
frequency, amplitude, phase

iv. Saw
frequency, amplitude, phase

v. Other

j. Simple or compound trajectories as modifiers
label

2. Operations on compound structures

a. Sequencing

b. Permutation

c. Interpolation (morphing)

3. Algorithmic defined functions based on externals

algo (label)

B. Global operations

1. Global scaling (space, time)

a. Linear

b. Non-linear

2. Sequence (Horizontal)

a. Loop

b. Cross

c. Tight

d. Pause

3. Superposition (Vertical)
    1. Synchronous start
    b. Delay
    c. Synchronous end

C. Cross-domain interaction
  1. Scaling (time)
  2. Synchronous start
  3. Delay
  4. Synchronous end

3.3 Further Taxonomy directions

Since the primary intention of the SSMN project is to provide a working prototype of a software package that can be tested by composers, each aspect of the taxonomy that has been addressed here undergoes verification by users. As indicated earlier, an open source score editor (MuseScore) has been targeted for graphic symbols implementation allowing real-time OSC messages to be transmitted to a rendering engine. The sound projection tool used for these experiments is an ambisonics spatialization system allowing the simulation of different multichannel projections in various formats as well as a binaural headphone version. The score editor is dubbed MuseScoreSSMN and sends all OSC spatialization information via a dedicated port to Max-based tools (e.g. the SSMN-Rendering-Engine) [19].

While the prototype is being prepared, tested and documented, further aspects that could be symbolized are being oriented towards questions of behavioral interactions between two or more sound sources affecting their spatial movement. A research project at the University of Zurich concerning data mining and visual analysis of movement patterns proposes a taxonomy of movement patterns [18] that can be investigated using sound sources and can be integrated into the spatial taxonomy. The following list of behavioral attributes and relationships make reference to this work (page numbers) and are presented here as a suggestion for further research:

A. Behavioral attributes
  1. Trend-setter: a sound source establishing movement patterns followed by other sources, p. 10
  2. Follower, p. 10
  3. Indifferent: autonomous (non-uniform) or random movement within a behavioral context. See also: «dispersion»: "non-uniform or random motion, opposite to concurrence", p. 8

B. Behavioral relationships
  1. Imitation: see also «concurrence»: "same values of motion attributes at a certain instant or duration", e.g. «synchrony», p. 7
  2. Coincidence: similar positions, full or lagged, p. 8
  3. Opposition: bi- or multi-polar arrangement, e.g. spatial splitting, p. 8
  4. Constancy: "movement patterns remain the same (...) for a particular duration", p. 8
  5. Convergence: synchronous or delayed, "movement to the same location". See «encounter», p.9
  6. Divergence, synchronous or delayed: movement away from the same location. See also «breakup», p. 9
  7. Attraction. See also: «pursuit», p. 10
  8. Repulsion. See also: «evasion», p. 10

4. SYMBOLS

4.1 Early SSMN Spatial Taxonomy and Symbolic representation research

Initial decisions about symbol design concerned the approach to symbolic representation. As the taxonomy was being developed a provisional set of symbols was defined based on ongoing comparative studies of 2-D and 3-D graphic representation of spatial motion. Additional pertinent authors were Trevor Wishart (1996) [24], Bijan Zelli (2001) [25], Larry Austin (2004) [26], Lasse Thoresen [27], Bertrand Merlier (2008) [28] and Vincent Verfaille (2003) [29]. An overall design concept was adopted with the primary criteria requiring clarity, legibility and rapid recognition through reliance on simple visual symbols such as cube, sphere, radar, perspective, arrows, colors, size, etc. (see figure 1).

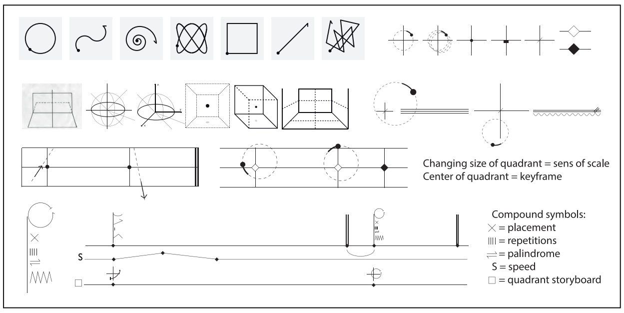


Figure 1. Example of early symbol design research.

This basic set was improved in subsequent design loops. The following major issues came up during the process:

- Defining "symbolic" as opposed to "descriptive" (i.e. icon versus image)
- Creating a grammar such as the creation of compound symbols (e.g. a circle with vibrato-type movement including acceleration) or determining a set of regrouped staves/tracks for which a common action is proposed)
- Determining parameters of SSMN symbols
- Establishing a timeline with key frames (e.g. a dedicated staff) allowing continuous activity of a symbol being reported on the timeline representation
- Pertinence of the use of a quadrant or grid to improve legibility (see figure 1, upper row, symbols 8-12)
- Creating tools for manuscript input to allow a degree of freedom for composers to deal with situations where the taxonomy would not provide the adequate tool for a specific idea (e.g. the utilization of a rubber-stamp for rapid manual input of composer's trajectory designs).

Several strategies of graphical possibilities had to be tested in view of integrating these symbols into the open source score editor MuseScore.

4.2 SSMN Symbol set

This process led to the actual symbol set consisting of the following categories:

- Physical performance space characteristics
- Initial physical placements of performers, microphones, loudspeakers and objects
- Position of sound sources (RS, PA)
- Trajectories / displacement of sound sources, microphones, loudspeakers and objects
- Operations
- Stop/End markers delimiting the time domain of symbols (see "Working with symbols" below)
- Inter-application communication resources (OSC, MIDI) for interaction with external programming environments

The last two are not explicitly contained in the taxonomy. They came up as a result of operational needs during the implementation stage.

Table 1 includes only symbols created according to the taxonomy. Some of them are already available within MuseScoreSSMN. Trajectories appear in two variations: single direction and back and forth.

Cube		Hemisphere	
Church		Other	
Performer		Perf_line	
Perf_arc		Music stand	
Audience		Microphone	
Loudspeaker		Swinging microph.	
Swinging loudsp.		Choreography	
Sound point RS		Sound point PA	
Group RS		Group PA	
Plane		Scale	
Straight		Polyline	
Poly_closed		Circle	
Slinky		Spiral	
Bézier		Bézier_spline	
Béziergon		Bernoulli	
Sinus		Triangle	
Square		Saw	
Random		Algo	

Table 1. Symbols designed according to the taxonomy

The symbols shown in Table 2 complement those referring directly to elements of the taxonomy. They specify further positions and movements of sources or address new elements and functionalities. The first two rows include additional types of movements of performers. The next two rows introduce stop markers for trajectories and modifiers as well as special markers for defining pauses within a trajectory without sound interruption. The next row presents symbols defining alternate movements of points and groups. The following two rows specify 3D positions of points and groups, the next one the position of planes. The symbols in the last row allow for the definition of inter-application communication and a dedicated SSMN staff respectively.

Perf_rotate		Perf_free	
Perf_to&from		Perf_other	
Trajectory_end		Modifier_end	
Pause_start		Pause_end	
Alternate_point		Alternate_group	
3D_point (RS)		3D_point (PA)	
3D_group (RS)		3D_group (PA)	
Root_plane back		Root_plane front	
Communication OSC		SSMN Staff	

Table 2. Additional symbols

4.3 Working with symbols

Figure 2 below illustrates the basic workflow within MuseScoreSSMN: (A) selection of a symbol from the "SSMN Palette"; (B) placement in the score; (C) definition of parameters in the "Inspector window" corresponding to the symbol chosen; (D, E) display of the trajectory or trajectories designed by the user in the interactive "Radar window". This window contains a top and a side view (E). Each circle corresponds to 10 spatial units to be scaled according to the real space.

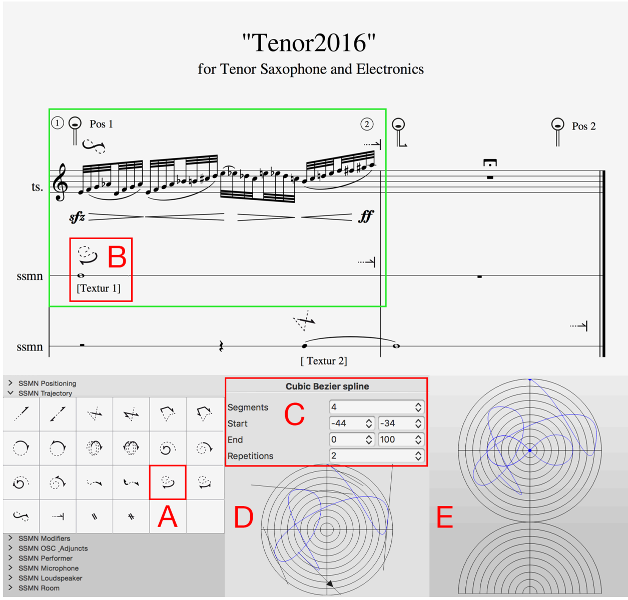


Figure 2. Workflow within MuseScoreSSMN.

In this example the phrase played by the tenor saxophone will be picked up by a microphone (projected audio) and spatialized according to the "Bernoulli" symbol (see Table 1) placed above the staff at the beginning of bar 1. The corresponding "Trajectory_end" symbol (see Table 2) at the end of the first bar marks the exact point in the timeline at which the trajectory ends, thus defining its effective duration. The initial position of the performer (root sound) is defined by the "Performer" symbol (see Table 1). After playing bar 1 the player is asked to move towards a new position defined by a similar symbol at the end of bar 2. The trajectory used is defined by the "Perf_line" symbol (see Table 2) at the beginning of bar 2.

Additionally, two dedicated SSMN staffs have been set to define the spatialization of pre-produced samples. The movement of the sample named "Textur 1" is defined by a "Bézier_spline" symbol (B, see also Table 1). The resulting trajectory is shown in the radar window (D). It contains 4 control points (marked with tangents) and will be repeated once. The corresponding parameters including start and end positions (x, y) are shown in the inspector window (C). The sample named "Textur 2" begins at the fourth beat of bar 1. It was defined as a polyline. Both samples have "Trajectory_end" symbols above the corresponding staff. An SSMN staff can be used independently of musical events and become a timeline for other kinds of information (e.g. choreography notation, film editing).

The green line indicates a selection including the saxophone staff and the upper SSMN staff. The radar window (E) shows the superposition of both trajectories. Here the ends of the trajectories are highlighted with a point. The position of the performer is not displayed in the Radar window since it is not relevant for rendering.

4.4 Current developments

A basic operative feature to be implemented in the near future is the possibility of saving movement patterns defined by descriptors and modifiers. Another issue is the question of symbol activity in the context of digital representation possibilities. On one hand, having a score in the digital domain allows for much greater latitude in providing continuous information through windowing, with or without animation. On the other hand it might be necessary to reduce the displayed information in the printed version of full scores and parts for reasons of clarity.

New possibilities appear when imagining interaction through integration of various software applications dedicated to facilitating artistic processes. A collaboration between the research teams of "inScore" and "Faust" at GRAME (Lyon) and SSMN has recently been undertaken with the expectation of creating tools to facilitate interaction on a local level and in web applications for visual display and audio rendering purposes. Other aspects being currently investigated are SpatDIF compatibility and the integration of SSMN Elements within the MusicXML protocol.

5. CONCLUSIONS

Results of the SSMN project have already been tested with composition students at the Zurich University of the Arts and presented at the Haute École de Musique of Geneva. This experience has revealed encouraging developments, such as increased awareness of spatialization possibilities within the composition process and augmented spatial listening acuity. The main intention of the project is to reflect on the ways we think of and work with spatiality in composition and to envision procedures that integrate spatiality from the very beginning. The software prototype is intended as a tool that facilitates the exploration of such procedures. Further tests and experiences should help to clarify if similar workflows can become practical and open enough to meet the necessities of different composers.

The taxonomy presented here reflects approaches to spatialization based mainly on geometrical and visual concepts such as lines, curves and planes. New organization paradigms can be envisioned by introducing time based dynamic movement patterns as observed in biological and social contexts. The persistent idea of sound as an object, closely related to visual and geometric concepts, could be challenged by an understanding of sound as a continuously changing field of energy, as the result of interacting information streams. Although the emergence of new notation paradigms will be supported by an evolving technology that already makes possible the integration of interactive interfaces in performance practice, it can be assumed that conceptual thinking in composition will remain the major source of aesthetic innovation of spatialization in electroacoustic music.

Acknowledgments

The SSMN research team is grateful for the assistance and support offered by the Swiss National Foundation for Scientific Research, the members of Institute for Computer Music and Sound Technology at the Zurich University of the Arts, the Computer Music division of the Haute École de Musique of Geneva, and the GRAME, Lyon.

6. REFERENCES

[1] K. Patton, “Morphological notation for interactive electroacoustic music”, *Organised Sound* 12(2), pp. 123-128, 2007.

[2] D. Besson, “La transcription des musiques électroacoustiques: que noter, comment et pourquoi?”, *Analyse musicale,* 3 trimestre, pp. 37-41, 1991.

[3] J. Chadabe, “Preserving Performances of Electronic Music”, *Journal of New Music Research*, 30:4, pp. 303-305, 2001.

[4] K. Stockhausen, “Oktophonie. Elektronische Musik aus DIENSTAG aus LICHT”, Stockhausen-Verlag, Kürten, 1994.

[5] M. Harley, “From Point to Sphere: Spatial Organization of Sound in Contemporary Music (after 1950)”, *Canadian University Music Review* 13, pp. 123-14, 1993.

[6] K. Stockhausen, “Helicopter Quartet”, Stockhausen-Verlag, Kürten, 1996.

[7] E. Bates, “The Composition and Performance of Spatial Music”, Trinity College, Dublin, 2009.

[8] A. Vandegorne, “L’espace comme cinquième paramètre musical”, in L. Pottier (Ed.), *La Spatialisation des musiques électroacoustiques*, pp. 53-80, Publications de l’Université de Saint-Étienne, Saint-Étienne, 2012.

[9] Ch. Dimpker, *Extended Notation: The Depiction of the Unconventional,* LIT Verlag, Münster, 2013.

[10] M. Trochimczyk, *Space and Spatialization in Contemporary Music: History and Analysis, Ideas and Implementations*, Moonrisepress, California. 2011.

[11] J. Kapralos, *Auditory Perception and Spatial (3D) Auditory Systems*, York University Press, Canada, 2003.

[12] J. J. Nattiez, *Music and Discourse: Toward a Semiology of Music,* Princeton University Press, New Jersey, 1980.

[13] G. Theile, “Wave Field Synthesis- A Promising Spatial Audio Rendering Concept”, in *Proceedings of 7th ICDAE* (DAF’04), Naples, 2004.

[14] J. Daniel, *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia.* Thèse de doctorat de l’Université, Paris 6., 2000.

[15] C. Agon, G. Assayag and J. Bresson (Eds), *The OM Composer's Book Vol. 1.* Collection Musique/Sciences, Editions Delatour France / IRCAM, 2006.

[16] A. Cont, J. L. Giavitto and F. Jacquemard, “From Authored to Produced Time in Computer-Musician Interactions”, CHI 2013 Workshop on Avec le Temps! Time, Tempo, and Turns in Human-Computer Interaction, Apr 2013, Paris, France. ACM, 2013. <hal-00787033>

[17] D. Fober, Y. Orlarey, S. Letz, “Inscore – An Environment for the Design of Live Music Scores”, in *Proceedings of the Linux Audio Conference 2012*, pp. 47-54, 2012.

[18] S. Dodge, et al “Towards a taxonomy of movement patterns”, in *Information Visualization*, 1-13 Palgrave/Macmillan Ltd. London, 2008.

[19] E. Ellberger, G. Toro Pérez, J. Schuett, et al, “Spatialization Symbolic Music Notation at ICST”, in *Proceedings ICMC|SMC|2014* Athens, pp. 1120-25, 2014.

[20] E. Ellberger, G. Toro Pérez, J. Schuett, et al (2015). “A Paradigm for Scoring Spatial Notation”, in *Proceedings: First International Conference on Technologies for Music Notation and Representation TENOR 2015*, Paris, pp. 99-102, 2015.

[21] D. Smalley, “Spectromorphology: explaining sound-shapes”, *Organised Sound* 2(2): pp. 107-126, Cambridge University Press, (1997).

[22] D. Smalley, “Space-form and the acoustic image”, *Organised Sound* 12(1): pp. 35-58, Cambridge University Press, (2007).

[23] J. Favori, “Les Unités Sémiotiques Temporelles”, *Mathematics and Social Sciences*- 45e année, n° 178, pp. 51-55, 2007.

[24] T. Wishart, *On Sonic Art*, ch. 10. Routledge, Taylor & Francis Group Ltd. Oxford, 1996.

[25] B. Zelli, *Reale und virtuelle Räume in der Computermusik: Theorien, Systeme, Analysen,* Techniche Universität Berlin, 2001.

[26] L. Austin, *John Cage’s Williams Mix (1951-3): The Restoration and New Realizations of and Variations on the First Octophonic, Surround-Sound Tape Composition.* Edited by Hall and Sallis. Cambridge University Press, 2004.

[27] L. Thoresen, and A. Hedman, “Spectromorphological Analysis of Sound Objects: an adaptation of Pierre Schaeffer's typomorphology”, *Organised Sound* 12 (2), pp. 129-141, Cambridge University Press, 2007.

[28] B. Merlier, *Vocabulaire de l’espace en musiques électroacoustiques,* Editions Delatour, France, 2008.

[29] V. Verfaille, “Effets audionumériques adaptatifs: théorie, mise en oeuvre et usage en création musicale numérique”, Doctoral Thesis, l’Université Aix-Marseille II, France, 2003.

MUSIC ANALYSIS THROUGH VISUALIZATION

Jia Li

narkli@ccrma.stanford.edu
CCRMA, Stanford University
Shanghai Conservatory of Music

ABSTRACT

In this paper analytic visualizations are used to selectively highlight salient musical features in four modern compositions, focusing on micro or macro structures: from motivic pitch contour to large-scale form. At a glance these visualizations allow a quick grasp of the structure and assist listeners to make connections between local features and global trends. Textures obscured by musical notation become more apparent when displayed in a graphical format, such as broad registral shifts, polyphonic streaming, as well as interplay between instruments. Pitch, timbre and voicing are plotted against time to show large-scale patterns that would otherwise be difficult to recognize in a musical score or compare between different works. Music analysis through compositional data visualization not only makes sense to musicians but also to non-musicians, facilitating collaboration and exchange with artists and technicians in other media.

1. INTRODUCTION

At the turn of the nineteenth century, technological change occurred rapidly. Many inventions such as photography, cinematography, sound recording, telephones and aviation were heralding the dawn of a new age. Artists sought new ways to reflect the modernity of the era, such as subjective perspective in Expressionism and multiple perspectives in cubism. Modern compositional tendencies reflected those of the visual arts, with music becoming less tied to traditional tonality and musical notation. Music became more conceptual and experimental, with a unique form for every work. Some composers even abandon traditional music notation, instead experimenting with graphical ones, such as John Cage's Concerto for Piano. This kind of breakdown in genre gives rise to a need for a tangible form of the structure to assist in understanding.

Early in the twentieth century, graphic techniques were used to illustrate analytic aspects of musical scores. It can be dated back at least to Alfred O. Lorenz (1924), who used graphs to show the modulatory scheme for Wagner's Ring Cycle [3]. He developed a method of exploring Wagner's musical texture as large, closed totalities. With the increasing popularity of computers in the 1970s, music visualization becomes more accessible; either from pitch and rhythm extracted from traditional scores or sound-based visualizations such as waveforms and sonograms.

While sound-based visualizations are useful for displaying performance information, this paper examines compositional structures that are best visualized from data. While the musical notation is optimal for conveying instructions to a performer, it is not easy to quickly read larger structures from a score, which may otherwise be obscured by crossing many pages or extra-musical directions. Pitch, rhythm and timbre are the clearest features that can be extracted from scores, so these features are used in the following visualizations[1]. The axes for the visualizations display pitch against time in a manner similar to piano rolls or MIDI sequencing software, with time on the horizontal axis and pitch on the vertical axis. Color and shape are used in some of the visualizations to highlight timbral or voice information.

2. A MOTIF AND ITS DEVELOPMENT

The motif, a device widely used in compositions, always has been one of the basic musical elements. Impressive examples include the four-note motivic cell in Beethoven's Symphony No.5 Movement I, and the atonal thematic motif of Aaron Copland's Piano Variations. Development of a motif is a common method of composition. Therefore, visualizing a piece's motif and subsequent development is the best way to show the benefit of score visualization.

He Xuntian's *Scent Dance I* (2009) for solo clarinet is constructed using a short motivic pattern that is varied and developed throughout the piece. Although the

[1]A good example of this sort of visualization can be seen in those of the Music Animation Machine. (http://www.musanim.com).

composition is written for a single instrument, the monophonic line perceptually splits into polyphonic voices as the pitch range expands, bifurcating into two interlocking forms of the motif. The piece's construction is that of a fractal, with the motive pattern-forming event the large-scale double-arched structure of the entire piece. The following visualizations are used to highlight the motivic structures that He uses to construct *Scent Dance I.*

2.1 The Motivic Pattern

Scent Dance I starts with a ten-note motif that is used to generate structure throughout the piece. This section illustrates how graphical notation can be used to analyze and explain the construction of the piece more clearly than can be done by looking at the traditional score.

Figure 1 shows the opening motif in traditional notation, and a more graphical interpretation is shown in Figure 2, which plots pitch against time. The individual notes are shown as dots in the graphical form, with dashed lines enhancing the motivic shape of a double arch, or a capital "M". Although the contour of the motif can be seen in the original score, other notational features such as barlines, dynamic markings, and slurs obscure the essence of the pattern. Thus, the graphical display in Figure 2 better highlights the importance of pitch contour. Exact pitches are more difficult to read from the graphical version of the pattern, but this helps to emphasize the contour rather than the pitches. Besides this, it also helps to find out how many notes and how many times of these notes have been used in the motif, even in the whole piece. This motif contains four pitch classes, each repeated a different number of times: 1 A, 2 Fs, 3 C-sharps and 4 Ds.

At a glance, the motif seems to be symmetric, but in fact it is slightly asymmetric. The motif consists of two slightly offset symmetries between pitch and rhythm. The C-sharp at the start of measure 4 is a point of symmetry for the first nine notes of the motif, while the middle of measure 4 forms a mirror point for symmetry in the ten rhythms of the motif. The offset between pitch and rhythm shows up in the graphical version as a distorted symmetry.

Figure 1 The motivic pattern of *Scent Dance I* (Bars: 1-7)

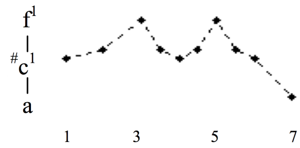


Figure 2 The visualization of the motivic pattern (Bars: 1-7)

2.2 The First Part with Motivic Development

Based on these elements of pitch, rhythm and contour, the motif is used to build the following phrase (Fig.3). Using the compositional data, we can plot a graph (Fig. 4) in which each pattern is clearly visible, showing 14 patterns including the opening motif. A quick examination of the graphical patterns shows that they are similar, but not exactly the same. Some contain wider intervals, while others have longer durations inside each motivic variation. The ending note of a motivic cell is not only the end of one cell, but also the beginning of the next. All of the patterns connect to each other, constructing the first part of the composition.

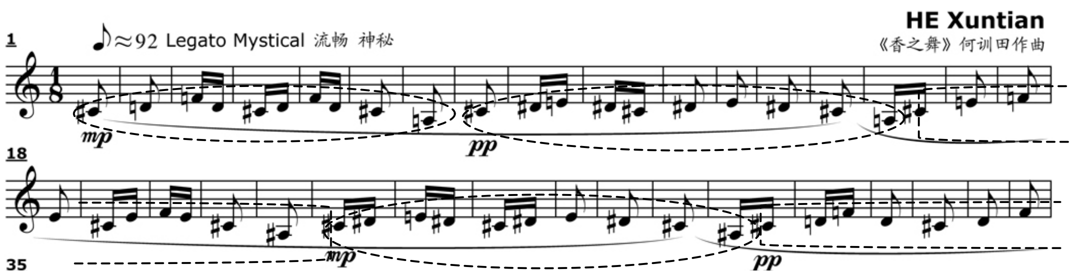



Figure 3 The first part with development on the motif (Bars: 1-34)

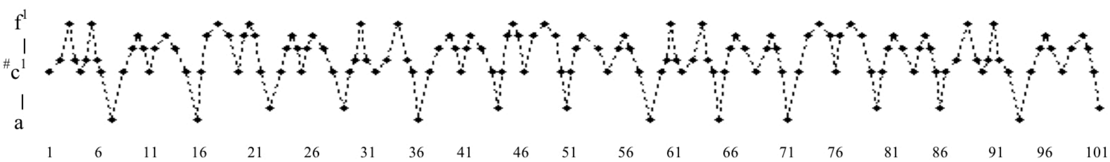


Figure 4 The visualization of the first part (Bars: 1-101)

2.3 The Second Part with Development

In second part of *Scent Dance I*, the development of the motif is more complex. Figure 5 gives the notation for the modified three-measure motif. On the graph in Figure 6, eight repetitions of this motif can be seen. The patterns in this part seem difference from the original motif. However, by carefully observing the shape of the new motif, we can identify patterns that use the “M” shape contour through the graph. The pattern changes into two interlocking forms of the original motif, each in a different register. It is no longer a monophonic melody, but rather a complex perceptual stream. One stream is outlined in Figure 6 with a blue line following the lowest notes of the pattern. This lower outline forms the same contour as the original motif (Figs. 1 & 2). Another stream is outlines by the red line following the contour of the upper pitches. This top stream can be viewed as an inversion of the lower one, a bit like a “W”, or it can be viewed as an offset version of the “M” contour. These interlocking patterns and their trails are not otherwise easy to recognize in the score due to the rhythmic activity and linear presentation of the notation.

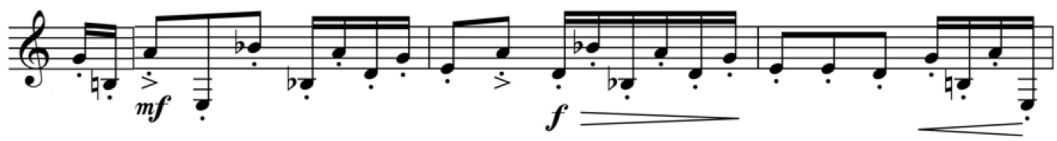



Figure 5 The second part with development on the motif (Bars: 123-25)

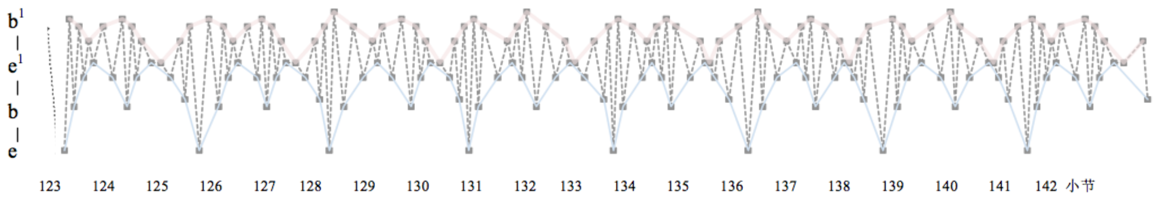


Figure 6 The visualization of the second part (Bars: 123-142)

2.4 The Fourth Part with Development

The technique of multi-layering in the second part continues within the fourth part, as the motif is further overlaid and streamed. It is more complex than ever before (Figs. 7 & 8). Besides the two layers, a mid-ranged pedal note and grace note has been added.

For the two layers of the pattern, we can see that the basic shape of the original motif has been kept in the lowest layer. The upper layer has more activity. At the end of the part, some notes change their register to the higher octave, giving the music more tension than before and leading to the climax of the piece.

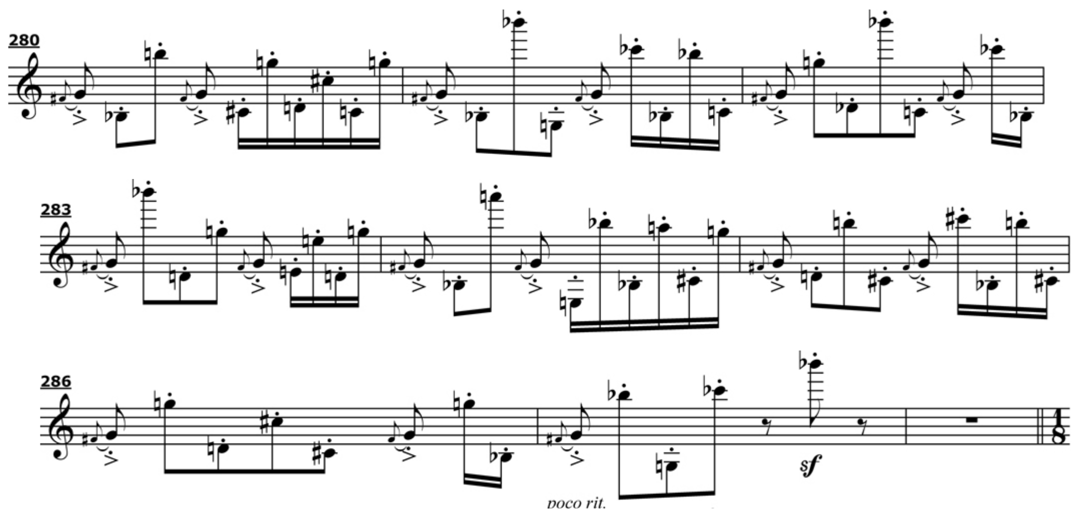


Figure 7 The fourth part with development on the motif (Bars: 280-287)

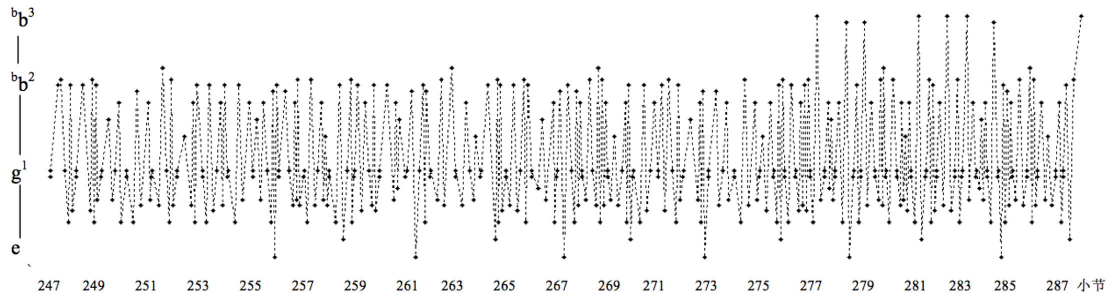


Figure 8 The visualization of the fourth part (Bars: 247-287)

2.5 The Whole Structure

In the visualization of the entire composition seen in Figure 9, we can clearly perceive the composers' thinking—the motif's transformation and recurrence in the original form, but in different registers, gives structure to the whole composition. The form of this piece is ABACA, plus a coda. In the whole graph of the piece, we can see the first, third and fifth parts are more active but also change register: the pitch ranges being a3–f4, a#4–f5, and b4–g5 / b3–g4. The interval in these three parts is about a sixth. In second part, the range is E3–Bb4; the fourth part has wider pitch range that is E3–Bb6; in the coda, the range is C4–C6. From these alternating narrow and wide pitch ranges, the whole structure of the piece has the same shape as a capital "M", mimicking the opening motif.

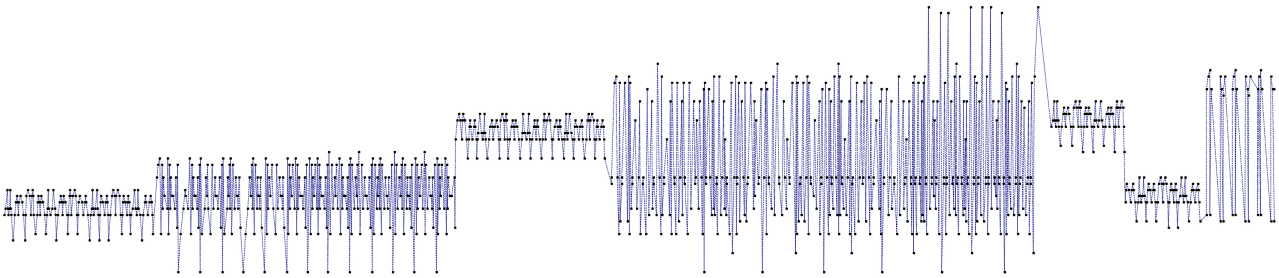


Figure 9 He Xuntian: *Scent Dance I*

3. VISUALIZING OTHER FEATURES

Data visualization not only shows the trail from the motif to the whole piece, but also shows many other aspects of a work.

"A good graphic representation enables us to perceive many aspects of the musical 'shape' of a composition" [4]. Even though the graphs only show pitch and time, other higher-level features such as pitch range and texture can be seen from micro to macro forms.

3.1 Pitch Range and Tendency

Viewing pitch usage in modern music on a graph can amaze an audience. The visual patterns are more obvious than in classical music. For example, Figure 10 shows a huge contrast of pitch range within a composition. This piece is for harpsichord by György Ligeti, written in 1968. The composer wanted to compose a piece "that would be a paradoxically continuous sound, something like *Atmosphères*, but that would have to consist of innumerable thin slices of salami". Ligeti's earlier works used a technique known as micropolyphony. Rather than a using large orchestra, this piece uses extremely rapid activity by a dense, rich stack of pitches with a solo instrument to create the impression of continuous sound and a monolithic image.

The graph also shows an attraction and repulsion of the two hands. In the outer parts of the composition the two hands are entangled in the same registers, but in the middle part they drift away from each other.

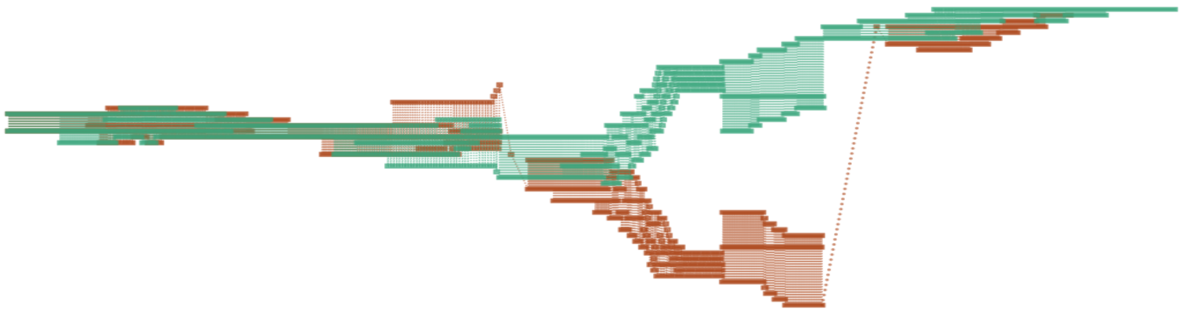


Figure 10 Ligeti: Continuum for Harpsichord

3.2 Texture and Voice

The texture generated by the two dimension of pitch and time represent the inter-stream correlations of a piece. Through an overview of this texture, broad categories of monophony, homophony, polyphony, and heterophony can be read from the graphs.

The following graph displays Webern's Symphony, movement 1, part 1 (Fig. 11). Schematic organization of pitch, rhythm, register, timbre, and melodic contour is Webern's innovation. His eagerness to redefine imitative contrapuntal techniques, such as canon and fugue, can be seen clearly using this graph.

Different colors represent each voice in the score. Some imitative patterns can be seen counterpoint relationships, such as the pink and green points, or the yellow and purple diamonds. These pairing have inversional symmetry.

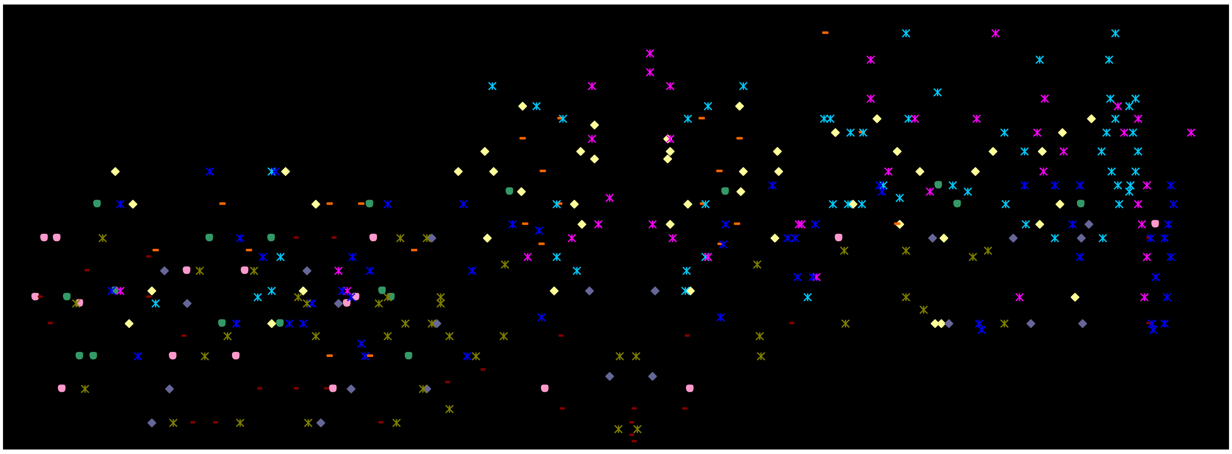


Figure 11 Anton Webern: Symphony (Op. 21) Mv.1 Part1.

4. MACRO STRUCTURE

In musical compositions, texture also delineates structural aspects. Data visualization is an ideal format for recognizing such structures. Let us examine the complete musical formal structure of a piece, which can be recognized in a glance from a visualization, showing both the dissimilarity and similarity in the macro structure.

The following example is Steve Reich's minimalist work "Octet" composed in 1979, later rescored as "Eight Lines" in 1983. The ensemble consists of string quartet, two pianos and two clarinets doubling both bass clarinet and flute as well as piccolo.

Figure 12 shows that the piece is organized into five sections. We can see that the two pianos (shown in orange and light green) build a dense background, using syncopated ostinatos. In first, third and fifth parts they cover a wide register, while in second and fourth parts they move to narrower higher section of the register. This gives space to the cello (purple), viola (blue) and bass clarinet (red) for their sustained lower tones. It also shows that the division between sections is very smooth with some overlapping between the sections. These characteristics show the compositional technology, such as repetitive figures, slow harmonic rhythm and canons.

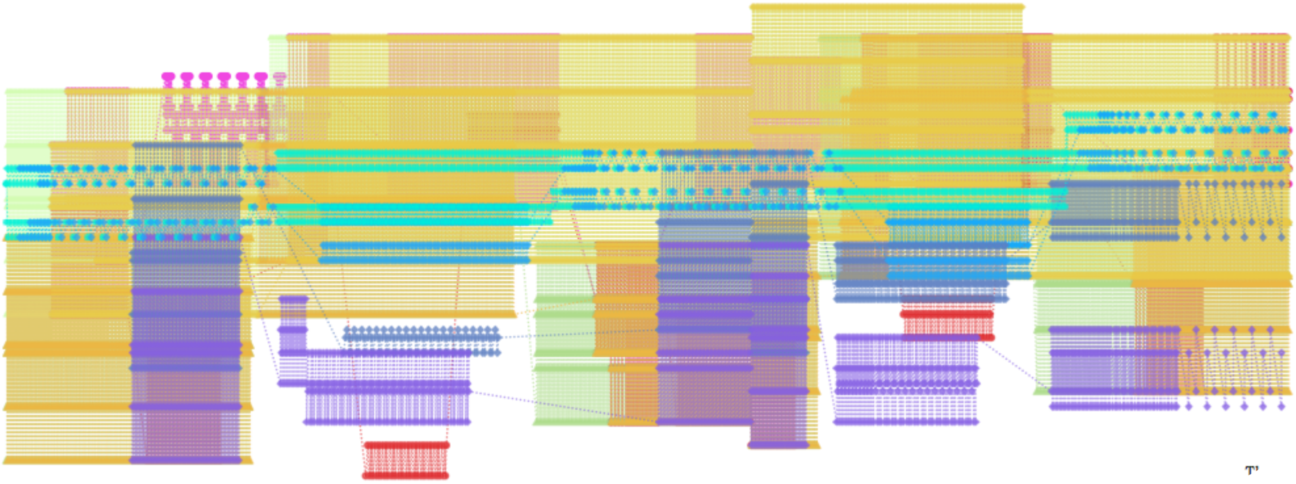


Figure 12 Steve Reich: *Eight Lines*

5. CONCLUSIONS

Above all, visualization graphs help in the recognition of motifs and their development through pitch contours, register, texture and the structure in both the pitch and time domains. Graphs can make the audience more aware of the overall structure in a composition rather than focusing on individual notes or phrases. They can find the details on graph to enhance their listening experience. Performers can also discover new relationships between the surface features of notes and the deeper structures of form by using such graphs; otherwise, the link between notes and a composers' overall intentions are difficult for individual band or orchestra members to intuit from the parts.

We have seen that visualization can produce a beautiful picture for a piece. Listening to the music while viewing the graphs, a listener will become a bit like the conductor who sees the whole score. People can foresee what will happen in the next passage and remember the music already passed through on the graph. Mapping time into space also allows non-musician to understand better the temporal aspects of the music. The artists and technicians can also use it as a sketch to interrelate with other media, such as dance, drama and animation.

For more example graphs, view the GraphMusic channel on YouTube at www.youtube.com/channel/UCnF-gtWPS520-C4-bu6WvQg .

6. REFERENCES

[1] Brian Evans, "The graphic design of musical structure: Scores for listeners: Incantation and mortuos plango, vivos voco." In EMS-05: *Electroacoustic Music Studies Conference 2005*, 2005.

[2] Tobias Kunze and Heinrich Taube, "See–a structured event editor: Visualizing compositional data in common music." In ICMC'96: *Proceedings of the International Computer Music Conference 1996*, pages 63–66, San Francisco, CA, USA, 1996. ICMA.

[3] Alexander R. Brinkman and Martha R. Mesiti: Computer-Graphic Tools for Music Analysis, in *Proceedings of the 1991 ICMC*, pp. 53-56 (Montreal, Canada: McGill University, 1991)

[4] Waters, Simon and Tamas Ungvary, "The Sonogram: A Tool for Visual Documentation of Musical Structure." *Proceeding of the 1990 ICMC*, 159-63. San Francisco: ICMA.

[5] Pope, S.T., "Music Notations and the Representation of Musical Structure and Knowledge." *Perspectives of New Music* 24(2): pp. 156-189. 1986

[6] Huron, David. "Characterizing Musical Texture." *Proceeding of the 1989 ICMC*, pp. 131-34. San Francisco: ICMA.

[7] He Xuntian. *Scent Dance I.* Schott Music, Mainz, Germany; 2009. ISMN: 979-0-001-19082-4

NOTATION AS TEMPORAL INSTRUMENT

Eric Maestri
LabEX GREAM, Université de Strasbourg
CIEREC, Université Jean Monet, Saint-Etienne
eric.maestri@gmail.com

ABSTRACT

In this paper the author proposes a descriptive musicological framework built on the notion of notation as temporal instrument in today's context of electronic music. The principal goal is to discuss a research categorization of musical notation that consider the performative character of musical writing in electronic music performance. In the intentions of the author, this framework could resume the multiple enhancement of the temporal dimension of notation implied by the new means of performance in electronic music.

1. INTRODUCTION

Claude Cadoz [1], Anne Veitl [2] and Chris Nash [3] define a notational system from the point of view of its usability and performability. In particular, for Veilt, *performability* and *causality* are two of the main characters of a writing system ("*système d'écriture*"), among *materiability*, *visibility*, *readability*, and *systemic character*. For Veilt programs for synthesis and sampling are at the same time instruments, in the sense of Mathews, and tablature-like scores. Thus, music notation becomes a concrete instrument for performance, exploiting its performative character. While traditionally music notation is used to write past events (with the principal objective of documentation, analysis and transmission), or future events (new compositions written for future performances), today, programming is an act, applied to perform electronic music in the present, in the studio or on the scene.

These considerations are related to the philosophical debate of notation of Goodman's theory of notation [4], *event theory* [5, 6, 7] and *embodied cognition* [8, 9]. However, the author is aware of the subtle difficulties that lies in these theories and ask the reader to consider them more as a theoretical reference than a philosophical discussion.

2. TEMPORAL INSTRUMENT: DELAYING AND PROJECTING MUSICAL ACTS

As claimed by Christopher Small, composers "provide materials for the performance" [9]. This statement denotes one dimension of notation, that is the prescription of actions oriented towards the creation of events on the scene. Thus notation could be seen, in part, as characterized by *projections*. This musical act could be seen at the light of Nelson Goodman's concept of *projectibility*:

> To learn and use any language it to resolve problems of projection. On the basis of sample inscriptions of a character we must decide whether other marks, as they appear, belong to that character; and on the basis of sample compliants of a character, we must decide whether other objects comply. Notational and discursive languages are alike in this respect. [4]

or Andrew Sorensen's notion of *act of programming* [13]. Today, if we accept, as Veitl suggest, that programs are scores, scores create events in performances becoming a particular kind of instrument, that could be played in order to create music in live performances. This hypothesis, near to the causal paradigm proposed by Veitl, incorporates two other theories. The first one is the one of "sound event" explained by O'Callaghan as following:

> Sounds stand in causal relations to the activities of objects and events that are sound sources, and they fulfill the causal requirement on any account of their veridical perception. Sounds thus occupy distinctive causal roles. Sounds are particular events of a certain kind. They are events in which a moving object disturbs a surrounding medium and sets it moving. The strikings and crashings are not the sounds, but are the causes of sounds. The waves in the medium are not the sounds themselves, but are the effects of sounds. Sounds so conceived possess the properties we hear sounds as possessing: pitch, timbre, loudness, duration, and spatial location. [7]

The second one is the physiological and perceptual implications of electronic music. Traditionally, the body is the trigger that gives energy, and sense, to sound events towards physical effort. Nowadays, in electronic music performance, this effort is transferred to the computer (and its interfaces). However, even if it is perceptually weakened, in this evolution the causal aspect of notation rests fundamental. In fact notation projects movements in time, prescribing causalities in the future and synthesizing possible causalities of the past: it embodies, in the case of human or digital performer, information for the performance. Notation, used as compositional instrument, is characterized by projections of movements in virtue of the absence of the physicality of the sound and of the performer. But, as traditionally, even in electronic music the composer projects sound objects in time in an intersubjective dimension. Thus the score includes, implicitly, the body of the performer (human or digital) in a unique musical act that starts from the composition of the score and ends in the public performance of the piece. Performers and composers are entrenched in the same form of projection characterized by different degrees of distance from the gestural and sonic output. The composer uses the score as an instrument, as a temporal and physical interface of abstract interaction in time and space with the body of the performer which aims for it to create the sound event: *scores are extensions of the body of the composer in the body of the performer via the projection of the instrument represented by the score* [10]. That creates a singular temporal dimension based on the absence and presence of the instrument: the composer constructs absences and the performer reconstructs the projected presences. In recent electronic music's performances the composer, which write the score on the scene programming the music, correspond with the performer. It seams to the author of this paper that writing become a performative instrument.

As instruments are spatially related with sound and connected directly to the body of the performer, notation is just behind the performative gesture, temporally related with the gestural causality of sound. In electronic music programming is, for instance in *live coding*, a performative act. Consequently emerges a new dimension of notation as instrument (in the sense that it relates sound to gesture) that is not only instrument of the past (memory) and future (projection) but also of the present (performance). This development, in the idea of the author, is due to the new dimension of prescription that electronic music means imply. These forms of prescription, in notation, express, anyway, two forms of causality realized traditionally for the human body and today for the electronic body, the loudspeaker.

3. TWO DIMENSIONS OF PRESCRIPTION: *ERGOGRAPHIC* AND *PHONOGRAPHIC*

In the case of a "traditional notation" the sound event is created towards prescription of body movement. In the electronic music programming prescribe via the computer the movements of loudspeaker's membrane. We propose the notions of *ergographic* and *phonographic* in order to highlight two different kind of notation: the first one is used to prescribe movement of the body, controlling the instrument; the second one is conceived to control the movement of the loudspeaker. *Ergographic* defines prescription of movement as causes of sound based on the notion of note. In *Ergographic notation* composer indicates implicitly (or explicitly in the case of tablatures) the movements that must be used, interpreted, performed to create the resulting sound, as result of a final musical act. *Phonographic notation*, prescribes the movement of the loudspeaker towards the elaboration of information by the machines, indicating the precise parameters that compose the resulted sound.

4. NOTATION AS INSTRUMENT OF THE PAST, OF THE PRESENT AND OF THE FUTURE: NEW TIMES OF MUSIC NOTATION

Starting from this framework the author will try, in the last part of the paper, to propose a topology of notations from the point of view of their performative aspect. We define the structure of instrument-notational intentionality into three temporal dimensions and associate new means for musical notation to the following categories: notation of the past, notation of the present, and of the future.

4.1 Notation of the past

Notation is used to reconstruct a possible origin of a recorded sound event. It has the objective to represent, extract and transmit informations from a past event, recorded or memorized. The programs that allows the translation of information are numerous and used as basis of MIR analysis. Recently: Sonic Visualizer, MirTool Box, Tony etc. [11, 12].

4.2 Notation of the present

Notation as instrument of the present is conceived as a concrete performative means. The intentionality, as a complex amalgam of informations and projections is realized in the very moment of the transcription. Thor Magnusson [13] and Andrew Sorensen [14] provide through *Live Coding* examples of this new kind of instrumental relationship with notation. In a similar manner the live notations of Chris Fischer [15], Ryan Ross Smith [15] and Richard Hoadley [16] are instrument-notations built on an improvisational environment based on the instantaneous interpretation on the sign projected on the screen. This new form in instrumental relationship based on short term projections of prescriptions to the performers exalt the scenic presence of the notational means and integrate the act of writing with the traditional act of performing. In this context we can define two sub-dimensions related with two types of direct performance of notations oriented to the creation of instantaneous sound events.

Notation as instrument of the present on composer's "table": it corresponds to the studio dimension related with composition, in which the composer works in a quasi-perfomative environment with the computer. Thus the distance between the writing and the result is diminished: traditional scores, sketches, representations that allow the composer to control simulations in the sense of present intentionality projected in the immediate future, delayed and scaled in the direction of the further future. In addition to the traditional instruments of CAC there are more recent examples: Pierre-Alexandre Tremblay's *thinking inside the box project* [17], Rodrigo Costanzo's *dfscore* [18], Daniele Ghisi and Andrea Agostini's *bach* [19] etc.

Notation as instrument of the present, on the scene. The other one is on the scene, in which the score is used to create performances: *Live Coding*, score generated, read and interpreted at the same time and live notations: Chris Fischer, Ryan Ross Smith , Richard Hoadley, Cat Hope, Lindsay Vickery [20] and Aaron Wyatt [21]. This is the case for *Distant Mirrors* by Jean-Baptiste Barrière who claims:

> In this way, *Distant Mirrors* intends to put the performers in a situation similar to the "state of dream", in which you can recognize some elements but not all of them, many being alternately close or strange, in which you never know exactly what is going to happen next, you just have to adapt to the enigmatic course of events, and create your own interpretation of dreams [22].

Coherently with these developments Andrew Sorensen and Henry Gardner state that

> An *act of programming* is usually considered to sit firmly within the context of "software development", with the latter being the active process behind the production of "software products". In this view, causal actions made by the program- mer target the design and notation of a formal specification for *future* action by a computing system. [14]

4.3 Notation of the future

This dimension was the standard one of notations in the whole history of music. This approach still characterizes the actual compositional practice, extended via the instantaneous interaction in the studio. However, this typical interaction in western music is still represented, as in TENOR 2015 by Carlo Laurenzi and Marco Stroppa [23] and Pedro Rebelo [24].

5. FURTHER DEVELOPMENTS

This research has the objective to present a perspective that considers the musical act of writing, notating, as part of the musical act of performing. From the point of view of musicological research this framework has the objective to provide three descriptive categories for notation as instrument musical practices in electronic music. The author think that the proposed categorization could be part of a larger project of classification caracterized by the temporal features of notational-instruments. In fact the new tools for music notation and representation emerge under the notion of temporal instrument, that include the notion of *projectibility*, defined by Goodman, the notion of *embodiment* and *sound event,* and Anne Veitl's researches about the causal paradigm that se proposed. The new tools of music representation and performance provide a large reservoir of examples capable to support this perspective highlighting the new possibilities of the projections of actions, differed intentionality, in musical writing. Author's hope is that this framework can help researches to categorize the recent musical practice and to define, in the future, a topology of new notations from the point of view of performativity and temporality. Next developments will be characterized by the analysis of the new tools for music notation in relation with the enhancement and the depth of the theoretical framework proposed here.

6. CONCLUSIONS

Notation was traditionally used to prescribe future musical events. The new means for performance and notation break the temporal dimension of standard notation enlarging an essential element of notations, that is *performability*. Our research has the objective to frame theoretically this general problematic resumed under the definition of *notation as temporal instrument.* This definition is a direct consequence of the remarkable reduction of the theoretical distinction between the inner space of intentionality and the outer space of performance. At the same time this new performative territory from musical notation is already inscribed in the evolution of electronic means since the development of Music III. Nowadays notation is used to project intentionality in the present, the future and to reconstruct the past. The author try, with these framework, that is still in progress, is an attempt to provide a simple categorial means that allow a topology and more in deep description of new means for notation and performance in electronic music. The new notational techniques enlarge the temporal possibilities of the Instrument defining it as an instrument for the performance, a veritable performative means.

Aknowledgments

I would like to thank LabEx GREAM, Pavlos Antoniadis for the very intelligent and smart reflexions and suggestions, Laura Odasso for the patience and the two anonymous reviewers.

7. REFERENCES

[1] C. Cadoz, "Musique, geste, technologie", in *Les Nouveaux gestes de la musique*, Marseille, Editions Parenthèses, 1999, p. 47-92.

[2] A., Veitl, "Notation écrite et musique contemporaine : quelles grandes caractéristiques des technologies numériques d'écriture musicale", *JIM07 (Journées d'Informatique Musicale),* 2007, http://www.tscimuse.org/biblios/veitl/technologiesecrituremusicale.pdf.

[3] C., Nash, "The Cognitive Dimensions of Music Notation", *Proc. Int. Conf on New Tools for music notation and representation - TENOR*, Paris, 2015.

[4] N. Goodman, *Languages of Art. An Approach to a Theory of Symbols*, Hacket Publishing Company, Inc., Indianapolis/Cambridge, 1976.

[5] R. Casati, J. Dokic, *La philosophie du son*, Paris, Jacqueline Chambon, 1998.

[6] W. W. Gaver, "What in the World do We Hear? An Ecological Approach to Event Auditory Perception", Ecological Psychology, 5:1, 1993a, p. 1-29.

[7] C. O'Callagham, Sounds: a Philosophical Theory, Oxford-New York, Oxford University Press, 2007.

[8] F. J., Varela, E. T., Thompson, E. Rosch, The Embodied Mind: Cognitive Sciences and Human Experience, Cambridge, MIT Press, 1993.

[9] C. Small, *Musicking. The Meaning of Performing and Listening,* Middletown, Connecticut, Wesleyan University Press, 1998.

[10] E. Maestri, P. Antoniadis, "Notation as Instrument: From Representation to Enaction", *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

[11] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, S. Dixon, "Computer-aided Melody Note Transcription Using Tony Software: Accuracy and Efficiency, *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR, Paris*, 2015.

[12] T. Magnusson, "Codes Scores in Live Coding, *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

[13] A. Sorensen, H. Gardner, "Programming with Time. Cyber-physical programming with Impromptu", *Proceedings of OOPSLA10 : ACM International Conference on Object Oriented Programming Systems Languages and Applications,* New York, 2010, p. 822-834.

[14] C. Fischer, "Understanding Animated Notation". *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

[15] R.R. Smith, "An Atomic Approach to Animated Music Notation", *Proc. Int. Conf. On New Tools for Music Notation and Representation* – TENOR, Paris, 2015.

[16] R. Hoadley, "Semaphore: Cross-domain Expressive Mapping with Live Notation", *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

[17] P.A. Tremblay, S. McLaughlin, "Thinking inside the box: A New Integrated Approach to Mixed Music Composition and Performance", *Proceedings ICMC 2009*, p. 379-386.

[18] R. Costanzo, "dfscore 2.0", http://www.rodrigoconstanzo.com/dfscore/, 2015.

[19] A. Agostini, D. Ghisi, "Real-Time Computer-aided composition with *bach*", *Contemporary Music Review,* vol. 32, part 1, p. 41-48, 2013.

[20] C. Hope, L. Vickery, "The Decibel Scoreplayer – a Digital Tool for Reading Graphic Notation", *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

[21] A. Wyatt, C. Hope, L. Vickery, S. James, "Animated Music Notation on the iPad (Or: Music stands just weren't designed to support laptops)". *Proceedings of the International Computer Conference, Perth*, WA, 2013, p. 201- 207.

[22] J-B., Barrière, *Distant Mirrors*, 1999.

[23] C. Laurenzi, M. Stroppa, "The Notation of Dynamic Levels in the Performance of Electronic Music, *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

[24] P. Rebelo, Composing with Graphics: Revealing the Compositional Process through Performance, *Proc. Int. Conf. On New Tools for Music Notation and Representation – TENOR*, Paris, 2015.

VISUAL CONFUSION IN PIANO NOTATION

Marion Wood
University of Muenster
mwood@uni-muenster.de

ABSTRACT

This series of Reaction Time experiments investigates how quickly notes can be read from a screen and immediately executed on a MIDI keyboard. This makes it possible to study pitch reading and motor coordination in considerable detail away from the customary confounds of rhythm reading or pulse entrainment. The first experiment found that reaction times were slower in extreme keys (3#, 4#, 3b, 4b), even for very experienced sight-readers, a large effect of clef in most individuals, and other results suggesting that, in this simple paradigm at least, reading notation presents more of a difficulty to execution than motor coordination. A second experiment found, in addition, an effect of order in which the notes were presented.

A clarified form of notation was devised that disambiguates visual confusion across key signatures, and to some extent across clefs. Initial results from an experiment to contrast traditional noteheads with the clearer ones found substantial improvements in both Reaction Time and accuracy for the clarified notation. The possible applications of improved notation to the wider field of piano playing are discussed.

1. INTRODUCTION

Existing research into piano sight-reading [1] suggests that expert sight-readers may process common musical configurations as 'chunks' to a greater extent than novices. This study looks at the question of how some common musical chunks are learned or recognised.

Musical notation, considered as a semiotic system, is not a very effective map of the physical space of the piano keyboard (Figure 1). It does not illustrate the octave repeating pattern of the keyboard, and identical visual symbols or clusters of symbols must be executed differently by the two different clefs/hands.

Simply tabulating the different possible responses to a single common triad, (Figure 1) we find no less than ten visual-to-spatial mappings, considered across two clefs and eleven key signatures. The mappings also have different musical meanings: major, minor and diminished are words describing the musical 'character' of a chord.

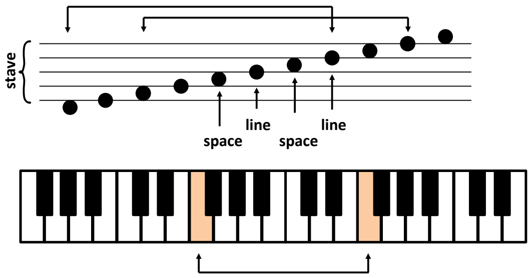


Figure 1. After seven notes of the scale, the keyboard repeats. Unfortunately the binary structure of the stave does not represent the number seven very effectively.

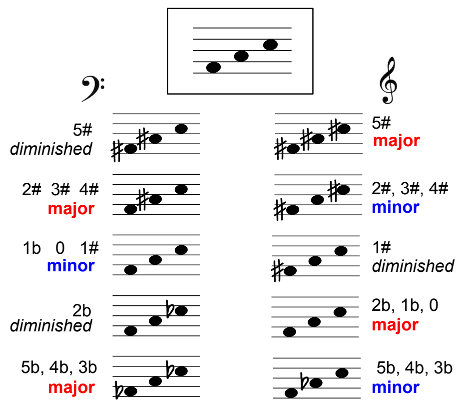


Figure 2. Ten different musical 'meanings', each with a specific motor response pattern, represented by a single visual fragment.

They all sound different, despite looking the same. The notation is not supporting 'chunk' learning or recognition by reflecting either execution mapping, or auditory mapping, or musical meanings.

Drawing a parallel from text reading, homographs and homophones cause particular difficulty for dyslexic readers [2]. Homographs are words that look the same, but whose sound and meaning are different: *lead*: *to go first* or *lead*: *a metal*. Conversely, homophones are words that sound the same, but whose meanings and visual presentations are different: *two, to, too*.

Even in non-dyslexic adult populations, homographs are read more slowly than singular control words, although homophones may be read marginally quicker [3]. Fortunately in most languages, these awkward words are the minority exceptions. By contrast, in piano music, any potentially recognisable musical 'word' – a chord, a scale fragment, or melodic pattern – can be classified both as a homophone and a homograph, having two separate execution patterns within any given key (Figure 3).

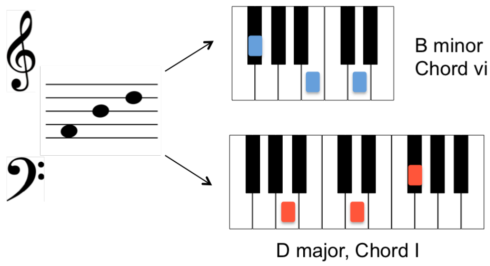


Figure 3a. The same visual fragment requires a different motor response in treble/bass clefs (right/left hands). Example from Key signatures of 1#, 2# or 3#.

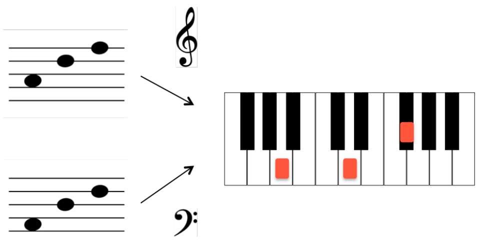


Figure 3b. A similar execution configuration and musical meaning requires two different visual presentations in treble/bass clefs (right/left hands). Example from Key signatures 1#, 2#, or 3#.

In mental chronometry research, visual processing is a topic of interest. Participants might be required to classify a visual stimulus according to various different rules, pressing one of two (or more) buttons in response, as quickly as possible. The time from stimulus presentation until the participant responds is the Reaction Time (RT).

Findings from this area include that are relevant to a discussion of sight-reading include: an increase in RT if the rules for responding are changed (a task-switch cost), longer RTs if the stimulus can be interpreted under two different rule-sets, a general increase in RTs when more than one rule set has to be held in mind at any one time [4], and the 'Simon effect' - an increase in RT if the buttons are arranged in an incongruent way, such as being required to press a right-hand button when a leftwards arrow is presented [5].

All of these factors may be considered to apply to music reading at the keyboard, where a left-right mapping on the keyboard is represented by low-to-high visual (and sound) mappings, and focusing on two different clefs requires us not only to hold two rule sets in mind, but also to switch between them frequently. The experiments below use standard RT paradigms to investigate these effects directly.

Some common musical patterns are normally taught to students of the piano under the topic of "Scales & Arpeggios", [6] although these exercises are often memorised. Thus while we may find that these patterns have been systematically rehearsed in their motor-execution, perhaps their recognition from visual presentation has not. Nevertheless, they represent the kind of chunks that we would expect expert sight-readers to recognise easily.

In the key signature 2#, for example, the chords of D major and B minor described in Figure 2 are so common in the musical literature that we would expect these patterns to become familiar very quickly to anyone who had played one or two tonal pieces in that key. Rather than asking how excellent sight-readers learn their skill, we should perhaps be asking why it is that so many pianists with years of experience do not. The hypothesis of this study is that overlapping visual representations may be part of the reason.

In summary, the experiments described below were designed to measure the reaction times of amateur and professional participants to visual musical stimuli in several keys and both clefs. Variations in reaction time were expected to reflect difficulties of processing the visual information, and/or motor coordination.

2. EXPERIMENT ONE

2.1 Method

Participants were requested to respond on a MIDI keyboard by playing series of 3-note combinations shown on a computer screen. Treble clef / right hand stimuli were shown in the top half of the screen, and Bass clef / left hand in the lower half, as seen in Figure 5. Participants were requested to play the notes in the order shown, as quickly as they were able to. There was no aural feedback (the keyboard was silent) but any errors were marked with red crosses on the screen after each trial.

The initial version of this experiment used a classic alternating task-switching paradigm [5], where two trials in one clef were followed by two trials in the other clef, in blocks of approximately 40 trials. The key signature remained the same for two blocks in a row and then was changed, with the whole experiment covering nine keys. (624 trials per participant). All stimuli were common triads in root position or inversion, all ascending, as shown in Figure 4.

The stimuli were grouped into three sets, with each block containing a mixture of two of these sets, while the other set rested. The order of presentation always followed a predictable pattern, *e.g.* (1)treble, (2)treble, (3)bass, (4)bass. Positions (1) and (3) are considered 'task-switch' positions where the clef has just been changed, and (2) and (4) 'task-repeat' conditions. Each stimulus was presented at some time during the block exactly once in each of these four positions. The order of stimulus presentations was otherwise randomised.

22 participants were recruited for the initial experiment by word of mouth from a variety of musical communities, in and around Exeter in Devon, UK.

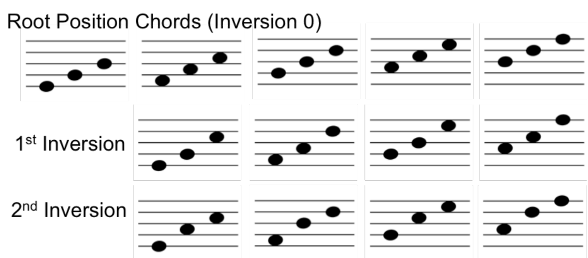


Figure 4. The 13 ascending triads that fall within the stave, used as stimuli in the initial experiment.

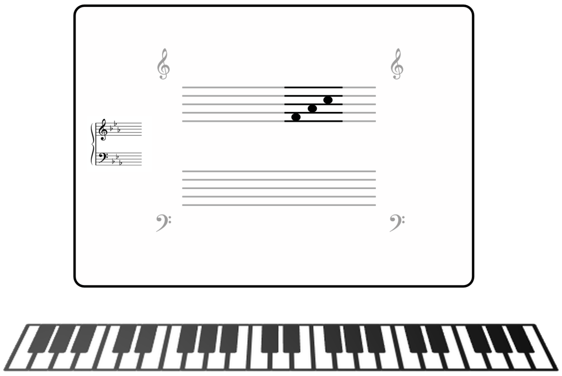


Figure 5. Experimental setup with screenshot of a repeat trial in the treble clef. A reminder key signature remains at the left during the whole block.

3. RESULTS AND DISCUSSION, EXPERIMENT ONE

3.1 Data Analysis

All 22 participants had a level of amateur involvement in music, and about half also had a professional component to their musical activities. They ranged in age from 18 to 74, and use the piano in a variety of different situations, including solo performance, teaching piano, teaching classroom music, teaching another instrument, accompanying another instrument, or learning music they later plan to sing.

The data analysis relies on averaging the mean RT over groups of participants across the cells of the design. Although it would be possible to normalise the data across all participants, there are some aspects of motor coordination and cognitive architecture which are common to all levels of competence. Reaction time is a direct reflection of a physical quantity (processing duration) and is consequently not usually transformed in reporting experiments of this type. Consequently the participants were divided into two main equal-sized groups, consisting of those with an average RT in the region 800-1500ms (11 participants) and those in the region 1500-2500ms (11 participants) with their data averaged in contrasting conditions of interest. As it turned out, this division into two groups on the basis of RT mapped onto a difference in musical history between those who have or had some professional component to their music and those for whom the piano was an adjunct to their other musical activities or a less serious hobby.

Various combinations of factors were grouped for analysis in repeated measures ANOVAs. Error scores of -1 were mostly single errors of execution in the correct hand in the right general area of the keyboard, whereas errors of type -3 were almost all mistakes of switching (the wrong hand used, or wrong clef read).

Main findings of the effect of clef, switch of clef, key signature, change of key signature and effect of the preceding presentation condition of a visual stimulus are reported in detail and discussed below. Other findings of the effect of inversion, diatonic chord and difficulty of hand execution are summarised more briefly.

3.2 Effect of Clef

3.2.1 Results

This contrast compared the mean reaction times found in the two clefs. In this experiment the treble clef was always played by the right hand and the bass clef by the left hand, and so any disparity might be caused either by differences in reading the clefs, or motor coordination differences between the hands, or a mixture of both. Across the expert group, the mean reaction times were treble/right, 1115ms, and bass/left, 1233ms: a difference of 118ms, $F(1,10)=31.77$, $p<0.001$. In the moderate group, these values were 1835ms and 2044ms respectively: a difference of 209ms, $F(1,10)=34.73$, $p<0.001$. Percentage errors were also greater in the bass clef for both groups, but this difference was not reliable in either group, either for the total or for any value of score.

Of 25 participants, three reported being left handed, and three would read music more often in bass clef outside of their piano playing, for example when playing the 'cello. One participant was in both of these groups. All these participants, however, performed significantly better in right hand/treble clef trials. In fact no participants were found for whom the left hand/bass clef showed an advantage compared to the right/treble.

3.2.2 Discussion

Left-handed participants expressed little surprise on being informed that their treble/right hand RTs were faster than their bass/left. They mostly reported the view that they had learned the treble clef first, and therefore had always felt more fluent reading it. In terms of accumulated reading practice, it is also the case that piano music tends to contain more notes in the right hand than the left. Consequently here is probably not the place for a wide-ranging discussion of handedness. However, this finding lends general support to the idea that reading the notation may be more of an issue than motor coordination.

3.3 Effect of Switch of Clef

3.3.1 Results

This comparison contrasted trials where the clef had just been 'switched' with those where the clef was repeated. Across the whole data set, a time cost of switching clef, as opposed to repeating the previous clef, was found. In the expert group the mean RT on clef switch trials was 1232ms, and 1186ms on repeat trials, a difference of 46ms, $F(1,10)=23.64$, $p=0.001$. In the moderate group these values were 2055ms and 1917ms; a difference of 138ms, $F(1,10)=13.75$, $p=0.004$. Levels of error were not significant in either group.

3.3.2 Discussion

During the course of the experiment, it became clear both by observation and self-report that a number of participants were finding it very difficult to maintain the pattern of two-trials-per-hand. Several experienced pianists appeared to be so thoroughly accustomed to alternating hands that they found it extremely hard, even after 20 minutes, to remember to repeat each clef. Eye-tracking studies of fluent sight-readers report a frequent alternation of saccades between clefs [7], [8]; this habitual pattern may be harder to shake off than expected.

Notwithstanding the unexpected difficulty in maintaining the predictable pattern of the experiment, a clear cost in reaction time of switching clef was found. Although not large in comparison to other effects found in this study, this result is interesting in the context of task-switching literature. After hundreds of hours training in task-switching labs, the question of whether participants can ever eliminate the switch-cost with sufficient practice is still hotly debated. This results suggests that switch-costs remain an issue in piano playing, even after thousands of hours of practice.

3.4 Effect of Key Signature

3.4.1 Results

Key signature as a whole was found to be highly significant. Participants generally performed more slowly in “extreme” keys, and faster in “central” keys: 1b, 0, 1#, as seen in Figure 6. In individuals, the pattern was influenced to a greater or lesser extent by favourite keys or recent experience, but the sensitivity to key signature was by far the most substantial effect seen in the experiment. Of all 22, 16 performed best in the key 0, and all but one of the others in either 1# or 1b.

In the expert group, performance was slowest in the key of 2#, with a mean RT of 1196ms, and fastest in the central key of 0, at 1011ms; a difference of 185ms ($F(8,80)=9.54$, $p<0.001$). In the moderate group the slowest average performance was in the key signature of 4#, (2298ms), and fastest in the key 0, (1678ms); a difference of 620ms ($F(8,80)=16.13$, $p=0.002$).

Individual preference or experience in key performance tended to cancel one another out in the means quoted above. In fact no participant’s individual variation between their best and worst key signatures was less than 200ms. In the expert group, the mean of individual differences between best and worst performance in a key signature was 359ms, with the individual differences lying between 200ms and 750ms. In the moderate group, the average individual difference between best and worst keys was 795ms, with a range individual differences varying from 285ms to 1205ms.

Error scores showed no statistically reliable effects, but single-note errors showed some sign of approaching significance, and mirrored the shape of the key change variable: for the expert group $F(8,80)=2.04$, $p=0.108$, and for the moderate group $F(1,10)=3.94$, $p=0.075$.

3.4.2 Discussion

This is a very substantial finding: although sensitivity to key signature varies greatly, apparently even the most proficient pianists are not immune to its effects. The most experienced professional in the experimental set, with thousands of hours experience in playing, sight-reading and accompanying, had a mean RT on correct trials in the central key of 831ms, rising to 1064ms in 4# and 1062ms in 4b: a difference of 233ms. Expressed as a percentage of best performance, the effect of key signature appeared to add some 25% to reaction time.

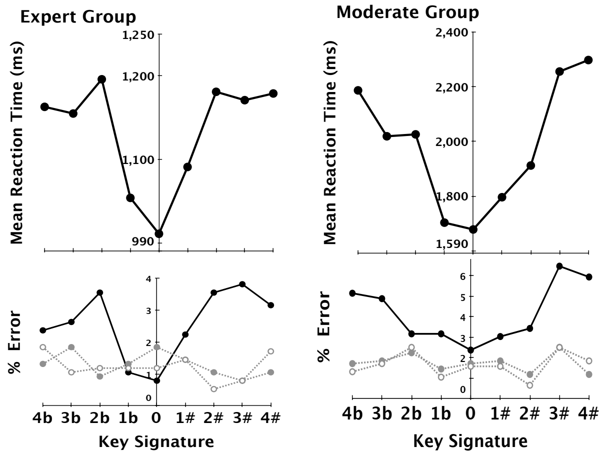


Figure 6. Average RT for Expert and Moderate groups across 9 common key signatures.

This is a result that would be surprising to most musicians, although perhaps not to researchers familiar with the mental chronometry literature. Pianists are generally supposed to become fluent “in all keys” with sufficient practice. The idea that an experienced professional might be as much as 25% slower processing pitch patterns in outer keys than in central ones runs counter to the prevailing view of practical proficiency in piano sight-reading.

In terms of a more nuanced pattern of key signature difficulty, it was seen that participants did not necessarily all find the outermost keys the most challenging. Indeed by self-report, when there were more than 3 modifiers in the key signature (4#, 4b), some participants used a strategy of remembering which black notes not to play (there are 5), rather than keeping track of all the modifiers. This resulted in some participants expressing the idea that outer keys of 4#, 4b, were actually easier than the “middle” keys of 3#, 3b. Participants of this type were more frequent in the expert group, which may be seen from the shape of the means plotted in Figure 6.

3.5 Effect of Changing Key Signature

3.5.1 Results

The key signature was changed every other block, and so reaction times could be contrasted between key change blocks, and those where the key remained as previously. Significantly higher average reaction times were found both the expert and moderate groups in the key change blocks, and so a finer grained analysis, dividing the blocks into thirds, was conducted.

A clear pattern of “settling into” the key signature was seen (Figure 7). In the expert group this was largely captured by a drop of 74ms in mean RT from 1308ms in the first third of a key change block to 1234ms in the next. In the moderate group the drop was 148ms, comparing 2205ms in first thirds of key change blocks to 2057ms in second thirds. The interaction of these effects was statistically reliable: in the expert group $F(2,20)=8.45$, $p=0.002$, and the moderate group $F(2,20)=9.18$, $p=0.001$. An analysis of errors did not reach significance.

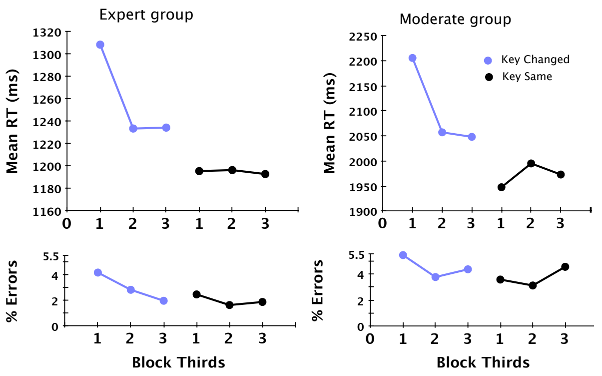


Figure 7. Means for each third of a block, for pairs of blocks in the same key.

3.6 Effect of Novel/Repeat Stimulus groups, and Clef-congruence of Previous Presentation

The effect of stimuli on one another within the experiment was analysed in two ways.

On a global scale, three subsets of stimuli were rotated so that half the trials in each block were from a 'repeat set' – i.e. they were also shown in the previous block, and half from a "novel set" that had been absent in the previous block.

At the local level, within each block, each stimulus appeared four times, once in each clef-switch/repeat condition, i.e. twice in each clef in each block. Investigating whether the RT of a stimulus is affected by its most recent previous appearance, trials were coded according to whether the stimulus had most recently been seen in the same (similar) clef, or in the other clef (different): see Figure 8. Stimuli most recently seen in a previous key signature were removed from this analysis.

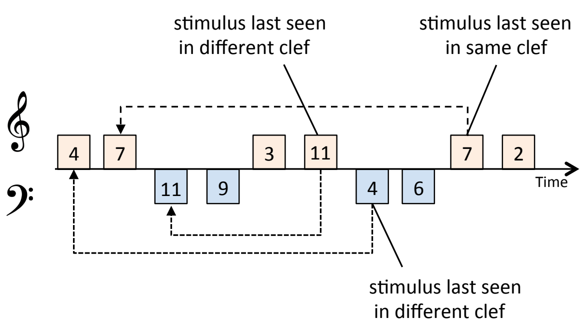


Figure 8. Illustration of last-seen-clef similarity. (Stimuli numbered arbitrarily).

3.6.1 Results

Comparing the two subsets of 'novel' and 'repeat' stimuli within blocks where the key signature remained the same, no significant effect was found in either the expert or moderate groups, or in the error rates. The variable describing 'last-seen-clef' congruence, however, was found to be highly significant in both expert and moderate groups. In the expert group, the mean RT of congruent last-seen-clef trials was 1176ms, whilst mean RT where the last-seen-clef had been different was 1220ms; a contrast of 45ms, $F(2,20)=27.91$, $p<0.001$. In the moderate group, the mean RT for congruence of last-seen-clef was 1910ms, and for incongruent last-seen-clef 2006ms; a contrast of 96ms, $F(2,20)=13.25$, $p=0.001$.

3.6.2 Discussion

This is an important finding. Practicing one visual stimulus (albeit with two different hand interpretations) might be expected to have an effect on the same stimulus in further blocks of the same key. Having either learned, or been 'reminded' of how a particular visual sign should be executed in both hands, we might reasonably expect an improvement in performance in the second block of the same key signature. The fact that any such improvement was not detectable in this experiment, whilst instead, the clef similarity of the most recent previous presentation did make a significant difference, suggests that visual confusion at the local level is active in a substantial way, and may be disrupting more general learning of patterns across both hands.

3.7 Other Effects

Other results are summarised in brief.

3.7.1 Effect of black notes

The number of black notes present in each chord could provide a simple reason for slower performance in outer keys, being perhaps harder to read or execute. Comparing triads with 0, 1, or 2 black notes required two contrasts, as not all types occur in every key. However no effect was found for the number of accidentals, except for a small but significant difference in the expert group, between chords with one and two accidentals, of 41ms, seen in Figure 9.

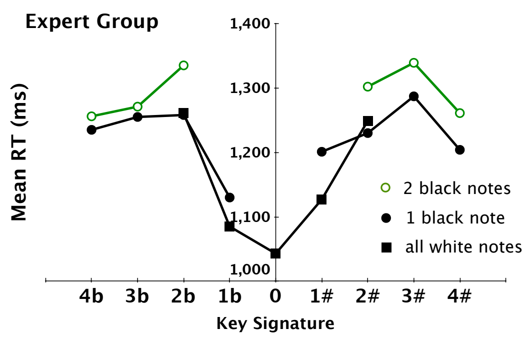


Figure 9. Mean RT in the Expert group of chords with 0, 1 and 2 accidentals.

3.7.2 Inversion

The effect of chord inversion (see Figure 4) was also analysed in combination with clef and the number of black notes in the chord. Unexpectedly, inversion turned out to be a significant factor in itself, and no reliable interaction was observed with hand (treble/bass clef) or number of black notes in the chord. For the expert group the mean RT between fastest (root position) and slowest (second inversion) was 29ms, and for the moderate group there was a much larger difference of 216ms. This effect was entirely unexpected, as inversion is relatively well-represented in the notation, with a slightly wider vertical gap in some triads (Figure 3) corresponding to a greater distance on the keyboard, and execution configurations not noticeably more complex.

Possible reasons for the effect of inversion include a bias towards recognising the chord by its root note, which in this experiment was the initial note presented in the case of the root position chord. Alternatively, it may be easier from a visual processing perspective to read three similar notes all on lines or all in spaces, than to distinguish a mixture of the two types.

3.7.3 Diatonic Chord

The seven chords of each key can be classified by musical type, commonly referenced in music theory by their roman numerals. In each key there are three major chords (I, IV, V), three minor chords (ii, iii, iv) and a single diminished chord (*vii*). Of these, I and vi are the 'naming' chords of each key. Means for these diatonic chord types are shown in Figure 10.

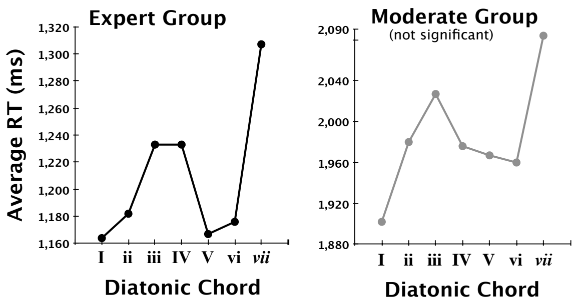


Figure 10. Variation in mean RTs according to diatonic chord. I is the major key chord, vi the minor key chord.

The difference between chord *vii* and the mean RT of all the other chords in the key was significant in both groups, with mean RT differences of 115s and 116s in the expert and moderate groups respectively. This chord is less common in the literature, and requires a slightly different hand configuration.

In the expert group pairwise comparisons between individual diatonic chords were significant for chords I, iii, IV and vi. The idea that chords I and vi may be more easily recognised, particularly by expert participants, is broadly encouraged by these results.

3.7.4 Single-Note Errors

Responses in which only one note was played incorrectly were collated, and showed that across the whole experiment, about 50% were caused by omitting the last accidental of the key signature – the 4th note of the scale in flat keys and the 7th note of the scale in sharp keys. This effect seemed to be irrespective the order in which key signatures were presented.

3.7.5 Unbalancing of the Design

The effects of Inversion and Diatonic Chord unbalanced the design of this experiment to some extent. In the total set of 13 stimuli, there were 5 root position chords, and only 4 of each of first and second inversions (Figure 3). The stimuli were drawn randomly to form subgroups that would rotate across blocks, and there was no control to ensure that an approximately equal mix of inversion types or diatonic chords fell in each block, or consequently each key signature. The larger effects of clef and key signature were likely to be valid findings, but balanced sets should be a requirement of any future experimental design.

3.8 Summary

Table 1 shows a summary of results from experiment one. As noted in the introduction, existing literature suggests that pattern recognition may play a part in sight-reading fluency, and this is broadly supported by these results. The effect of inversion appears to reduce substantially with expertise, and the differentiation between diatonic chords to increase. That clef and key signature should remain such large sources of variation in the expert group is at odds with the prevailing impression amongst musicians. The only good evidence for motor-coordination challenges is provided by the expert group in executing chords with two black notes. Other findings generally support the idea that decoding multivalent notation may be a substantial challenge faced by all pianists, more-or-less regardless of expertise.

Effect	**Expert**	**Moderate**
range of average RTs	*800-1500ms*	*1500-2500ms*
Clef	118 ms	209
Switch of clef	46	138
Key signature	185	620*
Effect of key change	65	111*
Last-seen-clef similarity	45	96
2 black notes	41	(n.s.)*
Inversion	29	216
Diminished chord	115	116*
Other diatonic chords	70*	(n.s.)*

* results that may be unbalanced by the effect of inversion/diatonic chord

Table 1. Summary of results from experiment 1, with differences in mean RTs given in ms.

4. EXPERIMENT TWO

4.1 Method

A follow-up experiment took place at Dartington Summer School, likewise recruiting interested volunteers by word-of-mouth, at a variety of expertise levels. Participants were sorted according to performance on a practice block, with 15 moderate participants completing a shorter, balanced version of the experiment with 252 trials.

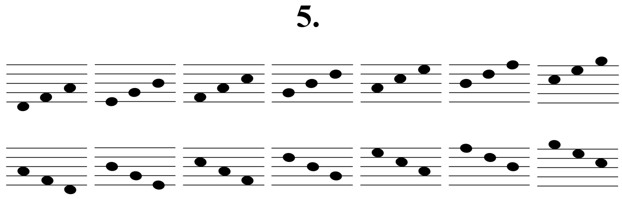


Figure 11. 14 musically balanced triads used as stimuli in a second experiment.

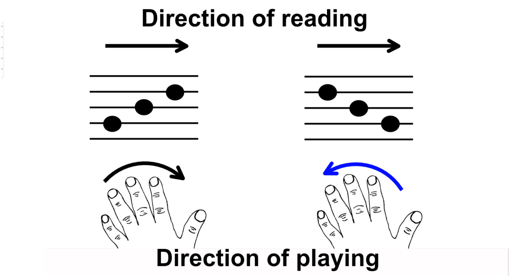


Figure 12. Congruence and incongruence of playing direction *vs* reading direction, in forward and reverse presentation of stimuli.

This experiment omitted the task-switching element of the design, (which, although statistically significant, was not very large), and instead presented trials alternately to each hand.

Seven root position chords were used (see Figure 11), and every stimulus was presented in every key. In a variation of the original experiment, each triad was presented both forward, and in reverse, *i.e.* with the highest note first. The hypothesis was that incongruence of direction (Figure 12) might provoke a 'Simon effect' [5], with descending Figures disadvantaged.

6. RESULTS & DISCUSSION, EXPERIMENT TWO

6.1 Data Analysis

15 moderate participants completed the experiment, with RTs in the range 1150-3300ms. 12 of these fell in the range 1400 – 2400ms. Given that the experiment was restricted to root position triads, this group forms a good comparison with the moderate group of Experiment 1.

6.2 Effect of Clef

6.2.1 Results

The mean reaction times for contrasting clefs were treble/right, 1722ms, and bass/left, 1726ms: a difference of 109ms, F(1,11)=15.69, p=0.002. Once again, three self-reported left-handers showed no left hand advantage.

6.3 Effect of Key Signature

6.3.1 Results

As before, key signature was found to be significant. The slowest average performance was in the key signature of 4b, (1973ms), and fastest in the key 0, (1488ms); a difference of 485ms (F(8,99)=6.10, p<0.001).

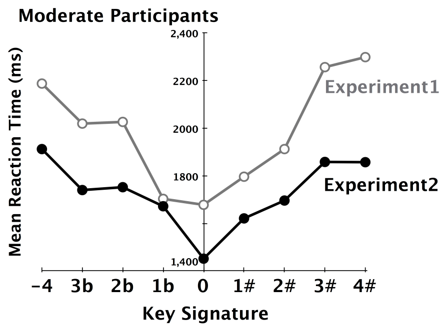


Figure 13. Mean RTs across 9 common key signatures; data from both experiments for comparison.

6.3.2 Discussion

Although, once again, individual key signature profiles showed great diversity, the averaged data for this balanced version of the experiment showed interesting similarities to data from the previous experiment, seen in Figure 13. The 'kink' in the flat keys suggesting an advantage for 3b, and what may be a corresponding disadvantage for 3# are apparently consistent features that would bear further investigation.

6.4 Effect of Order

6.4.1 Results

The order of presentation of the triads was statistically significant, with a mean reaction time of 1726ms in ascending triads, and 1830ms descending, a difference of 104ms F(1,11)=36.71, p<0.001. No interaction with clef was found. Across the whole experiment, single-note errors showed a tendency to be more common in the third note than the first two: of 84 such errors, 45 were in the third note.

6.4.2 Discussion

This result is similar in size to the effect of clef, and may to some extent reflect the cognitive architecture required to process a mirror rotation. While not discounting the possible hypothesis that presenting the naming note of the chord first confers an advantage, the pattern of errors (either in this experiment or the previous) did not show any evidence that the key note was preferentially recognised.

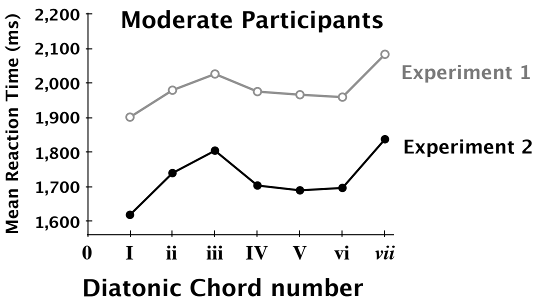


Figure 14. Mean RTs across 7 diatonic chords.

6.5 Effect of Chord Type.

In this experiment an analysis of diatonic chord was significant, with lowest mean reaction time for chord I 1662ms, and the highest for the diminished chord *vii* of 1895ms, a difference of 233ms, $F(6,33) = 8.57$, $p<0.001$. The graph of mean RTs for individual diatonic chords was again very similar to the corresponding previous data; see Figure 14.

6.6 Summary

The follow-up experiment clarified previous results of key signature and diatonic chord using a more rigorously balanced design, and found an effect of order of notes to be played. The particular shapes of the key signature and diatonic graphs are interesting and merit further exploration.

7. EXPERIMENT THREE: IMPROVED NOTATION SYSTEM

There have been various attempts to improve piano notation to be better suited to describing the execution of music at the keyboard[1]. There is great resistance to any kind of notation, however, that does not take account of the enormous canon of existing literature, or the years of investment by current professionals in the traditional system. Is any modification to 'standard' notation possible that might clarify the cognitive difficulties, whilst remaining legible to those accustomed to traditional notation? One suggestion is given in Figure 15.

Figure 15. Modified noteheads, showing the chords Bb major and D major. Notes with the left half filled are flat (b), and with the right half filled, sharp (#). Short barlines are also placed to clarify the clefs.

Modification of noteheads had been implemented in the testing program from the outset, partly to provide a fallback for those who found the main experiment beyond their capability. Assorted pilot data indicated that this notation improved RTs at every level of competence, and in the case of one or two dyslexic participants made a transformational difference. The improvement was difficult to quantify simply by contrasting complete runs of the experiment, as there was also a considerable learning effect whenever the experiment was taken more than once.

A third experiment recruited participants mainly from the German town of Münster, mostly in the age-group 18-30, from the University Choir or Institute for Music Education.

Blocks of the clarified notation shown in Figure 15 were presented alternately with blocks of the black noteheads used in previous experiments. Six 'difficult' key signatures (4b,3b,2b, 2#,3#,4#) plus the central key of 0 were arranged in one of four maximally confusing key orders. Each key signature was presented twice in each run of the experiment, in such a way that all six key signatures were seen in both notations. In other respects the procedure was identical to Experiment two, apart from including one extra practice block of the new notation. The overall design investigates the enabling or disruptive effect of key signatures on one another.

8. RESULTS & DISCUSSION, EXPERIMENT THREE

8.1 Data Analysis

At the time of writing, 10 moderate participants with average RTs between 1400 and 2450 had completed two contrasting runs of the experiment, as one quarter of a larger 4x4 design. (Data from a further thee expert participants, plus eight who completed one experiment are not reported here.) Runs were undertaken in the same session, with not more than 20 minutes break between them.

8.2 Summary of Replicated Effects

The effects of clef, order and key signature were consistent with the previous experiments, with an average clef difference of 203ms, a difference between slowest and fastest keys (4#, 0) of 623ms, a difference between rising and falling note orders of 151ms, all highly significant, and a difference between diatonic chords (I and *vii*) of 247ms. The diatonic profile showed a relative advantage for chord V compared with previous results, making it slightly more like the expert profile seen in experiment one: see Figures 17, 14 and 10.

8.3 Effect of Clarified Notation

The effect of clarified notation was analysed in combination with the other factors across the six difficult keys.[2] The average RT for traditional notation (all black circles) was 1970ms compared to 1693ms for clarified notation (see Figure 12), a difference of 277ms. ($F(1,60) = 22.1$, $p=0.001$). There was also a dramatic effect of notation on error scores. Results are shown in Figure 16.

[1] Klavar Notation, for example, is a well-developed alternative.

[2] Clarified notation could also be expected to improve performance in the central key, by removing the conflicting mappings from other keys, but not in an experiment where traditional notation is also being presented.

There was also no interaction with the effect of diatonic chord, shown clearly in Figure 17. Both clarified and traditional notation showed similarly ‘musical’ patterns.

8.4 Learning Effect

8.4.1 Results

There was a significant learning effect across the two experiments. Across the difficult keys, the average RT for the first run was 1923ms, and 1724ms for the second; a difference of 199ms, $F(1,54)=43.28$, $p<0.001$. (This compares to the learning effect in the central key (0#/b) that fell from 1459ms to 1304ms; a proportionately comparable drop of 155ms.) The effect did not show an interaction with the clarified notation, which appeared to confer a similar advantage across the two experiments of 279ms and 274ms.

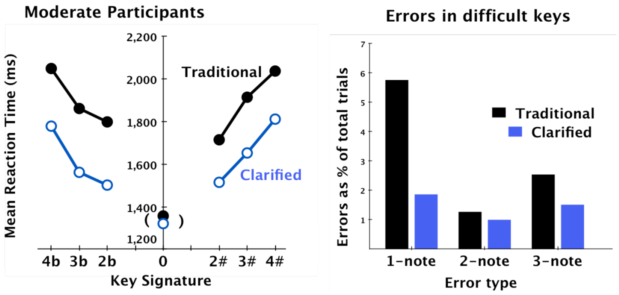


Figure 16. Results in difficult keys: comparing traditional and clarified notation.

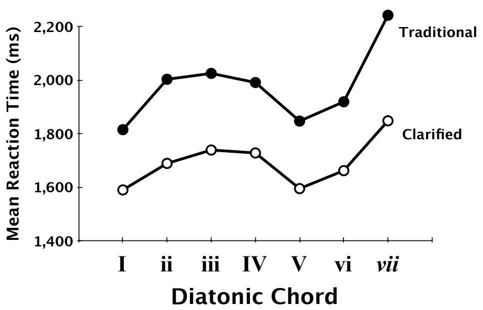


Figure 17. Results in difficult keys: Diatonic chords.

Rates of error also remained constant between the experiments, excepting 3-note-type errors, which fell slightly in the second experiment in both notations.

8.5 Participant feedback

8.5.1 Results

All participants commented that they found the clarified notation easier/faster to process, despite having been advised that the main aim of the altered notation in this experiment was to test the effect of key signatures on one another. Participants remarked that they not only performed more quickly, but were also more sure of their answers, and therefore felt less requirement to double check every response for errors before pressing the keys. A number of them made unprompted suggestions about how the clarifications could be introduced into ordinary piano music, notwithstanding the need to differentiate minims (whole notes) from crotchets (quarter notes).

8.6 Discussion

These are large effects in cognitive processing terms, with a gain of 15-20% on reaction times in difficult keys, and a halving of errors. More data is needed to complete the contrast of the particular key signatures being studied, but clearly from an experimental design perspective this is a useful way to separate some of the visual effects from other features of motor architecture or cognitive musical structure. It required very little acclimatisation, and conferred what appears so far to be a consistent advantage in difficult keys.

In terms of incorporating some clarifications into standard Piano music, the comments from participants are interesting. It may be argued that the visual disadvantage of overlapping pitch representations is somewhat overstated in these experiments, as there is so much extra contextual information on a real piano score. Looking at the question in reverse, however, freeing attention and working memory from the constant over-checking for pitch errors could leave room for sight-readers to take in more of that contextual information, resulting in bigger gains than those reported here.

It is encouraging that the diatonic key profile appears to persist in the clarified notation, as a further objection would be that clarifying the notation would reduce sight-reading to simple ‘button-pushing’ without the need for any musical understanding. In fact there was some indication from earlier experiments that at least part of this musical structure learning takes place outside of conscious theoretical understanding; some participants could not name either the major of minor keynote of most key signatures but nevertheless showed data of approximately this pattern. Clarifying the notation across the standard repertoire might simply have the effect of accelerating the pattern-learning process.

9. CONCLUSIONS AND DIRECTIONS FOR FURTHER STUDY

This study demonstrates that even sight-readers who excel have not achieved equal familiarity with every key, or parity of reading/execution between the clefs, and begins the process of investigating why this might be so.

The nature of the overlapping mappings, and the effect of inversion and diatonic chord make it complex to disentangle the effects of one clef on the other one, one key signature on another, or one visual pattern on another without finding a way to remove some of the confounds. Clarified notation creates a useful contrast to disentangle some of these effects and may itself provide either a training aid, or a structured alternative to traditional notation for those who find it useful.

9.1 Directions for Further Study

The third experiment uses clarified notation to investigate particular aspects of interference between one key and another, with data collection continuing at the time of writing. Further study is needed to discover whether the learning effect seen in traditional notation blocks is improved by interspersing blocks of clarified notation, or proceeds independently of it.

Further work also aims to involve the dyslexic / dyscalculic population, for whom Piano sight-reading often presents a disproportionate challenge, and for whom an alternative notation could offer particularly relevant benefits.

Presenting a form of clarified notation in a more realistic score format, and comparing attempts to sight-read simple pieces across more than one experimental session is also planned. Better sight-reading accuracy, and also better session-to-session retention in clarified notation is cautiously expected.

Acknowledgments

Many thanks to all the participants for their time, interest, suggestions and anecdotal insights, to Professor Stephen Monsell for his supervision in devising the experiment, and to the staff of Dartington Summer School for making the second experiment possible.

10. REFERENCES

[1] J. Sloboda, "The Eye-Hand Span-An Approach to the Study of Sight Reading," *Psychology of Music*, vol. 2, no. 2, pp. 4–10, Oct. 1974.

[2] Z. Breznitz, "Asynchrony of visual-orthographic and auditory-phonological word recognition processes: An underlying factor in dyslexia," *Reading and Writing*, vol. 15, no. 1–2, pp. 15–42, 2002.

[3] L. Gottlob, "Reading Homographs: Orthographic, Phonologic, and Semantic Dynamics," Journal of Experimental Psychology: Human Perception and Performance, Vol. 25, No.2, pp. 561–574, 1999.

[4] S. Monsell, "Task switching," *Trends in Cognitive Sciences*, vol. 7, no. 3, pp. 134–140, Mar. 2003.

[5] M. C. DeSoto, M. Fabiani, D. C. Geary, and G. Gratton, "When in Doubt, Do it Both Ways: Brain Evidence of the Simultaneous Activation of Conflicting Motor Responses in a Spatial Stroop Task," *Journal of Cognitive Neuroscience*, vol. 13, no. 4, pp. 523–536, May 2001.

[6] *Piano scales & arpeggios (from 2009). Grade 8. Grade 8.* London: ABRSM Pub., 2008.

[7] S. Furneaux and M. F. Land, "The Effects of Skill on the Eye-Hand Span during Musical Sight-Reading," *Proceedings: Biological Sciences*, vol. 266, no. 1436, pp. 2435–2440, Dec. 1999.

[8] Goolsby Thomas W., "Eye Movement in Music Reading: Effects of Reading Ability, Notational Complexity, and Encounters," *Music Perception: An Interdisciplinary Journal*, vol. 12, no. 1, pp. 77–96, Fall 1994.

FROM TRANSCRIPTION TO SIGNAL REPRESENTATION: PITCH, RHYTHM AND PERFORMANCE

Marie Tahon
IRISA
6 rue Kérampont
Lannion, France
marietahon@noath.net

Pierre-Eugène Sitchet
Paris-Sorbonne University, IReMus
2 rue de Louvois
Paris, France
zin@ginositson.com

ABSTRACT

Musical transcription is a real challenge, and more so in the case of folk music. Signal visualization tools may be of interest for this kind of music. The present paper is a comparison between a musical transcription and two signal representations (pitch and rhythm) applied to a song taken from the Gwoka repertoire. The study aims at finding similarities and differences in pitch, rhythm and performance features between the transcription and the signal visualization. Signal visualization is based on vowel segmentation, and on extraction of pitch and duration information. Transcription provides general characteristics about the music (harmony, tonality and rhythmic structure), while signal visualization provides performance-related characteristics. The main conclusion is that both approaches are of great interest for understanding folk music.

Keywords : Musical transcription, Signal visualization, Pitch, Rhythm, Performance, Gwoka.

1. INTRODUCTION

Transcription is described in ethnomusicology as a complex concept which overlaps musical analysis and musical culture [1]. Although transcription is a very difficult task, it may be a good support for analyzing musical features. Therefore, there is a growing interest in ethnomusicology for computational methods and their application to audio data collections [2]. Folk music recordings are very challenging because (i) instrument models do not necessarily exist ; (ii) audio quality is not always satisfactory ; (iii) sung languages have not necessarily been studied from a phonetic and cultural point of view. Automatic music transcription has been developed extensively during the past five years. This development is related to the huge amount of audio data recorded by everyone around the world and shared on databases. Thanks to multi-pitch detection and development of instrument physical models, automatic transcription is becoming more and more efficient. Added specific preliminary knowledge is of great importance for improving the efficiency of the system. As for example, knowledge related to morphological features of the sound of an instrument, sympathetic resonances, inharmonicity, harmonic transitions, tempo or rhythm. Automatic music transcription provides onsets, durations and pitch information. However such information is not sufficient for researchers who are looking for sound quality and music performance.

At present, many audio interfaces have been developed for analysis purposes. Among others, we can cite EAnalysis [3] and SonicVisualizer [4]. These tools are very powerful for giving a real-time representation of the music. They have been developed mainly for instrumental music. Some researchers in ethnomusicology try to use spectrograms for annotating timbral features [5], while others have used automatic transcription [6] and pitch analysis [7]. All these tools are a great help for analyzing music from scores or from audio recordings, but they are performance-specific. For analyzing music from recordings it is necessary to be able to separate performance features (musician, room, context, audio quality) from musical features (structure, harmony, tonality). In improvised folk music such a separation is very uneasy as these features are usually a combination of linguistic, musical and cognitive ones. In order to generalize the resulting analysis, many performances of the same song must be automatically analyzed.

The present work aims at validating an automatic analysis process by comparing manual transcription to signal representation obtained for a single performance. The present study has been carried out within the framework of a research project on Gwoka that aims at finding musical characteristic elements as well as, performance criteria of one specific singer.

The comparison between the manual transcription made by one of the authors and the signal representation obtained by the other underlines similarities and differences between elements such as pitch, rhythm or performance features. Energy level has not been investigated in the present study.

Section 2 summarizes the musical context of Gwoka. Transcription and signal representations are described in detail in section 3. Section 4 is devoted to the comparison between the two approaches. Conclusions and perspectives are drawn in the last section.

2. MUSICAL CONTEXT

Gwoka is a musical genre that emerged in the 17th century during the transatlantic slave trade. Today it has be-

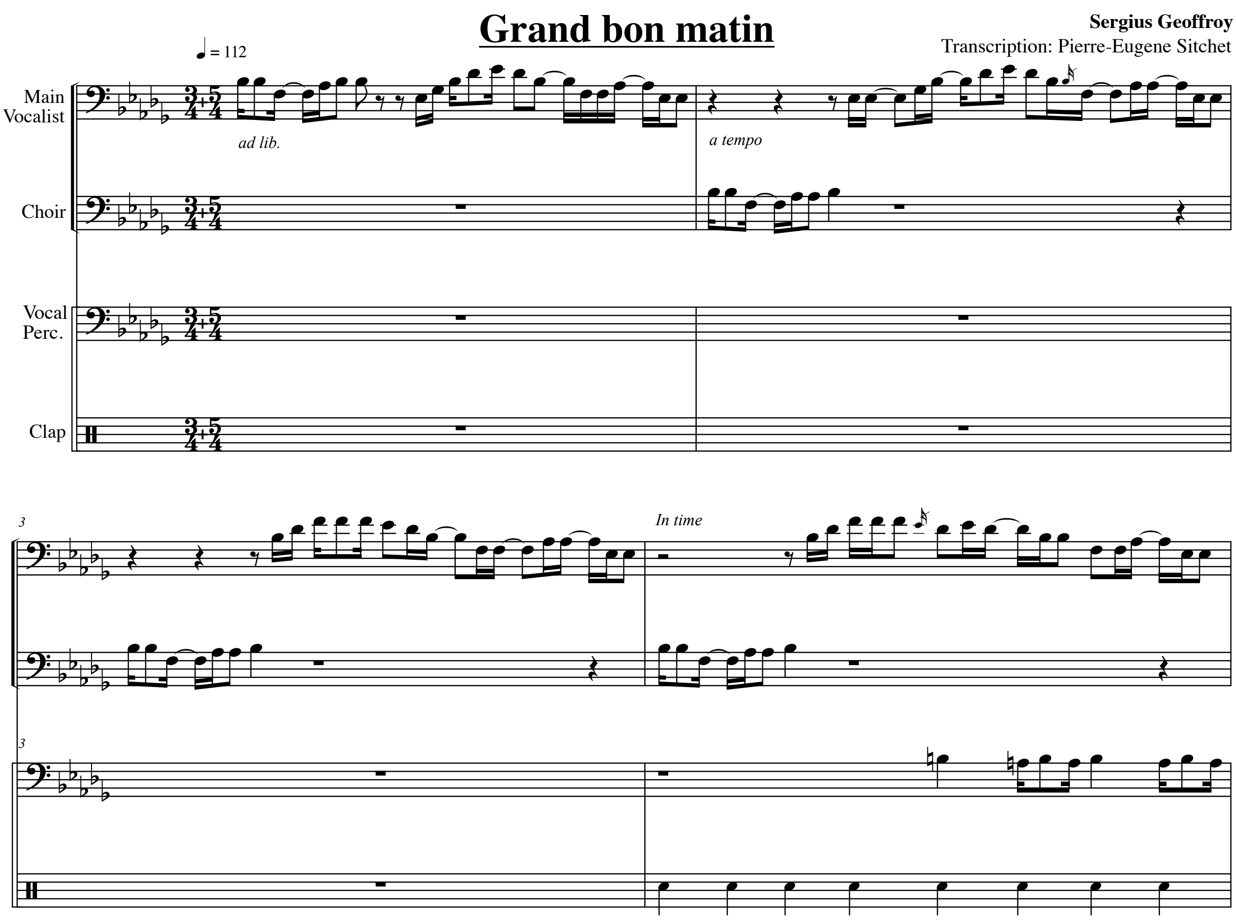



Figure 1. Musical transcription from Pierre-Eugène Sitchet of the song *Dimanche Gran Bon Maten* interpreted by Sergius Geoffroy. The lyrics are : *Dimanche gran bon maten / Gadé koudpyé nonm la té lansé an ka Ibana / Mwen di frè koudpyé nonm la té voyé an kaz a Ibana / Ka mandé lé répondè frapé lanmen an kaz a Ibana*

come an important element of the cultural heritage of Guadeloupe. Combining drum, song and dance, Gwoka is for those who practice it an artistic mode of expression, as well as an assertion of identity, a state of mind and a way of life. Through its history, Gwoka music has played —and still plays— an outlet role, offering an opportunity for freedom of expression. It has also become a sort of catharsis for Guadeloupeans [8].

The "Code Noir" forbade the use of any kind of drums in the practice of music. The slaves used a vocal technique, called Bouladjel, which imitates drums. While Gwoka was frowned for for years, it is now one of the most famous music and dance genres in Guadeloupe. Gwoka has been added to the UNESCO "Representative List of the Intangible Cultural Heritage of Humanity" on November 26, 2014. This musical genre is characterized by improvisation, syncopated rhythm and question-answer structure between a choir and the soloist.

Gwoka is also a kind of verbal joust between singers. Since it is a challenging performance [9] the singer has to hold the audience's attention for a long time. The singer's performance is generally assessed according to two criteria : "Santiman" (evaluation of the emotional expressivity of the singer) and "Lokans" (evaluation of the singer's power and his/her improvisation skills).

Gwoka is sung only in Creole [10]. Antillean Creole language came emerged during the slavery era. African slaves were forced to develop a new form of verbal communication by relying on what they heard from their French masters and from other African slaves. As a result, Creole is a combination of European words and of African expressions and sentence structure.

3. MATERIALS AND METHOD

In the present work the authors have made use of a song of Gwoka performed by Sergius Geoffroy and compared its transcription to the signal annotations. The entire song lasts 150 seconds. It alternates solo-improvised variations and a choir (répondé). The study focuses on the first four variations of the first 25 seconds.

3.1 Musical transcription

The musical transcription was based on a recording of the song. The first instants are reported in Figure 1. Four elements are considered : the soloist, the choir, vocal drums and claps. The choir is always singing the same melodic pattern repeating the title of the song : *Dimanche Gran Bon Maten*. The introduction is performed by the soloist ad libitum and the definitive tempo is reached at the third variation when claps come in. Bars have been chosen from a paradigmatic point of view. Each bar is a melodic, rhythmic and linguistic variation of the first one. The introduction and the first four variations contain 60 notes. The lyrics have also been transcribed, those transcriptions corresponding to the extract that was studied are shown in Figure 1.

Some "statistical" features —in opposition to "syntactical" features such as pitch and durations— are included in the score, for example : *ad libitum, a tempo*. Such annotations will never be able to provide information about voice quality or vocal techniques such as *vibrato, grupetto* because there are an infinite number of ways of doing them [11].

3.2 Signal visualization

3.2.1 Vowel segmentation

Because the study focuses on vocal music, speech and singing voice analysis features have been adapted to its specificities. The present contribution focuses on vowels only because these are homogeneous in time and spectral domains. Other speech sounds, such as voiced consonants, have been discarded because their spectral characteristics differ from those of vowels. Such tools are based on the extraction on vowels of different features. These features are related to pitch, formants, durations and voice quality. The extraction of such features from other speech signals (consonants) or instrumental signals (claps) would not have any physical meaning.

Creole speech is a highly voiced language in comparison with French, and an automatic segmentation based on voiced parts was not possible. Moreover, since this is a language that has not been widely studied, automatic syllable segmentation would probably result in many errors. For these reasons, the authors decided to segment and annotate manually vowels in sung and spoken tracks. Since the segmentation of vowels is mainly used to study their duration, they are annotated according to basic French vowels /a,e,i,o,u/. Nasals and diphthongs are annotated according to the closer basic vowel. Segmentation and annotation have been done with Transcriber [12]. A complete phonetic transcription would be interesting for studying pronunciation variants, but it has a higher annotation cost.

Pitch is extracted with Praat [13] tools every 10 ms on voiced signals only. Praat tools can make octave errors. To avoid such errors, only vowel signals that last more than 40 ms are considered. Duration corresponds to the length of the vowel signal that has been segmented. In the first four variations, 65 vowels were segmented. The number of notes and the number of vowels are not identical : some of the notes contain two vowels, some of the vowels are too short and therefore are not taken into account during the transcription process.

3.2.2 Pitch representation

Signal features are plotted on a graph in Figure 2. Continuous black lines correspond to the F key stave. Dotted black lines correspond to each semitone, the reference frequency being A4 at 440 Hz. The authors have added red notes that correspond to the pitches and durations obtained during the transcription process. Durations are computed on the basis of a 112 bpm tempo. The beginning of each note is synchronized with the beginning of each vowel.

3.2.3 Rhythm representation

Rhythm transcription has been done in order to highlight the global structure of the song. A bar is shown at the end of each melodic variation. In this section, the authors compare onsets and durations of each note or vowel. The transcribed score at relative level gives a note's duration ; the manually recognized tempo (112 bit per minute) gives their absolute values in seconds. These durations are discrete values. The manual segmentation process gives a vowel's duration. These durations are continuous values. It is important to consider that a note usually corresponds to a whole syllable (consonant and vowel) ; therefore the duration of a vowel will probably be shorter than that of a note.

Rhythm is visualized on a 2D graph (see Figure 3). Abscissa is time in seconds while ordinate is the duration value in seconds. Blue lines represent the duration of vowels and red lines the duration of notes. Onsets are used to plot each line at the right time. Black vertical lines join the beginning of a vowel to its assumed corresponding note. If a vowel has no corresponding notes it is linked to zero value.

At first the plot (not included here) showed that onsets and durations of extracted vowels and transcribed notes were not comparable. Onsets and durations of notes were thus adjusted to vowel onsets and durations (Table 1). The following two adjustments were made :

— The first note onset of each variation is synchronized with the first vowel onset.

— Tempo is adapted for each melodic variation using equation 1 where the transcribed variation duration is the number of beats of the variation and the vowel variation duration is the duration in seconds of the variation.

$$Tempo = 60 \times \frac{TranscriptedVarDuration}{VowelVarDuration} \quad (1)$$

4. COMPARATIVE ANALYSIS

In the present section, the authors discuss the comparison between transcription and signal representation.

4.1 Pitch analysis

4.1.1 Note decision

During the transcription process, the musical ear needs to decide the pitch of the note. This decision is not easy

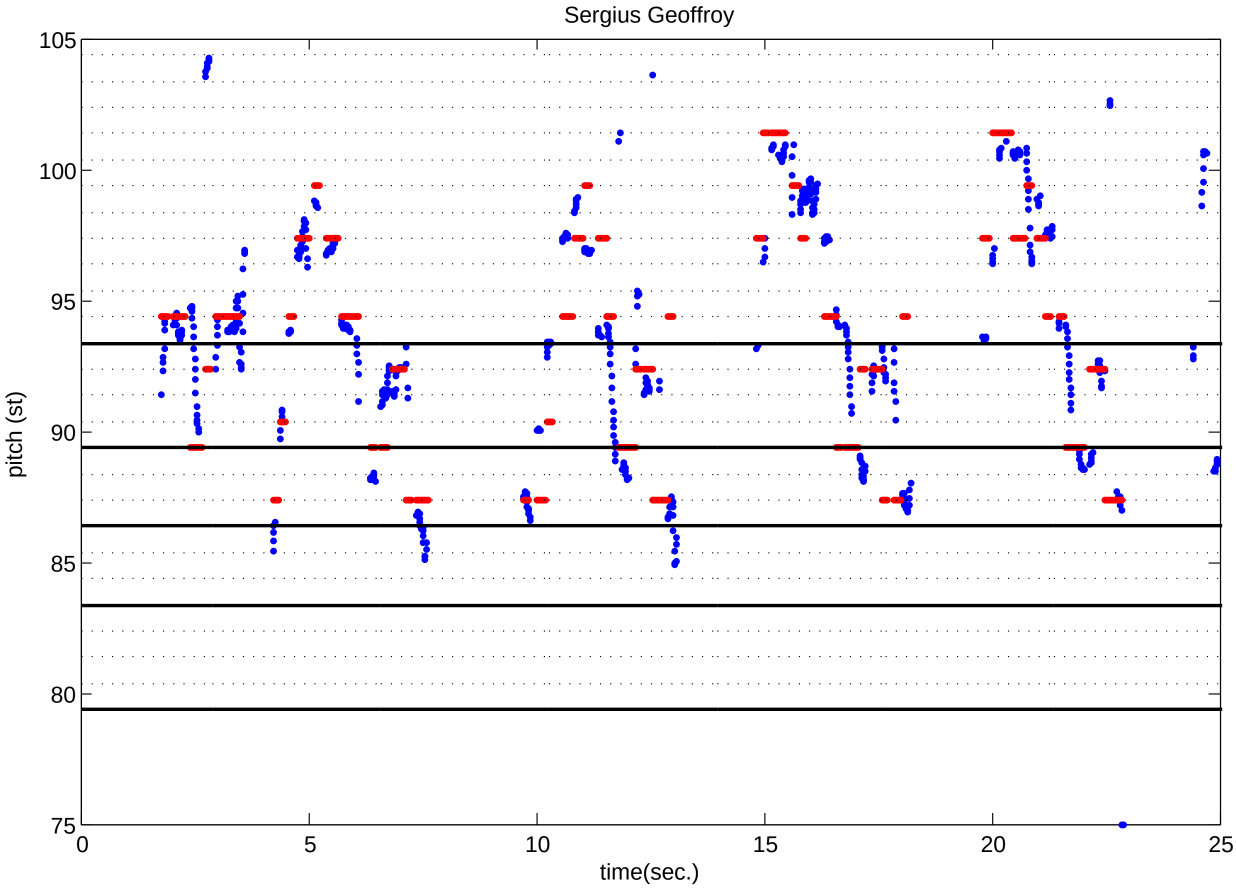


Figure 2. Pitch representation : extracted pitch values in blue and transcripted notes in red, according time.

Variation	Onset (sec.)	Tempo (bpm)
Intro	1.79	80.21
Var1	4.22	77.81
Var2	9.71	95.65
Var3	14.85	95.93
Var4	19.81	103.13

Table 1. Adjustments used for rhythmic comparison.

and is usually the result of a cognitive process that requires an interpretation of the song. There are some meaningful differences between transcribed notes and extracted pitch values. The transcriber usually chooses either :

— The last pitch he heard. It generally occurs after a glissando.
— The first pitch he heard. It generally occurs at the end of the variation.
— A kind of average of the pitches he heard. It generally occurs during a vibrato or even a glissando.

4.1.2 Tonal and harmonic a priori knowledge

Gwoka is a modal music with strong tonal characteristics [14]. The transcriber has in mind an a priori interpretation. For example, at the very begining of the second variation, the vocalist performs several glissandi from Db2 to D3, while the transcription shows successive notes from Eb2 to Eb3 from the dominant chord.

One can see that intonation is increasing with time. This evolution is discarded in the transcription, but is remarkable on the pitch signal representation. This information may be useful for understanding the performance, but is harmful for generalization purposes.

4.2 Rhythm analysis

4.2.1 Tempo stability

The tempo is not stable. In order to synchronize transcription with signal rhythm representation, one needs to adjust the tempo at each variation. During the performance different bars do not last the same time. Here again, this information is useful for understanding the performance and harmful for generalization purposes.

4.2.2 Vowel accentuation

Most vowels are shorter (in the signal representation) than the corresponding transcribed notes. However, some vowels are longer (for example the 6th vowel in Figure 3). This information helps the musicologist to locate accentuations in a given musical phrase. These usually occur at the same time as a vibrato, thus showing the importance of this particular vowel. Of course, these accentuations must be compared to the linguistic content.

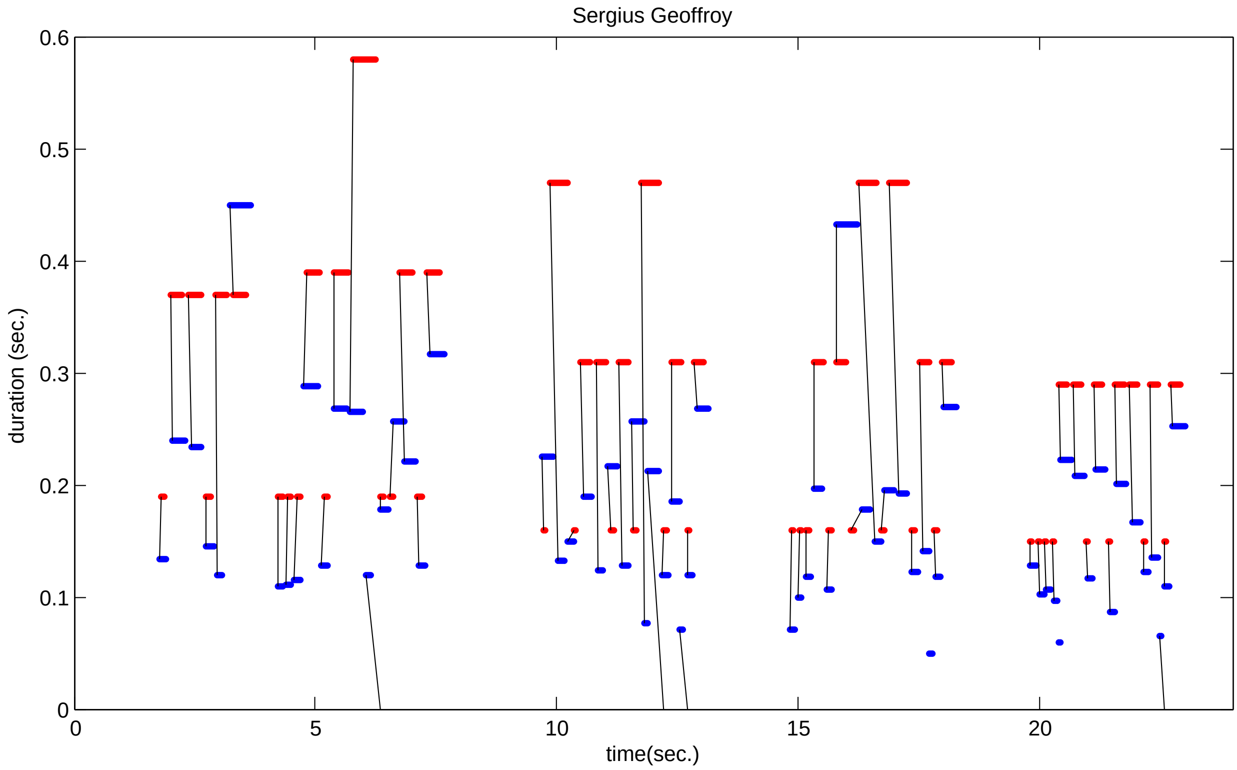


Figure 3. Rhythm visualization. Segmented vowel durations in blue and transcripted note durations in red according time.

4.3 Voice quality analysis

4.3.1 Vocal techniques

Annotation of vocal techniques is possible on the transcription score. An example is given in Figure 1, at the second variation. Some written annotations are usually added to the score in order to help musicians to interpret the score. However, the score has no vocation to show vocal quality or vocal techniques used by the singer during his performance [15].

Transcription in ethnomusicology aims at understanding and analyzing performances. Therefore indications on the techniques used are of great importance. Such indications are highly related to the musical context. Information about vibrato, glissando or grupetto can be seen on pitch graphs. Of course, a spectrogram of the signal greatly helps to detect these vocal techniques.

4.3.2 Linguistic content

In Gwoka, the linguistic content and the musical performance are strongly linked. Creole lyrics have also been manually transcribed (see Figure 1). However, the transcription has been made according to the guadeloupean creole alphabet and does not reflect the real pronunciation. An automatic speech recognition system trained for French Creole could enrich the transcription with phonological aspects.

For example, a previous work on Gwoka [16] shows that nasality is a specific-singer trait. Each singer has also his/her own way of saying the same word. Knowledge of the lyrics is very important to understand the song in the general context of Gwoka, while knowledge of phonological aspects helps the musicologist to understand the performance.

5. CONCLUSION AND DISCUSSION

This paper focuses on a single song of the Gwoka repertoire from Guadeloupe. It presents a comparison between a manual transcription and a signal representation through three main aspects : pitch, rhythm and performance.

Preliminary results validate an automatic transcription process thanks to a comparison with a manual transcription. While manual transcription generalizes the analysis of a specific song by introducing an a priori knowledge, automatic transcription is specific to one particular performance. Manual transcription highlights specific tonalities and rhythmic structures through a priori knowledge of harmony, rhythmic structure, etc. On the contrary, signal representation is entirely related to the song that is being analyzed (recording conditions, singer, context of the performance, etc.). The automatic process of statistical analysis can be done for a large number of performances of the same songs, making it possible to generalize the analysis. Thus it turns out to be a very nice tool for oral folk song analysis.

The present comparison between musical transcription and signal representations allowed the authors to describe some characteristic features of Gwoka and to understand how a singer performed his own interpretation. The singer's performance features result from several vocal techniques (grupetto, glissando, and vibrato), accentuation of some vowels, an increasing tempo and an increasing intonation.

The main conclusion is that both manual transcription and signal processing are complementary for oral music analysis. The fact that the two analyses were conducted independently avoids bias. Such an approach should provide researchers with tools for a better analysis of music.

Acknowledgments

The authors would like to thank all singers from Guadeloupe who have participated to the recordings.

6. REFERENCES

[1] B. Nettl, "I Can't Say a Thing until I've Seen the Score : Transcription" in *The study of ethnomusicology : Thirty-one issues and concepts.* Urbana, IL : University of Illinois Press. pp. 74-91, 2005.

[2] S. Canazza, A. Camurri and I. Fujinaga, "Special section : ethnic music audio documents : From the preservation to the fruition", *Signal Processing*, 90(4), pp. 977-1334, 2010.

[3] `http://logiciels.pierrecouprie.fr/?page_id=402`

[4] `http://www.sonicvisualiser.org/`

[5] A. Blackburn and J. Penny. "Timbral Notation from Spectrograms : Notating the Un-Notatable ?" in *Proc. of the Int. Conf. on New Tools for Music Notation and Representation TENOR 2015*, Paris, France, pp.218225, 2015.

[6] D. Cazau, O. Adam and M. Chemillier. "Automatic music transcription of the marovany zither, based on knowledge from musical acoustics." in *Proc of the 5th International Workshop on Folk Music Analysis - FMA2015*, Paris, France, pp. 10-12, 2015.

[7] S. Abdallah, A. Alencar-Brayner, E. Benetos, S. Cottrell, J. Dykes, N. Gold, A. Kachkaev, M. Mahey, D. Tidhar, A. Tovell, T. Weyde and D. Wolff. "Automatic transcription and pitch analysis of the British Library World & Traditional music collections". in *Proc of the 5th International Workshop on Folk Music Analysis - FMA2015*, Paris, France, pp. 10-12, 2015.

[8] M.H. Laumuno. *Gwoka et Politique en Guadeloupe, 1960-2003 : 40 ans de construction du pays . [Gwoka and politics in Guadeloupe, 1960-2003 : 40 years of local establishment.]* l'Harmattan, Paris, 2011.

[9] M.H. Pichette, M. Desroches. "Comprendre le Gwoka par la performance. [Understanding Gwoka with performance.]" in *Mise en scne des territoires musicaux : tourisme, patrimoine et performance [Musical territories staging : tourism, heritage and performance.]*, Presses de l'Universit de Montral, 2011.

[10] M.C. Lafontaine. "Terminologie musicale en Guadeloupe : ce que le crole nous dit de la musique. [Musical terminology in Guadeloupe : what creole tells about music.]" *Language et Socit*, vol. 32, pp. 7-24, 1985.

[11] L. B. Meyer,"A Universe of Universals", *The Journal of Musicology*, Vol. 16, No. 1, pp. 3-25, 1998.

[12] C. Barras, E. Geoffrois, Z. Wu, M. Liberman. "Transcriber : development and use of a tool assisting speech corpora production." *Speech Communication*, vol. 33 (1), pp. 5-22, 2000.

[13] `http://www.fon.hum.uva.nl/praat/`

[14] J. Camal. "From Gwoka modènn to jazz ka : music, nationalism and creolization in Guadeloupe", PhD thesis, Graduate School of Arts and Sciences of Washington University, 2011.

[15] P.-E. Sitchet and M. Tahon. "Le voisement dans le Gwoka : entre le parlé et le chanté" [Voicing in Gwoka : between speaking and singing], Journées d'Informatique Musicale, Albi, France, pp.153-159, 2016.

[16] M. Tahon and P.E. Sitchet. "Interest of a mixed approach combining paradigmatic method and signal processing. Case study : Analysis of two musical pieces of Gwoka from Guadeloupe" In : *International Colloquium : Innovant tools and methods for music and signal processing teaching*. St Etienne, France, 2015.